Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

A Survey on Network Calculus Tools for Network Infrastructure in Real-time Systems

BOYANG ZHOU[1], (Member, IEEE), ISAAC HOWENSTINE[2], (MEMBER, IEEE), SIRAPHOB LIMPRAPAIPONG, LIANG CHENG[3], (Senior Member, IEEE)

[1]Lehigh University, Bethlehem, PA 18015 (e-mail: boz319@lehigh.edu)
[2]Lehigh University, Bethlehem, PA 18015 (e-mail:ish219@lehigh.edu)
[3]Lehigh University, Bethlehem, PA 18015 (e-mail: cheng@lehigh.edu)

Corresponding author: Liang Cheng. Author (e-mail: cheng@lehigh.edu).

This material is based upon work supported by the National Science Foundation under Grant No. 1646458.

ABSTRACT Network Calculus is an established analytical framework which can provide deterministic estimates of latency and buffer requirements for feed-forward packet-switched communication networks. Researchers find it useful in analyzing network performance in real-time systems. As there are several Network Calculus tools and software packages for estimating network performance, it is important to select suitable Network Calculus tools for network analysis problems. This paper introduces twelve established tools for Network Calculus analysis and compares their advantages and disadvantages in terms of their applicability for various systems and analysis needs. We also provide recommendations for users to select the suitable tools for research. Five public tools, namely DiscoDNC, RTC Toolbox, Deborah, CyNC, and nc-tandem-tight, are chosen to analyze end-to-end delay in a series of tandem networks for their performance demonstration. Simulations of the same tandem networks are implemented to compare with the analytical results achieved by the Network Calculus tools. To our best knowledge, this is the first comprehensive study that compares existing tools of Network Calculus analysis for real-time system network infrastructure. Our aim is to provide quick guidance to research communities for selecting suitable tools to analyze delay performance of networks.

INDEX TERMS Network Calculus, DiscoDNC, RTC Toolbox, nc-tandem-tight, Deborah, CyNC, real-time systems

I. INTRODUCTION

Network Calculus is a set of theories and analytical methods to deterministically estimate the upper bound of the and end-to-end delay for data flows and minimum buffer size for switches in communication networks, which is widely used by researchers working on real-time systems such as real-time cyber-physical systems (CPS) [1]. Network Calculus provides a framework for modeling the critical aspects of the network to estimate bounds of the worst-case delay and backlog of the network. These aspects include the applications serviced by the network and their unique communication requirements, the capability of the network devices in the network, the topology of the network substrate, the data flow paths, and the scheduling algorithms.

As real-time system technologies evolve, they have begun to play vital roles in industrial applications, including smart factories, substation automation systems, avionics, and autonomous vehicles. It is essential to guarantee the delay performance of the supporting network, as they impact the safety and reliability of the operations of those systems. Network Calculus is a commonly used approach for analyzing the delay performance of the network infrastructure real-time systems.

A. EXISTING WORK

There are a lot of Network Calculus tools developed by researchers and private companies. In this paper, we introduce twelve tools, including seven public tools and five private ones, which are summarized in Table 1 and Table 2. Most of these papers cited in Table 1 are discussed in Section III.

These twelve tools are all tools implementing Network Calculus analyses we can find through Internet exploration. We can conclude that different tools have different application domains. In Table 2, there are different types of networks and schedulers mentioned using acronyms. We introduce what these acronyms represent in this section. TSN (Time-Sensitive Networking) is designed for time-triggered traffic to guarantee its hard delay requirements. AFDX (Avionics Full-Duplex Switched Ethernet) is used in aircraft to transmit data between end systems. TTEthernet (Time-Triggered Ethernet) is a time-critical network for industrial and avionics applications improved from AFDX [2]. LIN (Local Interconnect Network) is a protocol designed for vehicles to implement the data transmission. PROFINET (Process Field Net) is a standard used in industrial Ethernet, which is designed to collect data and control equipment. Scheduling algorithms mentioned in this paper include FP (Fixed-Priority), RR (Round-Robin), EDF (Earliest Deadline First), FIFO (First In, First Out), and WFQ (Weighted Fair Queue). SFA, TFA, PMOO, and LUDB are all analysis methods of Network Calculus. These methods are introduced in Section II-B. Pay Bursts Only Once (PBOO) in the table uses the concatenation theorem in Theorem 4 so that the burstiness of the arrival curve is amortized along all of the servers in the flow path.

B. CONTRIBUTION

There are two major contributions in this paper: (i) Providing a comprehensive reference and a quick lookup of Network Calculus tools for researchers. (ii) Comparing various Network Calculus tools and verifying their correctness. As shown in Table 1 and Table 2, there are many software tools for conducting Network Calculus analyses. The discussions and descriptions of these tools are scattered in a large number of papers. Thus, researchers may spend a great deal of effort to identify a proper tool for their research tasks considering many factors influencing the tool selection. These factors include the scheduling policies, the type of networks, and the support of optimization functions. We perform Network Calculus analyses in a case study using five public tools: RTC Toolbox, DiscoDNC, Deborah, CyNC, and nc-tandem-tight. We compare the results from the five tools to evaluate their tightness. Simulations using OMNeT++ are implemented to check the pessimism of the analytical results. To the best of our knowledge, this is the first paper that surveys available Network Calculus tools, describes their functionalities, discusses the advantages and disadvantages of the tools, and recommends suitable tools for different network analysis tasks.

This paper consists of six sections. Section II introduces key Network Calculus concepts and methods of analysis. Section III introduces different Network Calculus toolboxes. In Section IV, the publicly available software tools RTC Toolbox, DiscoDNC, Deborah, CyNC, and nc-tandem-tight are used to analyze the end-to-end latency in a series of tandem networks of various sizes. In section V, the results of a simulated tandem network in OMNeT++ are compared with the tightest analytical results from Section IV. In Section VI, the strengths and weaknesses of different Network Calculus tools are discussed in terms of the tightness of the bounds they produce, the range of application, the ease of implementation, and the control over the analysis. It will also include recommendations for selecting a software tool based on the needs of the network analyst or designer.

