Research Article

Analyzing Multimode Wireless Sensor Networks Using the Network Calculus

Xi Jin,[1] Nan Guan,[2,3] Jintao Wang,[1] and Peng Zeng[1]

[1]*Laboratory of Networked Control Systems, Shenyang Institute of Automation, Chinese Academy of Sciences,*
 114 Nanta Street, Shenyang 110016, China
[2]*Northeastern University, Shenyang 110819, China*
[3]*Uppsala University, 75105 Uppsala, Sweden*

Correspondence should be addressed to Xi Jin; jinxi@sia.cn and Peng Zeng; zp@sia.cn

The network calculus is a powerful tool to analyze the performance of wireless sensor networks. But the original network calculus can only model the single-mode wireless sensor network. In this paper, we combine the original network calculus with the multimode model to analyze the maximum delay bound of the flow of interest in the multimode wireless sensor network. There are two combined methods A-MM and N-MM. The method A-MM models the whole network as a multimode component, and the method N-MM models each node as a multimode component. We prove that the maximum delay bound computed by the method A-MM is tighter than or equal to that computed by the method N-MM. Experiments show that our proposed methods can significantly decrease the analytical delay bound comparing with the separate flow analysis method. For the large-scale wireless sensor network with 32 thousands of sensor nodes, our proposed methods can decrease about 70% of the analytical delay bound.

1. Introduction

Network calculus (NC) [1, 2] is a theory of deterministic queuing systems. NC analyzes the performance characteristics of the system based on min-plus algebra and has been widely used in numerous applications to provide deterministic delay and backlog bounds.

Some wireless sensor networks must meet high reliability and real-time requirements. For example, in industrial wireless sensor networks, sensor data must be transmitted to their destination before a deadline; otherwise, terrible disasters may happen. So the performance analysis of the data flow is very important at design time. NC is a powerful tool for this problem. There have been some works on using NC to analyze network performance, for example, [3–6].

However, these works only focus on single-mode systems, in which the workload and resource information is fixed. In this paper, our work is based on multimode wireless sensor networks. *Multimode* means that the workload and resource information is allowed to be different for different modes in a network, and the mode change is determined by the

system information or external operations. Multimode wireless sensor networks are commonly used in real applications. For example, Figure 1 shows the wireless sensor network of a cement manufacturer. The working process of the wireless sensor network consists of three modes. The first one is the normal mode, in which sensor nodes and all equipment normally work. Then, to improve yield, the system changes to the full-load mode, in which the high-speed wireless sensor network is used to monitor the full-load equipment. Finally, if the rotary kiln is overheated due to the full-load working, the system changes to the high-temperature mode, in which the temperature-related data (part A in Figure 1) are delivered more frequently. The system changes from the first mode to the third mode in order. When the system is in the third mode and the temperature exceeds a specified threshold, workers will handle the problem.

In multimode wireless sensor networks, a variety of workload and resource information need to be considered simultaneously. But the original network calculus cannot model the multimode model due to its natural way. Some researchers combine the multimode model with the real-time
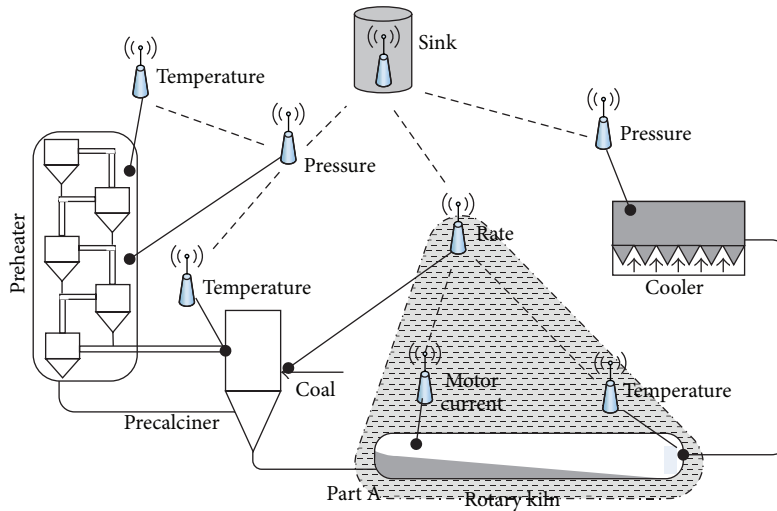
FIGURE 1: The wireless sensor network in a cement manufacturer.

calculus to analyze multimode embedded systems (e.g., [7–10]). The real-time calculus is the extended network calculus for performance analysis of real-time embedded system. But the system models of these works are not based on networks. Therefore, they cannot be used without modification. In this paper, we present a multimode wireless sensor network model and extend the network calculus to multimode networks. We propose two methods to compute the maximum delay bound of the flow of interest and discuss which one of the methods has the tighter delay bound. Experiments show that our proposed methods are effective and scalable.

The rest of paper is organized as follows: Section 2 introduces the related work. Section 3 reviews network calculus basics. Section 4 describes our multimode wireless sensor networks. Section 5 introduces our proposed end-to-end delay bound analysis methods and the corresponding comparative analysis. Section 6 shows evaluation results. Finally, Section 7 concludes this paper.

2. Related Work

The related work is classified into the following two categories: the analysis of networks and the analysis of multimode embedded systems.

2.1. The Analysis of Networks. Schmitt and Roedig present a series of works on the analysis of sensor networks. In [11], they extend the network calculus to sink-tree sensor networks and propose a general framework, called sensor network calculus, for analyzing the performance. The framework illustrates the various trade-offs between node power consumption, buffer requirement, and the data transfer delay bound. The main contribution of the general framework is the proposal that computes the total flow delay bound under the arbitrary multiplexing. The work in [12] focuses on the uncertain topologies. They compute the worst-case topology and then use the method of [11] to analyze sensor networks. In [13, 14], the sensor network calculus is extended to the networks with

multiple sink nodes and in-network processing, respectively. The work in [15] proposes the method Pay Multiplexing Only Once (PMOO) for arbitrary multiplexing networks, and in [16, 17] the method PMOO is illustrated more clearly. The work in [18] focuses on arbitrary multiplexing networks with FIFO per microflow. The work in [5] uses the linear program to compute the end-to-end delay bound.

In [6], Bouillard et al. also propose an analysis method based on the linear program, which can be used in any feed-forward network under blind multiplexing. In [19], the same authors study the exact worst-case delay for FIFO multiplexing tandem networks and propose the analysis method that only requires two linear problems to compute the upper and lower delay bounds.

In [4], the authors propose the separated flow analysis (SFA) method for sensor network calculus. SFA can be adopted in general feed-forward networks. It makes use of the concatenation theorem, which provides a tighter delay bound than additive per-node bounds. In [20, 21], the same authors extend the work of [4] to allocate the implicit guaranteed time slots and balance the trade-off between energy efficiency and delay guarantees.

In addition, some researchers make use of stochastic network calculus (e.g., [22–26]) to analyze the performance. But in this paper we only focus on the deterministic network calculus. Using the stochastic network calculus to analyze multimode networks will be studied in our future work.

These prior works do not consider the networks with multiple modes. In [27], sensor nodes exhibit different operative modes. Each mode has the given requirement in terms of energy and bandwidth. The service and arrival curves represent the incoming bandwidth and the bandwidth requirement. Then the bandwidth allocation and the mode assignment are formulated as two nonlinear optimization problems, and the service and arrival curves are used as the constraints of optimization problems. This work focuses on multimode sensor nodes but does not consider the performance analysis.

