Research Article

Performance Analysis of Flow-Based Traffic Splitting Strategy on Cluster-Mesh Sensor Networks

Huimin She,[1] Zhonghai Lu,[1] Axel Jantsch,[1] Dian Zhou,[2] and Li-Rong Zheng[1]

[1] School of Information and Communications Technology, KTH Royal Institute of Technology, 16440 Stockholm, Sweden
[2] School of Microelectronics, Fudan University, Shanghai 201203, China

Correspondence should be addressed to Huimin She, huimin@kth.se

Performance analysis is crucial for designing predictable and cost-efficient sensor networks. Based on the network calculus theory, we propose a flow-based traffic splitting strategy and its analytical method for worst-case performance analysis on cluster-mesh sensor networks. The traffic splitting strategy can be used to alleviate the problem of uneven network traffic load. The analytical method is able to derive close-form formulas for the worst-case performance in terms of the end-to-end least upper delay bounds for individual flows, the least upper backlog bounds, and power consumptions for individual nodes. Numerical results and simulations are conducted to show benefits of the splitting strategy as well as validate the analytical method. The numerical results show that the splitting strategy enables much better balance on network traffic load and power consumption. Moreover, the simulation results verify that the theoretic bounds are fairly tight.

1. Introduction

With the advances of wireless communications and microelectronics technologies, wireless sensor networks (WSNs) have received more and more attention over the last decades due to their potential in a variety of application domains [1–4], such as environment monitoring, human activity tracking, healthcare, and military assistance, and so forth.

A typical sensor network consists of a larger number of sensor nodes that are capable of sensing the environment and forwarding their observation values to a fusion center (sink) through multihop wireless links. Thus, the traffic pattern in sensor networks is usually in a many-to-one manner. The nodes near the sink may need to forward more data and thus require more energy consumptions and bigger buffers than the nodes far away. Consequently, the distributions of energy and buffer requirements in WSNs are extremely uneven. However, energy supplies and buffers are limited and expensive resources in typical WSNs, since sensor nodes are usually made as tiny devices with limited buffers and equipped with batteries that may not be convenient or economical for replacement. One way to address these challenges is applying traffic splitting strategies which have been adopted by many researchers for load balancing in communication networks [5, 6]. With traffic splitting, a main flow is divided into several subflows and forwarded to the destination through different routing paths. By distributing traffic over the network, the overall network load balance can be improved. It is shown in [7] that the spare capacity can be reduced and thus the overall performance of the system can be improved by splitting traffic across multiple disjoint paths.

One popular application of WSNs is real-time monitoring and tracking, such as logistic chain tracking [4] and healthcare application [8]. In such kind of applications, it is crucial to ensure sensor data delivered to the sink within time constraints so that appropriate actions can be made. In order to design a WSN with predictable delay, backlog, and energy consumptions, formal performance analysis is desired for analyzing a sensor network before its actual deployment. While simulation-based methods can offer high accuracy, it can be very time-consuming and tedious to find the worst-case performance. Each simulation run may take considerable time and evaluates only a single network configuration, traffic pattern, and load point. Hence, formal methods are desired to dimension sensor networks in an analytical way rather than case-by-case simulations. Starting with the initial

work by Cruz [9, 10], *network calculus* has been developed as a useful tool for the performance analysis of networked system [11]. In contrast to queueing theory, network calculus deals with performance bounds, such as worst-case delay and backlog bounds, rather than average values. It has been applied to sensor networks by many researchers recently [12–15].

In this paper, we propose a flow-based traffic splitting strategy and its analytical method for worst-case performance analysis on cluster-mesh sensor networks based on the network calculus theory. We introduce the flow-based traffic splitting strategy, which is useful in balancing not only network load but also power consumption. Aiming to evaluate the worst-case performance in terms of end-to-end least upper delay bound, least upper backlog bound, and power consumption, a splitting model is built for a single-node analysis and an analytical method is proposed for the network analysis. Through an example, we show that the performance analysis method is able to derive closed-form formulas of these bounds. The numerical results indicate that the backlog and power consumption can be balanced by applying the traffic splitting strategy. In addition, simulations are performed to validate the performance bounds of our analytical method. The results show that their tightnesses are satisfactory.

The rest of this paper is organized as follows. Section 2 introduces related work. Section 3 contains preliminaries including the cluster-mesh network topology and power consumption model, and basics of network calculus. Section 4 includes performance analysis of the flow-based traffic splitting strategy. An analysis example is given in Section 5. Numerical results and simulations are presented in Section 6. Finally, conclusions and directions for future work are given in Section 7.

2. Related Work

In general packet switching networks, network calculus provides methods to deterministically reason about timing properties and resource requirements. Based on the powerful abstraction of *arrival curve* for traffic flows and *service curve* for network elements, it allows computing the worst-case delay and backlog bounds. Systematic accounts of network calculus can be found in books [11, 16].

Network calculus has been extremely successful for ensuring performance bounds when applied to ATM, Internet, and other networks. It is recently extended and applied for performance analysis and resource dimensioning of WSNs by several researchers. In [12], Schmitt and Roedig firstly applied network calculus to sensor network and proposed a generic framework for performance analysis of WSNs with various traffic patterns. They further extended the general framework to incorporate computational resources besides the communication aspects of WSNs [14]. In [13], Kouba et al. proposed a methodology for the modeling and worst-case dimensioning of cluster-tree sensor networks. They derived plug-and-play expressions for the end-to-end delay bounds, buffering, and bandwidth requirements as a function of the WSN cluster-tree and traffic characteristics. Lenzini et al.

[17] proposed a method for deriving tight end-to-end *least upper delay bounds* in sink-tree networks. The least upper delay bound is defined as the minimum value of the upper delay bound. In [15], the authors presented a method for computing the worst-case delays, buffering, and bandwidth requirements while assuming that the sink node can be mobile.

Traffic splitting strategies have several common features with multipath routing protocols. There have been plenty of research works on multipath routing and traffic splitting for sensor networks [18–23]. The authors in [18] proposed a multipath routing scheme that finds several disjoint paths. In this scheme, the source node or an intermediate node chooses one path from the available paths to deliver the data to sink based on the performance requirements such as delay and throughput. An energy efficient multipath routing protocol for WSNs with multiple sinks is presented in [19]. The path construction is implemented by the source node sending route messages to its neighbors. Traffic is distributed over the multiple paths according to a load balancing algorithm. The results show that the proposed scheme results in a higher energy efficiency. In [20], authors proposed an N-to-1 multipath routing protocol, in which nodes are arranged in a spanning tree. Multipaths are constructed by traversing the tree. The multipath scheme is a combination of end-to-end multipath traffic dispersion and per-hop alternate path salvaging. Zou et al. [21] studied the interplay between data aggregation and flow splitting in WSNs and proposed a flow-based scheme. The flows are preserved until the aggregation point. The aggregated data is splitted into multiple flows on the rest of the path to the destination. The results show that the scheme can balance energy consumption and therefore prolong the lifetime of WSNs. In [22], the authors investigated a joint coding/routing optimization of network costs and capacity in WSNs. By combining the link rate allocation and network coding-based multipath routing, the total energy consumption of encoding power, transmission power and reception power can be reduced. A backpressure collection protocol (BCP) for sensor networks is presented in [23]. In this protocol, routing and forwarding decisions are made based on a per-packet basis. By using ETX optimization and floating LIFO queues, BCP is capable of improving throughput and delivery performance under static and dynamic settings, respectively.

This work applies the network calculus theory for analytical performance evaluation of a traffic splitting strategy on cluster-mesh sensor networks. Our work differs from others' works and contributes to state of the art in the following aspects. First, we address the particular problem of deriving performance bounds and resource requirements for a traffic splitting strategy on the cluster-mesh topology, which we believe are of great interest for time-sensitive WSN applications. Second, we introduce a flow-based traffic splitting strategy that can be used to balance traffic load in the network. We define a splitting model and set up an equivalent packet delivery model for the original network. Based on these models, the end-to-end least upper delay bounds and backlog bounds are derived. The results show that the variance of backlogs can be greatly reduced by applying the traffic

splitting strategy, which indicates better load balancing. Third, we conduct power consumption analysis, which is crucial for most applications of WSNs. The results indicate that the traffic splitting strategy also enables better balance on power consumptions. Since most applications of WSNs involve sensors with unreplaceable power supplies, better power balance would lead to longer lifetime of the whole network. On the other hand, even if the batteries of sensors can be recharged or replaced, better power balance would bring about less labor for recharging or replacing and thus can reduce the overall deployment costs. In addition, although our work focuses on performance analysis of a traffic splitting strategy on a particular network topology, we believe the intrinsic idea of the method is also very useful for analyzing sensor networks with other topologies and traffic planning policies.

We have described the traffic splitting scheme in our previous work [24, 25], from which we borrow many notations used in this paper. In this work, we have significantly extended and enhanced the previous work in the following aspects. First, in previous work, we only presented the analysis method without simulations. The analytical method in this work is validated through simulations which also prove the tightness of the delay and backlog bounds. Second, the end-to-end delay bound in [24, 25] is calculated by summing up the per-hop delay together. While, in this paper, the end-to-end delay bound is computed using the end-to-end equivalent service curve, the later method can get tighter bounds. A comparison between these two methods is shown in Figure 20. Third, we integrate power consumption analysis in this work, which we believe is of great importance for WSNs.

3. Preliminaries

This section presents system models, including the cluster-mesh network topology and power consumption model.

3.1. The Cluster-Mesh Topology. A wireless sensor network may consist of a large number of sensors that are densely deployed either inside the phenomenon of interest or close to it. These sensors can be organized in various topologies, such as mesh- and cluster-based topologies. The mesh networking has advantages like supporting path diversity which enables better balance on traffic load and energy consumption [26]. The cluster-based topologies are also quite suitable for WSNs with demanding requirements in terms of Quality of Service (QoS) support and real-time communications [13]. Considering these aspects, we adopt the *cluster-mesh* topology that merges advantages of mesh and cluster [3, 27]. It is a two-layered architecture with the mesh defining a backbone that consists of a set of *cluster heads* (CHs). A cluster is formed by grouping a number of sensors within a geographic neighborhood. We define the network composed by cluster heads and the sink as the layer-1 network and the network inside a cluster as the layer-2 network.

In summary, the cluster-mesh network contains three types of nodes: *sink*, *cluster head*, and *sensor*. Like in most
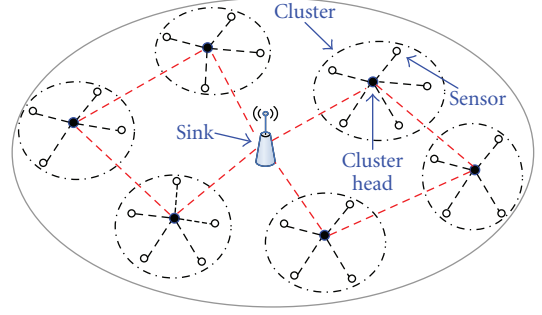

FIGURE 1: A cluster-mesh sensor network.

sensor networks, the sink is responsible for controlling the network and collecting data from all the other nodes. A cluster head and multiple sensors form a cluster. In order to reduce the cost and complexity, sensors do not communicate with each other and data generated by them is collected by their cluster head and delivered to the sink through neighbor cluster heads. For simplicity and conciseness, we consider cluster heads are static and they do not sense the environment and generate input data. However, this assumption can be easily relaxed, and the subsequent analysis is straightforward. In the mesh network composed by cluster heads and the sink, links are considered bidirectional.

Figure 1 shows an example of the cluster-mesh topology. A cluster-mesh network is a mesh network where each cluster head and its connected sensors form their own logical cluster. The layer-1 network can be modeled as a direct graph $G(\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ is the set of all sensor nodes and the sink and $\mathcal{L}$ is the set of all direct links in the network. In this paper, our work concentrates on analyzing the layer-1 network.