II. BACKGROUND KNOWLEDGE

A. NETWORK CALCULUS

Network Calculus analyses require certain information and assumptions, which include:

1) The topology of the network;
2) The properties of each data flow (the piecewise linear function called the arrival curve, and the data flow path through the network including the source and sink);
    a) The flow paths are assumed to be static during the analysis;
    b) The network induced by all the flow paths must be feed-forward (the definition of feed-forward networks is introduced in the next paragraph);
3) The properties of each switch in the network (the piecewise linear function called the service curve, the scheduling strategy that each switch uses).
4) The model to characterize the arrival curves of data flows entering the network;
5) The model to characterize the service curves of each switch in the network;
6) The level of service that the flow of interest, which is the flow we want to get the bound of the maximum delay for, receives along its path from source to sink;

The network topology (the graph representing the traffic pattern of the network, and the physical connections between end systems and switches) is fundamental to the network, which defines the boundary of analysis, including sources and sinks for any data flow. Besides paths of data flows, arrival curves of data flows are needed to implement Network Calculus analyses, which are described by piecewise linear curves usually based on the leaky bucket model consisting of a burstiness component and a rate component. In rare instances an arrival curve based on a dual or multiple token bucket curve may be used, made of several piecewise linear curves [32]. Network Calculus only works in deterministic routing networks where the path of each flow from source to sink is static. Moreover, Network Calculus only works with feed-forward networks, which is a restriction on how data must pass through the network. The formal definition of a feed-forward network is that there exists an ordinal numbering of switches of the network that can provide a monotonically increasing sequence of switch labels for all flow paths in the network. A thorough explanation of the feed-forward network is provided in [43]. Besides the flow information, information about the service curves of the

TABLE 1. Network Calculus Tools: Part I

Tool Name	Language/Interface	Tutorial/Documentation	Citations and Their Discussions	Private /Public
RTC Toolbox	MATLAB/Java	✓	[3]–[8], III-A.	Public
Deborah	C++	N/A	[9]–[12], III-B.	Public
DiscoDNC	Java	✓	[13]–[19], III-C.	Public
NC-maude	Maude	Manual & Example	[20], [21], III-D.	Public
CyNC	MATLAB/Simulink	Included in Download Package	[22]–[25], III-E.	Public
CATS	Eclipse	✓	[26], III-F.	Public
WOPANets	GUI	✓	[27]–[31], III-G.	Private
DIMTOOL	MATLAB	Example	[32] [33], III-H.	Private
DelayLyzer	GUI	NA	[34], [35], III-I.	Private
SINETPLAN	GUI	NA	[36] [37], III-J.	Private
RTaW-Pegase	GUI	✓	[38] [39], III-K.	Private
nc-tandem-tight	OCaml	✓	[40]–[42], III-L	Public

TABLE 2. Network Calculus Tools: Part II

Tool Name	Targeted Network Types	Analysis Method	Source
RTC Toolbox	Feed-forward Networks, Distributed Embedded Systems	Implemented by users	https://www.mpa.ethz.ch/Rtctoolbox/Overview
Deborah	FIFO Tandem Network	LUDB	http://cng1.iet.unipi.it/wiki/index.php/Deborah
DiscoDNC	FIFO and Arbitrary Multiplexing Feed-forward Network	TFA, SFA, PMOO	https://disco.cs.uni-kl.de/index.php/projects/disco-dnc
NC-maude	Embedded Real Time Systems	N/A	https://www.onera.fr/en/staff/marc-boyer
CyNC	Network with FP, RR, EDF and WFQ Schedulers	Implemented by users	http://kom.aau.dk/ henrik/old-control/CyNC$_2$.0/
CATS	Real-time Systems	N/A	http://www.it.uu.se/research/group/darts/times/cats
WOPANets	AFDX and Ethernet with FIFO, FP, WFQ and RR Scheduling Policies	Propogation Analysis, PBOO	N/A
DIMTOOL	Feed-forward Networks & Worst-case Simulation	SFA, TFA, PMOO	N/A
DelayLyzer	Feed-forward Networks	SFA, TFA, mTFA	N/A
SINETPLAN	PROFINET networks	N/A	https://new.siemens.com/global/en/products/automation /industrial-communication/PROFINET/portfolio/sinetplan.html
RTaW-Pegase	TSN, AFDX, TTEthernet, LIN, ARINC	N/A	https://www.realtimeatwork.com/software/rtaw-pegase/
nc-tandem-tight	Arbitrary Multiplexing Tandem networks	Linear Programming	https://www.di.ens.fr/ bouillar/NCbounds/

switches and the scheduling policies used by the switches is also needed to implement Network Calculus analyses.

Min-plus algebra performs a fundamental role in Network Calculus, which provides convolution and deconvolution operations of two functions f_1 and f_2 which are defined as:

$$convolution \quad (f_1 \otimes f_2)(d) = \inf_{0 \le s \le b} \{f_1(d-s) + f_2(s)\}, \quad (1)$$

$$deconvolution \quad (f_1 \oslash f_2)(d) = \sup_{u \ge 0} \{f_1(d+u) - f_2(u)\}. \quad (2)$$

Both operations are frequently used in Network Calculus computations.

In addition, the concepts of the arrival curve and the service curve are introduced. The arrival curve describes the worst-case data arrival of a data flow within any time period. The service curve represents the minimum ability of a switch to forward packets. The arrival curve should be a concave function, and the service curve should be a convex function [44]. To discuss these ideas further, definitions of an arrival process, the arrival curve, and the service curve are provided below.

Definition 1. *An arrival process $F(t)$ represents the total cumulative quantity of data that has arrived at a node at time t. The data amount is measured in bits. It is a non-decreasing function that describes what has happened at the node from time 0 to t.*

Definition 2. *An arrival curve $\alpha(t)$ is a bounding envelope that characterizes all possible arrival processes. Given an arrival process $F(t)$, a real-valued, non-negative, non-decreasing function $\alpha(t)$ defined for any $t \ge 0$ is an arrival curve of $F(t)$ if and only if*

$$\forall t \ge s \ge 0 : F(t) - F(s) \le \alpha(t-s) \quad (3)$$

One commonly used arrival curve is the leaky bucket arrival curve, which comprises two linear piecewise components, a rate component ρ and a burstiness component b. A leaky bucket arrival curve has a format shown in Equation 4.

$$\alpha_{\rho,b}(t) = \begin{cases} \rho t + b & t \ge 0 \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

Definition 3. *Given a flow passing a network node with an input arrival process $F^{in}(t)$ and an output process $F^{out}(t)$, a real-valued, non-negative, non-decreasing function $\beta(t)$ is the service curve of the node if and only if*

$$F^{out}(t) \ge F^{in}(t) \otimes \beta, where\ \beta(0) = 0 \quad (5)$$

To create the service curve for a switch, one needs to know both $F^{in}(t)$ and $F^{out}(t)$. The service curve that is used in Network Calculus computations is the maximum curve that satisfies the constraint in Definition 3, which provides the tightest bound for the end-to-end latency. We assume that the service curve is the property of the switch in this paper, which is independent of the arrival curves.

