Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2016, Article ID 2392149, 11 pages
http://dx.doi.org/10.1155/2016/2392149

*Research Article*

# Waterfalls Partial Aggregation in Wireless Sensor Networks

**Wuyungerile Li,[1] Bing Jia,[1] Shunsuke Saruwatari,[2] and Takashi Watanabe[3]**

[1] *College of Computer Science, Inner Mongolia University, Hohhot 010021, China*
[2] *Faculty of Information, Shizuoka University, Hamamatsu 4328011, Japan*
[3] *Graduate School of Information, Science and Technology, Osaka University, Osaka 5650871, Japan*

Correspondence should be addressed to Bing Jia; jiabing@imu.edu.cn

Received 8 December 2015; Revised 6 March 2016; Accepted 21 March 2016

Academic Editor: Miguel A. Zamora



In wireless sensor networks (WSNs), energy saving is a critical issue. Many research works have been undertaken to save energy. Data aggregation is one of the schemes that save energy by reducing the amount of data transmission. Normally, researchers focus on saving energy by aggregating multiple data or turning to achieving short transmission delay in data aggregation; few of them are concerned with network lifetime. This work achieves an optimum network lifetime by balancing energy consumption among nodes in network. Here, we propose a waterfalls partial aggregation, controlled by a set of waterfalls pushing rate vectors. The first contribution of this paper is to propose a waterfalls partial aggregation and to model it with queuing theory. The second contribution is that the optimum network lifetime is achieved mathematically and a near optimum algorithm is proposed for a given transmission delay. The results are compared with existing energy efficient algorithms and the evaluation results show the efficiency of proposed algorithm.

## 1. Introduction

In the recent development of wireless technology, wireless sensor networks (WSNs) [1] have attracted researchers' attention because of the applicability in many fields for effective collection of sensing data with low cost [2]. WSN consists of a large number of sensor nodes, where sensor nodes sense events and generate event data, then transmitting the data to a sink node via intermediate sensor nodes in a multihop manner [3]. Sensor nodes are battery-powered with limited energy supply; moreover, in many of the applications, sensor nodes are deployed in harsh nature environment or vast space so that the continuous energy supplement is impossible. In this case, if one of the nodes in the network exhausted energy, the network would break down and perform reorganization, where the reorganization of the network also consumes much energy and time. For these reasons, WSNs should be energy efficient. For energy saving, many researchers are working on Medium Access Control (MAC) protocols [4], routing protocols [5], topology control [6], and data aggregation [7] in WSNs.

In WSNs, data generated from neighboring sensor nodes are often redundant and highly correlated. Sensor nodes spend considerable energy for sending or relaying a large number of redundant data. Moreover, a large number of data transmissions cause data collisions and data congestion. All these lead to the turning up of data aggregation technique. Data aggregation is defined as the process of aggregating data from multiple sensor nodes to eliminate redundant transmission and provide fused information to a sink node. In data aggregation, there are mainly three kinds of aggregation ways. The first is clustering data aggregation where data are collected and aggregated at a cluster node and then transmitted to a sink node [8, 9]. The second is hop by hop aggregation, which means that data are aggregated at each intermediate node [10]. The third is partial aggregation, where data aggregation satisfies a time or energy threshold [11]. However, all these three kinds of data aggregations have their drawbacks. In clustering data aggregation, a cluster head always consumes much energy than others; WSNs always perform a cluster head selecting algorithm to decide a new cluster head, which wastes considerable time and energy [12].

In hop by hop aggregation, data suffer long transmission delay and nodes on the transmission way suffer unbalanced energy consumption [10]. Compared to these two, partial aggregation appears to be more flexible. In partial aggregation with scheduling technique always set a time period to collect data for nodes that have aggregating function. Here, time period is always decided by a given time threshold or requested data accuracy but always does not consider the energy consumption of nodes in the network [11, 13, 14]. As everyone knows, network lifetime is always decided by nodes' lifetime; when a node's energy is exhausted, the network has to perform reorganization. Hence, even though the partial aggregation can shorten the transmission delay, it still has the shortcoming of unbalanced nodes lifetime, which leads to short network lifetime.

In this paper, to achieve optimum network lifetime, at first, we propose a waterfalls partial aggregation (WPA), which is a kind of scheduling data aggregation scheme where optimum network lifetime is achieved by controlling the data transmitting period. Secondly, by analyzing the data transmission process of proposed scheme, we determined the formulations of energy consumption and transmission delay of the network, which causes advantage for future scientific studies. Then in Section 4, optimum network lifetime is analyzed mathematically and achieved by a heuristics algorithm. We present evaluation in Section 5. Finally, we show the conclusion in Section 6.

## 2. Related Work

WSNs have various applications. Some applications are required to send data as soon as possible, while in other applications energy saving is much more important. For example, in disaster monitoring or emergency rescue, immediate data transmission is more important than others; hence, data are always transmitted to neighboring nodes without aggregation; we call this nonaggregation. However, in nature monitoring, energy is more significant than transmission delay because the replacement of battery for sensor nodes is supposed to be impossible. In respect to energy saving, data aggregation technique is widely applied in WSNs that are composed of large number of sensor nodes and meanwhile it is not so easy to supply continuous energy. Hence, we call it full aggregation that processes data aggregation only for energy saving in the network. In some applications of WSNs, it is required to trade off node energy and transmission delay or some other properties. In this case, data aggregation should consider corresponding delay or some other required thresholds; we call this kind of data aggregation a partial aggregation. In this section, we introduce nonaggregation, full aggregation, and partial aggregation and then discuss their advantages and disadvantages.

*2.1. Nonaggregation.* Definition of nonaggregation is that a node transmits received data to an adjacent lower node immediately after receiving data from a neighboring node which means that all data are sent to a sink node one by one, as shown in Figure 1.

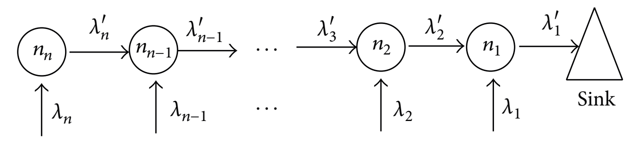


Figure 1: Nonaggregation.

From the definition of nonaggregation, we find that data transmission rate at a node becomes larger near a sink node because of data relaying properties of a node. Hence, energy consumption of nodes near a sink node is much more than that in nodes far from the sink. As a result, nodes near the sink node exhaust energy sooner than others, which results in short network lifetime. Moreover, when event data generation rate is large, data congestions occur at nodes around a sink node and they prolong transmission delay as well as let the network lifetime become worse [4–6, 15–18]. Sensor MAC [4] (S-MAC) is proposed for energy saving in wireless sensor networks based on IEEE 802.11 MAC protocols. In S-MAC, the network is always assumed to be less in amount of data transmission; data process and data aggregation can be performed in the networks where the transmission delay is considerably tolerant. S-MAC uses a listen/sleep model and divides the time into frames; each frame has the model of listen/sleep. Besides, topology control algorithms are studied for improving network lifetime, and the minimum spanning tree is the typical one. LMST [16] is a minimum spanning tree (MST) based topology control algorithm for multihop wireless networks, called local minimum spanning tree (LMST). The topology is constructed from each node, where a node builds its local MST independently (with the use of information locally collected) and keeps only one-hop on-tree nodes as its neighbors.

