Delay Bound Analysis in Real-Time Networks with Priority Scheduling Using Network Calculus

Jing Xie
Research & Innovation
Det Norske Veritas
Veritasveien 1, 1363 Hvik, Norway
Email: jing.xie@dnv.com

Min Xie
University Centre at Blackburn College
Blackburn, BB2 1LH, UK
Email: min.xie@ieee.org

Abstract—**Quality of Service (QoS) guarantees in large-scale real-time computing and communication networks are supported by enabling technologies such as real time scheduling, which can be implemented by priority scheduling. In order to guarantee the hard delay bound for delay sensitive applications, worst-case analysis has been developed. Network Calculus (NC) is a well known network analysis tool as it possesses a set of properties particularly suitable for large-scale networks. In NC, the arrival process and the service process of a system are bounded by the arrival curve and the service curve, respectively. These two curves are then used to derive the bounds of performance measures. However, due to the difficulty in properly choosing the curves, original NC based on min-plus algebra is not always able to provide tight performance bounds. In this paper, we develop a NC approach based on max-plus algebra and use it to derive the upper delay bound of a network system with non-preemptive priority scheduling. In max-plus NC, the arrival curve is defined as the lower bound on the cumulative inter-arrival time and the service curve gives the upper bound on the cumulative service time. Since we derive the worst-case cumulative service time for non-preemptive priority scheduling, these definitions allow us to choose the curves in such a way that they accurately capture the main characteristics of the arrivals and services. As a result, our max-plus NC approach generates tight delay bounds. In an example Controller Area Network (CAN) system, a series of numerical results for our delay bounds turn out to be as tight as the simulated worst-case delay. Therefore we prove that NC, as a suitable analytical method for large-scale real-time networks, is capable of providing accurate performance bounds .**

I. Introduction

Large-scale computing and communication systems such as sensor networks, cloud computing, and Internet of Things provide a new way for customers to access networks and share resources. In order to enable such evolutionary technologies, it is critical to optimize the network protocols and schemes to guarantee the Quality of Services (QoS). Since many present applications are delay-sensitive, the systems are required to provide delay guarantees, particularly the hard deadlines. Real-time scheduling is one of the enabling techniques guarantee delays and can be implemented by priority scheduling, in which the resource is allocated to users based on their priorities. Depending on how the priorities are assigned, there are various priority scheduling algorithms, *e.g.*, Earliest Deadline First (EDF) and Least Slack Time first (LST) scheduling[1]. For hard deadline guarantees, scheduling is designed to ensure that the deadlines are met even in the worst-case (WC) scenario, which is known as the worst-case analysis.

Several methods have been proposed to analyze the worst-case scenarios. The Worst-Case Response Time Analysis (WC-RTA) [2][3] was developed in the 1970s and has been widely applied to the analysis of worst case response time in real-time systems. It aims to calculate the maximum delay taken to deliver a packet from a source to the destination. Since WC-RTA was originally designed to analyze real-time systems rather than communication networking systems, its analysis is not scalable for large-scale complex network systems, particularly for random traffic.

Compared to the WC-RTA, Network calculus (NC)[4][5][6][7], a methodology to evaluate network performance, is relatively new with its first application in 2000 for the study of delay bounds in real time scheduling of communication networks[8]. In principle, NC uses min-plus and max-plus algebra to determine the bounds of performance measures like packet delay and flow backlog. The use of min-plus and max-plus algebra makes NC powerful at deriving performance bounds, characterizing flow aggregation, and determining per-flow service. As a result, NC is capable of handling complex network operations such as flow aggregations and end-to-end (e2e) transmissions and thus suitable for analyzing large-scale networks.

NC analysis is based on two basic elements, the *arrival curve* that defines the bound of the arriving traffic and the *service curve* that represents the bound of the service provided by the network. All the performance bounds are derived from these two critical curves that significantly affect the tightness of the derived bounds. On the one hand, the arrival curve and the service curve are often chosen as linear functions [9] to allow for tractable analysis in large-scale networks. On the other hand, linear approximation of the arrival and service process is poor for non-regular traffic and service and thus often results in loose delay bounds, as shown in [9][10].