The most commonly used service curve is called the rate-latency function, which is shown in Equation 6, with a delay component T and a rate component R. In Equation 6, $[a]^+$ equals to $max\{a, 0\}$.

$$\beta_{R,T}(t) = [R(t-T)]^+ \tag{6}$$

Knowing the service curve and the arrival curve, we can derive the bound of the worst-case delay of the flow of interest and the backlog of the switch.

Theorem 1. *Suppose the switch's service curve is β, and the arrival curve of the flow is α. The bound of worst-case delay can be calculated by Equation 7 [16].*

$$\begin{aligned} delay : \forall t \geq 0 : D(t) &\leq inf\{d \geq 0 | (\alpha \oslash \beta)(-d) \leq 0\} \\ &= h(\alpha, \beta) \end{aligned} \tag{7}$$

The backlog of the switch can be computed by

$$backlog : \forall t \geq 0 : B(t) \leq (\alpha \oslash \beta)(0). \tag{8}$$

Graphically, Figure 1 shows the delay and backlog. The backlog is the maximum vertical deviation between the service curve and the arrival curve, which means the maximum amount of data that could be accumulated in the buffer. Thus, the backlog determines the minimum buffer size needed by the switch. Assuming FIFO multiplexing at the switch, the worst-case delay bound is the maximum horizontal distance between the service curve and the arrival curve.

Besides FIFO multiplexing, arbitrary or blind multiplexing is another multiplexing strategy used by switches. The transmission order of packets does not totally depend on the arrival time of packets. Scheduling policies such as RR, FP, WFQ, and FP all belong to arbitrary multiplexing. In arbitrary multiplexing, packets belong to the same flow are still FIFO. When the switches use arbitrary multiplexing, the worst-case delay bound is the maximum horizontal deviation between the arrival curve and the service curve as well except using TFA, which is discussed in Section II-B. Instead, the time coordinate of the intersection of the arrival curve and the service curve, which is shown in Figure 1, can be used as the delay bound when using TFA under arbitrary multiplexing. It is a looser assumption and will lead to a larger upper bound for delay times caused by the switch.

The above computation applies to a one-node, one-flow system. To analyze systems with more than one node, one may need to account for the impact of intervening flows, compute the arrival curve of the outgoing flow for downstream nodes to process, or convert a sequence of nodes into a single abstracted switch using Network Calculus computations.

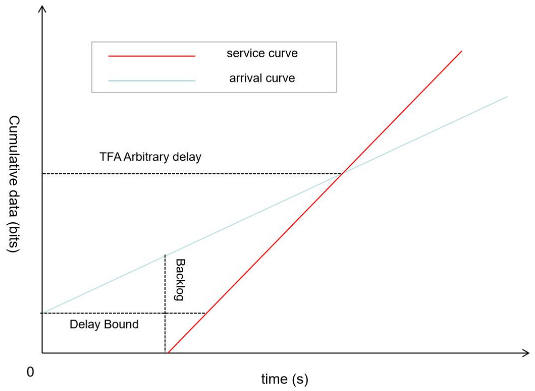


FIGURE 1. This figure shows the delay and the backlog graphically. There are two types of delays which are determined by the multiplexing strategy of the switch.

Theorem 2. *Given an incoming arrival curve α^{in} of a flow and a service curve β of a switch, the arrival curve of the outgoing flow can be calculated using Equation 9.*

$$\alpha^{out}(d) = (\alpha^{in} \dot{\oslash} \beta)(d) = \begin{cases} 0 & d = 0 \\ (\alpha^{in} \oslash \beta)(d) & otherwise \end{cases} \tag{9}$$

One needs to calculate the outgoing arrival curve because it will be the incoming arrival curve of the next switch in the path of the flow if the current switch is not the destination. .

If more than one flow passes the same switch, they share the service provided by the switch. Network Calculus defines the leftover service as the minimum level of service that a flow would receive in this situation. The following theorem discusses a two-flow example, which can be extended for additional competing flows.

Theorem 3. *Suppose flows f_1 and f_2 have arrival curves $\alpha_{\rho_1,b_1}(t)$ and $\alpha_{\rho_2,b_2}(t)$ respectively, and two flows pass a switch, which uses arbitrary multiplexing, with a service curve $\beta_{R,T}(t)$. Then the leftover service curve $\beta^1(t)$, which defines the minimum level of service given to flow f_1, can be calculated by Equation 10 [16]. The leftover service curves calculated by Equation 10 is the lower bound of the service received by the flow of interest. Arbitrary multiplexing covers various scheduling algorithms, such as round-robin and strict priority. If the specific scheduling algorithm is known, we can get more accurate leftover service curves compared with those calculated by Equation 10.*

$$\begin{aligned} \beta^1(t) &= [\beta_{R,T}(t) - \alpha_{\rho_2,b_2}(t)]^+ \\ &= [(R-\rho_2)(t-(RT+b_1)/(R-\rho_2))]^+ \end{aligned} \tag{10}$$

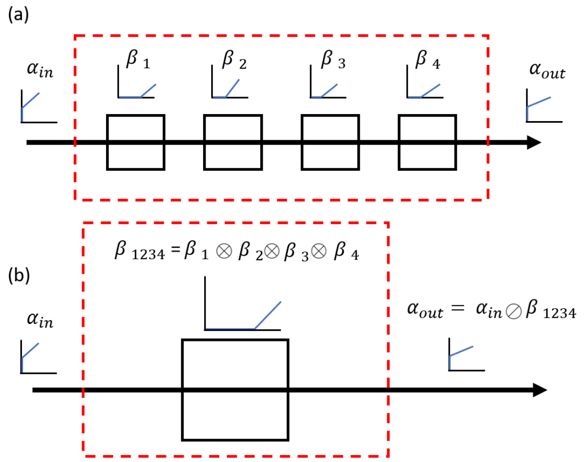

(a)
α_{in}
β_1
β_2
β_3
β_4
α_{out}
(b)
$\beta_{1234} = \beta_1 \otimes \beta_2 \otimes \beta_3 \otimes \beta_4$
α_{in}
$\alpha_{out} = \alpha_{in} \oslash \beta_{1234}$

FIGURE 2. A display of how concatenation can be used to abstract multiple nodes into effectively one node. The concatenation is done by using min-plus convolution of curves β_1, β_2, β_3 and β_4 from the sequence of nodes in figure (a). The service curve of the switch in (b) is equivalent to $\beta_1 \otimes \beta_2 \otimes \beta_3 \otimes \beta_4$. In part (b), the arrival curve and service curve could be laid on top of each other like in Figure 1 to calculate the max delay and buffer size needed. The arrival and service curves in these figures are not drawn to scale.