3.2. Power Consumption Model. In most types of sensor nodes, the energy consumption is mainly contributed by the transmitter, receiver, and computation module [28]. We consider the application scenario of sensor networks for fresh food monitoring in warehouses. In this scenario, sensors may perform tasks and send packets periodically. Consequently, the power consumption of the computation module can be considered as nearly constant denoted by p^c. Let ϵ^r denote the energy consumption of the receiver electronics for receiving one bit data. In practical applications, the power consumption of the receiving is usually stable [28]. So ϵ^r can be considered as a constant. According to the results in [29], the energy required to transmit a given amount of data is a convex and monotonically increasing function of the transmission rate, that is, the energy per bit can be expressed as (Figure 2)

$$\epsilon^t = \frac{N_0}{RG}\left(2^{R/\eta W} - 1\right), \tag{1}$$

where R is the transmission rate, W is the channel bandwidth, G is the channel gain, N_0 is the noise power, and $\eta \in (0, 1)$ is the probability that the information can be reliably transmitted at a given transmission rate. (In an optimal
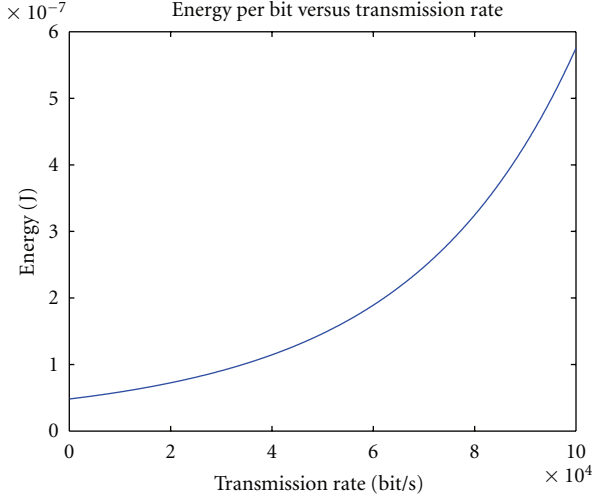
$\times 10^{-7}$

Energy per bit versus transmission rate

FIGURE 2: Energy per bit versus transmission rate [29]: $W = 20\,\text{kHz}$, $N_0 = -100\,\text{dB}$, $G_0 = -50\,\text{dB}$, $d = 5\,\text{m}$, $\theta = 3$.

channel coding scheme, a transmission rate $R = \eta C$ can be guaranteed for any $0 < \eta < 1$, where $C = W \log_2(1 + GP/N_0)$ is the Shannon capacity [29].) We adopt a simplified model for channel gain, that is, $G = G_0(d/d_0)$ (Section 2.6 of [30]), where θ is the path loss exponent ($2 \le \theta \le 4$), d is the distance between the transmitter and receiver, d_0 is a reference distance where the reference channel gain G_0 is measured.

Therefore, the total power consumption P_i of a node i can be expressed as

$$P_i = \epsilon^r \sum_{k \in N_{\text{in}}(i)} \rho_{ki} + \sum_{j \in N_{\text{out}}(i)} \epsilon^t_{ij} \rho_{ij} + p^c$$

$$= \epsilon^r \sum_{k \in N_{\text{in}}(i)} \rho_{ki} + \sum_{j \in N_{\text{out}}(i)} \frac{\rho_{ij} N_0}{R_{ij} G_{ij}} \left(2^{R_{ij}/\eta W} - 1 \right) + p^c, \tag{2}$$

where ρ_{ki} and ρ_{ij} denote the data rates on link ki and ij, respectively, R_{ij} denotes the transmission rate of node i sending data to node j, $G_{ij} = G_0(d_{ij}/d_0)^{-\theta}$ denotes the channel gain between node i and j, $N_{\text{in}}(i)$ denotes the set of nodes which are the direct sources of incoming data flows of node i, and $N_{\text{out}}(i)$ denotes the set of nodes which are the direct destinations of output data flows of node i.

3.3. Traffic Model and Service Model. As stated in the previous section, sensor nodes inside a cluster generate input data and then send them to their cluster head. A traffic *flow* is defined as an infinite stream of data from a source to a destination. Following network calculus, we model the input flow at a cluster head using its cumulative traffic $F(t)$, defined as the number of bits coming from the flow in time interval $[0, t]$. Furthermore, we use a wide-sense increasing function $\alpha(t)$ to constrain this cumulative traffic flow $F(t)$, defined as

$$F(t) - F(s) \le \alpha(t - s), \quad \forall t \ge 0, \ t \ge s, \tag{3}$$

where $\alpha(t)$ is called the arrival curve of the input flow $F(t)$ [11]. *Affine arrival curve* is one of the most commonly used

arrival curves, which has been adopted in many works [12–15]. The application scenario of this work is real-time monitoring, and the sensor nodes sense the environment and send packets periodically to the cluster heads. Therefore, the affine arrival curve model can be used to abstract the input traffic of cluster heads, defined as $\alpha(t) = \gamma_{\sigma, \rho} = \rho \cdot t + \sigma$, where σ and ρ represent the burst tolerance (in bits) and the average data rate (in bits/s), respectively. Figure 3(a) shows examples of a periodic cumulative flow $F(t)$ and an affine arrival curve $\alpha(t)$.

Service curve is an abstraction to model the processing capability of a node, depending on link layer characteristics, such as transmission rate, channel characteristics, and packet scheduling. The node and the channel together are modeled as a network element which provides a service curve β^s to the input flows. If the node forwards packets with the rate R (bits/s) and delays packets for $T(s)$ at maximum due to scheduling and queuing, it can be modeled by a *rate-latency* service curve [11] that consists of two components: a *rate service curve* and a *latency service curve*. The rate-latency service curve can be formally defined as $\beta(t) = \chi_{R,T} = R[t - T]^+$, where notation $[x]^+$ denotes $\max\{0, x\}$.

In wireless networks, data transmission over wireless channels is usually unreliable due to their inherent uncertainties. The actual transmission rate and success probability are influenced by the transmission power, path loss, noise power, and interference. In spite of these uncertainties, deterministic network calculus can still be useful in modeling wireless networks by making reasonable assumptions and abstractions. First, the uncertainties in some applications of WSNs are low. An example scenario for which our framework suits well is the process monitoring and tracking in logistics systems [4]. Second, the link unreliability and data loss rate can be mitigated by applying high transmission powers, especially for the cases with small distances between a transmitter and a receiver. Third, the interference between adjacent nodes can be alleviated by using appropriate MAC layer protocols. There are plenty of research works on designing TDMA-based link protocols which can create collision-free slot schedules [31, 32].

Based on these assumptions about link reliability and interference, we can abstract and approximate the service capability of a node by a deterministic service curve with the idea of effective transmission rate. From information theory, the Shannon capacity of a wireless channel can be expressed as $C = W \log_2(1 + GP/N_0)$, where W, G, N_0 are the same as those in (1). The service rate is defined as the rate that the information can be reliably transmitted. With an optimal or suboptimal channel coding scheme, a service rate of $R = \eta \cdot C$ can be guaranteed for any $0 < \eta < 1$ [29]. R defines a lower bound on the transmission rate. Therefore, the rate-latency service curve $\beta(t) = \chi_{R,T} = R[t - T]^+$ can be applied to model the service capability of a wireless channel, where T denotes the maximum possible processing/queuing delay.

3.4. Delay, Backlog, and Output Bounds. Given the arrival curve and service curve of a node, the least upper delay bound, least upper backlog bound, and output bounds can be derived according to network calculus [11]. The least
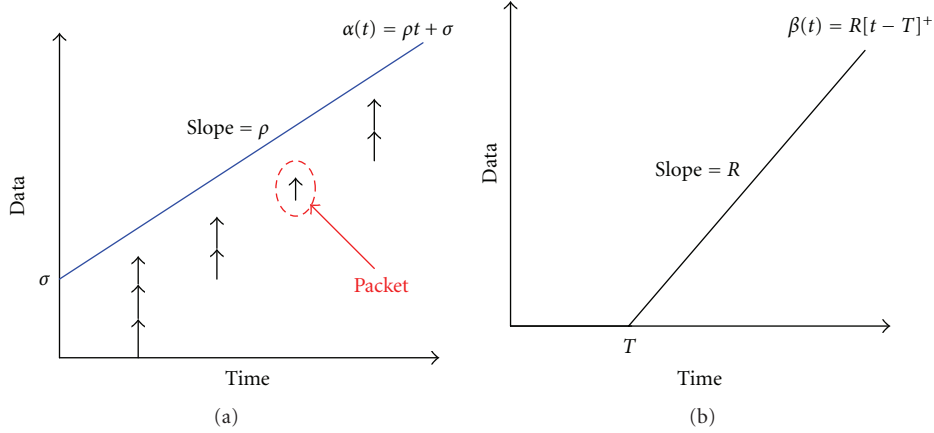
FIGURE 3: (a) An affine arrival curve: the arrows show the packet generation process. (b) A rate-latency service curve.

upper backlog bound is defined as the minimum value of the upper backlog bound. Consider a node i provides a service curve β_i^s to the input flow which is constrained by an arrival curve α_i. According to network calculus, the least upper delay bound of the flow can be computed by

$$D_i = h(\alpha_i, \beta_i^s) = \sup_{t \geq 0} \left\{ \inf_{\tau \geq 0} \{ \tau : \alpha_i(t) \leq \beta_i^s(t + \tau) \} \right\}. \quad (4)$$

Moreover, the least upper backlog bound of node i can be calculated by

$$B_i = v(\alpha_i, \beta_i^s) = \sup_{t \geq 0} \{ \alpha_i(t) - \beta_i^s(t) \}. \quad (5)$$

Additionally, the arrival curve of the departure flow can be derived by

$$\tilde{\alpha}_i = \alpha_i \oslash \beta_i^s = \sup_{\tau \geq 0} \{ \alpha_i(t + \tau) - \beta_i^s(\tau) \}. \quad (6)$$

4. Analysis of the Flow-Based Traffic Splitting Strategy

In this section, we first introduce the splitting and multiplexing models. Then, the formal performance analysis procedure is presented. After that, we discuss the scope and assumptions of our analysis approach.

4.1. The Splitting Model. To analyze the splitting strategy, we build a splitting model that identifies the relations of input, output, delay, and backlog for a single node. Without losing generality, we consider a main flow is split into two subflows. The node f_1 traverse is abstracted as the combination of a buffer plus a *splitter* depicted in Figure 4(a).

We consider that the node performs a weighted proportional splitting scheme, in which the main flow is split according to the configured weights, ϕ_i for subflow i. In each round, the splitter will try to forward ϕ_i packets to output link i before moving to the next one. The values of ϕ_i can be set either according to a predefined rule or randomly. Increasing the value of ϕ_i can result in increased packets to

output link i. By adjusting ϕ_i, the amount of traffic over each link can be controlled. If the service rate is R bits/s, the maximum length of a round is consequently equal to $\sum_i \phi_i l/R$ seconds and the time for packets of subflow i to be forwarded within a round is bounded by $\phi_i l/R$ seconds, where l is the packet length. In the weighted proportional splitting scheme, the worst case appears when the packets of a subflow just misses its turn in the current round. Consequently, it will have to wait for its turn at the next round. In the worst case, packets of the subflow i have to wait up to $\sum_{i \neq j} \phi_j l/R$ seconds to be served.