2.2. The Analysis of Multimode Embedded Systems. Some researchers combine the real-time calculus (RTC) with the multimode model to analyze the performance of embedded systems. Phan et al. have done a lot of works ([7–9, 28]) about it. In [7], the real-time calculus is used to describe the workload and service models, and the event count-automata (ECA) [29] explicates the state-based models. These heterogeneous models are connected by specified interfaces. The hybrid method is more expressive and efficient comparing to the pure RTC and ECA. In [8], the authors extend RTC to the timed automata, which is introduced to model the multimode information. The method combining RTC and the timed automata is more universal for all the components in embedded systems. But it faces the problem of state space explosion, since the workload and resource information of each processing element is modeled as a automata, respectively. In [9], the authors propose the richer processing semantics to represent different scheduling policies and the dynamic characteristics of data/event streams. In [28], the authors extend the work of [9] to the system that executes multiple multimode applications concurrently.

Lampka et al. propose a hybrid framework ([10, 30]), which describes the state-based component as the timed automata and the state-less component as the real-time calculus. The hybrid method can balance the expressiveness and scalability in the analysis of complex systems. And in [31], they introduce properties established to the component interface. This method can support a rigorous compositional design methodology for a complex system. Then in [32], the authors embed the methods described in [7, 10] into a framework for system-level performance analysis. However, these system models cannot be used in the analysis of wireless sensor networks.

3. Network Calculus Basics

Network calculus is a mathematical tool for analyzing the system performance. The functions involved in this paper belong to the space $\mathscr{F}$:

$$\mathscr{F} = \left\{ f : \mathbb{R}^+ \longrightarrow \mathbb{R}^+, \ \forall t \geqslant s : f(t) \geqslant f(s), \ f(0) = 0 \right\}. \tag{1}$$

They are nonnegative and wide-sense increasing. For a given data flow, the input cumulative function $R(t)$ ($R(t) \in \mathscr{F}$) is used to denote the number of bits that arrive in the time interval $[0, t]$. The output cumulative function $C(t)$ ($C(t) \in \mathscr{F}$) means the number of bits which the system resource can process in the time interval $[0, t]$. In a *backlogged period*, for any t belonging to the backlogged period $R(t)$ is greater than $C(t)$. Based on the two functions $R(t)$ and $C(t)$, NC uses *arrival curves* and *service curves* to model the system workload and resource information.

Definition 1 (arrival curve). Given a function $R(t)$, the arrival curve α ($\alpha \in \mathscr{F}$) satisfies

$$\forall s, \ 0 \leqslant s \leqslant t, \quad R(t) - R(s) \leqslant \alpha(t - s). \tag{2}$$
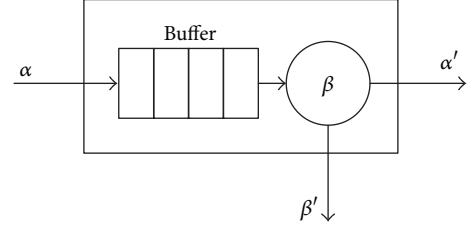

Figure 2: Component model.

Definition 2 (strict service curve). Given a function $C(t)$, the strict service curve β ($\beta \in \mathscr{F}$) satisfies

$$\text{for any backlogged period }]s, t[, \quad C(t) - C(s) \geqslant \beta(t - s). \tag{3}$$

The affine function $\gamma_{r,b}(t) = b + rt$ $(t > 0)$ and the rate-latency function $\beta_{R,T}(t) = R[t - T]^+$ are the typical arrival and service curves. The operator $[\ ]^+$ is defined as $[x]^+ = \max(0, x)$.

NC makes use of the min-plus operations convolution and deconvolution to analyze the system performance.

Definition 3 (min-plus convolution and deconvolution). Let f and g be two curves. The min-plus convolution $\otimes$ and the min-plus deconvolution $\oslash$ are defined as

$$(f \otimes g)(t) = \inf_{0 \leqslant s \leqslant t} \left\{ f(t - s) + g(s) \right\},$$

$$(f \oslash g)(t) = \sup_{s \geqslant 0} \left\{ f(t + s) - g(s) \right\}. \tag{4}$$

For the convolution operation, its rule of *convex functions* is represented as follows.

Rule (Convex Functions) [2]. Let $f, g \in \mathscr{F}$. If f and g are convex, then $f \otimes g$ is convex. In particular if f, g are convex and piecewise linear, $f \otimes g$ is obtained by putting end to end the different linear pieces of f and g, sorted by increasing slopes.

Figure 2 shows the relations between curves. The input data denoted by α are stored in data buffer until they are processed by β. The maximum processed data is bounded by $\alpha' = \alpha \oslash \beta$, and the left-over service is $\beta' = [\beta - \alpha]^+$. For a flow, we consider it to be served in FIFO order. So given the arrival curve α and the service curve β, the maximum delay bound $D(\alpha, \beta)$ is expressed as follows:

$$D(\alpha, \beta) = \sup_{s \geqslant 0} \left\{ \inf \left\{ t \in [0, s] : \alpha(s - t) \leqslant \beta(s) \right\} \right\}$$

$$= \sup_{s \geqslant 0} \left\{ \inf \left\{ t \in [0, s] : (\alpha \oslash \beta)(-t) \leqslant 0 \right\} \right\}. \tag{5}$$

Figure 3 shows an example, where the arrival curve is $\gamma_{1,10}$ and the service curve is $\beta_{2,10}$. $D(\alpha, \beta)$ is the maximal horizontal distance between the two curves.
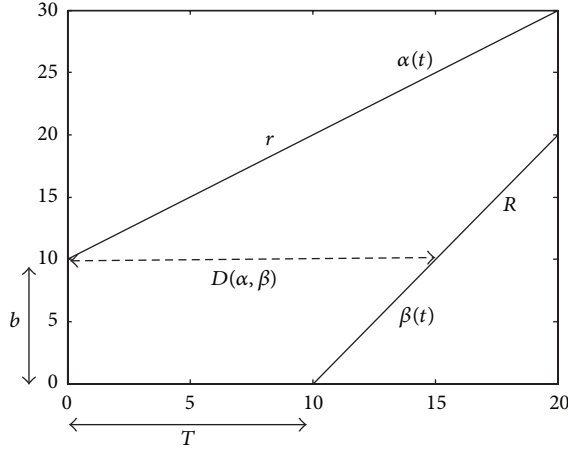
FIGURE 3: Illustration of $D(\alpha, \beta)$.

FIGURE 4: A mode graph.

4. System Model and Problem Statement

The key notations used in this paper are summarized in the Appendix.

In our system, a sink-tree wireless sensor network is characterised by a tuple $W = \langle N, E \rangle$.

(i) $N = \{n_1, n_2, \ldots\}$ is the set of sensor nodes, where n_1 is the root. A node n_i is characterised by $n_i = \langle \alpha_i, \beta_i, n_i^p, N_i^c \rangle$. We use the classical arrival curve γ_{r_i, b_i} and service curve β_{R_i, T_i} to describe the input data α_i and the supplied service β_i by the node n_i; that is, $\alpha_i = \gamma_{r_i, b_i}$ and $\beta_i = \beta_{R_i, T_i}$. Each node has a parent n_i^p and several children $N_i^c = \{n_1^c, n_2^c, \ldots\}$. Note that, for the sink node n_1, $n_1^p = \phi$, and, for leaf nodes, $N_i^c = \Phi$.