When the two flows pass a switch with FIFO multiplexing, the leftover service curve β^1 of flow f_1 can be calculated by Equation 11 [45].

$$\beta^1(t) = [(R - \rho_2)(t - T - b_2/R)]^+ \tag{11}$$

When a flow passes several switches, we can use the concatenation theorem to calculate the system's overall service curve, which is described as follows.

Theorem 4. *Consider a flow f passes a tandem of switches S_i, i = 1,...,n, where the service curve of switch S_i is β_i. Then, the tandem of switches can be treated as a single switch with an overall service curve defined as $\beta = \overset{n}{\underset{i=1}{\otimes}} \beta_i$.*

Figure 2 gives an example of how the concatenation theorem can be used to calculate the overall service curve of a tandem network and how to compute the outgoing flow's arrival curve with the overall service curve. Using the concatenation theorem can simplify the calculation of Network Calculus because the process of calculating the arrival curve of the outgoing flow for each switch can be eliminated. Moreover, using the concatenation theorem helps derive tighter worst-case delay bounds compared with methods that don't use this theorem, like TFA.

B. ANALYSIS METHODS

This paper covers four types of network analyses: Total Flow Analysis (TFA), Separated Flow Analysis (SFA), Pay Multiplexing Only Once analysis (PMOO), and Least Upper Delay Bound analysis (LUDB). These methods are used to find the upper bound of the worst-case delay for a particular flow of interest passing through a network. To conduct these analyses requires the network topology, the flow paths, the arrival curves of all flows at their entry point in the network, the service curves of all switches, and the multiplexing strategy of the switches. The tightness of the results of these methods depends on the topology and the traffic pattern of the network being analyzed.

The different techniques perform differently in terms of their computational processes and calculation results. A researcher or engineer would typically implement various analyses on the same network and use the result with the tightest upper bound.

1) Total Flow Analysis (TFA)

TFA finds a conservative estimate of the delay at each node along the flow of interest's path and sums these individual delays into one end-to-end latency bound. TFA does not use the concatenation theorem. For each node along the flow of interest, TFA finds the total cumulative arrival curve at that node, summing all incoming arrival curves together. Assuming FIFO multiplexing, the maximum horizontal distance between the cumulative arrival curve at each switch and the service curve of that switch is computed. Assuming arbitrary multiplexing, the point of intersection between the arrival curve and service curve for that switch must be computed, which is shown in Figure 1. The resulting set of delays are summed together to get the maximum delay that a packet in the flow of interest could experience. The backlog is the maximum nodal backlog in the path of the flow of interest.

Assuming that P is the path of the flow, s denotes a particular switch in that path, D_s is the nodal delay of the switch s, and B_s is the nodal backlog of the switch s, the total end-to-end delay D and the backlog B of the flow of interest can be expressed as

$$B = \max_{s \in P} B_s, \; D = \sum_{s \in P} D_s. \tag{12}$$

The nodal latency computations rely on the incoming arrival curves. Given any flow in the network that crosses the flow of interest, that flow's arrival curve as it crosses the flow of interest must be known. Thus, that crossflow's arrival curve must be calculated from source to sink, in order to get the arrival curve as the crossflow begins commingling with the flow of interest, and the arrival curve when the cross flow departs from commingling with the flow of interest.

The assumption that TFA treats all flows passing the current switch as one flow is the essence of this method. This assumption makes the delay bound of each node be independent of the flow of interest [46]. Thus, it will always give the same delay bound for flows with the same path, even if the flows have different arrival curves. TFA usually offers a looser bound than SFA, which is proved in [47]. Because of this, TFA is rarely used in practice, and many toolboxes do not provide TFA method for analyses.

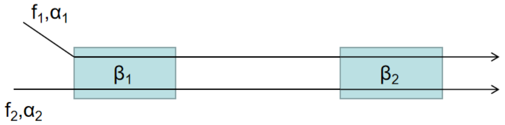


FIGURE 3. A simple network shows the difference between SFA and PMOO.

2) Separated Flow Analysis (SFA)

SFA computation is done in three steps: first, calculate the leftover service curve for the flow of interest at every node along the flow of interest's path; second, concatenate the leftover service curves to create an end-to-end service curve for the flow of interest's path by using the concatenation theorem; finally, compute the delay and the backlog based on the flow of interest's arrival curve and the concatenated leftover service curve [16].

The advantage of SFA is that it can provide different delay bounds for different flows. Unlike TFA, flows having the same path can have different delay bounds if they do not have the same arrival curve. SFA's use of the concatenation theorem contributes to a tighter delay bound.

3) Pay Multiplexing Only Once (PMOO)

PMOO is developed based on SFA and works similarly to SFA. There are some similarities between the two methods. They both calculate the leftover service curves and use the concatenation theorem to calculate the service curve for a tandem of switches. The difference between SFA and PMOO is that SFA calculates the leftover service for each switch before concatenation, while PMOO concatenates the service curves of switches ahead of calculating the leftover service allocated to the flow of interest.

To demonstrate the difference between SFA and PMOO, we utilize the network in Figure 3 with arbitrary multiplexing as an example. Using SFA, we can derive the overall leftover service for f_1 as

$$\beta^{SFA}_{overall} = [\beta_1 - \alpha_2]^+ \otimes [\beta_2 - (\alpha_2 \dot{\oslash} [\beta_1 - \alpha_1]^+)]^+. \quad (13)$$

Using PMOO, the overall leftover service for f_1 is

$$\beta^{PMOO}_{overall} = [\beta_1 \otimes \beta_2 - \alpha_2]^+. \quad (14)$$

This example clearly shows the difference between SFA and PMOO. PMOO provides a larger service curve, which will result in a tighter delay bound. PMOO may provide a tighter bound of the maximum delay than SFA when the multiplexing strategy is arbitrary, but this is not guaranteed because there is currently no formal proof to show that PMOO always generates a better service curve than SFA in general cases.

4) Least Upper Delay Bound (LUDB)

Another method to implement Network Calculus analyses is LUDB [48] [49] [50] [12]. A strength of the LUDB method is that in certain topologies like sink-trees, the analysis method provides a global minimum for the latency upper bound. However, this method only works on the nested tandem topology. In a nested tandem network with a set of n flows, given any pair of two flows identified by their starting and end nodes (i, j), and (h, k), if $i < h$, then $j < h$ or $k \leq j$. If two flows overlap, the boundaries (entry and exit) of one flow will be contained inside the boundaries of the other flow. See Figure 4 for a comparison of nested and non-nested networks.