Consider a main flow f_1 that is upper constrained by arrival curve $\alpha_1 = \gamma_{\sigma_1, \rho_1}$, be split into two subflows $f_{1.1}$ and $f_{1.2}$ according to weights ϕ_1 and ϕ_2, where σ_1 and ρ_1 denote the burstiness and average data rate of f_1, respectively. Burstiness is defined as the amount of data inputted/outputted to/from a system or a node at one time. Consequently, it should be equal or bigger than the packet size l. Let $\alpha_{1.1} = \gamma_{\sigma_{1.1}, \rho_{1.1}}$ and $\alpha_{1.2} = \gamma_{\sigma_{1.2}, \rho_{1.2}}$ denote the arrival curves of $f_{1.1}$ and $f_{1.2}$, respectively. Then, we have

$$\rho_{1.1} = \frac{\phi_1}{\phi_1 + \phi_2} \rho_1, \qquad \sigma_{1.1} = \max\left(\left\lceil \frac{\phi_1 \sigma_1}{(\phi_1 + \phi_2) l} \right\rceil \cdot l, l \right),$$

$$\rho_{1.2} = \frac{\phi_2}{\phi_1 + \phi_2} \rho_1, \qquad \sigma_{1.2} = \max\left(\left\lceil \frac{\phi_2 \sigma_1}{(\phi_1 + \phi_2) l} \right\rceil \cdot l, l \right),$$

$$(7)$$

where $\lceil \cdot \rceil$ denotes the minimum integer equal to or bigger than the number inside.

Let the splitter provide a service curve $\beta^s = \chi_{R,T}$. Since the splitter serves a subflow at one time, the service rate for each subflow also equals R. Therefore, the equivalent service curve for subflow $f_{1.1}$ can be derived by

$$\hat{\beta}_1^s = \beta^s \otimes \delta_{\phi_2 l/R} = \chi_{R, T + \phi_2 l/R}. \quad (8)$$

Analogously, the equivalent service curve for $f_{2.2}$ is

$$\hat{\beta}_2^s = \beta^s \otimes \delta_{\phi_1 l/R} = \chi_{R, T + \phi_1 l/R}. \quad (9)$$

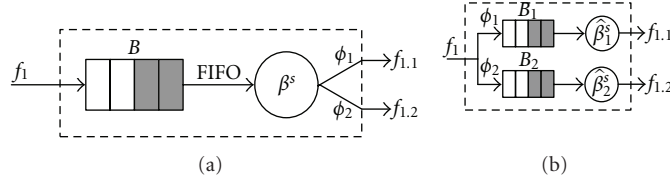FIGURE 4: (a) The main flow f_1 is split into two subflows $f_{1.1}$ and $f_{1.2}$. (b) The equivalent model.

Furthermore, the equivalent bounds on backlogs can be calculated by

$$B_1 = \sigma_{1.1} + \frac{\phi_1 \rho_1}{\phi_1 + \phi_2}\left(T_1 + \frac{\phi_2 l}{R}\right),$$
$$B_2 = \sigma_{1.2} + \frac{\phi_2 \rho_1}{\phi_1 + \phi_2}\left(T_1 + \frac{\phi_1 l}{R}\right). \qquad (10)$$

Therefore, the least upper bound of the total backlog is computed by

$$B = B_1 + B_2 = \sigma_1 + \rho_1\left[T + \frac{2\phi_1\phi_2 l}{R(\phi_1 + \phi_2)}\right]. \qquad (11)$$

The least upper delay bounds consist of three parts: the processing time, the time to serve input burstiness, and the scheduling delay. Let $D_{1.1}$ and $D_{1.2}$ denote the delay bounds of subflow $f_{1.1}$ and $f_{1.2}$, respectively. They can be computed by

$$D_{1.1} = T + \frac{\sigma_{1.1}}{R} + \frac{\phi_2 l}{R}, \qquad D_{1.2} = T + \frac{\sigma_{1.2}}{R} + \frac{\phi_1 l}{R}. \qquad (12)$$

Furthermore, the departure arrival curves of $f_{1.1}$ and $f_{1.2}$ can be derived by

$$\tilde{\alpha}_{1.1} = \frac{\phi_1 \rho_1}{\phi_1 + \phi_2}t + \sigma_{1.1} + \frac{\phi_1}{\phi_1 + \phi_2}\left(\rho_1 T + \frac{\rho_1 \phi_2 l}{R}\right),$$
$$\tilde{\alpha}_{1.2} = \frac{\phi_2 \rho_1}{\phi_1 + \phi_2}t + \sigma_{1.2} + \frac{\phi_2}{\phi_1 + \phi_2}\left(\rho_1 T + \frac{\rho_1 \phi_1 l}{R}\right). \qquad (13)$$

4.2. The Multiplexing Model.

In order to analyze resource sharing when multiple input flows share the bandwidth of a link at a node, we propose a multiplexing model. We shall use this model for analyzing a network with various traffic flowing scenarios.

Without loss of generality, let us consider a node serve two flows f_1 and f_2 in the FIFO order as shown by Figure 5(a). And its equivalent model is drawn in Figure 5(b). Let the node provide a service curve β^s to the aggregating flows, and f_1 and f_2 have α_1 and α_2 as arrive curves, respectively. We define $\hat{\beta}_1^s = \kappa(\beta^s, \alpha_2)$ as the *equivalent service curve* [17] provided to flow f_1, where $\kappa(\cdot, \cdot)$ is an operator to compute the equivalent service and τ is an intermediate argument. Thus, the departure arrival curve of f_1 can be derived by $\tilde{\alpha}_1 = \alpha_1 \oslash \kappa(\beta^s, \alpha_2)$, and its least upper delay bound is computed by $h(\alpha_1, \kappa(\beta^s, \alpha_2))$, and the least upper backlog bound of the node is $v(\alpha_{\{1,2\}}, \beta^s)$,

where $\alpha_{\{1,2\}}$ denotes the arrival curve of the aggregating flow $f_{\{1,2\}}$. Similarly, the equivalent service curve provides to flow f_2 can be derived by $\hat{\beta}_2^s = \kappa(\beta^s, \alpha_1)$, and its delay and backlog bounds, and departure arrival curve can be derived accordingly.

We give an example to show how to compute $\kappa(\cdot, \cdot)$. Let $\beta^s(t) = \chi_{R,T} = R[t - T]^+$ and $\alpha_2(t) = \gamma_{\sigma_2, \rho_2}(t) = \rho_2 t + \sigma_2$, then, applying Corollary 4.5 in [17], the equivalent service curve for f_1 can be calculated by

$$\hat{\beta}_1^s = \kappa(\beta^s, \alpha_2) = \gamma_{R\cdot\tau, R-\rho_2} \otimes \delta_{T+\sigma_2/R+\tau} \quad (\tau \geq 0), \qquad (14)$$

where τ is an intermediate argument for calculating the least upper delay bound, and $\delta_T(t) = +\infty$ for $t > T$, and 0 otherwise.

The least upper delay bound of f_1 is calculated by

$$D_1 = h\left(\alpha_1, \hat{\beta}_1^s\right) = \inf_{\tau \geq 0}\left\{T + \frac{\sigma_2}{R} + \frac{\sigma_1 - R\tau}{R - \rho_2} + \tau\right\}. \qquad (15)$$

Furthermore, the least upper backlog bound of the node can be derived by

$$B = \sigma_1 + \sigma_2 + (\rho_1 + \rho_2)T. \qquad (16)$$

Additionally, the arrival curve of the departure flow of f_1 is computed by

$$\tilde{\alpha}_1 = \alpha_1 \oslash \hat{\beta}_1^s = \rho_1 t + \sigma_1 + \rho_1\left(T + \frac{\sigma_2}{R} + \tau\right), \qquad (17)$$

where τ is the same as the value obtained in (15).

4.3. The Splitting-Based Performance Analysis Procedure.

There have been several research works on the traffic splitting strategy in packet networks [6, 7], due to its efficiency in load balancing. In the flow-based splitting strategy, a traffic flow is split into multiple subflows at its source node, and these subflows are forwarded to the sink through different routing paths. The source node decides how the subflows are split. Given the traffic patterns, service models, the routing protocols, and the splitting strategy, we then detail the general performance analysis procedure as follows.

Step 1. Based on the traffic pattern, routing protocols, and the traffic splitting strategy, construct a performance analysis model that converts the original network into an equivalent network.

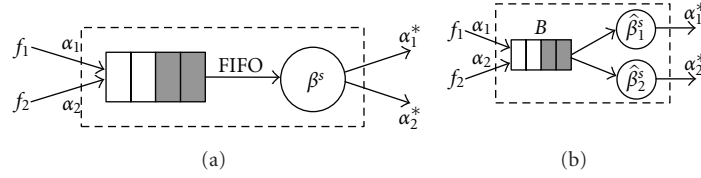*Step 2.* Derive the input and departure arrival curves of all nodes in the network based on network calculus.

FIGURE 5: (a) A node serves two input flows. (b) The equivalent model.

Step 3. Derive the end-to-end equivalent service curves for the subflows; then compute the end-to-end least upper delay bound for the main flow using $D = h(\alpha, \hat{\beta})$, where α denotes the input arrival curve and $\hat{\beta}$ denotes the end-to-end equivalent service curve.

Step 4. Using the results in Step 2, compute the least upper backlog bound of each node $B_s = v(\sum_{i \in \mathcal{I}(s)} \alpha_i^s, \beta^s)$, where $\mathcal{I}(s)$ represents the set of input flows of node s.

Step 5. Compute the power consumption of a node s by $P_s = \epsilon^r \sum_{i \in N_{in}(s)} \rho_{is} + \sum_{j \in N_{out}(s)} \epsilon_{sj}^t \rho_{sj} + p_s^c$, where $N_{in}(i)$ and $N_{out}(i)$ denote the set of nodes which are the direct sources of incoming data flows and the direct destinations of output data flows of node i, respectively.

4.4. Discussions

4.4.1. Flow-Based Splitting versus Multipath Routing. Basically, a traffic splitting process consists of two stages: establishing multiple routing paths and allocating traffic on each path according to the splitting strategy. Multipath routing is a technique exploiting routing diversity by using multiple source-destination pairs. It has been receiving plenty of research attentions [33, 34]. There are plenty of works in the literature on how to set up multiple routing paths in ad hoc networks [18, 34, 35]. Our work focuses on analyzing the performance of the splitting strategy rather than finding multipath routes. We assume that multiple paths have already been established between source nodes and the sink.

There are several common features between multipath routing and flow-based traffic splitting: first, both of them use multipaths to explore routing diversities; second, both of them aim for improving load balance. Apart from these common features, there exist significant differences between them. In multipath routing, routing decisions are made on a per-packet basis, this is, each packet chooses its routing path and is forwarded to the destination individually. Multipath routing is mainly used for improving network performance in terms of reliability and robustness [33, 34]. While in flow-based splitting strategy, the routing and forwarding is made on a per-flow basis. So it is capable of realizing a controlled splitting and providing quality of service. For example, if there are two paths between a source and a destination, a flow may be split half to one path and half to the other. So the delay guarantees can be reasoned about.

4.4.2. Flow-Based Splitting versus Node-Based Splitting. According to the way that a traffic flow split, the traffic splitting strategy can be classified into two categories: flow-based splitting and node-based splitting.

In flow-based splitting, the source node decides how the subflows are scheduled and split. The subflows can be identified after splitting. As shown in Figure 6(a), the traffic flow f_i is split into two subflows ($f_{i.1}$ and $f_{i.2}$) only at its source node s_1 and forwarded to the sink (s_6) through two routing paths: $\{s_1, s_2, s_4\}$ and $\{s_1, s_3, s_5\}$. In the node-based traffic splitting strategy (Figure 6(b)), the traffic flow can be split at the intermediate nodes that have multiple output links (such as nodes s_1 s_2 s_3 in Figure 6(b)), and these nodes decide how their input flows are allocated to their output links. An example of node-based routing strategy is the backpressure-based routing protocol (BCP) [23]. Implementing a flow-based traffic splitting strategy is more complex than a node-based traffic splitting strategy in practice, but a flow-based traffic splitting strategy also has its own advantages. In the flow-based splitting strategy, the routing and forwarding decisions are made on a per-flow basis. So it is capable of realizing a controlled splitting and satisfying quality of service requirements.