(ii) Matrix $E : N \times N$ is the set of links. If n_i sends data to n_j, the element e_{ij} in E is equal to 1; otherwise, $e_{ij} = 0$.

We use the source node to tag a data flow. The input data flow of the node n_i is f_i. The flow f_i is transmitted from the node n_i to its parent n_i^p and then to the parent of the node n_i^p, until the flow arrives at the sink node. The path of the flow f_i is denoted by $p_i = \{n_i, n_i^p, \ldots, n_1\}$.

The wireless sensor network consists of multiple work modes. External signals or the amount of running time triggers the mode change. A mode graph can fully express this system information. The multimode wireless sensor network is modeled by the mode graph $G = \langle M, \mathscr{E}, U, \Lambda \rangle$.

(i) M is the set of all modes. In each mode m_i, two curve sets α^i and β^i are used to represent all sensor nodes' workload and resource information. The curves of the node n_j in the mode m_i are represented by α_j^i and β_j^i. m_0 is the initial mode.

(ii) $\mathscr{E}$ denotes the set of external signals, for example, high-temperature alarms and manual operations.

(iii) U is the set of the upper bounds of running time. Each element U_i represents the maximum amount of time that is spent in the mode m_i. Note that U_i can be infinity.
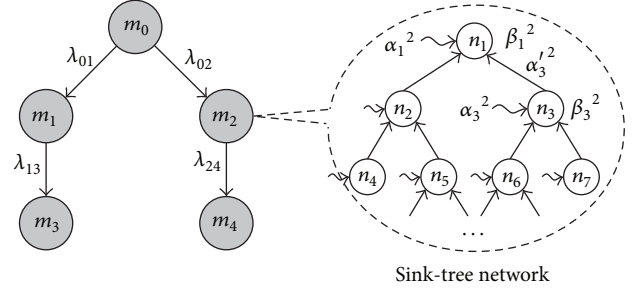
(iv) $\Lambda \subseteq M \times \mathscr{E} \times U \times M$ is the set of mode change relations. An element $\lambda_{ij} = (m_i, \mathscr{E}_i, U_i, m_j)$ denotes a change from the mode m_i to the mode m_j. If the amount of time spent in the mode m_i exceeds its upper bound of running time U_i, the wireless sensor network must change to the mode m_j; otherwise, only external signals $\mathscr{E}_i$ can trigger the mode change.

Figure 4 shows an example. The mode graph consists of five modes. Each mode corresponds to the given arrival and service curves. The network starts from the mode m_0 and then changes to the mode m_1 or m_2 due to external signals $\mathscr{E}$ or too much running time. Similarly, other mode changes can be triggered by some conditions.

The problem that we address is stated as follows. Given the wireless sensor network W and the corresponding mode graph G, our objective is to analyze the maximum delay bound of the flow of interest, such that the system performance can be tested at design time.

5. Analysis

In this section, we combine the network calculus and the multimode model to analyze the end-to-end delay bound of multimode wireless sensor networks. We first introduce using the network calculus to analyze one component with multiple modes and then extend it to the wireless sensor network.

5.1. Network Calculus for One Component with Multiple Modes. Assume that in our system model there is only one component, which works in different modes. The mode changes are described in the mode graph G.

The component in each mode has the given arrival and service curves. When the component enters the mode m_j from the mode m_i, the backlog and the supplied resources by the mode m_i will impact on the component execution in the mode m_j. So the original arrival and service curves cannot fully represent the component's information. In this paper, we call the curves that describe the maximum workload and minimum service information in any time interval before the component leaves the mode m_j the *multimode curves* and use α^* and β^* to denote them.

When the component is in the initial mode m_0, its multimode arrival and service curves are the original curves; that is, $\alpha^{*0} = \alpha^0$ and $\beta^{*0} = \beta^0$. Then we compute multimode

curves of other modes based on mathematical induction. We define the path from the initial mode to the mode m_j as $\mathscr{P}_j = \{m_0, \ldots, m_j\}$. Assume that when the component is in the mode m_i we have obtained its multimode curves α^{*i} and β^{*i}. If there exists the mode change λ_{ij}, then the multimode arrival and service curves, when the component is in the mode m_j, are denoted as follows:

$$\alpha^{*j}(t) = \max_{0 \leqslant s \leqslant U^i, 0 \leqslant t-s \leqslant U_j} \left\{ \alpha^{*i}(s) + \alpha^j(t-s) \right\}, \quad (6)$$

$$\beta^{*j}(t) = \min_{0 \leqslant s \leqslant U^i, 0 \leqslant t-s \leqslant U_j} \left\{ \beta^{*i}(s) + \beta^j(t-s) \right\}, \quad (7)$$

where $U^i = \sum_{\forall m_g \in \mathscr{P}_i} U_g$ represents the latest time instance when the component enters the mode m_j; that is, it is the maximum time interval before the component enters the mode m_j. The maximum amount of data that arrive in the last t time units consists of two parts. The first part is the arrival data in the last s time units before the component enters the mode m_j and can be obtained by the multimode arrival curve α^{*i}. The s time units must not be larger than the maximum time units U^i. The second part is the arrival data when the component is in the mode m_j. The $(t - s)$ time units cannot be larger than the maximum amount of time that is spent in the mode m_j; otherwise the component will change to the next mode. The maximum sum of the two parts is the upper bound of the arrival data in any t time units before the component leaves the mode m_j; that is, α^{*j} is the multimode arrival curve when the component is in the mode m_j. Similarly, the multimode service curve β^{*j} can be computed using the same way.

Assume that some data arrive when the component is in the mode m_i. Due to the mode change, these data may be processed in the mode m_i or its any following mode. We use D^{ig} to denote the delay bound when the data are processed in the mode m_g. So

$$D^{ig}\left(\alpha^{*i}, \beta^{*g}\right)$$

$$= \sup_{0 \leqslant t+d \leqslant U^g} \left\{ \inf \left\{ d : \alpha^{*i}(t) \leqslant \beta^{*g}(t+d), \; t \in \left[0, U^i\right] \right\} \right\}. \quad (8)$$

We use ζ_i to denote all following modes of the mode m_i. Then the end-to-end delay bound of the data flow is denoted as

$$\max_{\forall m_i \in M, \forall m_g \in \{\zeta_i + m_i\}} \left\{ D^{ig} \right\}. \quad (9)$$

In the following, we analyze the end-to-end delay bound of the multimode wireless sensor networks based on the above result.

5.2. End-to-End Delay Bound Analysis. The *separate flow analysis* (SFA) [4] is the classical delay bound analysis method for general single-mode feed-forward networks. For the flow of interest, SFA first computes the left-over service curves
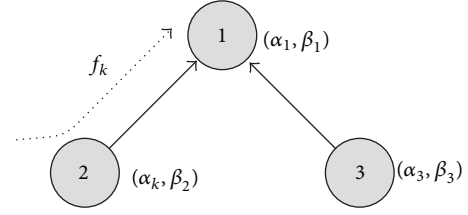

FIGURE 5: A sink-tree network.

for all nodes on the path of the flow and then computes the convolution of them as the service curve of the flow of interest. For example, a simple sink tree is shown in Figure 5. According to the analysis method SFA, the end-to-end delay bound of the flow of interest f_k is $D_{\text{SFA}} = D(\alpha_k, \beta'_2 \otimes \beta'_1)$.

Note that the left-over service curve is independent of the arrival curve of the flow of interest f_k. For brevity, in the remaining of the Section 5.2 the arrival curves of the flow α_k are not involved in the computation of left-over service curves.