*2.2. Full Aggregation.* Full aggregation is that a node processes data aggregation when new data generated at itself. In more detail, the node does not transmit any received data from neighboring node immediately; all arrival data from neighboring nodes wait for new generated data. When there are new data at a node, the node aggregates all arrival data with its own data and then transmits them to its lower node. Figure 2 illustrates the full aggregation in detail. It is clear to see from the definition that full aggregation can save much energy by aggregating data at every relay node. The only condition for data transmission is that a new event data occurs at a node, which means that the data transmission rate is the same with data generation rate. If we assume data generation rates at all nodes are the same, then the number of data transmissions for each node is the same, so that full aggregation can achieve balanced energy consumption among nodes.

However, full aggregation has its inevitable defect that transmission delay is too long for data occurring at nodes far from a sink node because of the waiting time for generating data at every intermediate node. And what is more, the transmission delay is longer when data generation rate is low at nodes. PEGASIS (Power-Efficient Gathering in Sensor Information Systems) [10] is one of the energy efficiency

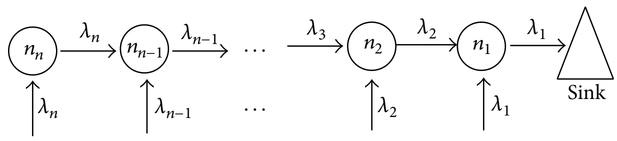


Figure 2: Full aggregation.

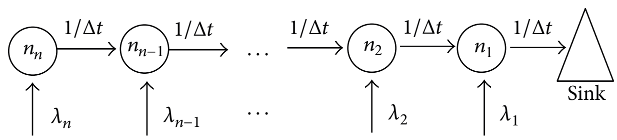


Figure 3: Partial aggregation.

chain based data aggregation protocols that employs a greedy algorithm. There are two assumptions in this scheme: one is that all nodes are far from a sink node; the other is that nodes except the end nodes fuse the received data and their own data and then aggregate them into one packet before transmitting them to another neighboring node. The main idea of PEGASIS is forming a chain among the sensor nodes so that each node will receive from and transmit to fused data only with a close neighbor. The fused data are sent from node by node, and all the nodes take turns to be the leader for transmission to the sink node. The disadvantage of PEGASIS is that transmission delay is too long for nodes that are at the end of a chain.

*2.3. Partial Aggregation.* In partial aggregation, data always wait a period of time to be transmitted for collecting more data at a node; all collected data are aggregated into one or several representative data and then transmitted to an upper neighboring node. The waiting time at nodes is adjustable and is always decided by corresponding applications [11, 13, 19–21]. Figure 3 illustrates a most simple class of partial aggregation in which data wait $\Delta t$ time for aggregation at a node before they are transmitted ($1/\Delta t$ is data transmission rate at a node). When waiting time $\Delta t$ is out, a node aggregates all generated data and received data into one type of representative data and then sent it out. In partial aggregation, energy is traded off for improving transmission delay, data accuracy, network lifetime, and so forth. Since full aggregation leads to long transmission delay and nonaggregation consumes much energy, partial aggregation can balance energy with other performances in WSN. As the waiting time at nodes is always decided by applications, one can set the waiting time very short when they want data with short transmission delay and otherwise the opposite.

In in-network cascading timeout data aggregation [13] and ATS-DA (Adaptive Timeout Scheduling for Data Aggregation) [11], a sink initially broadcasts a request to all nodes. Each node waits for a certain time period to receive data from their child nodes. The timeout period of each node is set based on the position of the node in the data aggregation tree. However, these studies are not applicable when real time data or short delay data are required. Because all data at a node have to wait at least $\text{SHD}_{\text{avg}}$ time, no matter whether the data are important or not. Moreover, when the data arriving rate is irregular at nodes, the node energy consumption is also irregular among nodes, which result in unbalanced network. In particular, in large scale WSN, it is very difficult to achieve fairness for all nodes from the points of transmission delay and energy consumption because of data collision and congestion as well as retransmission.

## 3. Waterfalls Partial Aggregation

*3.1. Sensor Network Model.* In this research, we apply queuing theory to analyze and model wireless sensor network. As in mathematical analysis, too complex network model makes it too sophisticated to get the analytical result and formulation; therefore, we define the network model to be the most basic and simplest model of tandem sensor network; however, the results can be extensible to other more complex topologies. The structure and transmission principle of tandem network are shown in Figure 4.

In tandem networks, all the nodes deployed in a flat are allocated omnidirectional antennas with the same transmission range. Data generated at the nodes are transmitted to a sink node in multihop manner. The distance between two neighboring nodes is the same, and all the nodes are within the transmission range of their neighboring nodes:

(i) $n_i$ denotes the $i$th node from the sink.

(ii) $n$ is a set of all nodes; $N$ is natural number set.

(iii) $E_i$ denotes the energy at node $n_i$.

(iv) $L_{\max}$ denotes maximum network lifetime.

(v) $D_n$ denotes the total transmission delay.

(vi) $\mu_i$ is waterfalls pushing rate at node $n_i$.

(vii) $t_i$ is data transmission delay for each node.

*3.2. Definition of Waterfalls Partial Aggregation.* To shorten the data transmission delay and to lengthen the network lifetime of the network as well as to suppress data congestion around a sink node, we propose a waterfalls partial aggregation (WPA). In WPA, data are transmitted to a neighboring node in two conditions: (a) if there are new local generated data at a node or (b) after waiting a holding time $\Delta t$ at a node. The inverse of the holding time $\Delta t_i$ at node $n_i$ we call waterfalls pushing rate and it is denoted as $\mu_i$ where $\mu_i$ becomes smaller to nodes nearer a sink node; it means $\mu_i > \mu_j$, $i > j$, as shown in Figure 5.

In other words, data tends to be aggregated rarely at nodes far from a sink node to suppress delay. When there are new generated data at a node or the holding time at this node is over, all data at this node are aggregated and wait for further transmission. In more detail, the holding time for nodes nearer a sink node is longer than the ones far from the sink, so that it results in the decrease of the amount of data transmission near a sink node. Finally, it can achieve an equal amount of data transmission at each node by controlling the waterfalls pushing rate vectors. The flow chart of data aggregation and transmission is shown in Figure 6.

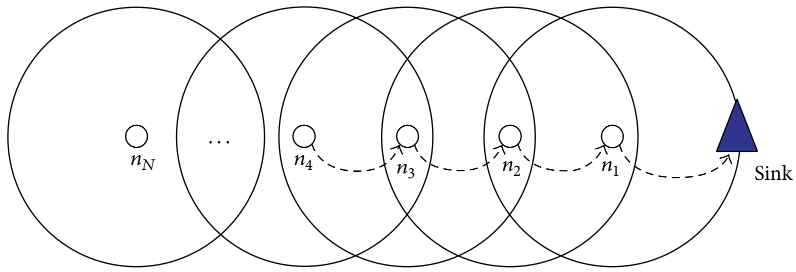


FIGURE 4: Tandem sensor network.

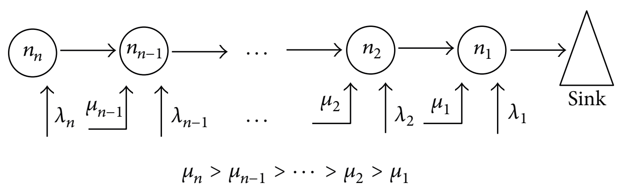


FIGURE 5: Waterfalls partial aggregation.