4.4.3. How to Set Splitting Parameters? Our work mainly provides a framework for quality of service analysis of the flow-based splitting strategy. Another research issue is on splitting parameter exploration, that is, how to set splitting factors? One way is to utilize static network state information, such as link capacity and buffer length, to set splitting parameters. For example, in Figure 6(a), source node S_1 can select appropriate splitting factors based on the link capacity information of it downstream links. A larger amount of traffic can be allocated to the links with higher capacity. In [36], authors presented an explicit rate-based flow control scheme, in which each route ran a proportional max-min fair bandwidth sharing algorithm to divide the measure bandwidth among the passed flows. Alternatively, the splitting decision can be made based on dynamical network state information. Each node records its current or historic buffer lengths, and the information is sent to the source node, so that the source node can select appropriate splitting parameters. For example, authors in [37] proposed a congestion-aware routing scheme which could redirect a certain amount of traffic to other paths under heavy traffic load. In this scheme, the congestion status information at each route is detected depending on the average MAC layer utilization and queue length. If congestion happens, traffic is split into other paths according to its services type.
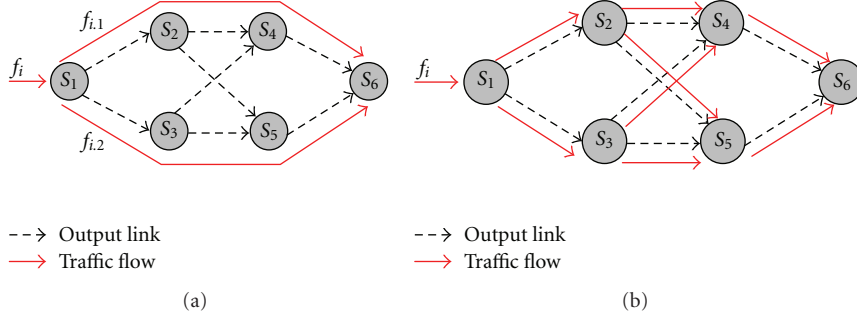
FIGURE 6: An example of splitting strategies. s_1 is the source node of the traffic flow, and s_6 is the sink. (a) Flow-based traffic splitting strategy; (b) node-based traffic splitting strategy.

5. An Analysis Example

In this section, we exemplify the general performance analysis and derive close-form formulas for of delay bounds, backlog bounds, and power consumptions under the conditions of affine arrival curve and rate-latency service curve. Consider a network consists of four nodes as shown in Figure 7. We define the *tagged main flow* as the main flow for which we shall derive the end-to-end delay bound. In this example, we choose f_1 as the tagged main flow. Let f_1 be constrained by the arrival curve $\alpha_1 = \gamma_{\sigma_1,\rho_1}$, and split into two subflows $f_{1.1}$ and $f_{1.2}$ and, respectively, traverse two different routing paths $\mathcal{R}(f_{1.1}) = \{s_1, s_2, s_4\}$ and $\mathcal{R}(f_{1.2}) = \{s_1, s_3, s_4\}$ from the source node s_1 to the sink. ϕ_1 and ϕ_2 are the splitting weights of $f_{1.1}$ and $f_{1.2}$, respectively. f_2 is the contention flow that is modeled by the arrival curve $\alpha_2 = \gamma_{\sigma_2,\rho_2}$. Let $\alpha_j^{s_i}$ and $\widetilde{\alpha}_j^{s_i}$, respectively, denote the input and departure arrival curves of flow j at node s_i. Next, we need to derive the end-to-end least upper delay bound for flow f_1 and the least upper backlog bound and power consumption for each node.

In order to compute the end-to-end least upper delay and the least upper backlog bounds, we first need to derive the input and departure arrival curves of each node.

5.1. Arrival Curves of Input and Output.
According to the results of the splitting model, the arrival curves of departure flows at node s_1 can be derived as

$$\widetilde{\alpha}_{1.1}^{s_1} = \frac{\phi_1 \rho_1}{\phi_1 + \phi_2} t + \sigma_{1.1} + \frac{\phi_1}{\phi_1 + \phi_2}\left(\rho_1 T_1 + \frac{\rho_1 \phi_2 l}{R_1}\right),$$
$$\widetilde{\alpha}_{1.2}^{s_1} = \frac{\phi_2 \rho_1}{\phi_1 + \phi_2} t + \sigma_{1.2} + \frac{\phi_2}{\phi_2}\left(\sigma_1 + \rho_1 T_1 + \frac{\rho_1 \phi_1 l}{R_1}\right). \tag{18}$$

Node s_2 has two input flows, with arrival curves α_2 and $\alpha_{1.1}^{s_2} = \widetilde{\alpha}_{1.1}^{s_1}$. According to the multiplexing analysis results (Section 4.2), we can derive the arrival curves of the departure flows of node s_2 as following

$$\widetilde{\alpha}_2^{s_2} = \rho_2 t + \sigma_2 + \rho_2\left(T_2 + \frac{\sigma_{1.1}^{s_2}}{R_2} + \tau_1\right),$$
$$\widetilde{\alpha}_{1.1}^{s_2} = \widetilde{\alpha}_{1.1}^{s_1} + \frac{\phi_1 \rho_1}{\phi_1 + \phi_2}\left(T_2 + \frac{\sigma_2}{R_2} + \tau_2\right), \tag{19}$$

where $\sigma_{1.1}^{s_2} = \sigma_{1.1} + (\phi_1/(\phi_1 + \phi_2))(\rho_1 T_1 + \rho_1 \phi_2 l/R_1)$, τ_1 and τ_2 are defined by

$$\arg\min_{\tau_1} \zeta_1(x) = \left\{\tau_1 \geq 0 : \frac{\sigma_2 - R_2 \tau_1}{R_2 - \phi_1 \rho_1/(\phi_1 + \phi_2)} + \tau_1\right\},$$
$$\arg\min_{\tau_2} \zeta_2(x) = \left\{\tau_2 \geq 0 : \frac{\sigma_{1.1}^{s_2} - R_2 \tau_2}{R_2 - \rho_2} + \tau_2\right\}. \tag{20}$$

For node s_3, the arrival curve of its input flow is $\alpha_{1.2}^{s_3} = \widetilde{\alpha}_{1.2}^{s_1}$, and it provides a service curve $\beta^{s_3} = \chi_{R_3,T_3}$. Consequently, the arrival curve of its departure flow is

$$\widetilde{\alpha}_{1.2}^{s_3} = \widetilde{\alpha}_{1.2}^{s_1} + \frac{\phi_2 \rho_1}{\phi_1 + \phi_2} T_3. \tag{21}$$

According to the connection relations, we can get the arrival curves of three input flows at node s_4, which are $\alpha_2^{s_4} = \widetilde{\alpha}_2^{s_2}$, $\alpha_{1.1}^{s_4} = \widetilde{\alpha}_{1.1}^{s_1}$, and $\alpha_{1.2}^{s_4} = \widetilde{\alpha}_{1.2}^{s_3}$. Since it is not necessary to compute the arrival curves of the departure flows at node s_4, we omit the derivation here.

5.2. The End-to-End Delay Bound.
In order to compute the end-to-end delay bound, we first need to derive the service curve provided by individual nodes. Let $\widehat{\beta}_k^{s_i}$ represent the equivalent service curve provided by node s_i to its kth input flow. As shown in Figure 7, node s_2 serves two flow $f_{1.1}$ and f_2. According to the multiplexing analysis in Section 4.2, the equivalent service curve for $f_{1.1}$ at s_2 is $\widehat{\beta}_2^{s_2} = \kappa(\beta^{s_2}, \alpha_2)$. Node s_4 serves three flows, and the equivalent service curve for $f_{1.1}$ is $\widehat{\beta}_1^{s_4} = \kappa(\kappa(\beta^{s_4}, \alpha_2^{s_4}), \alpha_{1.2}^{s_4})$. Thus, the end-to-end equivalent service curve for $f_{1.1}$ can be derived by

$$\beta_{1.1}^{e2e} = \widehat{\beta}_1^{s_1} \otimes \widehat{\beta}_2^{s_2} \otimes \widehat{\beta}_2^{s_4}$$
$$= \beta^{s_1} \otimes \delta_{\phi_2 l/R_1} \otimes \kappa(\beta^{s_2}, \alpha_2) \otimes \kappa(\kappa(\beta^{s_4}, \alpha_2^{s_4}), \alpha_{1.2}^{s_4})$$
$$= \chi_{R_1,T_1} \otimes \delta_{\phi_2 l/R_1} \otimes \gamma_{R_2\tau_2, R_2-\rho_2} \otimes \delta_{T_2+\sigma_2/R_2+\tau_2} \otimes \gamma_{R_4\tau_3, R_4-\rho_4'}$$
$$\otimes \delta_{T_4+\sigma_4'/R_4+\tau_3}$$
$$= \chi_{R_1,T_1} \otimes \delta_{\phi_2 l/R_1 + T_2+\sigma_2/R_2+\tau_2+T_4+\sigma_4'/R_4+\tau_3} \otimes \gamma_{R_2\tau_2, R_2-\rho_2}$$
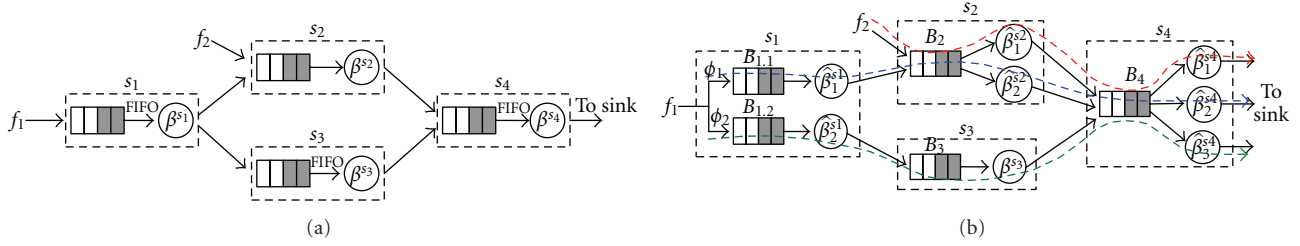$$\otimes \gamma_{R_4\tau_3, R_4-\rho_4'}, \tag{22}$$

FIGURE 7: (a) A network analysis example: the main flow f_1 is split into two subflows $f_{1.1}$ and $f_{1.2}$. f_2 is the contention flow. (b) The equivalent analysis network: the blue, green, and red dashed lines show the routing path of flow $f_{1.1}$, $f_{1.2}$, and f_2, respectively.

where τ_1 and τ_2 are the same as those in (19), τ_3 is calculated by $\arg\min_{\tau_3}\{\tau_3 \geq 0 : (\sigma_{1.1}^{s_4} - R_4\tau_3)/(R_4 - \rho_4') + \tau_3\}$, $\rho_4' = \rho_2 + \phi_2\rho_1/(\phi_1 + \phi_2)$ and

$$\sigma_4' = \rho_2\left(T_2 + \frac{\sigma_{1.1}^{s_2}}{R_2} + \tau_1\right) + \sigma_{1.2} \\ + \frac{\phi_2[\rho_1(T_1 + T_3) + \rho_1\phi_1 l/R_1]}{\phi_1 + \phi_2} + \sigma_2. \tag{23}$$

Analogously, the end-to-end equivalent service curve for $f_{1.2}$ can be derived by

$$\beta_{1.2}^{e2e} = \hat{\beta}_2^{s_1} \otimes \beta^{s_3} \otimes \hat{\beta}_3^{s_4} \\ = \beta^{s_1} \otimes \delta_{\phi_1 l/R_1} \otimes \beta^{s_3} \otimes \kappa(\kappa(\beta^{s_4}, \alpha_2^{s_4}), \alpha_{1.1}^{s_4}) \\ = \chi_{R_1,T_1} \otimes \delta_{\phi_1 l/R_1} \otimes \chi_{R_3,T_3} \otimes \gamma_{R_4\tau_4, R_4 - \rho_4''} \otimes \delta_{T_4 + \sigma_4''/R_4 + \tau_4}, \tag{24}$$

where τ_4 is defined by $\arg\min_{\tau_4}\{\tau_4 \geq 0 : (\sigma_{1.2}^{s_4} - R_4\tau_4)/(R_4 - \rho_4'') + \tau_4\}$, $\rho_4'' = (\rho_2 + \phi_1\rho_1/(\phi_1 + \phi_2))$ and

$$\sigma_4'' = \frac{\phi_1\rho_1}{\phi_1 + \phi_2}\left(T_1 + T_2 + \frac{\sigma_2}{R_2} + \tau_2\right) + \frac{\rho_1\phi_1\phi_2 l}{R_1(\phi_1 + \phi_2)} \\ + \rho_2\left(T_2 + \frac{\sigma_{1.1}^{s_2}}{R_2} + \tau_1\right) + \sigma_{1.1} + \sigma_2, \tag{25}$$

where τ_1 and τ_2 are the same as those in (19).