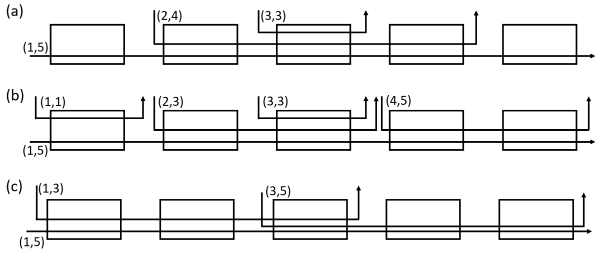


FIGURE 4. (a) perfectly nested tandem network, (b) nested tandem network, (c) non-nested tandem network. Notice in (c) that flows (1,3) and (3,5) both cross node 3. Given $(i, j) = (1, 3)$ and $(h, k) = (3, 5)$, $i < j$ but $h = j$ so this violates the definition of a nested traffic flow. This network must be cut into three sections: (1,2) (3), and (4,5).

Although the LUDB method considers a nested tandem network, it can be expanded to non-nested tandem networks. The network used for analysis in this paper is not nested, and a discussion about using LUDB-based method for analyzing non-nested tandems is provided at the end of this subsection.

The LUDB method is based on the following theorem from [48], defining an equivalent service curve for a flow sharing a service node with one flow.

Theorem 5. *Consider a lossless node serving two flows, 1 and 2, in FIFO order. Assume that the node guarantees a minimum service curve β to the aggregate of the two flows and that flow 2 has α_2 as an arrival curve. Define the family of functions:*

$$\beta^{eq,1}(t, \tau) = [\beta(t) - \alpha_2(t - \tau)]^+ \times 1_{\{t>\tau\}} \quad (15)$$

where $1_{\{t>\tau\}}$ is equal to 1 if $t > \tau$ and zero otherwise. For any $\tau \geq 0$ such that $\beta^{eq,1}(t, \tau)$ is wide-sense increasing, then flow 1 is guaranteed the (equivalent) service curve $\beta^{eq,1}(t, \tau)$.

This describes an infinity of equivalent service curves, instances of which are obtained with a specific value of τ. LUDB uses the idea of "equivalent service curves" [12], similar to the leftover service curve. This is calculated by iteratively removing the cross flows from the nodes passed through by the flow of interest until all cross flows have been accounted for in the updated "equivalent service curve." For nested flows, this means iteratively removing the innermost nested flows (typically the flows that are spanning the fewest number of switches), computing the "equivalent service" and convolving the equivalent service curves with adjacent service curves. This is done until there is a single equivalent service curve for the flow of interest. In some limited cases (sink-tree networks), the LUDB method provides the exact worst-case delay [48]. However typically, this method will introduce pessimism.

In the case where the network flows are not nested, the flows must be cut into nested sub-portions. This allows for the LUDB method to work on each of the subnetworks, and then aggregating the along each of those subnetworks. This results in larger delay bounds because the total network delay must be separated, separately computed and then aggregated together.

III. OVERVIEW OF TOOLS

Since Network Calculus is widely used to analyze the performance of real-time systems, many software tools have been created and used to conduct Network Calculus analyses. Tables 1 and 2 list some of these available tools. These tools are further discussed in this section.

A. RTC TOOLBOX

A common tool used to conduct Network Calculus analysis is RTC (Real-Time Calculus) Toolbox [3]. Real-Time Calculus is an extension of Network Calculus that is directed at embedded systems. It works in a similar way to Network Calculus. It takes arrival and service curves and provides deterministic bounded results. RTC Toolbox is based on Java and Matlab-based [3]. The tutorial available [51] is useful for picking up the basics. Network Calculus analyses can be done in Matlab using the RTC Toolbox package. RTC Toolbox also has a Java library.

Users construct piecewise linear arrival and service curves to upper and lower bound the cumulative arrivals and departures of data bits. The way these units move through the system is defined by users according to the network design, while the network described must be feed-forward. The package can conduct Network Calculus computations like min-plus and max-plus convolution and deconvolution.

RTC Toolbox can be applied in numerous situations and put users in complete control of how to implement their analysis. Users can implement TFA, SFA, PMOO, and LUDB with the tools in RTC Toolbox. A high level of control exists for users. RTC Toolbox allows for the creation, manipulation, and computation of arrival and service curves of arbitrary complexity. It is useful for giving users intimate knowledge of how to implement these analyses because they must implement the network and the accompanying analysis directly.

An additional advantage of RTC Toolbox is that it provides piecewise arrival curves and service curves, which can support many more types of curves other than the leaky bucket and rate-latency curves, which can bound arrival curves and service curves more accurately. This increases the applicability of RTC Toolbox, where higher resolution curves are required. There is no predefined way to calculate leftover service curves in RTC Toolbox, which can be defined by users independently for each flow and each switch. It means that RTC Toolbox can be applied to all types of feed-forward networks and most of the scheduling algorithms. Zhao et al. [7] use this library to analyze a mixed-criticality TSN network. Chakraborty et al. [8] also used RTC Toolbox to analyze the performance of systems-on-chip (SoC). RTC Toolbox also allows users to implement the Modular Performance Analysis (MPA) [4] [5]. The combination of MPA and RTC can analyze hard real-time systems. The flexibility of allowing users to define leftover service curves is a limitation of RTC Toolbox because users need to spend a lot of time on calculating leftover service curves.

However, it is cumbersome to compute a complete analysis as the complexity of the network increases. Given a particular network, the setups for utilizing three methods are totally different, so there is no reuse of code. Thus, conducting multiple analyses for the same network takes a significant increase in time and effort. For simple network topologies, this may not be a big issue, but when the system is complicated, it will take much effort. Another problem is that there is no GUI in RTC Toolbox, so one should pay close attention to keep track of complex network topologies and traffic patterns. Furthermore, RTC Toolbox does not provide a function for computing the time coordinate of the intersection of the arrival curve and the service curve as shown in Figure 1, making the computation of the delay for arbitrary multiplexing using TFA more difficult.

B. DEBORAH

Deborah is a toolbox used to derive the least upper and lower bound of the worst-case delay in tandem networks with FIFO policies primarily using the LUDB method [9].