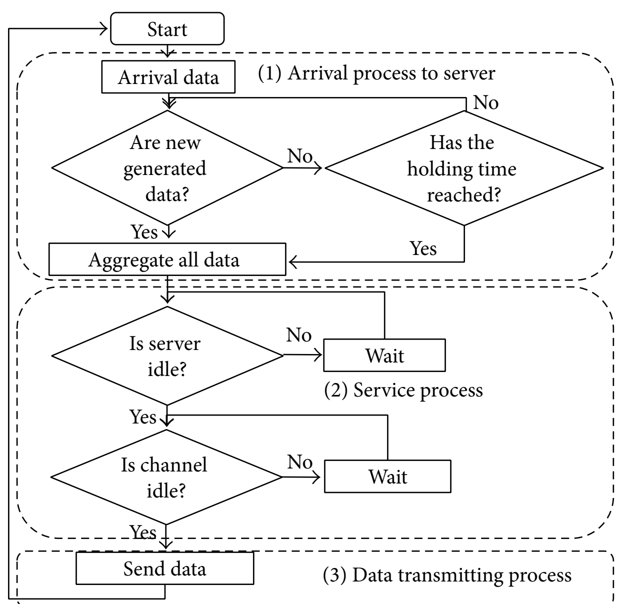


FIGURE 6: Flow chart of waterfalls partial aggregation.

In the flow chart, after data arriving from neighboring node, node $n_i$ starts to check if there are new generated data at itself. If yes, node $n_i$ aggregates all arrival data with the local generated data and then waits for service process; if no, node $n_i$ checks the holding time; if time is run up, it aggregates all arrival data and then goes to service process; otherwise, it checks new generated data. This process we call data arrival process to server. After aggregating data, node $n_i$ checks if the server is idle or not. If there are no other data waiting for transmitting, we say the server is idle; otherwise, server is busy; data queue up in the server and wait for further transmitting. When server is idle, node checks if the channel is idle or not. When channel is idle, data transmission process starts; otherwise, node $n_i$ checks the channel until it turns to idle. Data are sent to a neighboring node if the channel is idle.

*3.3. Problem Definition.* In WPA, the waterfalls setting of holding time is for balancing energy consumption among nodes as well as suppressing load. As we mentioned in Section 2 that network lifetime is always decided by the shortest lifetime node in the network; therefore, to lengthen the shortest node lifetime is to extend network lifetime. When tolerable maximum transmission delay is given by corresponding application, we set the holding time $\Delta t_i$ for nodes in the network according to WPA algorithm. The reason that set a maximum tolerable transmission delay is because in a finite network there would have several sets of waterfalls pushing rates vectors corresponding to different transmission delay. According to our definition, waterfalls pushing rate $\mu_i$ is inverse of holding time. With larger waterfalls pushing rate $\mu_i$, a node transmits more data and results in much energy consumption. Thus, the problem becomes how to set the waterfalls pushing rates among nodes so that all nodes keep having the same energy consumption.

In other words, how long should the holding time of arrival data for data aggregation at a node be to meet the conditions of balanced energy consumption among nodes and given transmission delay? For a node, if required transmission delay is given as $D_n$, then what our algorithm should do is let

$$D_n = t_1 + t_2 + \cdots + t_i + \cdots + t_n \quad (n \in N). \tag{1}$$

Here, $t_i$ is transmission delay for node $n_i$ and we determine the formulation of transmission delay for each node as well as total transmission delay of the network in Section 3.4. In WPA, data have to wait for $\Delta t$ time at a node for gathering more data; hence, the transmission delay for a node is mainly decided by this $\Delta t$. On the other hand, to balance the energy consumption among nodes, we first obtain total energy consumption $E_i$ for node $n_i$ which is composed of energy of data reception, data transmission, and overhearing. We show the solution of energy consumption for node $n_i$ and determine the formulations in Section 3.4. For node $n_i$, data transmission rate is decided by itself and data receiving rate is decided by its upper neighboring node $n_{i+1}$ and the overhearing rate is decided by its lower neighboring node $n_{i-1}$. For node $n_i$, data transmission rate $\lambda_i'$ is decided by data generation rate $\lambda_i$ and waterfalls pushing rate $\mu_i$. If we set the maximum network lifetime as $L_{\max}$, then we obtain (2)

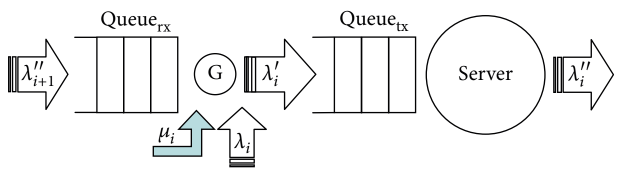


FIGURE 7: Analytical model of waterfalls partial aggregation at node $n_i$.

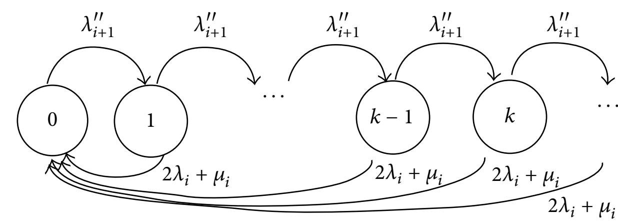


FIGURE 8: State transition of Queue$_{rx}$.

which means $L_{\max}$ is decided by nodes lifetime $L_{i \text{ or } j}$ when two arbitrary nodes $i$ and $j$ are balanced on energy $E_i = E_j$:

$$L_{\max} = L_{i \text{ or } j}\left(E_i = E_j\right) \quad (1 \le i, j < n). \tag{2}$$

Therefore, for achieving balanced energy among nodes, we first determine the optimum set of waterfalls pushing rates $\mu_i$ that correspond to given $L_{\max}$ in the network:

$$\mu_i = \arg\max\left(L\left(\mu_i\right)\right). \tag{3}$$

From the above analysis, it is clear that we are trying to obtain a set of waterfalls pushing rates:

$$\left\{\mu_i \mid \mu_i < \mu_{i+1},\ \mu_i \in |N|,\ 1 < i < n\right\}. \tag{4}$$

For obtaining optimum holding time for each node, it is necessary to know total delay of the network and energy consumption for a node. In the following sections, we determine the total delay and node energy consumption.

*3.4. Analytical Model of WPA*

*3.4.1. Queuing Theory Analysis.* For determining the transmission delay, we model the data arrival process and data transmission process of WPA by queuing theory. The analytical model of node $n_i$ is shown in Figure 7.

Before explaining the model, we introduce Queue$_{rx}$, Queue$_{tx}$, G, and Server. Queue$_{rx}$ denotes the arrival data queue in which arrival data are waiting for data aggregation at node $n_i$. Queue$_{tx}$ denotes the data queue after aggregation in which aggregated data are waiting for transmission to a neighbor node. G is assumed as a virtual gate between Queue$_{rx}$ and Queue$_{tx}$. Data transmitting process of data in Queue$_{tx}$ is accomplished via Server.

In the analytical model, $\lambda''_{i+1}$ is arrival data rate from upper node to Queue$_{rx}$ of node $n_i$ and is approximately abided by Poisson distribution. At node $n_i$, event generation rate $\lambda_i$ is assumed to be Poisson distribution. Waterfalls pushing rate $\mu_i$ is assumed to be exponential distribution. In Queue$_{rx}$, all arrival data and generated data are aggregated and then join Queue$_{tx}$ via gate G. $\lambda'_i$ is data arrival rate to Queue$_{tx}$ in which data are waiting for further transmitting. $\lambda''_i$ is the data rate upon exiting from the Server. Next, we analyze the above model start from data receiving at a node to data that are transmitted to a neighboring node.