The original NC is concentrated on min-plus algebra in which the arrival and service curve are defined to characterize the *cumulative amount* of arrival and service data[6][7]. In [11], we proposed a NC approach based on max-plus algebra, which, unlike the original NC, defines the arrival curve as a lower bound on the cumulative *inter-arrival time* of traffic and

978-1-4673-3122-7/13/$31.00 ©2013 IEEE

the service curve as an upper bound on the cumulative *service time* provided by the system.

In this paper, we investigate the potential of employing NC to analyze large-scale networks and achieve tighter performance bounds by modifying its network modeling approach. To do so, we apply our new max-plus-based NC approach to the analysis of non-preemptive priority scheduling in real-time networks. The new arrival and service curve enables us to derive a delay bound with the same accuracy as the simulated worst-case delay provided in [9]. In other words, our analysis provides an accurate description of the worst-case delay in the networked system and thus paves the way for guaranteeing hard delay deadlines in real-time networks.

The paper is organized as follows. First, we briefly review max-plus algebra and the NC approach applied in this paper. Then, Section III describes how the delay bound is derived for non-preemptive priority scheduling. In Section IV, the delay bound derived in Section III is applied to a commonly used automotive network protocol, Controller Area Network (CAN) bus, and compared with the delay bounds derived by the original NC approach and the simulation results of the WC delay. We conclude the paper in Section V.

II. Network Calculus based on Max-plus Algebra

NC is customized as a framework for worst case performance analysis of networked systems. It aims to provide the bounds on performance measures, *e.g.*, the upper bound on the e2e delays. It is built upon the cumulative functions that describe the input and output process of a queueing system. The cumulative functions $f(\cdot)$ are real-valued, non-negative, and wide-sense increasing as defined in the following:

$$\mathcal{F} = \left\{ f(\cdot) : \mathbb{R}^+ \to \mathbb{R}^+ : \forall x \geq y : f(x) \geq f(y), f(0) = f(0) \right\}.$$

In this paper, the original NC, developed on the basis of min-plus algebra, is called **min-plus NC**. Max-plus algebra, like min-plus algebra, is a sub-category of tropical algebra. Based on max-plus algebra, NC is designed to model the system with the cumulative inter-arrival time and the cumulative service time [7][12][11]. In this paper, all the models and results based on max-plus algebra are thus called **max-plus NC**. The basic operations in max-plus algebra include

- max-plus 'addition $+$' that represents *supremum* or *maximum* when it exists;
- max-plus 'multiplication $\times$ ' that is '+' in algebra;
- max-plus 'convolution $\bar{\otimes}$ ' of functions $f, g \in \mathcal{F}$, defined as
$$(f \bar{\otimes} g)(n) = \sup_{0 \leq k \leq n} \left\{ f(k) + g(n-k) \right\}$$
- max-plus 'de-convolution $\bar{\oslash}$ ' of functions $f, g \in \mathcal{F}$, defined as
$$(f \bar{\oslash} g)(n) = \inf_{k \geq 0} \left\{ f(n+k) - g(k) \right\}$$

Consider a traffic flow of packets. Newly arriving packets that cannot be served immediately are buffered in a queue. For the nth packet ($n = 1, 2, ...$), its arrival time, denoted by $a(n)$, is the moment when the packet arrives to a node. The departure time, denoted by $d(n)$, is defined as the moment when the packet departs the queueing system of the node and successfully received by the destination. If the packets in the queue are served by First-In-First-Out (FIFO), then the delay for the nth packet is defined by

$$D(n) = d(n) - a(n). \tag{1}$$

Max-plus NC, similar to min-plus NC, is based on the two elements, the arrival curve and the service curve. Their main difference lies in the definition of these two curves. In max-plus NC, the arrival curve $\lambda(\cdot)$ is defined as a *lower bound* on the cumulative inter-arrival time.