We extend the topology of sink-tree network to any depth. The computation of left-over service curves in the path p_k is shown as follows:

$$\forall n_i \in p_k, \quad \beta'_i = \left[\beta_i - \alpha_i - \sum_{\forall n_j \in N_i^c} \alpha'_j \right]^+, \quad (10)$$

where

$$\alpha'_j = \left(\alpha_j + \sum_{\forall n_g \in N_j^c} \alpha'_g \right) \oslash \beta_j. \quad (11)$$

Recall that $\alpha_i = \gamma_{r_i, b_i}$ and $\beta_i = \beta_{R_i, T_i}$. Then we obtain

$$\forall n_i \in p_k, \quad \beta'_i = R'_i \left[t - T'_i \right]^+.$$

$$\alpha'_i = b'_i + r'_i t, \quad (12)$$

where

$$R'_i = R_i - r_i - \sum_{\forall n_j \in N_i^c} r'_j, \quad (13)$$

$$r'_j = r_j + \sum_{\forall n_g \in N_j^c} r'_g, \quad (14)$$

$$T'_i = \frac{R_i T_i + b_i + \sum_{\forall n_j \in N_i^c} b'_j}{R'_i}, \quad (15)$$

$$b'_j = b_j + \sum_{\forall n_g \in N_j^c} b'_g + r'_j T_j. \quad (16)$$

For example, in Figure 5,

$$\beta'_1 = (R_1 - r_1 - r_3) \left[t - \frac{R_1 T_1 + b_1 + b_3 + r_3 T_3}{R_1 - r_1 - r_3} \right]^+. \quad (17)$$

SFA focuses on signal-mode wireless sensor networks. We combine it with the multimode model to analyze multimode wireless sensor networks.

The delay bound of the flow of interest f_k is determined by its multimode arrival curve α_k^{*i} and the supplied service in each mode. The multimode arrival curve relates only to the node n_k. The node can be modeled as a multimode component. So the computation of its multimode arrival curves is the same as that of one multimode component (shown in Section 5.1); that is, we can use (6) and the mode graph G to compute multimode arrival curves α_k^{*i} ($\forall m_i \in M$).

The computation of the supplied service relates to all nodes in the path p_k. We can model either each of these nodes or all of them as a multimode component. This leads to two methods N-MM and A-MM. In the following, we will introduce them and discuss which one of the methods is more effective.

5.2.1. N-MM. In this method, each node in the path p_k is modeled as a multimode component. We first compute their multimode service curves and then convolute the curves of all nodes in the same mode as the multimode service curve of the whole network in the mode. We use $\overline{\beta}_i^j$ to denote the multimode service curve of the node n_i in the mode m_j and $\overline{\beta}^j$ to denote the multimode service curve of the whole network in the mode m_j. We formulate the steps of the computation as follows.

Step 1. For all $n_i \in p_k$, $m_j \in M$,

$$\overline{\beta}_i^j(t) = \min_{0 \leqslant x \leqslant U^{j-1}, 0 \leqslant t-x \leqslant U_j} \left\{ \overline{\beta}_i^{j-1}(x) + \beta_i'^j(t-x) \right\}, \quad (18)$$

where we assume that the mode m_{j-1} is the parent node of the mode m_j. $\beta_i'^j$ denotes the left-over service curve of n_i in the mode m_j and is computed based on (10) and the given α^i, β^i in the network model.

Step 2. For all $m_j \in M$,

$$\overline{\beta}^j(t) = \bigotimes_{\forall n_i \in p_k} \overline{\beta}_i^j(t). \quad (19)$$

From the definition of the convolution operation, we know that $\overline{\beta}_i^j$ can also be represented as $\beta_i'^j \otimes \overline{\beta}_i^{j-1}$, where their ranges are limited in $[0, U_j]$ and $[0, U^{j-1}]$, respectively. Then, we have

$$\overline{\beta}_i^j(t) = \bigotimes_{\forall m_g \in \mathscr{P}_j} \beta_i'^g(t) \quad \left(\text{for each } \beta_i'^g(x), \ x \in [0, U_g] \right), \quad (20)$$

$$\overline{\beta}^j(t) = \bigotimes_{\forall m_g \in \mathscr{P}_j, \forall n_i \in p_k} \beta_i'^g(t) \quad (21)$$

$$\left(\text{for each } \beta_i'^g(x), \ x \in [0, U_g] \right).$$

So, based on (8) and (9), the end-to-end delay bound of the flow f_k is

$$\max_{\forall m_j \in M, \forall m_g \in \{\varsigma_j + m_j\}} \left\{ D^{jg}\left(\alpha_k^{*j}, \overline{\beta}^g \right) \right\}. \quad (22)$$

5.2.2. A-MM. Contrary to the method N-MM, the method A-MM first convolutes all of original left-over service curves that are in the same mode; that is, it models the whole network as a multimode component. Then the computation of multimode curves is the same as that in a component. We use β'^j to denote the left-over service curves of the whole network in the mode m_j and $\widehat{\beta}^j$ to denote the whole multimode service curve, which corresponds to $\overline{\beta}^j$ of the method N-MM. The formulations are shown as follows.

Step 1. For all $m_j \in M$,

$$\beta'^j(t) = \bigotimes_{\forall n_i \in p_k} \beta_i'^j(t). \quad (23)$$

Step 2. For all $m_j \in M$,

$$\widehat{\beta}^j(t) = \min_{0 \leqslant x \leqslant U^{j-1}, 0 \leqslant t-x \leqslant U_j} \left\{ \widehat{\beta}^{j-1}(x) + \beta_i'^j(t-x) \right\}, \quad (24)$$

where the mode m_{j-1} is the parent node of the mode m_j.

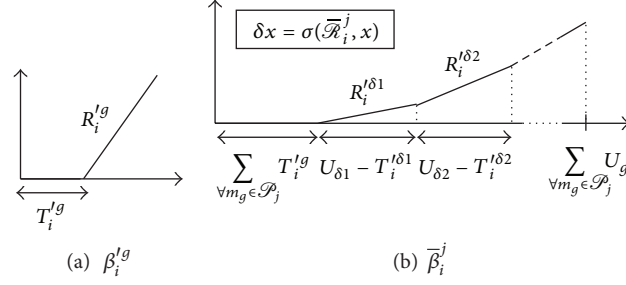Similarly, according to the definition of the convolution operation, we have

$$\widehat{\beta}^j(t) = \bigotimes_{\forall m_g \in \mathscr{P}_j} \beta'^g(t), \quad \left(\text{for each } \beta'^g(x), \ x \in [0, U_g] \right). \quad (25)$$

Then, the delay bound of the flow f_k computed by the method A-MM is

$$\max_{\forall m_j \in M, \forall m_g \in \{\varsigma_j + m_j\}} \left\{ D^{jg}\left(\alpha_k^{*j}, \widehat{\beta}^g \right) \right\}. \quad (26)$$

In the following, we discuss which one of the methods is more effective. The delay bound is the maximal horizontal distance between the arrival curve and the service curve. The arrival curves used in the two methods are the same. So the service curves of the two methods determine which one of the methods is effective.