*3.4.2. Event Waiting Time.* In this subsection, we decide the time interval that data wait in Queue$_{rx}$ for aggregating at node $n_i$; we call this time interval as event waiting time and denote it by $\tau_e(i)$ and it is counted from when the first data arrive to Queue$_{rx}$ until the data are aggregated and join Queue$_{tx}$. To determine $\tau_e(i)$, we first determine the amount of data waiting for event data in Queue$_{rx}$ at node $n_i$. The amount of data is denoted as $Q_{rx}(i)$ and we describe state transition rate diagram of Queue$_{rx}$ to determine it in Figure 8. Note that, at node $n_i$, the arrival data rate $\lambda''_{i+1}$ is approximately Poisson distribution, and $\lambda_i$ is event data generation rate at node $n_i$ and is approximately Poisson distribution. In data aggregation, service time in queuing theory corresponds to data aggregating time, which is time interval from last data aggregation until next data aggregation. The basic idea of the analysis is that data waits in Queue$_{rx}$ for the duration according to the exponential distribution of average $1/2\lambda_i$; since data aggregation rate is inverse of the mean aggregating time, hence data aggregation rate is $2\lambda_i$, and when considering with waterfalls pushing rate, data aggregating rate is

$$\lambda_{\text{agg}} = 2\lambda_i + \mu_i. \tag{5}$$

In the diagram, the state variable is the number of data waiting for an event. Assume that a number of data are waiting for an event data at node $n_i$ in Queue$_{rx}$. If we let $P_{i,0}$ and $P_{i,k}$ be the probability when the number of data waiting for an event is 0 and $k$ in Queue$_{rx}$ at node $n_i$, at state 0, according to Figure 8, we obtain

$$P_{i,0}\lambda''_{i+1} = \sum_{j=1}^{\infty} P_{i,j}\lambda_{\text{agg}}. \tag{6}$$

At state $k$, we obtain

$$P_{i,k}\lambda''_{i+1} = P_{i,k-1}\lambda''_{i+1} - P_{i,k}\lambda_{\text{agg}}. \tag{7}$$

Hence, we obtain

$$\begin{aligned} P_{i,k} &= \frac{\lambda''_{i+1}}{\lambda''_{i+1} + \lambda_{\text{agg}}} P_{i,k-1} \\ P_{i,k-1} &= \frac{\lambda''_{i+1}}{\lambda''_{i+1} + \lambda_{\text{agg}}} P_{i,k-2} \\ &\vdots \\ P_{i,1} &= \frac{\lambda''_{i+1}}{\lambda''_{i+1} + \lambda_{\text{agg}}} P_{i,0}. \end{aligned} \tag{8}$$

Bringing these equations into (6), we determine the relationship between $P_{i,k}$ and $P_{i,0}$ that

$$P_{i,k} = \left(\frac{\lambda''_{i+1}}{\lambda''_{i+1} + \lambda_{\text{agg}}}\right)^k P_{i,0}. \tag{9}$$

As we know,

$$\sum_{k=1}^{\infty} P_{i,k} = 1. \tag{10}$$

Let

$$\alpha = \frac{\lambda''_{i+1}}{\lambda''_{i+1} + \lambda_{\text{agg}}}. \tag{11}$$

Then we determine

$$\sum_{k=1}^{\infty} \alpha^k P_{i,0} = 1. \tag{12}$$

As

$$\sum_{k=1}^{\infty} \alpha^k = \frac{1}{1-\alpha}, \tag{13}$$

hence,

$$\sum_{k=1}^{\infty} \alpha^k P_{i,0} = \frac{1}{1-\alpha} P_{i,0} = 1. \tag{14}$$

Accordingly, we obtain

$$P_{i,0} = 1 - \alpha. \tag{15}$$

Substituting $\alpha$, we determine

$$P_{i,0} = \frac{2\lambda_i + \mu_i}{\lambda''_{i+1} + \lambda_{\text{agg}}}, \tag{16}$$

$$P_{i,k} = \frac{\left(\lambda''_{i+1}\right)^k + \lambda_{\text{agg}}}{\left(\lambda''_{i+1} + \lambda_{\text{agg}}\right)^{k+1}}. \tag{17}$$

Now, we determine the amount of data that are received in $\text{Queue}_{\text{rx}}(i)$ as

$$Q_{\text{rx}}(i) = \sum_{k=0}^{\infty} k P_{i,k}. \tag{18}$$

Taking (17) into the above equation, we obtain

$$Q_{\text{rx}}(i) = \sum_{k=0}^{\infty} k \frac{\left(\lambda''_{i+1}\right)^k + \lambda_{\text{agg}}}{\left(\lambda''_{i+1} + \lambda_{\text{agg}}\right)^{k+1}}. \tag{19}$$

We express the above equation as

$$Q_{\text{rx}}(i) = \sum_{k=0}^{\infty} k \left(\frac{\lambda''_{i+1}}{\lambda''_{i+1} + \lambda_{\text{agg}}}\right)^{k+1} \frac{\lambda_{\text{agg}}}{\lambda''_{i+1}}. \tag{20}$$

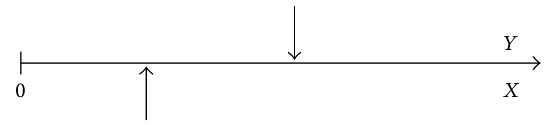


Figure 9: Property distribution of $X, Y$.

Let $\beta = \lambda''_{i+1}/(\lambda''_{i+1} + \lambda_{\text{agg}})$, and $\gamma = \lambda_{\text{agg}}/\lambda''_{i+1}$; then, we simplify (20) as

$$\begin{aligned} Q_{\text{rx}}(i) &= \sum_{k=0}^{\infty} k\beta^{k+1}\gamma = \sum_{k=0}^{\infty} k\beta^{k-1}\beta^2\gamma = \beta^2\gamma \sum_{k=0}^{\infty} k\beta^{k-1} \\ &= \beta^2\gamma \frac{\partial}{\partial\beta} \sum_{k=0}^{\infty} \beta^k = \beta^2\gamma \frac{\partial}{\partial\beta} \frac{1}{1-\beta} = \frac{\beta^2\gamma}{(1-\beta)^2}. \end{aligned} \tag{21}$$

As $\beta = \lambda''_{i+1}/(\lambda''_{i+1} + \lambda_{\text{agg}})$ and $\lambda_{\text{agg}} = 2\lambda_i + \mu_i$, so we get that

$$1 - \beta = \frac{2\lambda_i + \mu_i}{\lambda''_{i+1} + 2\lambda_i + \mu_i}. \tag{22}$$

Therefore, we determine that

$$\begin{aligned} &\frac{\beta^2\gamma}{(1-\beta)^2} \\ &\quad = \frac{\left(\lambda''_{i+1}/\left(\lambda''_{i+1} + 2\lambda_i + \mu_i\right)\right)^2 (2\lambda_i + \mu_i)/\lambda''_{i+1}}{\left((2\lambda_i + \mu_i)/(\lambda''_{i+1} + 2\lambda_i + \mu_i)\right)^2}. \end{aligned} \tag{23}$$

By calculating the above equations, we determine that

$$Q_{\text{rx}}(i) = \frac{\lambda''_{i+1}}{2\lambda_i + \mu_i}. \tag{24}$$

According to Little's formula, we determine the event waiting time as follows:

$$\tau_e(i) = \frac{Q_{\text{rx}}(i)}{2\lambda_i + \mu_i} = \frac{\lambda''_{i+1}}{(2\lambda_i + \mu_i)^2}. \tag{25}$$

*3.4.3. Arrival Process to $Queue_{tx}$.* From the analytical process of $\text{Queue}_{\text{rx}}(i)$, we find that arrival data rate $\lambda'_i$ to $\text{Queue}_{\text{rx}}(i)$ is decided by the waterfalls pushing rate $\mu_i$ and event data generation rate $\lambda_i$ at node $n_i$. The event generation rate $\lambda_i$ and waterfalls pushing rate $\mu_i$ abide by discrete distribution and they are independent of one another. To determine the formulation of $\lambda'_i$, we calculate the property distribution of $\lambda_i$ and $\mu_i$. At first, we define that $\lambda_i$ and $\mu_i$ are independent distribution $X$ and $Y$ as shown in Figure 9. We define the values of $X$ as $1, 2, \ldots, E_X$ and the values of $Y$ as $1, 2, \ldots, E_Y$. Here, we prove that the property $Y$ is bigger than $X$.