After we get the end-to-end service curves, the least upper delay bounds of $f_{1.1}$ and $f_{1.2}$ can be, respectively, computed by $h(\alpha_{1.1}, \beta_{1.1}^{e2e})$ and $h(\alpha_{1.2}, \beta_{1.2}^{e2e})$,

$$h(\alpha_{1.1}, \beta_{1.1}^{e2e}) \\ = T_1 + T_2 + T_4 + \frac{\phi_2 l}{R_1} + \frac{\sigma_2}{R_2} + \frac{\sigma_4'}{R_4} \\ + \inf_{\substack{\tau_2 \geq 0 \\ \tau_3 \geq 0}}\left\{\tau_2 + \tau_3 + \left[\frac{\sigma_{1.1}}{R_1} \vee \frac{\sigma_{1.1} - R_2\tau_2}{R_2 - \rho_2} \vee \frac{\sigma_{1.1} - R_4\tau_3}{R_4 - \rho_4'}\right]\right\},$$

$$h(\alpha_{1.2}, \beta_{1.2}^{e2e}) \\ = T_1 + T_3 + T_4 + \frac{\phi_1 l}{R_1} + \frac{\sigma_4''}{R_4} \\ + \inf_{\tau_4 \geq 0}\left\{\tau_4 + \left[\frac{\sigma_{1.2}}{R_1} \vee \frac{\sigma_{1.2}}{R_3} \vee \frac{\sigma_{1.2} - R_4\tau_4}{R_4 - \rho_4''}\right]\right\}. \tag{26}$$

Hence, the end-to-end least upper delay bound for the flow f_1 equals the maximum of the delays of two subflows, namely,

$$D_{f_1} = \max\{h(\alpha_{1.1}, \beta_{1.1}^{e2e}), h(\alpha_{1.2}, \beta_{1.2}^{e2e})\}. \tag{27}$$

5.3. The Backlog Bound.
Let B_{s_i} denote the backlog bound of node $s_i(i = 1, \ldots, 4)$. As we have already derived the arrival curves of input and output flows at each node, its least upper backlog bound can be calculated very easily. According to the result in (11), we have

$$B_{s_1} = B_{1.1} + B_{1.2} = \sigma_1 + \rho_1\left[T_1 + \frac{2\phi_1\phi_2 l}{R_1(\phi_1 + \phi_2)}\right]. \tag{28}$$

Node s_2 has two input flows α_2 and $\alpha_{1.1}^{s_2}$, so its least upper backlog bound is computed by

$$B_{s_2} = \sigma_{1.1} + \sigma_2 + \frac{\phi_1(\rho_1 T_1 + \rho_1\phi_2 l/R_1)}{\phi_1 + \phi_2} + \left(\frac{\phi_1\rho_1}{\phi_1 + \phi_2} + \rho_2\right)T_2. \tag{29}$$

Analogously, the least upper backlog bounds of node s_3 and s_4 can be derived by

$$B_{s_3} = \sigma_{1.2} + \frac{\phi_2}{\phi_1 + \phi_2}\left[\rho_1(T_1 + T_3) + \frac{\rho_1\phi_1 l}{R_1}\right],$$

$$B_{s_4} = \rho_1(T_1 + T_4) + \sigma_1 + \rho_2\left(T_2 + T_4 + \frac{\sigma_{1.1}^{s_2}}{R_2} + \tau_1\right) + \sigma_2 \\ + \frac{2\rho_1 l\phi_1\phi_2}{R_1(\phi_1 + \phi_2)} + \frac{\phi_2\rho_1 T_3}{\phi_1 + \phi_2} + \frac{\phi_1\rho_1}{\phi_1 + \phi_2}\left(T_2 + \frac{\sigma_2}{R_2} + \tau_2\right). \tag{30}$$

5.4. Power Consumption.
According to the power model (Section 3.2), the total power consumption of a node is

contributed by the radio transmitter, radio receiver, and computation electronics. Thus, the power consumptions of all the nodes can be computed by

$$P_{s_1} = \epsilon^r \rho_1 + \epsilon^t_{s_1 s_2} \rho_{1.1} + \epsilon^t_{s_1 s_3} \rho_{1.2} + p^c$$

$$= \rho_1 \left[\epsilon^r + \frac{\phi_1 N_0 (2^{R_{s_1 s_2}/\eta W} - 1)}{R_{s_1 s_2} G_{s_1 s_2} (\phi_1 + \phi_2)} + \frac{\phi_2 N_0 (2^{R_{s_1 s_3}/\eta W} - 1)}{R_{s_1 s_3} G_{s_1 s_3} (\phi_1 + \phi_2)} \right]$$

$$+ p^c,$$

$$P_{s_2} = \epsilon^r (\rho_{1.1} + \rho_2) + \epsilon^t_{s_2 s_4} (\rho^{s_4}_{1.1} + \rho^{s_4}_2) + p^c$$

$$= \left[\epsilon^r + \frac{N_0 (2^{R_{s_2 s_4}/\eta W} - 1)}{R_{s_2 s_4} G_{s_2 s_4}} \right] \left(\frac{\phi_1 \rho_1}{\phi_1 + \phi_2} + \rho_2 \right) + p^c,$$

$$P_{s_3} = \left[\epsilon^r + \frac{N_0 (2^{R_{s_3 s_4}/\eta W} - 1)}{R_{s_3 s_4} G_{s_3 s_4}} \right] \frac{\phi_2 \rho_1}{\phi_1 + \phi_2} + p^c,$$

$$P_{s_4} = \left[\epsilon^r + \frac{N_0 (2^{R_{s_4 s_0}/\eta W} - 1)}{R_{s_4 s_0} G_{s_4 s_0}} \right] (\rho_1 + \rho_2) + p^c,$$

$$(31)$$

where $d_{s_4 s_0}$ denotes the distance between node s_4 and the sink node s_0. Here, we assume that the link capacity can meet the requirements of the traffic bandwidth, that is, the service rate is bigger than the sum of input data rates.

6. Performance Evaluation

To show benefits of the traffic splitting strategy and validate the network calculus-based performance analysis method, we provide numerical results and simulations under the scenario of a fresh food monitoring application. In the numerical results, the end-to-end least upper delay bounds, the least upper backlog bounds, and power consumptions are compared under two scenarios: *general routing with no traffic splitting (NOS)* and *flow-based splitting strategy (FBS)*. In the simulations, we compare the results obtained by the analytical method with the simulation results also under these two scenarios.

The numerical results are based on an application example of a real-time fresh food monitoring system deployed in a warehouse [3, 4]. As shown in Figure 8, one sink and 9 cluster heads are uniformly distributed in a 20 m × 10 m warehouse. Each cluster head connects with 5 sensor nodes. The coordinates of cluster heads and sink (s_0) are $s_0(0, 0)$, $s_1(17.2, 1.7)$, $s_2(14.1, 5.5)$, $s_3(11, 0.5)$, $s_4(14.8, -3.6)$, $s_5(8.3, 4)$, $s_6(9.1, -4.4)$, $s_7(2.5, 4.7)$, $s_8(4.2, 0.8)$, $s_9(3.3, -3.6)$. We consider the application scenario of real-time monitoring, where sensor nodes periodically generate packets when there is a request.

6.1. Numerical Results. Assume sensor nodes in cluster C_1, C_2, and C_3 are requested to send packets to the sink via cluster head s_1, s_2, and s_3, respectively. Consequently, there are three traffic flows in the network: f_1, f_2, and f_3. They are characterized by arrival curves $\alpha_1 = \sum_{n_j \in C_1} \alpha^j_1$, $\alpha_2 = \sum_{n_j \in C_2} \alpha^j_2$, $\alpha_3 = \sum_{n_j \in C_3} \alpha^j_3$, where $\alpha^j_i (i = 1, 2, 3)$ denotes

TABLE 1: Parameters.

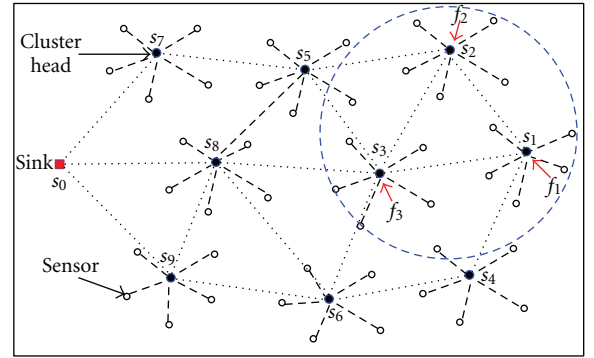| Parameter	Notation	Value	Unit
Packet length	l	400	bits
Computation power	p^c	10	uJ/s
Path loss factor	θ	3	—
Service delay	$T_i (i = 1, \ldots, 9)$	0.1	s
Data rate of f_2	ρ_2	1.2	kbps
Data rate of f_3	ρ_3	1.6	kbps
Burstiness	$\sigma_1, \sigma_2, \sigma_3$	400	bits
Channel bandwidth	W	20	kHz

FIGURE 8: A cluster-mesh sensor network: s_0 is the sink. An event happens in the blue circle, and three traffic flows f_1, f_2, f_3 are generated.

the arrival curve of traffic generate by sensor node j in cluster C_i. For periodic packets generation applications, the arrival process can be characterized by the affine arrival curve [12], that is, $\alpha_1 = \gamma_{\sigma_1, \rho_1}$, $\alpha_2 = \gamma_{\sigma_2, \rho_2}$, and $\alpha_3 = \gamma_{\sigma_3, \rho_3}$. Moreover, assume cluster head s_i provides a rate-latency service curve $\beta^{s_i} = \chi_{R_i, T_i}$. f_1 is the tagged main flow, and f_2, f_3 are the contention flows. Other parameters are listed in Table 1. We shall derive the end-to-end least upper delay bound for f_1, the least upper backlog bounds, and power consumptions of all nodes in two scenarios: NOS and FBS (Figure 9).

From the power model in (1) and Figure 2, the energy per bit is a monotonically increasing function of the transmission rate [29] if other parameters are fixed. Thus, it is better to use low transmission power for the sake of energy efficiency. On the other hand, the delay bound is a monotonically decreasing function of the service rate. (e.g., given an arrival curve $\alpha(t) = \rho t + \sigma$ and service curve $\beta(t) = R[t - T]^+$, the delay bound is $T + \sigma/R$ when $R \geq \rho$.) Therefore, there is a tradeoff between energy consumption and delay. In order to study this tradeoff, we implement numerical experiments in two scenarios: (1) uniform service rate: the service rate of all cluster heads are the same and fixed; (2) heterogeneous service rate: in order to guarantee a limited delay and backlog, the service rate should be equal to or bigger than the data arrival rate. So we set the service rate R equal to the arrival rate ρ.