Deborah runs in the command line, taking in a configuration file that defines the topology and traffic patterns of the flows, as well as tags that modify the standard LUDB analysis method. Deborah only supports tandem networks. Thus, the network topology is defined as a list of nodes. The configuration file takes the number of nodes or switches accompanied by their respective service curves, the sources and destinations of flows, and the arrival curves of the flows. There are samples associated with the software for user's reference. This program only supports rate-latency service curves and the leaky bucket arrival curves. Deborah does not provide a GUI. Deborah provides predefined functions for Network Calculus computations and it may need to be manually compiled.

The application range of Deborah is quite limited. It can only analyze the best-effort networks, which are work-conserving networks treating all packets in the same way, because the way to compute the leftover service curves is fixed. The network topology cannot be modified, and the multiplexing strategy can only be FIFO. Deborah is used to perform the LUDB analysis method and cannot perform TFA, SFA, or

PMOO computations. Furthermore, the results Deborah provides may not be clear without the necessary understanding of the LUDB analysis. Bouillard et al. [12] [10] use Deborah to analyze FIFO tandem networks and compare its results with the results of the optimization method. Bisti et al. [11] evaluates the running time of Deborah.

In conclusion, Deborah is not applicable to most circumstances since most networks are not tandem networks. However, it is convenient to use Deborah if the tool is applicable. Besides the upper bound, Deborah can provide a lower bound as well. Thus, for users who want to get both upper and lower bounds of the worst-case delay of a FIFO tandem network, Deborah is a suitable tool.

C. DISCODNC

DiscoDNC is a publicly available toolbox written in Java. Along with the release of a recent version of DiscoDNC, a paper introduces the tool and provides an example of how to implement a network and its analysis [16]. The example code in the article has been deprecated, but the package itself contains several templates and examples.

DiscoDNC implements three Network Calculus methods: TFA, SFA, and PMOO, which are used to calculate the backlog and delay of the network. The tool automates most of the steps of the analyses. Its analysis requires creating a *servergraph* object in Java. The *servergraph* object holds the structure of the network and paths of flows. It can represent arbitrarily complex networks with feed-forward flow paths. The analyst needs to set up the system and the necessary specifications, including service curves of switches, arrival curves of flows, the network topology, and the flow paths. The codebase is modularized, which makes it easier to use and maintain. Additionally, no external libraries are needed to use the latest version of the tool. Although the absence of external libraries may increase the amount of code, it eliminates the dependencies on external libraries and increases the toolbox's stability.

DiscoDNC supports both FIFO and arbitrary multiplexing strategies for latency and backlog calculations. The assumption of arbitrary multiplexing in switches increases the bound of the worst-case delay of the network. This assumption in DiscoDNC indicates that any flow passing through the switch will receive the lowest level of leftover service.

It is straightforward and quick to run PMOO, TFA, and SFA compared with RTC Toolbox, where the implementation details are left to users. The normal process is to run the three methods and then compare results to find the tightest bound.

There are several papers using DiscoDNC to analyze the delay performance of different kinds of networks. Cattelan et al. [18] applies DicoDNC to AFDX networks. Schon et al. [19] investigate a potential unification of DiscoDNC and RTC Toolbox. Bondorf et al. [14] extend DiscoDNC with a generalized version of the concatenation theorem so that DiscoDNC can be applied to analyze sensor networks.

One problem is that DiscoDNC only supports FIFO and arbitrary multiplexing. Toolboxes such as CyNC and WOPANets support more specific kinds of scheduling policies. RTC Toolbox can do this as well, although it must be implemented directly. For example, suppose that α_1 and α_2 are arrival curves for f_1 and f_2 respectively and β is the service curve of a node. Both of the two flows pass through the single node. Leftover service for f_1 is generally calculated using the formula $\beta_1^{l.o.} = [\beta - \alpha_2]^+$ under arbitrary multiplexing in DiscoDNC. For tools which support schedulers like WFQ, the minimum leftover service for f_1 may be calculated using the formula: $\beta_1^{l.o} = w_1 \times \beta$, where w_1 is the weight for f_1. A more accurate leftover service can be calculated by using tools supporting multiple scheduling policies. Another drawback of DiscoDNC is that it does not provide a GUI. Tools like WOPANets give a user-friendly GUI that can show the topology of the network created, which is straightforward for users to check the network configuration. Without it, users need to examine every line of code to determine whether the network's implementation is correct.

DiscoDNC is a suitable tool for researchers who have a basic understanding of Network Calculus and want to quickly construct the network and implement an analysis. Since DiscoDNC can support any network topology and any flow path, it is easier to use and has more extensive usage than most of the free tools introduced in this paper. As long as the traffic arbitration being used is either arbitrary multiplexing or FIFO, DiscoDNC is one of the best public tools for Network Calculus.

D. NC-MAUDE

NC-maude is an open and extensible tool written for Network Calculus [20]. When the author developed NC-maude, there were already several popular tools for Network Calculus. Thus, the development of NC-maude aims to solve some existing problems that previous tools encountered. NC-maude takes advantage of the rules in convolution and deconvolution to make the computation easier. For example, the convolution of two concave functions is the minimum of the them. NC-maude uses rationals instead of floating numbers. It gives more tractability to the results than the previous tools. Boyer et al. [21] use NC-maude to test the results of RTaW-Pegase.

NC-maude provides basic operations for Network Calculus as RTC Toolbox does. However, NC-maude can only provide rate-latency service curves and the leaky bucket arrival curves, while RTC Toolbox offers more types of piecewise linear curves. NC-maude can build network topologies like DiscoDNC as well.

The tool is developed based on Maude, which is a high-performance reflective language [52]. Users might need to learn the basic syntax of Maude to read the source code of NC-maude. It may increase the learning difficulty of the tool.

In summary, NC-maude is a tool that provides embedded functions to simplify the operations in Network Calculus. It requires users to have a basic knowledge of Maude. The lack of GUI may increase the difficulty of constructing the network topology in NC-maude.

E. CYNC

CyNC, which is short for Cyclic Network Calculus, is a performance-analysis tool for Network Calculus based on Matlab/Simulink environment [22]. There are many types of network elements supporting various scheduling algorithms, including fixed priorities, FIFO, round-robin/token passing, EDF, and WFQ embedded in the toolbox. None of the tools previously introduced in this section contain embedded functions of different scheduling algorithms. Examples of using these scheduling algorithms are given in the toolbox and [22]. When one wants to perform an analysis with the scheduling policies mentioned above, using CyNC can save a lot of time. CyNC provides a workflow generator for flows in the network. The generator takes numerical data as the inputs and outputs a lower bound and an upper bound of cumulative bits of the flow.The workflow generator also takes the jitter of the flow into account when calculating the upper bound and the lower bound of the arrival curve. Compared with most Network Calculus tools taking the arrival curve as inputs, CyNC provides a method to calculate the upper and lower bounds of arrival curves of the flow. The tool offers modules to calculate the output arrival curves, leftover service curves, and the overall service curve using the concatenation theorem.