We determine the property as follows. $P_X$ and $P_Y$ denote the properties when $X = i$ and $Y = i$:

$$\begin{aligned} P[Y > X] &= P_X(X=1)\,P_Y(Y=2) + P_Y(Y=3) + \cdots + P_Y(Y = E_Y) \\ &+ P_X(X = 2)\{ +P_Y(Y=3) + \cdots + P_Y(Y = E_Y)\} \\ &\vdots \\ &+ P_X(X = E_X)\{ + \cdots + P_Y(Y = E_X + 1)\} \\ &= \sum_{i=1}^{E_X} P_X(X = i)\left\{\sum_{j=i+1}^{E_Y} P_Y(Y = j)\right\}. \end{aligned} \quad (26)$$

When applied to continuous distribution, we obtain

$$P[Y > X] = \int_0^{\infty} P_X(t)\left[\int_0^{\infty} P_Y(z)\,dz\right] dt. \quad (27)$$

Here, if $P_X(t)$ and $P_Y(t)$ are exponential distribution, then

$$\begin{aligned} P_X(t) &= \lambda_X e^{-\lambda_X t}, \\ P_Y(t) &= \lambda_Y e^{-\lambda_Y t}, \\ P[Y > X] &= \int_0^{\infty} P_X(t)\left[\int_0^{\infty} \lambda_Y e^{-\lambda_Y z} dz\right] dt \\ &= \int_0^{\infty} P_X(t)\left[-e^{-\lambda_Y z}\right]_t^{\infty} dt \\ &= \int_0^{\infty} P_X(t)\left(e^{-\lambda_Y t}\right) dt \\ &= \int_0^{\infty} \lambda_X e^{-\lambda_X t} e^{-\lambda_Y t} dt \\ &= \int_0^{\infty} \lambda_X e^{-(\lambda_X + \lambda_Y)t} dt \\ &= \frac{\lambda_X}{\lambda_X + \lambda_Y}\left[-e^{-(\lambda_X+\lambda_Y)t}\right]_0^{\infty} = \frac{\lambda_X}{\lambda_X + \lambda_Y}. \end{aligned} \quad (28)$$

Accordingly, we determine the data arrival rate of $\text{Queue}_{\text{tx}}$:

$$\lambda_i' = \lambda_i + \frac{\lambda_i}{\lambda_i + \mu_i}\mu_i. \quad (29)$$

*3.4.4. Service Process.* Since the data generation rate is Poisson distribution and the waterfalls pushing rate abides by exponential distribution, the data arrival rate to $\text{Queue}_{\text{tx}}(i)$ approximates to Poisson distribution and each node has one server. The ACK packet transmission time is not considered. Data aggregating time is very short and negligible. Therefore, the service time is one-hop data transmission time $\tau$. In our work, data transmission rate is $V_c$ and generated data size is $S_i$. Hence, the service time for generated data is

$$\tau = \frac{S_i}{V_c}. \quad (30)$$

Since $V_c$ and $S_i$ are constant, hence, the service time for data is fixed and constant. According to the above analysis, we can determine that the queuing system on server approximates to $M/D/1$ model. As data transmission time is determined and server utilization ratio is $\eta = \lambda_i' \times \tau_{\text{tx}}(i)$, hence, the average data transmission time $\tau_{\text{tx}}(i)$ is

$$\tau_{\text{tx}}(i) = \frac{\tau}{1 - \lambda_i'\tau}. \quad (31)$$

We figure up the probability density function of the server time by means of Laplace transform in queuing theory and evaluate server waiting time $\tau_s(i)$ that data wait for further transmitting in server:

$$\tau_s(i) = \frac{\lambda_i'(\tau_{\text{tx}}(i))^2}{2(1 - \lambda_i'\tau_{\text{tx}}(i))}. \quad (32)$$

*3.4.5. Channel Waiting Time.* In general queuing system, customers can be served if there are no other customers waiting in front of them. However, when we apply queuing theory to model wireless communication, it is necessary to consider the impact of wireless channel caused by its properties. In wireless sensor networks, because of the overhearing of omnidirectional antenna, node has to wait a period of time if its upper or lower neighbors are transmitting data. Here, we call this time as channel waiting time $\tau_c(i)$ and it is determined as follows:

$$\tau_c(i) = 2 \times \eta \times \tau_{\text{tx}}(i) = 2\lambda_i' \times \tau_{\text{tx}}^2(i). \quad (33)$$

*3.4.6. Total Delay.* Total delay $D_n$ in WPA is the time interval from when an event data is generated at a node until the data are received by a sink node in $N$ hops network. The formulation is as follows:

$$D_n = \sum_{i=1}^{N}\left(\tau_e(i) + \tau_s(i) + \tau_c(i) + \tau_{\text{tx}}(i)\right). \quad (34)$$

*3.4.7. Energy Consumption.* The energy consumption $E_i$ for node $n_i$ is the sum of transmission energy consumption, reception energy consumption, and overhearing energy consumption. Now, we determine the three kinds of energy consumption in the following paragraphs. Note that $P_{\text{tx}}$ and $P_{\text{rx}}$ are energy consumption of transmitting or receiving for each data packet and $E_{\text{tx}}(i)$, $E_{\text{rx}}(i)$, and $E_{\text{oh}}(i)$ are energy consumption of data transmitting, data receiving, and overhearing of node $n_i$ in $D_n$ time. As we are trying to obtain the balanced energy consumption among nodes, hence we calculate energy consumption per $D_n$ time for each node. Therefore, we get the data transmission energy consumption according to Little's formula as follows:

$$E_{\text{tx}}(i) = \lambda_i' D_n P_{\text{tx}}. \quad (35)$$

The reception energy consumption is

$$E_{\mathrm{rx}}(i) = \lambda''_{i+1} D_n P_{\mathrm{rx}}. \tag{36}$$

Here, $\lambda''_{i+1}$ is data arrival rate from upper neighboring node $n_{i+1}$; as the service time is one data packet transmission time, hence the data rate getting out from server is same with data transmission rate, which means

$$\lambda''_{i+1} = \lambda'_{i+1}. \tag{37}$$

The overhearing energy consumption is

$$E_{\mathrm{oh}}(i) = \lambda'_{i-1} D_n P_{\mathrm{rx}}. \tag{38}$$

Here, $\lambda'_{i-1}$ is data transmission rate of node $n_{i-1}$. Hence, we determine total energy consumption $E_i$ for a node in time $D_n$ as

$$E_i = E_{\mathrm{tx}}(i) + E_{\mathrm{rx}}(i) + E_{\mathrm{oh}}(i). \tag{39}$$

## 4. Optimum Network Lifetime

*4.1. Mathematical Solution.* According to the analysis in Section 3.2, the longest network lifetime is decided by the total transmission delay and waterfalls pushing rate vectors. Here, we illustrate the relationship between transmission delay, energy consumption, and network lifetime, so that at last we determine the formulations of a set of waterfalls pushing rates. Accordingly, energy consumption can be written as follows:

$$\begin{aligned} E_i &= \left(\lambda'_i P_{\mathrm{tx}} + \lambda''_{i+1} P_{\mathrm{rx}} + \lambda'_{i-1} P_{\mathrm{rx}}\right) \\ &= \lambda'_i P_{\mathrm{tx}} + \left(\lambda''_{i+1} + \lambda'_{i-1}\right) P_{\mathrm{rx}}. \end{aligned} \tag{40}$$