Recall that our system model is based on the typical affine arrival curve and rate-latency service curve. So from (20), (21), (23), (25), and the rule *Convex functions*, we know that the two methods are based on the convolution operation and all service curves are piecewise linear and convex. Then, based on the rule *Convex functions*, we analyze $\overline{\beta}^j$ and $\widehat{\beta}^j$. For the method N-MM, Figure 6 shows service curves $\beta_i'^g$ and $\overline{\beta}_i^j$. $\overline{\beta}_i^j$ is the convolution of $\beta_i'^g$ that is limited in a range. So the length of the first segment is the sum of all delay $T_i'^g$ ($\forall m_g \in \mathscr{P}_j$ and $\forall n_i \in p_k$). The rate of the second segment is the minimum rate among the rate set $\overline{\mathscr{R}}_i^j = \{ R_i'^g \mid \forall m_g \in \mathscr{P}_j \}$,

$$\boxed{\delta x = \sigma(\overline{\mathscr{R}}_i^{\,j}, x)}$$

(a) $\beta_i'^{\,g}$

(b) $\overline{\beta}_i^{\,j}$

FIGURE 6: Illustration of $\beta_i'^{\,g}$ and $\overline{\beta}_i^{\,j}$.

$$\boxed{(\delta x, hx) = \sigma(\overline{\mathscr{R}}^{\,j}, x)}$$
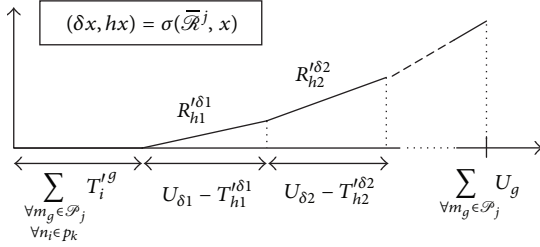
FIGURE 7: Illustration of $\overline{\beta}^{\,j}$.

where $R_i'^{\,g}$ can be computed based on (13). We use $\sigma(\overline{\mathscr{R}}_i^{\,j}, x)$ (δx for short) to denote the mode $m_{\delta x}$ with the xth minimum rate in the rate set $\overline{\mathscr{R}}_i^{\,j}$. So the rate of the xth segment in Figure 6(b) is $R_i'^{\,\delta(x-1)}$, and the length of the projection of the xth segment onto the horizontal axis is the difference between the range $U_{\delta(x-1)}$, which is limited in (20), and the delay $T_i'^{\,\delta(x-1)}$. Similarly, we can obtain $\overline{\beta}^{\,j}$ (shown in Figure 7) based on $\overline{\beta}_i^{\,j}$ ($\forall n_i \in p_k$). The length of the first segment is also the sum of all delay $T_i'^{\,g}$. For the rate set $\overline{\mathscr{R}}^{\,j} = \{\overline{\mathscr{R}}_i^{\,j} \mid \forall n_i \in p_k\}$, $\sigma(\overline{\mathscr{R}}^{\,j}, x)$ returns $(\delta x, hx)$, which denotes that the xth minimum rate belongs to the mode $m_{\delta x}$ of the sensor node n_{hx}.

For the method A-MM, $\beta'^{\,j}$ is not limited by any time constraint (shown in Figure 8(a)), and its rate $R'^{\,j}$ is the minimum rate among all rates of the mode m_j. The computation of $\widehat{\beta}^{\,j}$ limits the range of $\beta'^{\,j}$. So $\widehat{\beta}^{\,j}$ also contains multiple segments as shown in Figure 8(b), where $\widehat{\mathscr{R}}^{\,j} = \{R'^{\,g} \mid \forall m_g \in \mathscr{P}_j\}$.

Comparing with the two methods, we obtain three properties as follows.

Property 1. The two first segments of $\overline{\beta}^{\,j}$ and $\widehat{\beta}^{\,j}$ are the same and they are equal to $\sum_{\forall m_g \in \mathscr{P}_j, \forall n_i \in p_k} T_i'^{\,g}$.

$\overline{\mathscr{R}}^{\,j}$ is the rate set for all sensor nodes before the network leaves the mode m_j, and $\widehat{\mathscr{R}}^{\,j}$ is a subset of $\overline{\mathscr{R}}^{\,j}$. We can get the following properties.

Property 2. The xth minimum rate of $\overline{\mathscr{R}}^{\,j}$ must be less than or equal to the xth minimum rate of $\widehat{\mathscr{R}}^{\,j}$.

In $\overline{\beta}^{\,j}$, on the horizontal axis the length of the segment with $R_b'^{\,a}$ is $U_a - T_b'^{\,a}$, and in $\widehat{\beta}^{\,j}$ the length of the segment is $U_a - \sum_{\forall n_i \in p_k} T_i'^{\,a}$. We know that $n_b \in p_k$. So $T_b'^{\,a} \leqslant \sum_{\forall n_i \in p_k} T_i'^{\,a}$. Then we have the following.

Property 3. If a rate $R_b'^{\,a}$ occurs in both $\overline{\beta}^{\,j}$ and $\widehat{\beta}^{\,j}$, the corresponding segment's length on the horizontal axis in $\overline{\beta}^{\,j}$ is longer than or equal to that in $\widehat{\beta}^{\,j}$.

According to the three properties, we can prove the following theorem.

Theorem 4. *Given a wireless sensor network and its corresponding mode graph, the maximum delay bound computed by the method A-MM is tighter than or equal to that computed by the method N-MM.*

Proof. According to Property 1, the first segments of the two methods are the same. And the rates of their second segments are also the same. Since $\overline{\mathscr{R}}^{\,j}$ can be written as $\{R_i'^{\,g} \mid \forall n_i \in p_k, \forall m_g \in \mathscr{P}_j\}$, then

$$\sigma\left(\overline{\mathscr{R}}^{\,j}, 1\right) = \arg\left(\min_{\forall m_g \in \mathscr{P}_j, \forall n_i \in p_k} \left\{R_i'^{\,g}\right\}\right), \tag{27}$$

while

$$\sigma\left(\widehat{\mathscr{R}}^{\,j}, 1\right) = \arg\left(\min_{\forall m_g \in \mathscr{P}_j} \left\{R'^{\,g}\right\}\right)$$
$$= \arg\left(\min_{\forall m_g \in \mathscr{P}_j, \forall n_i \in p_k} \left\{R_i'^{\,g}\right\}\right). \tag{28}$$

And according to Property 3, the length of the second segment of $\overline{\beta}^{\,j}$ is longer than or equal to that of $\widehat{\beta}^{\,j}$. So in the first and second segments of $\overline{\beta}^{\,j}$, $\overline{\beta}^{\,j} \leqslant \widehat{\beta}^{\,j}$.

After the second segment, we assume that there exists z such that $\overline{\beta}^{\,j}(z) > \widehat{\beta}^{\,j}(z)$ (shown in Figure 9). Before z, there must exist a value $y = \sup\{z' \mid z' \leqslant z, \overline{\beta}^{\,j}(z') = \widehat{\beta}^{\,j}(z')\}$, which belongs to the $\overline{x}$th segment of $\overline{\beta}^{\,j}$ and the $\widehat{x}$th segment of $\widehat{\beta}^{\,j}$, and the rate of the $\overline{x}$th segment is larger than that of the $\widehat{x}$th segment. According to Property 2, we know $\overline{x} > \widehat{x}$. Then