CyNC is a tool developed based on RTC Toolbox, providing Simulink interfaces. Thus, CyNC provides the same result as RTC Toolbox when using the same analytical method. It inherits advantages from RTC Toolbox. Moreover, it has plenty of embedded functions so that users can directly get the performance results of specific scheduling algorithms. The drawbacks of CyNC are similar to those of RTC Toolbox. When the network is large, it is hard to use CyNC since network topologies, and traffic patterns cannot be implemented easily, although CyNC has a Simulink interface.

F. CATS

CATS is a tool designed for real-time systems. It can analyze the performance of the systems using both time automata and real-time calculus. It uses the concept of arrival curves in Network Calculus to describe the transducer derived from time automata. The tool is developed as a plugin for Eclipse, which is an integrated development environment.

CATS focuses on the timing analysis of real-time tasks. It can provide both the best and the worst-case response times of tasks after giving the task pattern and the computational resources. Although it uses Network Calculus knowledge, it is not a toolbox to analyze the network performance. One may follow a step-by-step tutorial [53] to learn how to use this tool. However, this tool is rarely used in real-time system papers. Although using the knowledge of Network Calculus, CATS is not a tool used to analyze network performance. Thus, we do not use CATS to estimate the worst-case delay bound in this paper.

G. WOPANETS

WOPANets is short for WOrst-case Performance Analysis of Embedded Networks [27]. WOPANets provides a GUI which is easy to use. Its GUI can show the network topology and plots of network information, which improves users' experience of the software. Similar to CyNC, WOPANets supports different scheduling policies.

Moreover, WOPANets implements different MAC (Media Access Control) mechanisms such as TDMA, Master/Slave, and Token Ring. Besides the delay and the backlog, the tool gives the result of the network load and loss rate as well. Flows in the network topology can be both periodic and aperiodic with or without jitter. Burstiness, deadline, period, jitter, priority, and length of the flow can be set through GUI or in an XML file. The tool supports unicast, multicast, and broadcast communication. It provides functions for Ethernet, AFDX, and SpaceWire [54] [55]. Daigmorte et al. and Ayed et al. [29] [30] apply WOPANets to analyzing AFDX networks.

There is an optimization analyzer in WOPANets, which is able to specify the variables, constraints, and the optimization problem's objective function. The tool uses the simplex [56] or heuristic algorithms, such as a genetic algorithm, to solve the optimization problem.

DiscoDNC, RTC Toolbox, Deborah, NC-maude, and CyNC are all public tools, while WOPANets is a private tool. Users need to pay attention to the cost of the software when they decide to use WOPANets. WOPANets has a broader range of applications than most of the public tools. It provides a GUI, data collector, and even protocols of different network types. The tool's highlight feature is that it provides an optimization analyzer to minimize the delay or maximize network utilization. One problem in this tool is the way in which it implements Network Calculus. The program uses a so-called propagation analysis for FIFO networks. The pseudocode of the algorithm is shown in [27]. As we can see in this algorithm, the algorithm works similarly to TFA, and it does not use the concatenation theorem. Thus, the result of the calculation may not be tight enough because TFA always gives looser delay bounds than SFA.

H. DIMTOOL

DIMTOOL is a private tool, and a major difference between DIMTOOL and previous tools is that DIMTOOL supports both Network Calculus analysis and simulations [32]. In communication networks, three primary methods are used to determine the delay bounds of real-time systems: (i) analytical methods, (ii) network simulations, and (iii) measurements. Usually, a combination of the three methods is applied to find the correct results. Network Calculus is one of the analytical methods.

DIMTOOL uses CSV files to describe network topologies. The arrival curves can be the leaky bucket or the dual bucket curves. The tool uses DiscoDNC, so DIMTOOl can support TFA, SFA, and PMOO. In this paper, we evaluate the performance of DiscoDNC. The results of DiscoDNC can represent the performance of DIMTOOL to some extent.

DIMTOOL implements the WCS (Worst-Case Simulation) mentioned in [57] to get the worst-case delay in the simulation. Thus, DIMTOOL can provide better worst-case scenario descriptions than most network simulators. Network simulators such as OMNeT++ [58], OPNET [59] and ns3 provide discrete-event simulations, which cannot guarantee to get the worst-case delay. Simulation results may reveal an inaccuracy of the analysis if the end to end latency of the simulation surpasses the analytical bound. This tool can be used to confirm the validity of the analytic model without need for a supplementary software tool.

The disadvantages of DIMTOOL are that it doesn't support different types of networks like AFDX and TSN, and doesn't support different types of scheduling algorithms, such as weighted round-robin. Thus, the application of the tool is restricted to limited domains.

I. DELAYLYZER

DelayLyzer is a private tool that provides a user-friendly GUI [34]. It can use SFA, TFA, and a combination of them to implement Network Calculus analyses. After the calculation, the delay can be visualized through the GUI.

Although DelayLyzer is a private tool, it only provides basic functions of Network Calculus. This tool supports a modified TFA (mTFA), which is a combination of SFA and TFA [60] [34]. However, DelayLyzer does not contain functionalities as rich as those provided by other private tools.

J. SINETPLAN

SINETPLAN is short for Siemens Network Planner. It is developed for designing and analyzing PROFINET. In SINETPLAN, there is a Network Calculus Engine that uses piecewise linear arrival curves and rate-latency service curves. SINETPLAN can do the simulation and optimize the network by calculating the network load down to the port level. It also offers a cost-optimization that can reduce the cost of wires.

SINETPLAN is specifically designed for PROFINET. Thus, the application of SINETPLAN is limited. However, this tool is suitable for the industry because it provides built-in models of industrial devices in the software. Thus, users do not need to collect and set the parameters of industrial equipment by themselves in SINETPLAN.

K. RTAW-PEGASE

RTaW-Pegase is a private tool written in Java that can provide network analyses and optimization. It can support many technology types, such as TSN, AFDX, and TTEthernet. Network Calculus is used in the toolbox to offer tight delay bounds and buffer utilization. Moreover, RTaW-Pegase provides simulations for networks as well. It offers a GUI to show the network topology so that the construction of the system can be simplified.