To balance energy among nodes, we have

$$E_1 = E_2 = \cdots = E_i = \cdots = E_n. \tag{41}$$

As the service time is one-hop data transmitting time, hence

$$\lambda''_i = \lambda'_i. \tag{42}$$

Here, $E_i$ is the energy consumption for node in time $D_n$. Hence, we assume the energy consumption for each unit of time is $E$. We assume that $\lambda_i = \lambda$ and the network in a finite network. Hence, there are no arrival data to the last node in the network so that it does not need a waterfalls pushing rate, and moreover, there is no overhearing for the first node that counted from the sink node. Therefore, for node $n_n$, we have

$$\begin{gathered} \mu_n = 0, \\ \lambda'_n = \lambda, \\ E_{\mathrm{rx}}(n) = 0, \\ E_{\mathrm{oh}}(1) = 0. \end{gathered} \tag{43}$$

Accordingly, we obtain equations as follows:

$$\begin{gathered} \left(\lambda + \frac{\lambda\mu_1}{\lambda+\mu_1}\right) P_{\mathrm{tx}} + \left(\lambda + \frac{\lambda\mu_2}{\lambda+\mu_2}\right) P_{\mathrm{rx}} = E \\ \left(\lambda + \frac{\lambda\mu_2}{\lambda+\mu_2}\right) P_{\mathrm{tx}} + \left(2\lambda + \frac{\lambda\mu_1}{\lambda+\mu_1} + \frac{\lambda\mu_3}{\lambda+\mu_3}\right) P_{\mathrm{rx}} = E \\ \vdots \\ \left(\lambda + \frac{\lambda\mu_i}{\lambda+\mu_i}\right) P_{\mathrm{tx}} \\ + \left(2\lambda + \frac{\lambda\mu_{i-1}}{\lambda+\mu_{i-1}} + \frac{\lambda\mu_{i+1}}{\lambda+\mu_{i+1}}\right) P_{\mathrm{rx}} = E \\ \vdots \\ \lambda P_{\mathrm{tx}} + \left(\lambda + \frac{\lambda\mu_{n-1}}{\lambda+\mu_{n-1}}\right) P_{\mathrm{rx}} = E. \end{gathered} \tag{44}$$

By calculating the above equations, we have

$$\mu_{n-1} = \frac{\lambda P_{\mathrm{rx}}\left(\lambda + \lambda\mu_{n-2}/\left(\lambda+\mu_{n-2}\right)\right)}{\left(\lambda\mu_{n-2}/\left(\lambda+\mu_{n-2}\right)\right) P_{\mathrm{rx}} - \lambda P_{\mathrm{tx}}} \tag{45}$$

$$\mu_{n-2} = \frac{\lambda P_{\mathrm{rx}}\left(\lambda\mu_{n-3}/\left(\lambda+\mu_{n-3}\right)\right) + \lambda\left(P_{\mathrm{rx}} - P_{\mathrm{tx}}\right)\left(\lambda\mu_{n-1}/\left(\lambda+\mu_{n-1}\right)\right)}{\left(P_{\mathrm{rx}} - P_{\mathrm{tx}}\right)\left(\lambda - \lambda\mu_{n-1}/\left(\lambda+\mu_{n-1}\right)\right) - P_{\mathrm{rx}}\left(\lambda\mu_{n-3}/\left(\lambda+\mu_{n-3}\right)\right)} \tag{46}$$

$$\begin{gathered} \vdots \\ \mu_i = \frac{\lambda^2\mu_{i+2}\left(P_{\mathrm{rx}} - P_{\mathrm{tx}}\right)/\left(\lambda+\mu_{i+2}\right) - \lambda^2\mu_{i+1}\left(P_{\mathrm{rx}} - P_{\mathrm{tx}}\right)/\left(\lambda+\mu_{i+1}\right) - \lambda^2\mu_{i+3}P_{\mathrm{rx}}/\left(\lambda+\mu_{i+3}\right)}{\lambda\mu_{i+3}P_{\mathrm{rx}}/\left(\lambda+\mu_{i+3}\right) - \lambda\mu_{i+2}\left(P_{\mathrm{rx}} - P_{\mathrm{tx}}\right)/\left(\lambda+\mu_{i+2}\right) + \lambda\mu_{i+1}\left(P_{\mathrm{rx}} - P_{\mathrm{tx}}\right)/\left(\lambda+\mu_{i+1}\right) - \lambda P_{\mathrm{rx}}} \\ \vdots \\ \mu_1 = \frac{\lambda^2\mu_2\left(P_{\mathrm{rx}} - P_{\mathrm{tx}}\right)/\left(\lambda+\mu_2\right) - \lambda^2\mu_3P_{\mathrm{rx}}/\left(\lambda+\mu_3\right) - \lambda^2 P_{\mathrm{rx}}}{\lambda\mu_2\left(P_{\mathrm{tx}} - P_{\mathrm{rx}}\right)/\left(\lambda+\mu_2\right) - \lambda\mu_3P_{\mathrm{rx}}/\left(\lambda+\mu_3\right) + 2\lambda P_{\mathrm{rx}} - \lambda P_{\mathrm{tx}}}. \end{gathered} \tag{47}$$

Input: $\lambda, \mu_{n-1}, \eta, D_{\text{app}}$
Output: $[\mu_1 : \mu_{n-1}]$
$\mu_{\max} = 2\lambda$
$\mu_{n-1} = \mu_{\max}$
$[\mu_1 : \mu_{n-1}] = f(\mu_{n-i})$
$D_{\text{heu}} = g(\mu_1 : \mu_{n-1})$
While $|D_{\text{app}} - D_{\text{heu}}| > \eta$ do
  $\mu_{n-2} = \mu_{n-1} - \Delta$
  $[\mu_1 : \mu_{n-1}] = f(\mu_{n-i})$
  $D_{\text{heu}} = g(\mu_1 : \mu_{n-1})$
End
Return $[\mu_1 : \mu_{n-1}]$

Algorithm 1: Heuristics algorithm.

From the above equations, we obtained the relationship among waterfalls pushing rate. Equation (45) shows that if $\mu_{n-1}$ is given, then $\mu_{n-2}$ can be determined. In (46), if $\mu_{n-1}$ and $\mu_{n-2}$ are given, then $\mu_{n-3}$ can be determined. In this way, all $\mu_i$ can be determined if $\mu_{n-1}$ is known. To determine the values of $\mu_i$, we designed a heuristics algorithm as follows. In light of real applications of WSN where the threshold of transmission delay is always given, in the heuristics algorithm, we assume that the transmission delay is predefined by application. Hence, we set the value of $\mu_{n-1}$ and then determine the others. However, the value of $\mu_{n-1}$ being too large or too little makes the algorithm too complex to process. Therefore, it is critical to determine the maximum value of $\mu_{n-1}$.

*4.2. Heuristics Algorithm.* In a given application, we assume that data generation rates at all nodes are known and are $\lambda$. Thus, at node $n_{n-1}$, if the arrival data from node $n_n$ just does not keep pace with the generated data, then the node needs waterfalls pushing rate to transmit the arrival data. However, the processing period of waterfalls pushing rate is decided by node $n_{n-1}$; hence, there is again the case that the arrival data just does not catch up the pace of the waterfalls pushing rate $\mu_{n-1}$. Therefore, we determine simply that, for the arrival data, the average catching up rate of waterfalls pushing rate at node $n_{n-1}$ is $\lambda/2$. Accordingly, for transmitting all arrival data, the maximum value of $\mu_{n-1}$ at node $n_{n-1}$ is

$$\mu_{\max} = 2\lambda. \quad (48)$$

Hence, a heuristics algorithm is described as shown in Algorithm 1.