Definition 1. *(Arrival Curve [11]). Given a flow with arrival time $a(n)$, a function $\lambda(n) \in \mathcal{F}$ is an arrival curve for $a(n)$ iff*

$$\forall n, k > 0, k \leq n : a(n) - a(n-k) \geq \lambda(k).$$

In max-plus NC, the service curve $\gamma(\cdot)$ represents the maximum service time guaranteed to an arrival flow, *i.e.*, $\gamma(\cdot)$ is an *upper bound* on the cumulative service time consumed by the individual packets.

Definition 2. *(Service Curve [11]). Consider a system with arrival time $a(n)$ and departure time $d(n)$. The system is said to provide a service curve $\gamma(n) \in \mathcal{F}$ to the arrival flow iff*

$$\forall n > 0 : d(n) \leq a \bar{\otimes} \gamma(n).$$

The performance bounds are then derived from the arrival curve and the service curve. An important step of max-plus NC analysis is to properly decide the arrival curve $\lambda(\cdot)$ and the service curve $\gamma(\cdot)$ such that

- the arrival and service processes are tightly bounded; and
- the performance bounds can be computed explicitly.

In a word, the selection of these curves depends on the traffic pattern, service scheduling, and computational complexity.

III. NC Analysis of Non-Preemptive Priority Scheduling

In this section, we use max-plus NC to derive a delay bound for a network with non-preemptive priority scheduling. As a starter, we focus on the single-hop delay.

A. System Description

Consider a single node in a network system. The arrival process to the node is an aggregation of $N+1$ flows, each of which is assigned a distinct priority i. Let 0 be the highest priority and N be the lowest priority. The transmission order is determined by the priority of each flow. The flow with the highest priority among all non-empty queues is always served first. In non-preemptive priority scheduling [13], the transmission of low priority packets should not be interrupted by the new arrival of high priority packets.

Let the flow with priority i be F_i. The flow consists of fixed-length packets that arrive randomly. Without loss of generality, we assume a minimal inter-arrival time T_i for each flow. The arrival curve and service curve of F_i are $\lambda_i(\cdot)$ and $\gamma_i(\cdot)$, respectively. The queue length of flow F_i is denoted by Q_i.

Denote the nth packet of F_i by $\mathrm{p}_i(n)$. The arrival and departure times of $\mathrm{p}_i(n)$ are $a_i(n)$ and $d_i(n)$, respectively. In a single queue Q_i, all packets have the same priority and are served in FIFO. As defined in (1), delay $D_i(n)$ of $\mathrm{p}_i(n)$ is the difference between $a_i(n)$ and $d_i(n)$. In our queueing analysis, the calculation of $d_i(n)$ mainly takes into account two factors,

- transmission time C_i for packets in flow F_i; and
- head of line (HOL) queueing delay $w_i(n)$, the period during which $\mathrm{p}_i(n)$ waits in Q_i after it reaches the HOL.

The calculation of $w_i(n)$ needs to consider whether there is a packet under service when $\mathrm{p}_i(n)$ arrives and becomes the HOL. If so, $w_i(n)$ includes the residual transmission time of the packet under service. If the served packet is of priority q, $q \in [0, N]$, let the residual transmission time be R_q. Note that the residual transmission R_q is upper bounded by the transmission time, *i.e.*, $R_q \leq C_q$.

B. Network Calculus Analysis

Accurate NC analysis is based on the properly selected arrival curve and service curve. Given the minimum inter-arrival time T_i, we select an arrival curve

$$\lambda_i(n) = T_i \cdot n. \tag{2}$$

Then for any $0 < k \leq n$, the cumulative inter-arrival time is bounded by

$$a_i(n) - a_i(k) \geq \lambda_i(n) - \lambda_i(k) = T_i \cdot (n-k). \tag{3}$$

Unlike the arrival curve, the service curve is difficult to decide because of its dependence on the cumulative service time, which is determined by scheduling. In priority scheduling, the service availability for $\mathrm{p}_i(n)$ is only affected by the queues of higher priorities F_j where $0 \leq j < i$. In order to choose a tight service curve, we need to investigate the queueing status of all higher-priority queues observed by $\mathrm{p}_i(n)$ at its arrival time $a_i(n)$. Two scenarios are discussed: i) empty queues and ii) non-empty queues.