We compare the end-to-end least upper delay of f_1, the backlog bounds, and power consumptions of all nodes in two
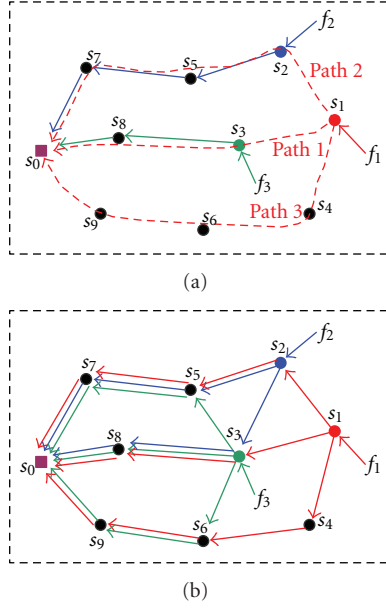
(a)

(b)

FIGURE 9: (a) General routing with no splitting (NOS): the tagged main flow f_1 chooses one of path 1, 2, or 3, and the routing paths of f_2 and f_3 are shown by the blue and green line, respectively. (b) Flow-based splitting (FBS): all three flows are split as shown by the red, blue, and green lines, respectively.
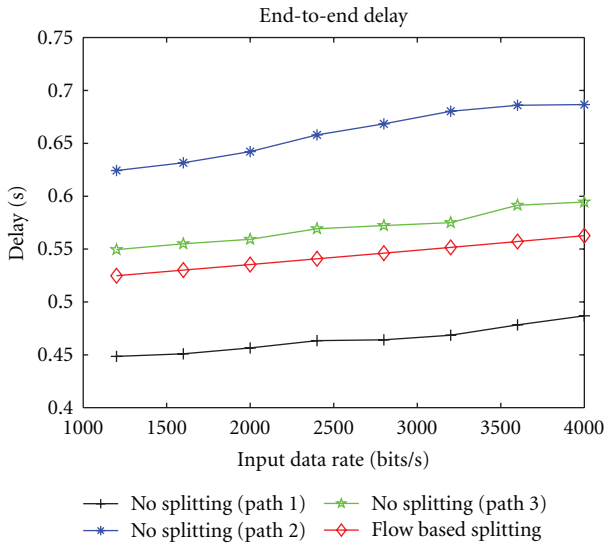

FIGURE 10: End-to-end least upper delay bound.

scenarios: NOS and FBS. In NOS, f_1 chooses one of the three paths: path 1—$\{s_1, s_3, s_8\}$, path 2—$\{s_1, s_4, s_6, s_9\}$, and path 3—$\{s_1, s_2, s_5, s_7\}$. In FBS, f_1 is evenly split into three subflows, and they are allocated on these three paths.

6.1.1. Uniform Service Rate. In the first numerical example, we choose a fixed service rate $R_i = 9.6$ kbps ($i = 1, \ldots, 9$).

Figure 10 shows the comparison of the end-to-end least upper delay bounds of the tagged main flow f_1 in NOS and

FIGURE 11: Least upper backlog bounds (in NOS: flow 1 chooses path 3).

FBS scenarios. In the NOS scenario, we compute the end-to-end delays of f_1 going through three different routing paths (Figure 9(a)). The input data rate of f_1 varies from 1.2 kbps to 4 kbps. From this figure, we can see that the end-to-end delays in all scenarios increase with the input data rates. Furthermore, on average, the delays in FBS are 23.6% and 9.4% less than those of path 2 and path 3 in NOS, respectively. And the delay in FBS is 12.4% bigger than those of path 1 in NOS. This is because path 1 is shorter than other two paths.

Figure 11 shows the least upper and average backlog bounds in the FBS and NOS scenarios with input data rates vary, where "Max-B(NOS)" means the maximum backlog in NOS which denotes the maximum value of backlogs among all nodes and "Ave-B(FBS)" means the average backlog in FBS which denotes the average value of backlogs over all nodes. From this figure, we can find that both the maximum and average backlogs in FBS are less than those in NOS. It indicates that the traffic splitting strategy can reduce backlogs. Moreover, the differences between the maximum backlogs of the two scenarios are much bigger than those of average backlogs. The average backlogs in FBS are 14.5% less than those in NOS on the average. While the maximum backlog in FBS is 23.4% less than that in NOS when the input data rate is 1.2 kbps, and the value increases to 40% when the input data rate is 4 kbps. We can also observe similar reduction in the variance of maximum backlogs (as shown in Figure 12), where the variance of backlogs in NOS is much bigger than that in FBS. It means that in NOS some nodes have very small backlogs, but some nodes have very large backlogs. Since the buffer size of a sensor node is basically determined by the value of maximum backlog, larger backlog would bring higher hardware cost. Therefore, applying the flow-based splitting strategy can bring better load balance and thus reduce overall cost.
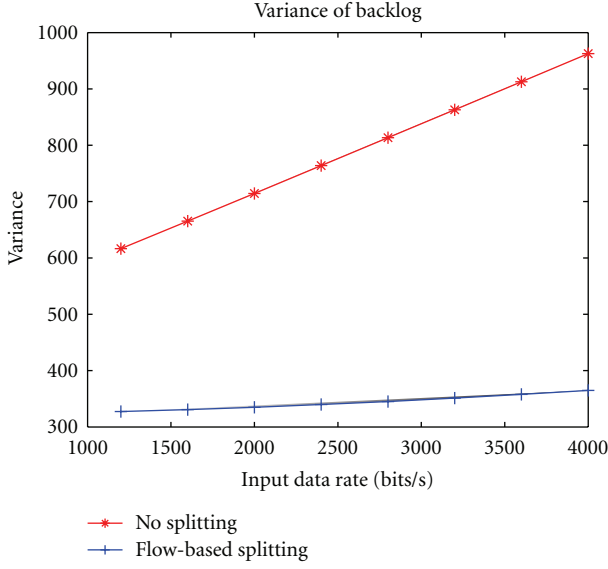
FIGURE 12: Variance of least upper backlog bounds (in NOS: flow 1 chooses path 3).
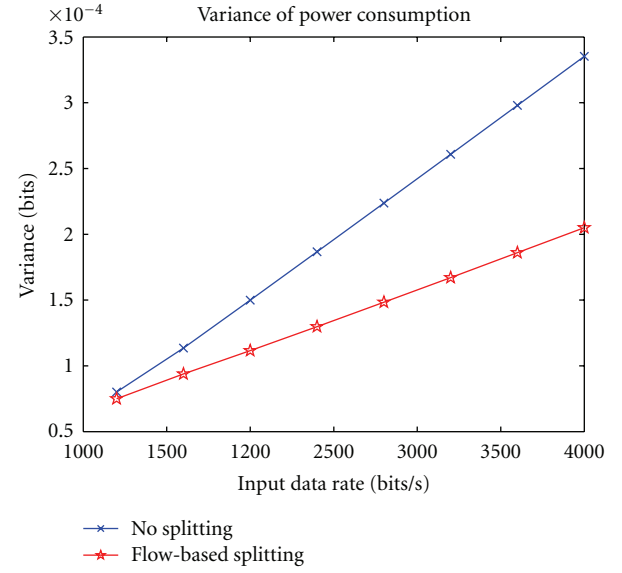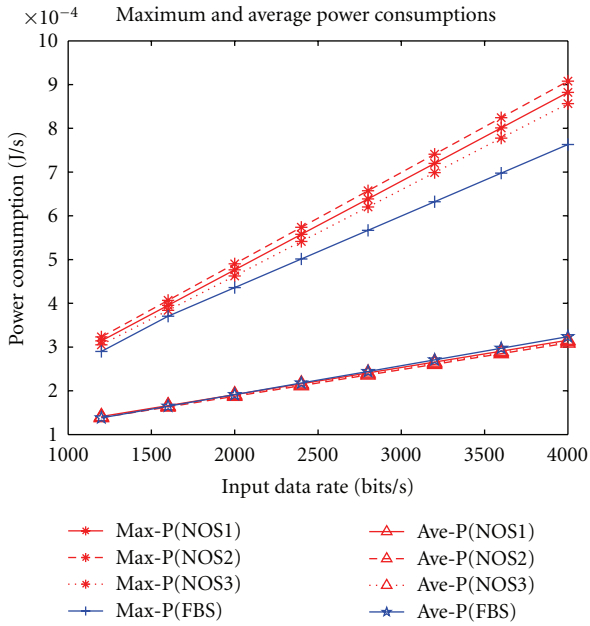

FIGURE 14: Variance of power consumption.

FIGURE 13: Power consumption (In NOSi: flow 1 chooses path i, where $i = 1, 2, 3$).

The maximum and average power consumptions of all nodes in NOS and FBS are shown in Figure 13, where "Max-P(NOS)" and "Ave-B(NOS)," respectively, denote the maximum and average power consumptions in the NOS scenario. First, we see from the figure that all the power consumptions increase with the input data rates. Furthermore, when the data rate increases, the average power consumptions in the NOS and FBS are almost the same. However, the maximum power consumption in NOS increases much faster than that in FBS, with the maximum differences between FBS and

NOS increasing from 0.8% to 12%. It indicates that the power consumptions of nodes are uneven in NOS. We can also see this from Figure 14 showing the variance of power consumption of all nodes. From this figure, we can find the variance in NOS increases much faster than that in FBS. Usually, the lifetime of a WSN is determined by the first node exhausting its energy. Hence, the flow-based splitting scheme can be used for balancing power consumption and consequently increasing the lifetime of the network.

6.1.2. Heterogeneous Service Rate. The data rate of f_1 varies from 1.2 to 4 kbps. The data rate of f_2 and f_3 is given in Table 1. The service rate of each node is equal to its arrival rate.

Being different from Figure 10, the end-to-end delays of FBS are basically bigger than those of NOS in this case (Figure 15). Moreover, when the input data rate increases, the end-to-end delay decreases. The reason is that the service rate increases with the input data rate. While the input burstiness is the same, so the delay would decrease.

Figure 16 shows the comparison of backlog bounds in the FBS and NOS scenarios. From this figure, we can find that the average backlogs in FBS and NOS are almost the same. While the maximum backlog in FBS is −2.4% less than that in NOS when the input data rate is 1.2 kbps, the value gradually increases to 12.8% when the input data rate gradually increases to 4 kbps. Similar to Figure 12, from Figure 17, we can see that the variance of backlogs in NOS is bigger than that in FBS. It also means that in NOS some nodes have very small backlogs, but some nodes have very large backlogs.

In the case of heterogeneous service rates, we see from the figure (Figure 18) that all the power consumptions increase with the input data rates. Furthermore, the average power consumptions in the NOS are approximately 11.4% bigger
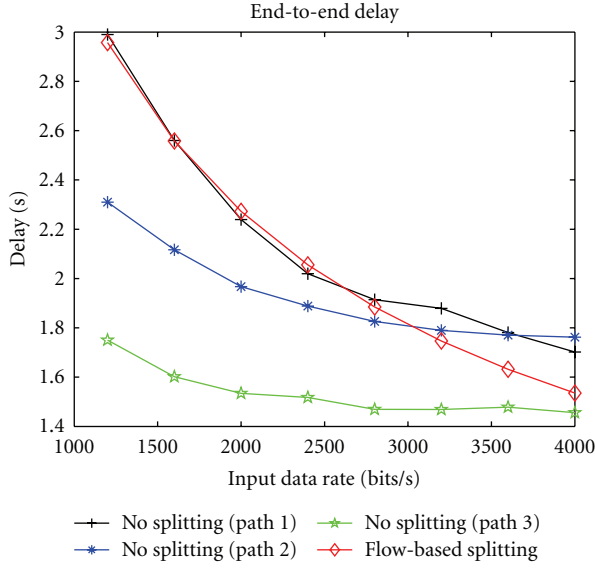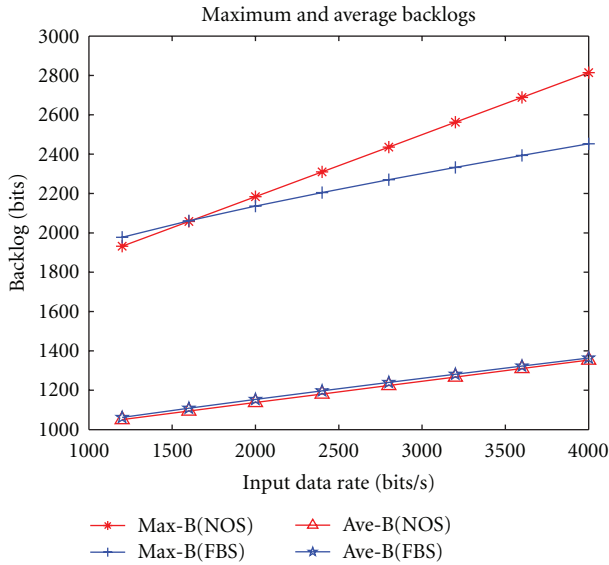
FIGURE 15: End-to-end delay.