## 5. Evaluation

*5.1. Validation of the Queuing Theory Model.* We implement WPA in the original network simulator written in C++ language. Analytic results in the previous section are shown as well as simulated result. The parameters are shown in Table 1.

In the simulation, data occurs at each node randomly and independently. Buffer size of each node is infinite. Although analytic model assumes that transmission error is negligible, transmission errors and retransmission may occur in the simulation. We set data generation rate $\lambda = 5$ and waterfalls pushing rate as $\mu_i = \{1, 2, 3, 4, 5\}$ for 5-hop tandem network.

Table 1: Simulation parameters of WPA.

| | |
|---|---|
| Node distance | 10 (m) |
| Transmission range | 11 (m) |
| Transmission rate | 250 (kbps) |
| Data size | 4096 (bit) |
| Current consumption for transmission | 17.4 (mA) |
| Current consumption for reception | 19.7 (mA) |
| MAC | CSMA/CA |
| Routing | DSR |

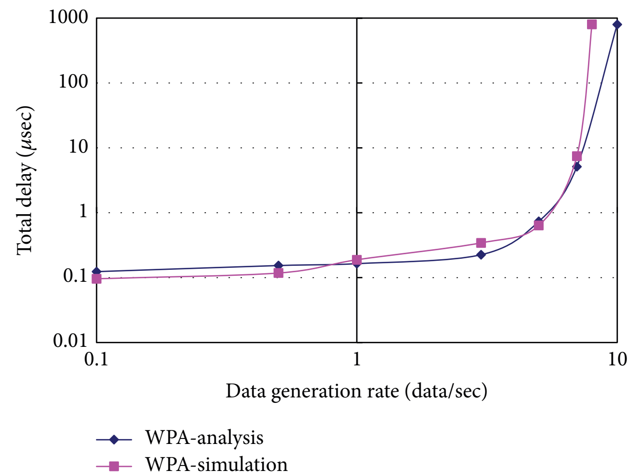


Figure 10: Total delay of WPA.

From Figures 10 and 11, we find that our analytical model meets with simulation results, so that we basically confirm the correctness of queuing theory model of WPA.

*5.2. Evaluation of Optimum Network Lifetime.* In this section, we evaluate the effectiveness of the proposed waterfalls partial aggregation. Here, we consider network environment as nature monitoring network, so that we obtain the maximum network lifetime, which is the same with network lifetime of full aggregation. We set $E$ in (44) which is equal to $E_{\text{ful-largest}}$, where $E_{\text{ful-largest}}$ denotes the energy consumption of the node that consumes most energy in the network. The optimal results can be achieved by numerical packets; however, the mathematical calculation results are too complex. For more understanding of our algorithm, we obtain the pushing rate vectors applying Excel when the event generation rate is $\lambda = 1$ for 5-hop tandem network. Nodes are counted from the sink node, which means that the nearest node with the sink is node $n_1$ and its waterfalls pushing rate is $\mu_1$. The optimum waterfalls pushing rates are shown in Table 2 when WPA has the same network lifetime with full aggregation. The analytical model of nonaggregation and full aggregation is realized by queuing theory; however, for the brevity of the paper, we omitted the derivation process.

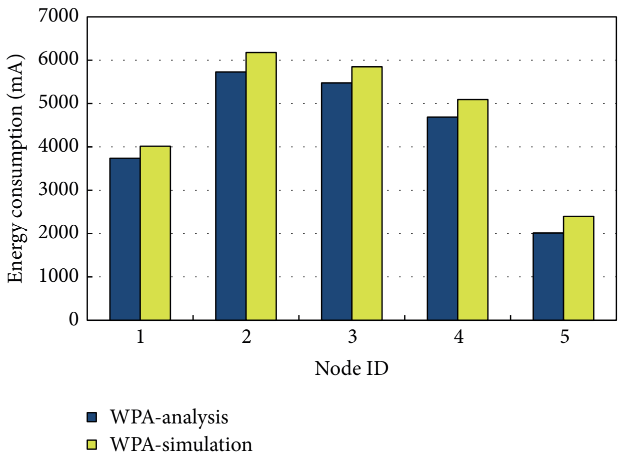


FIGURE 11: Node energy consumption of WPA.

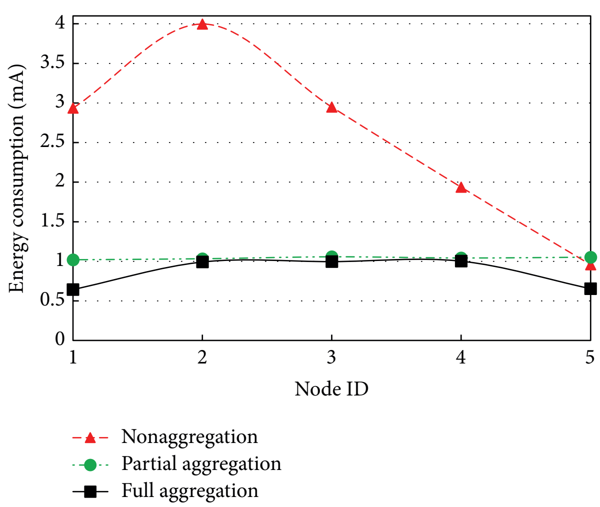


FIGURE 12: Lifetime of nodes.

TABLE 2: Waterfalls pushing rate for optimum network lifetime.

| Waterfalls pushing rate | Values |
|---|---|
| $\mu_1$ | 0.02 |
| $\mu_2$ | 0.04 |
| $\mu_3$ | 0.08 |
| $\mu_4$ | 0.1 |
| $\mu_5$ | 0.5 |

Figure 12 shows the lifetime of all nodes in the network where we set the waterfalls pushing rate of nodes as the same with Table 2. From the figure, we find that all nodes almost have the same lifetime in WPA. In this case, we can say that the node energy is utilized optimally in WPA. In full aggregation, the second and the third node consumed most energy and the other nodes have less energy consumption which results in the suboptimum utilization of energy. The nonaggregation has very large energy consumption at the second node, which results in the short lifetime of the entire network. We find from the figure that nonaggregation and full aggregation have the largest energy consumption at their second node. The reason is that the first node has no adjacent upper node, which means that it does not consume overhearing energy compared with others.

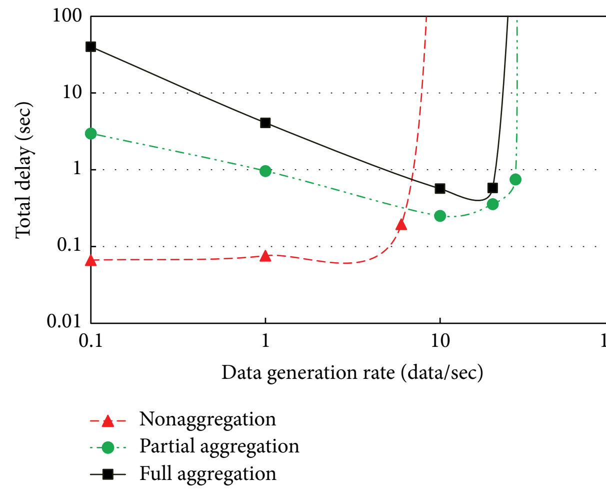


FIGURE 13: Transmission delay of network.