Scenario I – Empty Queues: when packet $\mathrm{p}_i(n)$ arrives, all high priority queues Q_j including Q_i $(0 \leq j \leq i)$ are empty but a packet of any priority q is currently served with a residual transmission time R_q. Then $\mathrm{p}_i(n)$ is inserted at the head of Q_i and waits for the current transmission to be completed. During the waiting period R_q, it is possible that higher priority packets may arrive. If so, $\mathrm{p}_i(n)$ has to wait until all the newly arriving higher-priority packets depart the system. In this case, the departure time of $\mathrm{p}_i(n)$ is given by

$$d_i(n) = a_i(n) + w_i(n) + C_i. \tag{4}$$

The HOL queueing delay $w_i(n)$ is bounded by the number of higher priority packet arrivals during $w_i(n)$ as follows

$$w_i(n) \leq R_q + \sum_{j=0}^{i-1} \left\lceil \frac{w_i(n)}{T_j} \right\rceil C_j. \tag{5}$$

The bound in (5) considers the worst-case scenario, in which the maximum number of priority j packets arrive during $w_i(n)$. The maximum number can be calculated from the minimum inter-arrival time T_j by $\lceil w_i(n)/T_j \rceil$. Plugging (5) into (4), we have the worst-case departure time $d_i(n)$

$$d_i(n) \leq a_i(n) + R_q + \sum_{j=0}^{i-1} \left\lceil \frac{w_i(n)}{T_j} \right\rceil C_j + C_i \tag{6}$$

Scenario II – Non-Empty Queues: when packet $\mathrm{p}_i(n)$ arrives, the higher priority queues Q_j including Q_i $(0 \leq j \leq i)$ are not all empty. Then $\mathrm{p}_i(n)$ is inserted into the end of Q_i, waits for all queued packets to be transmitted and then moves to the HOL. In this case, the departure time $d_i(n)$ of $\mathrm{p}_i(n)$ can be calculated recursively from the departure time $d_i(n-1)$ of the previous HOL packet $\mathrm{p}_i(n-1)$:

$$d_i(n) = \big[d_i(n-1) - C_i\big] + w_i(n) + C_i. \tag{7}$$

$\mathrm{p}_i(n)$ becomes the HOL when $\mathrm{p}_i(n-1)$ is moved out of the queue and served by the system, at time $d_i(n-1) - C_i$. Similar to Scenario I, $\mathrm{p}_i(n)$ will be served after the system has served all higher priority packets arriving during $w_i(n)$, as well as packet $\mathrm{p}_i(n-1)$ that is already under service. Then $w_i(n)$ is bounded by

$$w_i(n) \leq C_i + \sum_{j=0}^{i-1} \left\lceil \frac{w_i(n)}{T_j} \right\rceil \cdot C_j. \tag{8}$$

The departure time $d_i(n)$ is upper bounded as follows

$$d_i(n) \leq d_i(n-1) + \sum_{j=0}^{i-1} \left\lceil \frac{w_i(n)}{T_j} \right\rceil \cdot C_j + C_i \tag{9}$$

The overall worst-case departure time is then obtained by combining the two scenarios in (6) and (9)

$$\begin{aligned} d_i(n) \quad \leq \quad & \max\Big\{a_i(n) + R_q,\ d_i(n-1)\Big\} + \\ & \sum_{j=0}^{i-1} \left\lceil \frac{w_i(n)}{T_j} \right\rceil \cdot C_j + C_i \end{aligned} \tag{10}$$

Start with $d_i(0) = 0$. Applying (10) recursively gives us

$$\begin{aligned} d_i(n) \quad \leq \quad & \sup_{0<k\leq n} \Big\{a_i(k) + R_q + (n-k+1) \cdot C_i \\ & + \sum_{j=0}^{i-1} \left\lceil \frac{w_i(k)}{T_j} \right\rceil \cdot C_j \Big\}. \end{aligned} \tag{11}$$