FIGURE 17: Variance of least upper backlog bounds (in NOS: flow 1 chooses path 3).

FIGURE 16: Least upper backlog bounds (in NOS: flow 1 chooses path 3).

FIGURE 18: Power consumption (in NOSi: flow 1 chooses path i, where $i = 1, 2, 3$).

than those in FBS. However, the maximum power consumption in NOS increases much faster than that in FBS, with the maximum differences between FBS and NOS increasing from 23.9% to 33%. The differences of maximum power consumption in this case are much bigger than those in uniform service rate case Figure 13. We can also see this from Figure 19 showing the variance of power consumption. From this figure, we can find the variance in NOS increases much faster than that in FBS.

From all those results and comparison, we can have the following conclusions: first, applying FBS strategy can balance traffic load and power consumption, so as to reduce overall system cost and increase the network lifetime. Second, there is a tradeoff between power consumption and system performance. Under uniform service rate, the end-to-end
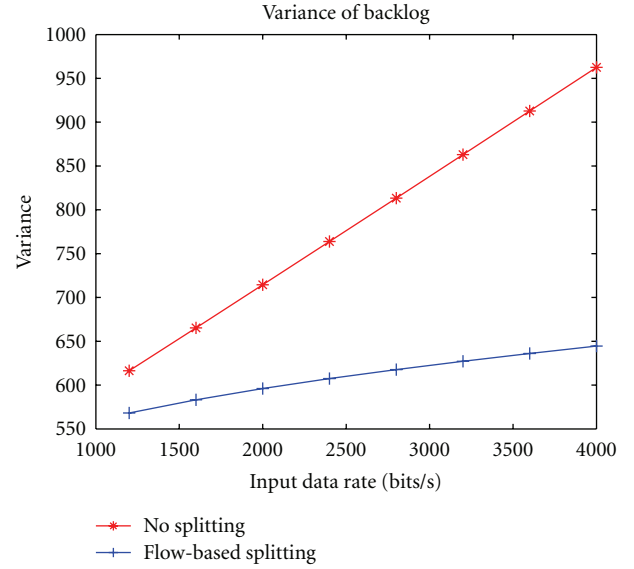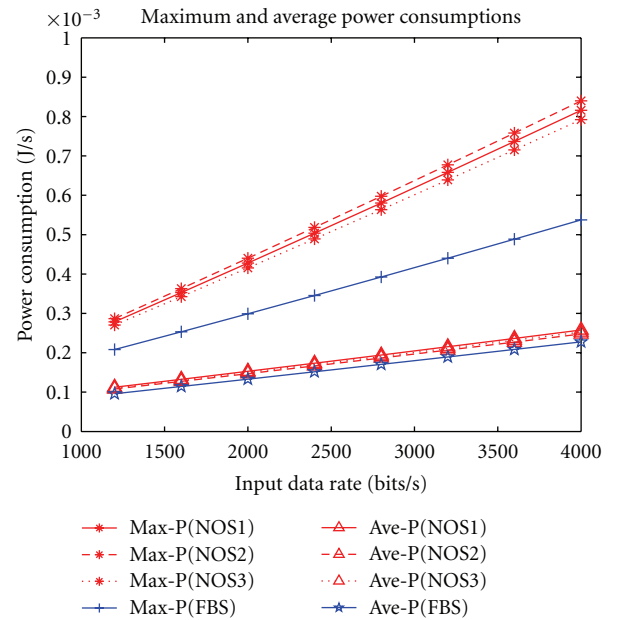
delays of FBS are less than those of NOS in most cases, and the power consumptions of FBS are slightly less than those of NOS. While under heterogeneous service rates, the end-to-end delays of FBS are generally bigger than those of NOS, but the power consumptions of FBS are much less than those of NOS. It means that the decreasing of power consumption is obtained at the cost of increasing delay.

6.1.3. Comparison of End-to-End and Hop-by-Hop Methods. As stated in [24], there are two ways to compute the end-to-end delay bound. The first method is summing up the per-hop delay together. The main idea of the other method is
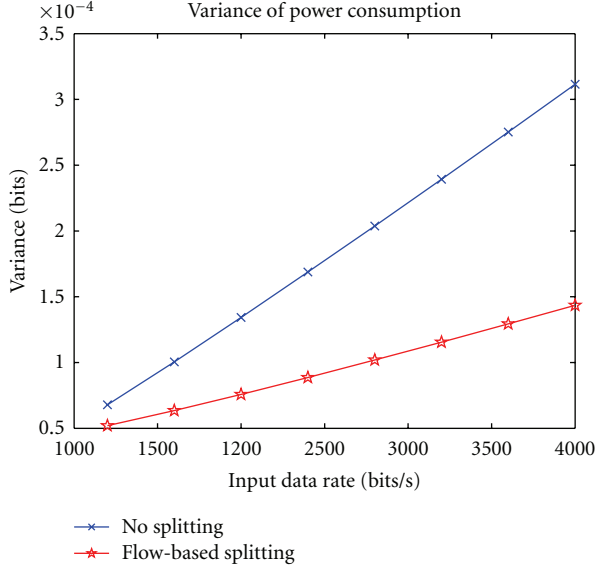
FIGURE 19: Variance of power consumption.

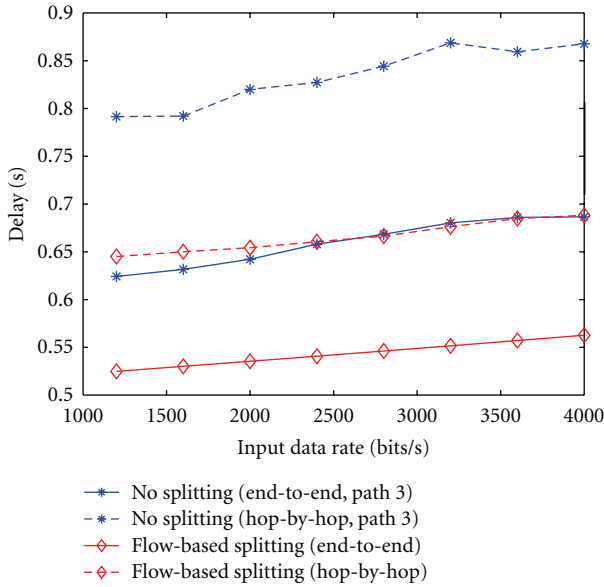FIGURE 21: End-to-end delays in NOS.

FIGURE 20: Compare the end-to-end delay computed by two methods.

to derive an equivalent service curve for a given traffic flow. And then the end-to-end delay bound is calculated using the equivalent service curve. In [24], we use the first method (hop-by-hop). While in this paper, we adopt the second method (end-to-end). **Figure 20** illustrates the comparison of these two methods in the scenario of FBS and NOS. In average, the hop-by-hop delay in NOS is 26.4% bigger than that of the end-to-end delay. And, in FBS, the hop-by-hop delay is 22.5% bigger than that of the end-to-end delay. Therefore, the end-to-end method can get tighter bound than the hop-by-hop method.

6.2. Simulation Results. Since a simulation environment allows us to create a realistic sensor network behavior while
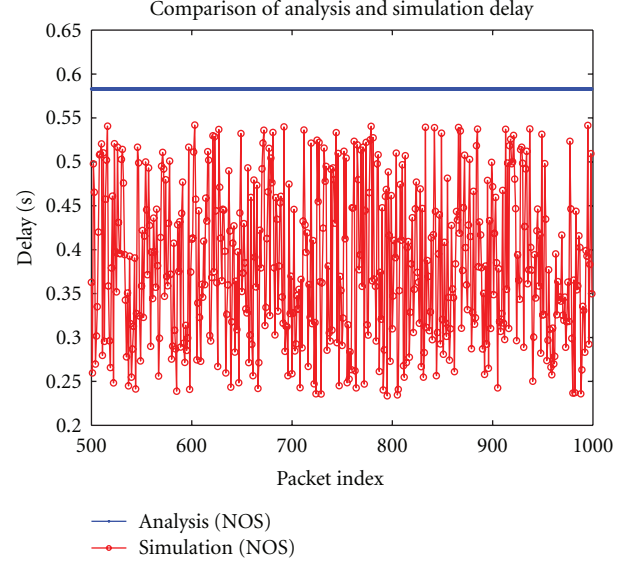
still controllable, we conduct experiments in a simulation environment based on OMNeT++ 3.3 rather than in a field trial. We define *tightness* as the ratio of maximum simulation value divided by the analytical value.

In the simulation, we use a most common log-normal path loss model [38]. This model can provide more accurate multipath channel models than Rayleigh and Nakagami models for indoor environments [39]. The simulation is also based on the application scenario shown in **Figure 8**. Parameters used in simulations are the same as those in Table 1. Other parameters used are $d_0 = 1$ m, reference channel gain $G_0 = 10^{-5}$ (-50 dB), noise power $N_0 = 10^{-10}$ (-100 dB), and the channel noise is subject to a Gaussian random variable with deviation 4. We conduct 50 simulation runs. In each run, the total simulation period is 25000 cycles and the source generates one packet every cycle. The packet generation rate is based on the predefined data rate. For example, if the predefined data rate is 2.8 kbps, 7 packets are generated in every second, and the length of a cycle is 1/7 s. In order to bypass the initial nonstationary stage, the data of first 5000 cycles are omitted. In each run, the delay of every packet is recorded and the value of backlog is recorded in every cycle. Since we want to compare the analytical results with the worst-case simulation results, we select the results of runs leading to maximum delay and maximum backlog as the simulation results.