Figure 13 shows the corresponding total delay of the network where we set the same waterfalls pushing rates with Table 1. From the figure, we find that when it has the same network lifetime with full aggregation, WPA can shorten the transmission delay by considerable amount. When data generation rate becomes larger (larger than 6 types of data/sec), transmission delay of nonaggregation gets longer rapidly due to data congestion at nodes. However, in WPA, the rapid increase of transmission delay arises later (at data generation rate of 30 types of data/sec) than nonaggregation and full aggregation on data generation line, which indicates that WPA can relieve data congestion at nodes. This is because, with waterfalls pushing rates, proposed WPA can adjust the number of data transmissions. Moreover, we find from the figure that with the given waterfalls pushing rates there is minimum transmission delay and we can determine it in WPA.

*5.3. Summary of Evaluation.* Firstly, from the presentation of Section 5.1, we conclude that our analytical model of WPA meets with simulation results; the diversity of energy consumption is that we did not consider retransmission in analytical model and considered it in simulation. Secondly, in Section 5.2, compared to nonaggregation and full aggregation, proposed WPA has the superiorities as follows: nonaggregation consumes much energy and data congestion occurs easily when data generation rate is larger. In WPA, after data are sent to a neighboring node, the data have to be aggregated together or aggregated with local data for further transmitting; hence, WPA is more energy efficient than nonaggregation. When data generation rate is larger at a node, data congestion occurs at nonaggregation, which leads to long transmission delay; however, proposed WPA can relieve data congestion via reducing the number of data transmissions by aggregating received data at nodes. On the other hand, full aggregation can save much energy consumption and suffer long transmission delay. Compared to full aggregation, proposed WPA has similar energy consumption and is superior in balancing energy consumption among nodes. Moreover, with waterfalls pushing rates, WPA can control data waiting time more reasonably and efficiently than full aggregation at nodes, so that transmission delay of WPA is superior to full aggregation.

## 6. Conclusion

In this paper, we proposed a waterfalls partial aggregation that can achieve optimum network lifetime in WSN via applying data aggregation. At first, we modeled WPA in queuing theory and then determined transmission delay of the network and energy consumption of a node. Then, we continued to model balanced energy consumption among nodes. The evaluation parts show the correctness of our analytical model and demonstrate the superiorities of proposed WPA.

## Competing Interests

The authors declare no competing financial interests.

## Acknowledgments

This work is supported in part by the Natural Science Foundation of Inner Mongolia Autonomous Region of China under Grant no. 2014BS0603 and no. 2014BS0601 and the Program of Higher-Level Talents of Inner Mongolia University under Grant no. 30105-135134.

## References

[1] C.-C. Shen, C. Srisathapornphat, and C. Jaikaeo, “Sensor information networking architecture and applications,” *IEEE Personal Communications*, vol. 8, no. 4, pp. 52–59, 2001.

[2] S. D. Glaser, “Some real-world applications of wireless sensor nodes,” in *Smart Structures and Materials 2004: Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems*, vol. 5391 of *Proceedings of SPIE*, pp. 344–355, San Diego, Calif, USA, March 2004.

[3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey,” *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[4] W. Ye, J. Heidemann, and D. Estrin, “Sensor-MAC (S-MAC): medium access control for wireless sensor networks,” in *Proceedings of the 21st International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM ’02)*, New York, NY, USA, June 2002.

[5] L. J. G. Villalba, A. L. S. Orozco, A. T. Cabrera, and C. J. B. Abbas, “Routing protocols in wireless sensor networks,” *Sensors*, vol. 9, no. 11, pp. 8399–8421, 2009.

[6] P. Santi, “Topology control in wireless ad hoc and sensor networks,” *ACM Computing Surveys*, vol. 37, no. 2, pp. 164–194, 2005.

[7] R. Rajagopalan and P. K. Varshney, “Data-aggregation techniques in sensor networks: a survey,” *IEEE Communications Surveys and Tutorials*, vol. 8, no. 4, pp. 48–63, 2006.

[8] W.-S. Jung, K.-W. Lim, Y.-B. Ko, and S.-J. Park, “A hybrid approach for clustering-based data aggregation in wireless sensor networks,” in *Proceedings of the 3rd International Conference on Digital Society (ICDS ’09)*, pp. 112–117, IEEE, Cancun, Mexico, February 2009.

[9] M. Kumar, S. Verma, and P. S. Pratap, “Clustering approach to data aggregation in wireless sensor networks,” in *Proceedings of the 16th IEEE International Conference on Networks*, pp. 1–6, IEEE, New Delhi, India, December 2008.

[10] S. Lindsey, C. Raghavendra, and K. M. Sivalingam, “Data gathering algorithms in sensor networks using energy metrics,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, no. 9, pp. 924–935, 2002.

[11] J. W. Baek, Y. J. Nam, and D.-W. Seo, “ATS-DA: adaptive timeout scheduling for data aggregation in wireless sensor networks,” in *Proceedings of the International Conference on Information Networking*, pp. 375–384, Estoril, Portugal, 2008.

[12] H. Chen, H. Mineno, and T. Mizuno, “Adaptive data aggregation scheme in clustered wireless sensor networks,” *Computer Communications*, vol. 31, no. 15, pp. 3579–3585, 2008.

[13] I. Solis and K. Obraczka, “In-network aggregation trade-offs for data collection in wireless sensor networks,” *International Journal of Sensor Networks*, vol. 1, no. 3-4, pp. 200–212, 2006.

[14] F. Mirian and M. Sabaei, “A delay and accuracy sensitive data aggregation structure in wireless sensor networks,” in *Proceedings of the International Conference on Information Management and Engineering (ICIME ’09)*, pp. 231–235, IEEE, Kuala Lumpur, Malaysia, April 2009.

[15] P. Sun, X. Zhang, Z. Dong et al., “A novel energy efficient wireless sensor MAC protocol,” in *Proceedings of the 4th International Conference on Networked Computing and Advanced Information Management*, pp. 68–72, Gyeongju, South Korea, September 2008.

[16] N. Li, J. C. Hou, and L. Sha, “Design and analysis of an mst-based topology control algorithm,” in *Proceedings of the IEEE Societies 22nd Annual Joint Conference of the IEEE Computer and Communications (INFOCOM ’03)*, vol. 3, pp. 1702–1712, San Francisco, Calif, USA, April 2003.

[17] B. Yin, H. Shi, and Y. Shang, “A two-level topology control strategy for energy efficiency in wireless sensor networks,” *International Journal of Wireless and Mobile Computing*, vol. 4, no. 1, pp. 41–49, 2010.

[18] A. Sharif, V. M. Potdar, and A. J. D. Rathnayaka, “LCART: lightweight congestion aware reliable transport protocol for WSN targeting hetero-geneous traffic,” in *Proceedings of the 17th International Conference on Neural Information Processing (ICONIP ’10)*, Sydney, Australia, November 2010.

[19] B. Yu, J. Li, and Y. Li, “Distributed data aggregation scheduling in wireless sensor networks,” in *Proceedings of the 28th Conference on Computer Communications (IEEE INFOCOM ’09)*, pp. 2159–2167, Rio de Janeiro, Brazil, April 2009.

[20] M. X. Cheng and L. Yin, “Energy-efficient data gathering algorithm in sensor networks with partial aggregation,” *International Journal of Sensor Networks*, vol. 4, no. 1-2, pp. 48–54, 2008.

[21] A. A. Allen, *Probability, Statistics, and Queueing Theory with Computer Science Applications*, Gulf Professional Publishing, 1990.