(a) β'^j

(b) $\widehat{\beta}^j$
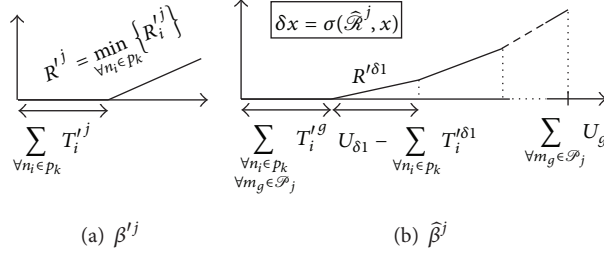
FIGURE 8: Illustration of β'^j and $\widehat{\beta}^j$.
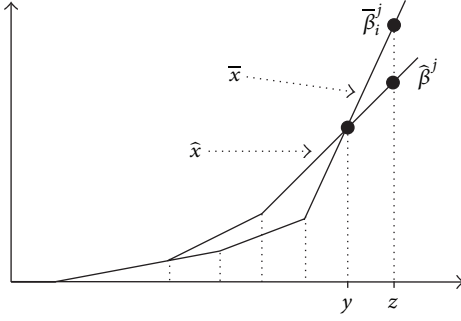
FIGURE 9: Comparing with $\overline{\beta}^j$ and $\widehat{\beta}^j$.

we have $\overline{x} - 1 \geqslant \widehat{x}$. Since $\widehat{\mathscr{R}}^j$ is the subset of $\overline{\mathscr{R}}^j$, the rates of the second segment to $\widehat{x}$th segments of $\widehat{\beta}^j$ must have occurred in prior $\overline{x} - 1$ segments of $\overline{\beta}^j$. Then, according to Property 3, the sum of lengths of $\overline{x} - 1$ segments of $\overline{\beta}^j$ is longer than or equal to the sum of lengths of $\widehat{x}$ segments of $\widehat{\beta}^j$; that is, the $\overline{x}$th segment and the $\widehat{x}$th segment cannot intersect at y. Therefore, there are no y and z.

So, for each i on the horizontal axis, $\widehat{\beta}^j(i) \geqslant \overline{\beta}^j(i)$. The horizontal distance between the multimode arrival curve and $\widehat{\beta}^j(i)$ is less than or equal to the horizontal distance between the multimode arrival curve and $\overline{\beta}^j(i)$; that is, the maximum delay bound computed by the method A-MM is tighter than or equal to that computed by the method N-MM. $\square$

We further illustrate the effectiveness of the method A-MM as follows. In the method N-MM, the multimode service curve $\overline{\beta}_i^j$ represents the minimum service of the node n_i in all modes of the path $\mathscr{P}_j$. When multiple nodes are convoluted, the segments of different nodes in the same mode may be put together. It means that a flow is supplied for the minimum service of a sensor node in the mode m_j, and in other sensor nodes it also suffers from the minimum service of the same mode m_j. However, there are no constraints to limit the sum of the amount of time, in which the flow is supplied for the service of the mode m_j. This leads to the fact that the sum of the amount of time may exceed the upper bound U_j. And the cause of this case is that the service supplied by the mode m_j is minimum. But, in the method A-MM, the flow cannot suffer from the minimum service that is supplied after the time instance U_j, because of the time interval constraints of

(25). So the multimode service curve of the method A-MM is larger than or equal to that of the method N-MM.

For the example of Figure 5, we set that $R_1'^1 = 2$, $R_1'^2 = 4$, $R_2'^1 = 1$, $R_2'^2 = 3$, $U_1 = U_2 = 10$, and all T' are equal to 2. The computations of $\widehat{\beta}^2$ and $\overline{\beta}^2$ are shown in Figures 10(a) and 10(b). From these figures, we know that in the method N-MM the two segments of the mode 1 are used in $\overline{\beta}^2$, and the sum of the amount of time exceeds the maximum amount of time U_1. This case is impossible in the method A-MM. The method N-MM introduces more pessimism.

6. Evaluation

There are no previous works on multimode networks. Therefore, we compare our proposed analysis methods with the original single-mode analysis method SFA [4]. These methods are implemented in RTC Toolbox [33]. RTC Toolbox is a set of MATLAB libraries and provides basic min-plus and max-plus operations. All experiments run on a desktop computer with 3.4 GHz Intel Core i7 processor and 4 GB memory.

We adopt the fully occupied binary tree, where in different modes all nodes have the same latency T and the same burst b. The arrival rate $r = \omega \times i$, where i denotes the ith mode and ω is a coefficient. We use the service rate R to limit the node's utilization. For each node, its service rate R is equal to the quotient of its arrival rate (14) and the utilization. The mode graph consists of three successive modes, and all modes have the same U_i. The method SFA cannot analyze multimode networks. In order to use it in the evaluation, the multimode model must be changed to the single-mode model, in which each node consists of one service curve and one arrival curve. So the arrival curve of the single-mode network is the maximum arrival curve between the three modes and the service curve is the minimum service curve. Then the method SFA can analyze the delay bound of the flow of interest based on the single-mode model. Only in this way the delay analyzed by the method SFA is the reliable delay bound.

Figure 11 shows the delay bound comparison between the different configurations when the tree depth varies. In the single-mode binary tree network, the arrival rate r is $3 \times \omega$ and the service rate R is the quotient of ω and the utilization. In order to make the network analyzable by the method SFA, the service rate R must be greater than the arrival rate r. So the
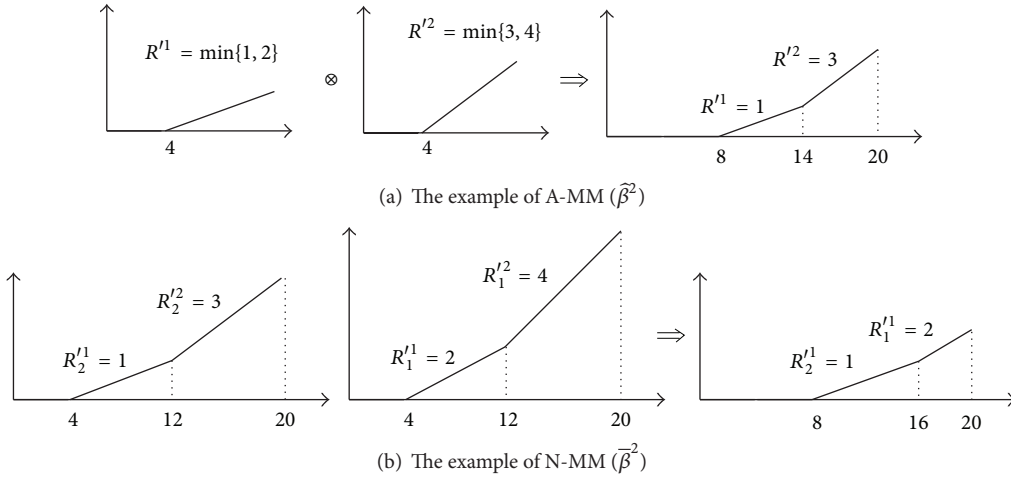
(a) The example of A-MM ($\widehat{\beta}^2$)

(b) The example of N-MM ($\overline{\beta}^2$)

Figure 10: Comparing with A-MM and N-MM.

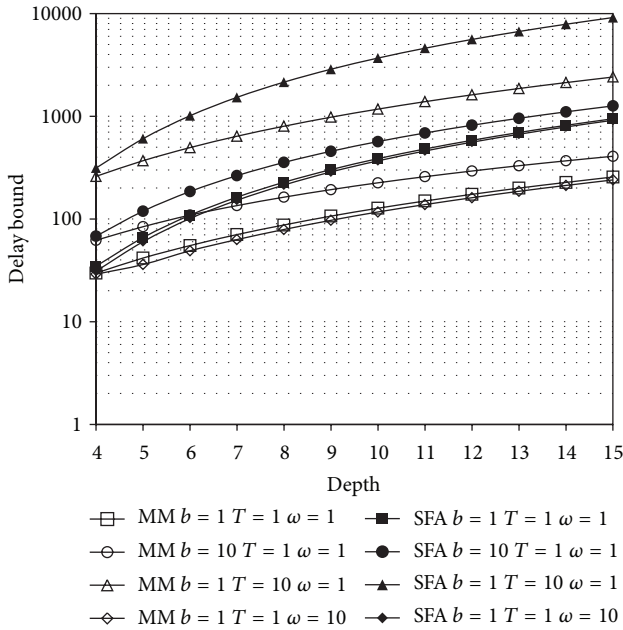Figure 11: The end-to-end delay bound comparison with different depths.