Figures 21 and 22 show the comparison of simulation results and analytical results of end-to-end delays of flow f_1 in the NOS and FBS scenarios, respectively. In NOS, the arrival rate of flow 1 is 2.8 kbps, and it selects path 3 as its routing path. In order to make the figure easy to read, we do not illustrate the delays of all packets but extract 500 values from them. From these two figures, we observe that all the simulation values are bounded by the analytical results. And the tightness in NOS and FBS are 91% and 93.2%, respectively. This indicates our analysis performs well on bounding the end-to-end data delivery delay.
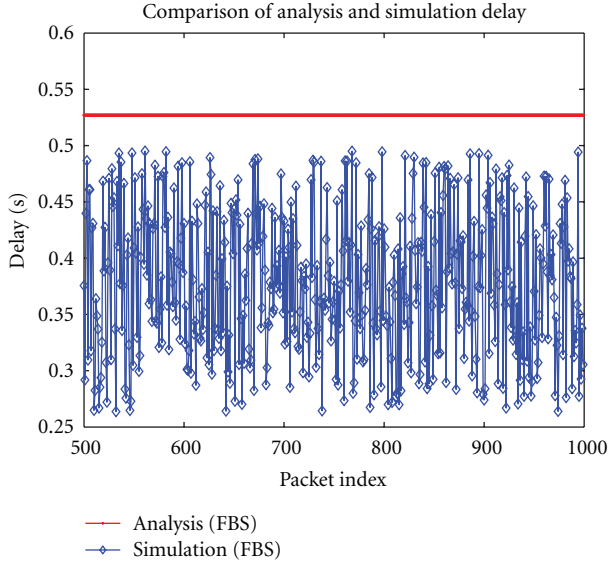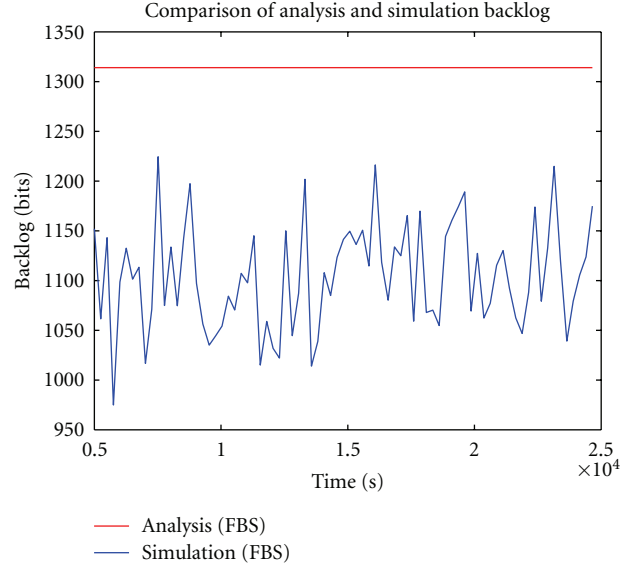
FIGURE 22: End-to-end delays in FBS.
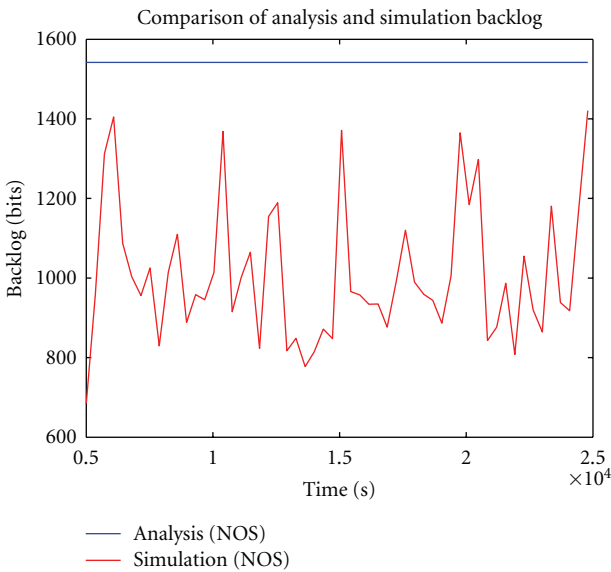

FIGURE 24: Nodes' backlogs in FBS.

FIGURE 23: Nodes' backlogs in NOS.

7. Conclusions and Future Work

Dimensioning timing-critical sensor networks requires formal methods to ensure performance and cost in any conditions. In this work, we present a network-calculus-based analysis method to compute the worst-case end-to-end delay bounds for individual flows, backlog bounds, and power consumptions for individual nodes. Based on network calculus and the splitting model, we are able to compute per-flow equivalent service curve provided by the tandem of visited nodes and the input and departure arrival curves of each node. Consequently, we can derive the performance bounds for the network which applies the flow-based traffic splitting strategy. Under the assumptions of affine arrival curve and rate-latency service curves, closed-form formulas of these bounds are computed. The numerical results for the example scenario show that, by applying the splitting strategy, the end-to-end delay can be reduced in most cases, the maximum backlog can be reduced up to 40%, and the power consumption can be reduced up to 15%. Furthermore, the simulation results verify that the theoretical bounds of our analysis are valid and fairly tight.

As stated in Section 4.4, there are several directions for future work. First, we will study the problem of designing a splitting scheme, this is, how to select splitting parameters based on network state information. Another research issue is to explore the optimized design space with given buffer sizes, performance requirements, and energy constraints for specific applications.

For the backlog analysis, node s_8 is chosen as the observation node. Its backlogs in different time points are recorded and compared as shown in Figures 23 and 24. In NOS, the arrival rate of flow 1 is 2.8 kbps, and it selects path 3 as its routing path. We can see that all the simulation values of backlogs are within the scopes of the analytical values. Moreover, the backlogs in the FBS scenario are less than those in NOS, which also indicates that the flow-based splitting scheme can reduce the maximum backlogs by balancing traffic load over the network. Additionally, the tightness of the analytical results in NOS and FBS is 93.5% and 92.1%, respectively. In summary, the proposed analysis method is correct on deriving the backlog bound and the tightness is satisfactory.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.

[2] D. Culler, D. Estrin, and M. Srivastava, "Overview of sensor networks," *IEEE Computer*, vol. 37, no. 8, pp. 41–49, 2004.

[3] Z. Pang, J. Chen, D. Sarmiento et al., "Mobile and wide area deployable sensor system for networked services," in *Proceedings of the 8th Annual IEEE Conference on Sensors*, Christchurch, New Zealand, 2009.

[4] Z. Zhang, Q. Chen, T. Bergarp et al., "Wireless sensor networks for logistics and retail," in *Proceedings of the 6th International Conference on Networked Sensing Systems (INSS '09)*, Pittsburgh, Pa, USA, 2009.

[5] S. Tai, R. Benkoczi, H. Hassanein, and S. Akl, "A performance study of splittable and unsplittable traffic allocation in wireless sensor networks," in *Proceedings of the IEEE International Conference on Communications (ICC '06)*, Istanbul, Turkey, July 2006.

[6] A. Meddeb, "Benefits of multicast traffic split routing in packet switched networks," in *Proceedings of the IEEE International Conference on Communications (ICC '04)*, June 2004.

[7] A. Zalesky, H. L. Vu, and M. Zukerman, "Reducing spare capacity through traffic splitting," *IEEE Communications Letters*, vol. 8, no. 9, pp. 594–596, 2004.

[8] F. Hu, Y. Xiao, and Q. Hao, "Congestion-aware, loss-resilient bio-monitoring sensor networking for mobile health applications," *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 4, Article ID 4909283, pp. 450–465, 2009.

[9] R. L. Cruz, "A calculus for network delay, part I: network elements in isolation," *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 114–121, 1991.

[10] R. L. Cruz, "A calculus for network delay, part II: network analysis," *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 132–141, 1991.

[11] J. Boudec and P. Thiran, *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet*, Springer, LNCS 2050, Berlin, Germany.

[12] J. B. Schmitt and U. Roedig, "Sensor network calculus—a framework for worst case analysis," in *Proceedings of the IEEE/ACM International Conference on Distributed Computing in Sensor Systems (DCOSS '05)*, vol. 3560, 2005.

[13] A. Koubaa, M. Alves, and E. Tovar, "Modeling and worst-case dimensioning of cluster-tree wireless sensor networks," in *Proceedings of the 27th IEEE International Real-Time Systems Symposium (RTSS '06)*, Rio de Janeiro, Brazil, December 2006.

[14] J. B. Schmitt, F. A. Zdarsky, and L. Thiele, "A comprehensive worst-case calculus for wireless sensor networks with in-network processing," in *Proceedings of the 28th IEEE International Real-Time Systems Symposium (RTSS '07)*, Tucson, Ariz, USA, December 2007.

[15] P. Jurcik, R. Severino, A. Koubaa, M. Alves, and E. Tovar, "Real-time communications over cluster-tree sensor networks with mobile sink behaviour," in *Proceedings of the 14th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA '08)*, 2008.

[16] C. S. Chang, *Performance Guarantees in Communication Networks*, Springer-Verlag, Berlin, Germany, 2000.

[17] L. Lenzini, L. Martorini, E. Mingozzi, and G. Stea, "Tight end-to-end per-flow delay bounds in FIFO multiplexing sink-tree networks," *Performance Evaluation*, vol. 63, no. 9-10, pp. 956–987, 2006.

[18] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, "Highly-resilient, energyefficient multipath routing in wireless sensor networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 5, no. 4, pp. 11–25, 2001.

[19] Y. M. Lu and V. Wong, "An energy-efficient multipath routing protocol for wireless sensor networks," in *Proceedings of the 64th IEEE Vehicular Technology Conference (VTC-06-Fall)*, Montreal, QU, Canada, September 2006.

[20] W. Lou, "An efficient N-to-1 multipath routing protocol in wireless sensor networks," in *Proceedings of the 2nd IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS '05)*, Washington, DC, USA, November 2005.

[21] S. Zou, I. Nikolaidis, and J. Harms, "Aggregation vs. load balancing in WSNS," in *Proceedings of the 18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '07)*, Athens, Greece, September 2007.

[22] C. Li, J. Zou, H. Xiong, and Y. Zhang, "Joint coding/routing optimization for correlated sources in wireless visual sensor networks," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '09)*, Honolulu, Hawaii, USA, December 2009.

[23] S. Moeller, A. Sridharan, B. Krishnamachari, and O. Gnawali, "Routing without routes: the backpressure collection protocol," in *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN '10)*, April 2010.

[24] H. She, Z. Lu, A. Jantsch, L.-R. Zheng, and D. Zhou, "Deterministic worst-case performance analysis for wireless sensor networks," in *Proceedings of the International Wireless Communications and Mobile Computing Conference (IWCMC '08)*, Crete Island, Greece, August 2008.

[25] H. She, Z. Lu, A. Jantsch, L.-R. Zheng, and D. Zhou, "Analysis of traffic splitting mechanisms for 2D mesh sensor networks," *International Journal of Software Engineering and Its Applications*, vol. 2, no. 3, pp. 25–37, 2008.

[26] N. Baker, "Real world wireless mesh sensor network solutions," in *Proceedings of the IEE Seminar on Industrial Networking and Wireless Communications for Control*, 2006.

[27] S. I. Lee and J. S. Lim, "Hybrid cluster mesh scheme for energy efficient wireless sensor networks," *IEICE Transactions on Communications*, vol. E91-B, no. 8, pp. 2610–2617, 2008.

[28] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava, "Energy-aware wireless microsensor networks," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 40–50, 2002.

[29] B. Prabhakar, E. U. Biyikoglu, and A. El Gamal, "Energy-efficient transmission over a wireless link via lazy packet scheduling," in *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '01)*, pp. 386–394, 2001.

[30] A. Goldsmith and A. Nin, *Wireless Communications*, Cambridge University Press, New York, NY, USA, 2005.

[31] S. C. Ergen and P. Varaiya, "Pedamacs: power efficient and delay aware medium access protocol for sensor networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 7, Article ID 1637439, pp. 920–930, 2006.

[32] A. Rowe, R. Mangharam, and R. Rajkumar, "Rt-link: A time-synchronized link protocol for energyconstrained multi-hop wireless networks," in *Proceedings of the 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks (SECON '06)*, vol. 2, pp. 402–411, Reston, Va, USA, 2006.

[33] Y. Charfi, N. Wakamiya, and M. Murata, "Adaptive and reliable multi-path transmission in wireless sensor networks using forward error correction and feedback," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '07)*, pp. 3684–3689, Kowloon, Hong Kong, 2007.

[34] P. Djukic and S. Valaee, "Reliable packet transmissions in multipath routed wireless networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 5, pp. 548–559, 2006.

[35] S. De and C. Qiao, "On throughput and load balancing of multipath routing in wireless networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '04)*, vol. 3, pp. 1551–1556, 2004.

[36] K. Chen, K. Nahrstedt, and N. Vaidya, "The utility of explicit rate-based flow control in mobile ad hoc networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '04)*, vol. 3, pp. 1921–1926, 2004.

[37] H. Jiao and F. Y. Li, "A service-oriented routing scheme with load balancing in wireless mesh networks," in *Proceedings of the IEEE International Symposium on Wireless Communication Systems (ISWCS '08)*, pp. 658–662, Reykjavik, Iseland, October 2008.

[38] T. S. Rappaport, *Wireless Communications: Principles and Practice*, Prentice Hall, New York, NY, USA, 2002.

[39] M. Z. Zamalloa and B. Krishnamachari, "An analysis of unreliability and asymmetry in low-power wireless links," *ACM Transactions on Sensor Networks*, vol. 3, no. 2, 2007.