In (11), the residual transmission time R_q is unknown but is upper bounded by the transmission time C_q, which can be used to further bound the departure time $d_i(n)$

$$\begin{aligned} d_i(n) &\le \sup_{0<k\le n} \Big\{ a_i(k) + \\ & \qquad \underbrace{C_q + (n-k+1)\cdot C_i + \sum_{j=0}^{i-1} \left\lceil \frac{w_i(k)}{T_j} \right\rceil \cdot C_j}_{S_i(k,n)} \Big\}, \\ &= a_i \bar{\otimes} S_i(n) \end{aligned} \tag{12}$$

where $S_i(k,n)$ is the worst-case cumulative service time and the right side of (12) corresponds to max-plus convolution. According to Definition 2, it is natural to choose $S_i(n)$ as the service curve, *i.e.*,

$$\gamma_i(n) = S_i(n) = C_q + \sum_{j=0}^{i-1} \left\lceil \frac{w_i(n)}{T_j} \right\rceil \cdot C_j + n \cdot C_i. \tag{13}$$

Using the arrival curve $\lambda_i(n)$, the service curve $\gamma_i(n)$ and the departure time $d_i(n)$, we bound the delay $D_i(n)$ by max-plus NC in Theorem 1.

Theorem 1. *(Delay Bound). Consider a FIFO system that provides service bounded by a service curve $\gamma(n)$ to an arrival flow characterized by an arrival curve $\lambda(n)$. The delay bound for the flow is:*

$$D(n) \le \sup_{0<k\le n} \big\{ \gamma(n-k+1) - \lambda(n-k) \big\}. \tag{14}$$

Proof: Use flow F_i as an example. Combining (1), (3), (12), and (13), we have

$$\begin{aligned} D_i(n) &= d_i(n) - a_i(n) \\ &\le a_i \bar{\otimes} \gamma_i(n) - a_i(n) \\ &= \sup_{0<k\le n} \Big[\gamma_i(n-k+1) - [a_i(n) - a_i(k)] \Big] \\ &\le \sup_{0<k\le n} \Big[\gamma_i(n-k+1) - \lambda_i(n-k) \Big] \end{aligned} \tag{15}$$

■

In communication networks, if the minimum inter-arrival time T_i and the maximum transmission time C_i for each traffic flow are known, (15) provides an upper bound on the packet delay. In the following section, we use CAN as an example to demonstrate how to derive the delay bound iteratively.

IV. Numerical Results

In order to evaluate the tightness of the delay bound (15) derived via max-plus NC, we compare it with the bounds derived by min-plus NC and the simulated worst-case delay generated in [9][10]. The objective is to find out how close are our max-plus NC bounds to the simulated worsts-case delay.

The simulation results and min-plus NC analysis were thoroughly compared for CAN communication systems [14] in [9]. For a fair comparison, we also use CAN as an example. In the numerical analysis, the same CAN parameter settings as in [9] are chosen because those simulation parameters are realistically chosen based on the data collected from Audi.

CAN is a broadcast bus designed to operate at a rate up to 1 Mbits/s. Data is transmitted in the form of *message* that is composed of payload up to 8 bytes, and an 11-bit or 29-bit unique identifier[1]. Message transmissions are controlled by non-preemptive priority scheduling, in which the priority is represented by the unique identifier. In a CAN system, a message can be invoked either by an event (event-trigger) or by an application task (time-trigger). In this paper, we choose the time-trigger with a minimum time interval, the same as the invoking technique used in [9].

A. System Setting of Small CAN

We start with a small CAN, in which 5 distinct priorities are assigned to CAN messages that are transmitted at the rate of 500k bits/sec. All messages have a maximum length of 130 bits[2]. Prior to transmitting, each CAN node senses the bus and starts transmission only if the bus has been idle for at least 6 bit times. Thus, the maximum message transmission time C_i for all $i \in [0,4]$ is $C_i = C = 0.272$ms for 136 bits. The minimal inter-arrival time of each priority is given in Table I.