Figure 12: The ratio of MM to SFA with different depths.

utilization is set as 0.3. Under this utilization configuration, the results of A-MM and N-MM are the same. So we use MM to tag them in Figure 11. U_i varies as the depth increases, and it is set as just enough for which any flow can be delivered to the destination in any two successive modes. From the figure, we have the following.

(i) The delay bound of MM is tighter than that of SFA, since MM supplies the fine-grained model for analysis.

(ii) The more the tree depth, the longer the end-to-end delay bound. It is because that the increase of nodes leads to the long latency T' (15) and burst b' (16). From
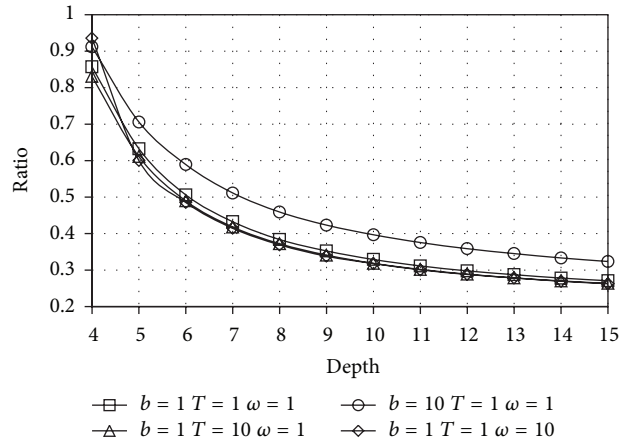
Figure 3, we know that these parameters relate closely to the delay bound.

(iii) The parameter T has more impact on the delay bound. It is because that in (15) and (16) its coefficient is r'_j/R' ($r'_j \geq 1$), while the coefficients of other parameters are $1/R'$ and T_j/R' (when r becomes $T_j = 1$).

(iv) Only the increase of ω leads to the shorter delay bound. In this case, b' increases, but R' expands ten times and T' decreases. So the delay bound decreases slightly when ω changes from 1 to 10.

Figure 12 shows the ratio of MM to SFA. SFA introduces more pessimism to the complex system. So the ratio decreases as the depth increases. The delay bound of MM is about 30% of the delay bound of SFA when the large-scale network contains 32 thousands of sensor nodes. The execution time of all programs is less than 1 minute. Therefore, our proposed analysis methods are thigh and scalable.
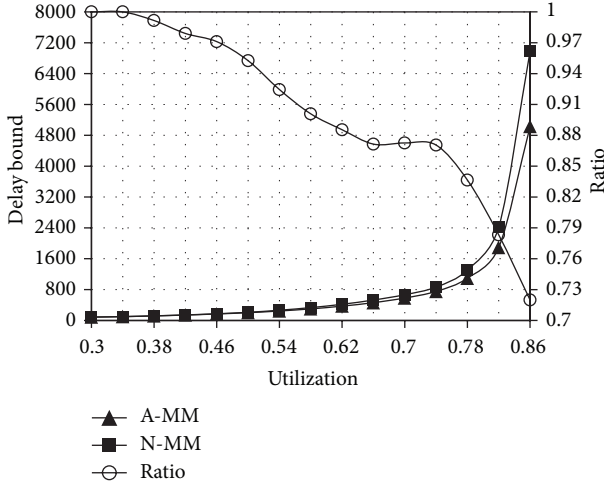
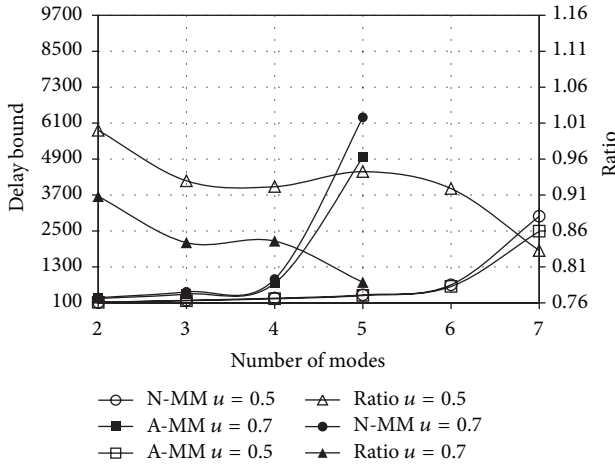FIGURE 13: The end-to-end delay bound comparison and the ratio of A-MM to N-MM with different utilizations.

FIGURE 14: The comparison with different numbers of modes.

first method N-MM models each sensor node in the path of the flow of interest as a multimode component; the second method A-MM models the whole network as a multimode component. Comparing with the two methods, we obtain three properties. According to these properties and the rule of convolution operation, we prove that the delay bound computed by the method A-MM is tighter than or equal to that computed by the method N-MM. Experiments show that, comparing with the analysis method SFA, our proposed methods can be used in high utilization and large-scale wireless sensor networks and can significantly decrease the analytical delay bound of the flow of interest (about 70% for the network with 32 thousands of sensor nodes). And in the future work we will extend the stochastic network calculus to multimode wireless sensor networks.

Appendix

Key Notations

$N = \{n_1, \dots\}$: The set of sensor nodes

p_i: The path from the root n_1 to n_i

$M = \{m_1, \dots\}$: The set of modes

$\mathscr{P}_i$: The path from m_0 to m_i

U_i: The maximum amount of time spent in the mode m_i

U^i: The latest time instance when the component leaves m_i

λ_{ij}: The change from m_i to m_j

α^{*i}, β^{*i}: Multimode curves for one component in the mode m_i

α_k^{*j}: The multimode arrival curve of the flow f_k in the mode m_j

$\beta_i'^{j}$: The left-over service curve of n_i in the mode m_j

$\overline{\beta}_i^{j}$: The multimode service curve of n_i in the mode m_j

$\overline{\beta}^{j}$: The multimode service curve of the whole network computed by N-MM

β'^i: The left-over service curve of the whole network

$\widehat{\beta}^i$: The multimode service curve of the whole network computed by A-MM

$\overline{\mathscr{R}}_i^{j}$: The set of the appeared rates before n_i leaves the mode m_j

$\overline{\mathscr{R}}^{j}$: The set of all rates computed by N-MM

$\widehat{\mathscr{R}}^{j}$: The set of all rates computed by A-MM.

In the following, we compare our proposed methods A-MM and N-MM. There is no SFA, since SFA cannot analyze the system with high utilization. Figure 13 shows the delay bound comparison with different utilizations when the tree depth is 8. And Figure 14 shows the delay bound comparison with different numbers of modes. The utilization is short for u. As our analyzed results, the delay bound of A-MM is tighter than that of N-MM, and N-MM is more pessimistic in the complex system.

7. Conclusion

Multimode wireless sensor networks are commonly used in real applications. In this paper, we combine the separate flow analysis, which is the classical delay bound analysis method based on the original network calculus for general single-mode feed-forward networks, and the multimode model to analyze the end-to-end delay bound of multimode wireless sensor networks. We propose two combined methods: the

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

References

[1] R. L. Cruz, "A calculus for network delay. I. Network elements in isolation," *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 114–131, 1991.