Priority (i)	Minimum Interarrival Time (T_i)
0	50ms
1	10ms
2	100ms
3	20ms
4	30ms

TABLE I
CAN SYSTEM PARAMETER SETTING [9]

B. Max-Plus NC Bounds for Small CAN

In a small CAN, the max-plus NC arrival curve is simply derived from (2) by using T_i in Table I. The derivation of the service curve $\gamma_i(n)$ is complex since (13) requires the knowledge of an unknown parameter $w_i(n)$. We use an iterative method to solve $w_i(n)$. Considering that R_q is upper bounded by C_q and $C_i = C = 0.272$ms for all priorities i, (5) is reduced to (8). Then $w_i(n)$ can be iteratively calculated from (8) by the following equation

$$w_i^{l+1}(n) = \left[1 + \sum_{j=0}^{i-1} \left\lceil \frac{w_i^l(n)}{T_j} \right\rceil \right] \cdot C, \tag{16}$$

where $w_i^l(n)$ represents the lth iteration. Assume the initial condition $w_i^0(n) = 0$. Notice that for the highest priority 0, $w_0(n) \equiv 0$. The iterative method given in (16) converges quickly. For example, for $i = 1$, $w_i^l(n)$ converges to 0.544 after two iterations.

After $w_i(n)$ is iteratively calculated, the service curve $\gamma_i(n)$ can be directly derived from (13) as well. For this small CAN

[1] Standard frames have 11 bit message identifiers and extended frames have 29 bit identifiers.

[2] In addition to all fields defined in the standard frame format, 3 bits of Intermission Frame Space (IFS) and 19 bits of stuff are also included [9].

system, $\gamma_i(n)$ for different i is summarized as follows:

$$\gamma_i(n) = 0.272 \cdot n + W_i \tag{17}$$

where W_i is a priority-related constant listed in Table II.

Priority i	Constant W_i
0	0.272
1	0.544
2	0.816
3	1.088
4	1.36

TABLE II
CONSTANT W_i FOR THE SERVICE CURVE

With the arrival curve and the service curve decided, we then use (15) to compute the delay bound. Table III lists the calculated delay bound $D_i(n)$ for priorities $i \in [0, 4]$.

Priority i	Delay Bound $D_i(n)$ (ms)
0	0.544
1	0.816
2	1.088
3	1.36
4	1.632

TABLE III
DELAY BOUNDS

The max-plus NC delay bounds in Table III are plotted in Figure 1 against the priority index. They are compared with the min-plus NC bounds that are provided by [9][3]. It shows that, as upper delay bounds, our max-plus NC bounds are tighter than those min-plus NC bounds, particularly for low priority packets. They will become tighter as the number of priority increases and the influence of higher priority flows is more significant. The reason is two-fold. Firstly, min-plus NC uses the linear approximation to represent both the arrival curve and service curve, which may introduce a certain gap for nonlinear arrival or service processes, like the setting considered in this section. Secondly, min-plus NC derives the delay bound geometrically. This method involves division operations and is susceptible to computational round off error[9]. The numerical bounds are therefore deviated from the accurate delay bound. On the contrary, our max-plus NC approach defines the arrival curve directly and chooses the worst-case cumulative service time as the service curve. Such a selection is sufficiently accurate to reflect the analytical solution. In addition, the computational operations of the delay bound in (15) are simple and thus avoids the significant round off error. Consequently, the numerical results using max-plus NC are more accurate than those obtained via min-plus NC approach.

C. System Setting of Large CAN

In [9], a large CAN system with 56 priorities was simulated. The system settings are summarized in Table IV, where the service time is $C_i = C = 0.272$ms for all priorities $i \in [0, 55]$.

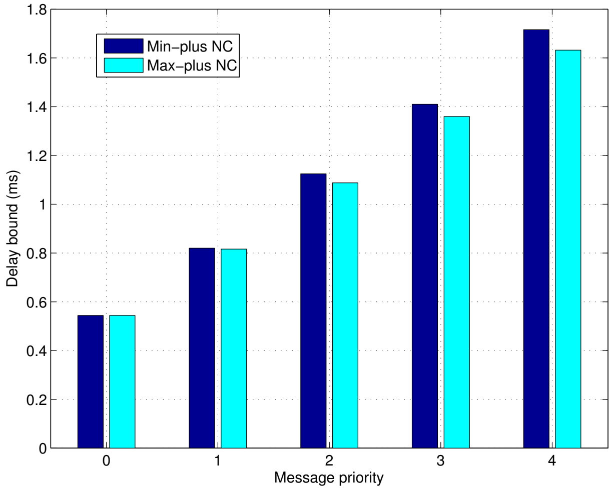

Fig. 1. Comparison of min-plus NC bound and max-plus NC bound.

Priority (i)	Period/Minimum Interarrival Time (T_i)
0, 16, 23-28	50ms
1-7, 13	10ms
8-12, 14-15, 17-21	20ms
22, 29-30	40ms
31-32, 35-41	100ms
33-34, 52	80ms
42-43, 45-46, 53	200ms
44, 47, 50-51	500ms
48, 54-55	1000ms
49	2000ms

TABLE IV
WORST CASE SIMULATION PARAMETER SETTING

Again, the arrival curve $\lambda_i(n) = T_i \cdot n$ can be directly expressed by using T_i from Table IV. The service curve is derived using the same method as in Section IV-B, which generates

$$\gamma_i(n) = \begin{cases} 0.272n + 0.272(i+1) & \text{if } 0 \le i \le 35, \\ 0.272n + 0.272(i+9) & \text{if } 36 \le i \le 55. \end{cases}$$

The delay bounds are

$$D_i(n) = \begin{cases} 0.272(i+2) & \text{if } 0 \le i \le 35, \\ 0.272(i+10) & \text{if } 36 \le i \le 55. \end{cases} \tag{18}$$

The max-plus NC bounds in (18) demonstrate a piece-wise linearity in two sectors $0 \le i \le 35$ and $36 \le i \le 55$ with a discontinuity between priority 35 and 36.

The simulated worst-case delay and the min-plus NC bounds [9] are selectively displayed in Figure 2 for priorities $i = 0-3, 27, 54, 55$. The max-plus NC bounds are obviously tighter than the min-plus bounds. For priorities $0-3$, the delay bounds produced by max-plus NC are equivalent to the simulation results. For priority 27, 54, and 55, our approach yields delay bounds $D_{27} = 7.888$ms, $D_{54} = 17.408$ms, and $D_{55} = 17.68$ms, different from the simulation result $D_{27} = 7.616$ms, $D_{54} = 17.136$ms, and $D_{55} = 17.408$ms by exactly one message transmission time $C = 0.272$ms.

[3] Due to the page limitation, we do not provide the detail procedure of computing the analytical delay bounds using the min-plus NC and the simulation model of WC-RTA.

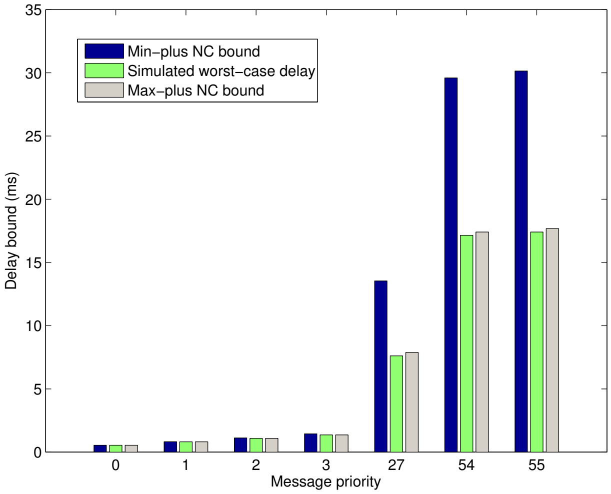

Fig. 2. Delay bound (ms) comparison

Moreover, for priorities $i \in [4, 26]$ and $i \in [28, 53]$, Figure 16 in [9] also demonstrated a piece-wise linear relationship between the worst-case delay and the priority index, similar to our finding in (18). In addition, Figure 16 in [9] showed an obvious discontinuity between priority 36 and 37. Interestingly, in our numerical results (18), a discontinuity appears between priority 35 and priority 36. Considering that there is a difference of $C = 0.272$ms between our bounds and the simulation result for priority $i \geq 27$, it is reasonable to speculate that the simulation results presented in Table 2 of [9] may have some errors which cause such a difference of $C = 0.272$ms for priorities $i \geq 27$. If that is the case, then our max-plus NC approach yields delay bounds that accurately represent the real worst-case delays.