[2] J.-Y. Le Boudec and P. Thiran, *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet*, Springer, 2001.

[3] M. Fidler, "A survey of deterministic and stochastic service curve models in the network calculus," *IEEE Communications Surveys & Tutorials*, vol. 12, no. 1, pp. 59–86, 2010.

[4] 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)*, pp. 412–421, December 2006.

[5] J. B. Schmitt, F. A. Zdarsky, and M. Fidler, "Delay bounds under arbitrary multiplexing: when network calculus leaves you in the lurch...," in *Proceedings of the 27th IEEE International Conference on Computer Communications (INFOCOM '08)*, pp. 1669–1677, April 2008.

[6] A. Bouillard, L. Jouhet, and É. Thierry, "Tight performance bounds in the worst-case analysis of feed-forward networks," in *Proceedings of the IEEE 29th International Conference on Computer Communications (INFOCOM '10)*, pp. 1–9, March 2010.

[7] L. T. X. Phan, S. Chakraborty, P. S. Thiagarajan, and L. Thiele, "Composing functional and state-based performance models for analyzing heterogeneous real-time systems," in *Proceedings of the IEEE International Real-Time Systems Symposium (RTSS '07)*, pp. 343–352, December 2007.

[8] L. T. X. Phan, S. Chakraborty, and P. S. Thiagarajan, "A multi-mode real-time calculus," in *Proceedings of the 29th Real-Time Systems Symposium (RTSS '08)*, pp. 59–69, Barcelona, Spain, December 2008.

[9] L. T. Phan, S. Chakraborty, and I. Lee, "Timing analysis of mixed time/event-triggered multi-mode systems," in *Proceedings of the 30th Real-Time Systems Symposium (RTSS '09)*, pp. 271–280, December 2009.

[10] K. Lampka, S. Perathoner, and L. Thiele, "Analytic real-time analysis and timed automata: a hybrid method for analyzing embedded real-time systems," in *Proceedings of the 7th ACM International Conference on Embedded Software (EMSOFT '09)*, pp. 107–116, ACM, October 2009.

[11] J. B. Schmitt and U. Roedig, "Sensor network calculus—a framework for worst case analysis," in *Proceedings of the 1st IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS '05)*, pp. 141–154, July 2005.

[12] J. B. Schmitt and U. Roedig, "Worst case dimensioning of wireless sensor networks under uncertain topologies," in *Proceedings of the 1st Workshop on Resource Allocation in Wireless NETworks (RAWNET '05)*, Trento, Italy, 2005.

[13] J. B. Schmitt, F. A. Zdarsky, and U. Roedig, "Sensor network calculus with multiple sinks," in *Proceedings of the Performance Control in Wireless Sensor Networks Workshop at the IFIP Networking Conference*, pp. 6–13, Coimbra, Portugal, 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)*, pp. 193–202, December 2007.

[15] J. B. Schmitt, F. A. Zdarsky, and I. Martinovic, "Improving performance bounds in feed-forward networks by paying multiplexing only once," in *Proceedings of the 14th GI/ITG Conference on Measuring, Modelling and Evaluation of Computer and Communication Systems (MMB '08)*, pp. 1–15, April 2008.

[16] U. Roedig, N. Gollan, and J. B. Schmitt, "Validating the sensor network calculus by simulations," in *Proceedings of the 3rd International Conference on Wireless Internet*, pp. 34–41, October 2007.

[17] N. Gollan and J. B. Schmitt, "Energy-efficient TDMA design under real-time constraints in wireless sensor networks," in *Proceedings of the 15th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, pp. 80–87, October 2007.

[18] J. B. Schmitt, N. Gollan, S. Bondorf, and I. Martinovic, "Pay bursts only once holds for (some) non-FIFO systems," in *Proceedings of the 30th International Conference on Computer Communications*, pp. 1970–1978, IEEE, Shanghai, China, April 2011.

[19] A. Bouillard and G. Stea, "Exact worst-case delay for FIFO-multiplexing tandems," in *Proceedings of the 6th International ICST Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS '12)*, pp. 158–167, October 2012.

[20] A. Koubâa, M. Alves, E. Tovar, and A. Cunha, "An implicit GTS allocation mechanism in IEEE 802.15.4 for time-sensitive wireless sensor networks: Theory and practice," *Real-Time Systems*, vol. 39, no. 1–3, pp. 169–204, 2008.

[21] A. Koubaa, M. Alves, and E. Tovar, "Energy and delay trade-off of the GTS allocation mechanism in IEEE 802.15.4 for wireless sensor networks," *International Journal of Communication Systems*, vol. 20, no. 7, pp. 791–808, 2007.

[22] Y. Jiang and Y. Liu, *Stochastic Network Calculus*, Springer, 2008.

[23] Y. Jiang, "A basic stochastic network calculus," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 4, pp. 123–134, 2006.

[24] D. Khan, B. Nefzi, L. Santinelli, and Y. Song, "Probabilistic bandwidth assignment in wireless sensor networks," in *Wireless Algorithms, Systems, and Applications*, vol. 7405 of *Lecture Notes in Computer Science*, pp. 631–647, Springer, Berlin, Germany, 2012.

[25] K. Angrishi, "An end-to-end stochastic network calculus with effective bandwidth and effective capacity," *Computer Networks*, vol. 57, no. 1, pp. 78–84, 2013.

[26] H. Al-Zubaidy, J. Liebeherr, and A. Burchard, "A (min, ×) network calculus for multi-hop fading channels," in *Proceedings of the 32nd IEEE Conference on Computer Communications*, pp. 1833–1841, April 2013.

[27] L. Santinelli, M. Chitnis, C. Nastasi, F. Checconi, G. Lipari, and P. Pagano, "A component-based architecture for adaptive bandwidth allocation in wireless sensor networks," in *Proceedings of the 5th International Symposium on Industrial Embedded Systems (SIES '10)*, pp. 174–183, Trento, Italy, July 2010.

[28] L. T. Phan, I. Lee, and O. Sokolsky, "Compositional analysis of multi-mode systems," in *Proceedings of the 22nd Euromicro Conference on Real-Time Systems (ECRTS '10)*, pp. 197–206, July 2010.

[29] S. Chakraborty, L. T. X. Phan, and P. S. Thiagarajan, "Event count automata: a state-based model for stream processing systems," in *Proceedings of the 26th IEEE International Real-Time Systems Symposium (RTSS '05)*, pp. 12–23, IEEE, Miami, Fla, USA, December 2005.

[30] K. Lampka, S. Perathoner, and L. Thiele, "Analytic real-time analysis and timed automata: a hybrid methodology for the performance analysis of embedded real-time systems," *Design Automation for Embedded Systems*, vol. 14, no. 3, pp. 193–227, 2010.

[31] K. Lampka, S. Perathoner, and L. Thiele, "Component-based system design: analytic real-time interfaces for state-based component implementations," *International Journal on Software Tools for Technology Transfer*, vol. 15, no. 3, pp. 155–170, 2013.

[32] S. Perathoner, K. Lampka, and L. Thiele, "Composing heterogeneous components for system-wide performance analysis," in *Proceedings of the Design, Automation & Test in Europe Conference & Exhibition*, pp. 842–847, March 2011.

[33] E. Wandeler and L. Thiele, "Real-time calculus (rtc) toolbox," http://www.mpa.ethz.ch/Rtctoolbox.