V. Conclusion

In this paper, we apply the max-plus NC approach to model networking systems with non-preemptive priority scheduling and analyze the worst-case delay for real-time applications. Since the arrival curve in max-plus NC is concentrated on the cumulative inter-arrival time rather than the cumulative amount of arriving data, this max-plus NC approach can accurately characterize the arrival process without the need of approximation as long as the packet arrivals are constrained by a minimum inter-arrival time. This is the first reason why max-plus NC produces better delay bounds than the min-plus NC approach. Secondly, the service curve in max-plus NC is defined for the cumulative service time. In non-preemptive priority scheduling, we have derived the worst-case cumulative service time, which then allows us to define the service curve directly, reducing the risk of using any approximation for the service curve that may potentially result in loose delay bounds. Moreover, in our approach, both the arrival and service curve of each flow is determined by those flows that have been served prior to the flow under consideration. Therefore, this approach can be also applied to other ordering scheduling algorithms such as Generalized Processor Sharing.

By using the CAN system as an example for numerical analysis, we not only prove that the max-plus NC approach achieves tighter delay bounds than the min-plus NC approach but also demonstrate that such tight bounds could be the same as the real worst-case delay if the NC curves are properly chosen. Similar to min-plus NC, this max-plus NC approach has a potential to provision tight e2e performance by taking advantages of the concatenation property of NC when the flows travel through a multi-hop network. In the future, we will further explore the properties of max-plus NC and investigate the e2e performance of large-scale networks such as delay and backlog. It is then reasonable to conclude that NC could facilitate more efficient design of real-time communications networks for large-scale delay-sensitive applications.

References

[1] J. W. S. Liu, *Real-time Systems*. Prentice Hall, 2000.

[2] C. L. Liu and J. W. Layland, "Scheduling algorithms for multiprogramming in a hard real time environment," *Journal of the ACM*, vol. 20, no. 1, pp. 46–61, 1973.

[3] K. W. Tindell and A. Burns, "Guaranteeing message latencies on controller area network (CAN)," in *Proc. 1st International CAN Conference*, Sept. 1994.

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

[5] ——, "A calculus for network delay. part II: Network analysis," *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 132–141, Jan. 1991.

[6] J.-Y. L. Boudec and P. Thiran, *Network Calculus: A Theory of Deterministic Queueing Systems for the Internet*. Springer Verlag, 2012.

[7] C.-S. Chang, *Performance Guarantees in Communication Networks*. Springer, 2000.

[8] L. Thiele, S. Chakraborty, and M. Naedele, "Real-time calculus for scheduling hard real-time systems," in *Proc. Symposium on Circuits and Systems ISCAS*, March 2000.

[9] T. Herpel, K.-S. Hielscher, U. Klehmet, and R. German, "Stochastic and deterministic performance evaluation of automotive CAN communication," *Elsevier Computer Networks*, vol. 53, no. 8, pp. 1171–1185, Feb. 2009.

[10] A. Koubaa and Y.-Q. Song, "Evaluation and improvement of response time bounds for real-time applications under non-pre-emptive fixed pirority scheduling," *International Journal of Production Research*, vol. 42, no. 14, pp. 2899–2913, July 2004.

[11] J. Xie and Y. Jiang, "Stochastic service guarantee analysis based on time-domain models," in *Proc. 17th IEEE/ACM International Symposium on Modelling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, 2009.

[12] M. Fidler and S. Recker, "Conjugate network calculus: A dual approach applying the legendre transform," *Elsevier Computer Networks*, vol. 50, no. 8, pp. 1026–1039, June 2006.

[13] D. P. Bertsekas and R. G. Gallager, *Data Networks*. Prentice Hall, 1992.

[14] "International Standard ISO 11898, Road Vehicles - Interchange of Digital Information - Controller Area Network (CAN) for High Speed Communication, 1st ed., International Organization for Standardization, 1994, ISO Reference Number ISO 11898," 1993.