RTaw-Pegase is the only toolbox that supports TSN analyses and simulations. The tool offers a unique optimization mechanism called ZeroConifg-TSN (ZCT), which can dramatically reduce TSN's development time. The tool's optimization method can design the number and location of switches, the routing rules, and the allocation of the software functions. This method can both increase the performance of the network and decrease the cost of the system. Another important function of the tool is the Topology Stress Test (TST), which helps users choose the initial network topology without having full knowledge of network requirements. It is a very useful function for designers to start the design.

Since RTaW-Pegase has built-in functions for most types of networks, it supports most network protocols such as IEEE 802.1Q, IEEE 802.1CB, and IEEE 802.1Qbu. IEEE 802.1Qbu frame preemption performs a vital role in networks. When there are flows with different priorities, frame preemption can reduce the analytical delay bounds for high priority flows, while delay bounds for low priority flows may not be influenced significantly. RTaW-Pegase supports different scheduling algorithms such as FIFO, priority, Time-Aware Shaper (TAS), and Credit-Based Shaper (CBS). The Gate Control List (GCL) of TAS and parameters of CBS can be modified, which means that users can precisely control TAS and CBS. Migge et al. [39] analyze the performance of TSN and AVB networks using RTaW-Pegase. RTaW-Pegase can support both periodic and sporadic flows in the simulation and the Network Calculus calculation. Routing algorithms are provided by the toolbox to realize the load balancing for the network. Moreover, the toolbox offers built-in functions for Network-on-Chip (NoC), a special network that provides data exchange service between different processors on one chip. It means that the application range of RTaW-Pegase is quite extensive.

In summary, RTaW-Pegase is one of the most powerful tools among all public and private tools introduced in this paper. It is the only toolbox introduced in this paper that provides functions for TSN, TTEthernet, and NoC. Other tools would need to be extended to provide such capability. RTaW-Pegase supports different scheduling algorithms, as well. It provides both auxiliary design functions and optimization for networks. Users can save much time using this tool to design and optimize different types of networks. A GUI is provided to give users a clear view of networks. Many large companies, such as Airbus Group, Mercedes Benz, Renault

Group, and Robert Bosch, are customers of the tool. This tool contains the most comprehensive function compared with other Network Calculus toolboxes, and it can save effort for designers.

L. NC-TANDEM-TIGHT

Nc-tandem-tight is an executable program that is created based on the paper by Bouillard and Thierry [61] written in Ocaml. It is used to compute delay bounds under arbitrary multiplexing. Although nc-tandem-tight does not use methods mentioned in Section II-B, it uses Network Calculus concepts. Nc-tandem-tight uses Linear Programming to find the worst-case delay bound.

Nc-tandem-tight transforms information describing the network structure, nodal service curves, and flow arrival curves from the input text-file into a linear programming problem with constraints and an objective function. To get the worst-case delay bound, the objective function of the optimization problem is to maximize the gap between the packet's arrival time and the packet's departure time.

All the constraints will be automatically written into LP format files. Users can use a linear programming solver to get the result. The objective function can be chosen between maximizing end-to-end delay for a specific flow or maximizing the backlog of a specific server in the network.

An advantage of this method is that users do not need to know much about programming knowledge before using it because nc-tandem-tight auto-generates LP instances from the input. The resulting LP instances file can be solved using any standard solver.

A key disadvantage of this tool is that it only works for tandem networks. A way of working around this limitation is by "flattening" feed-forward networks in all the possible ways to create a set of tandem networks created by using a topological sort-based algorithm. This conversion of one feed-forward network topology to multiple tandem networks for analysis must be performed by users. Another potential disadvantage is that the executable is not self-contained: it needs an external LP to solve the optimization problem. As a result, users may need to write a script to automate the pipeline to solve several problems in a row.

In summary, nc-tandem-tight is a tool that is easy to use for researchers who understand the methods used in the paper by Bouillard et al. [61], and who are comfortable using LP problem solver software. As long as an arbitrary multiplexing tandem network is the subject of analysis, this is a suitable tool. Moreover, this tool provides a different view of deriving bounds of the delay and backlog of systems besides using Network Calculus.

In this section, we have discussed some well-known public and private Network Calculus toolboxes. In sections IV and V, we will use five public tools to implement analyses and verify the analytical results using OMNeT++ simulations.

IV. CASE STUDIES OF TANDEM NETWORK ANALYSIS

In this section, we show the analytical results of the worst-case delay in tandem networks using RTC Toolbox, DiscoDNC, nc-tandem-tight, Deborah, and CyNC. The reason for choosing these five tools is that they are all public toolboxes. Although NC-maude is a public tool as well, it is not chosen because (i) NC-maude can only handle service curves and arrival curves with integer parameters while we do use real numbers in our case studies, (ii) it uses maude language, which might be new to most users and lead to a steep learning curve, and (iii) NC-maude is not superior to other public tools in terms of functionalities and application domains. CATS is a public tool as well, but it is not designed to analyze the network performance.

Tandem networks are used because they are widely used in the real world. Mover, Deborah and nc-tandem-tight can only be applied to tandem networks. An example of the tandem network is represented in Figure 5.

The template for this tandem scenario comes from the paper by Bouillard and Thierry [61]. Ten different networks are analyzed: tandem networks of length 1, 2, 3, etc., up through 10. These tandem networks are used in [40] to evaluate the performance of nc-tandem-tight. Ten switches are enough for most tandem networks. Argibay-Losada et al. [62] propose a ring network with ten switches, which is used to manage vehicular traffic in a metropolitan area, in 5G scenarios. Soni et al. [63] evaluate an industrial-size network with eight switches. Hotescu et al. [64] conduct research on an AFDX network with seven switches. Thus, networks with ten switches can cover most applications of networks. As switches are added to the tandem network (from 0 up to 10), the pattern of flow paths continues. The flow of interest is the central flow, which passes through every switch. FIFO and arbitrary multiplexing are both used in the evaluations. In this section, we only derive the bound of the maximum end-to-end delay for the flow of interest because researchers generally concern more about latency than backlog.

To use these toolboxes, we need to predefine the service curves of all switches and arrival curves of all flows. Each switch has the same rate-latency service curve, with a latency of 1ms, and a service rate of 1Gbps. Each flow entering the network has a burstiness of 0.008Mb and an arrival rate of 2Mbps. The results of different analyses and different tools are shown in Figure 7. The reason for using this arrival curve is that it can be achieved by the periodical flows in OMNeT++. We use the simulation results of OMNeT++ to evaluate the delay bounds calculated by different tools. Figure 7 (a) shows the results of DiscoDNC, (b) shows the results of CyNC and RTC Toolbox, and (c) shows the results of Deborah and nc-tandem-tight.

A. DISCODNC RESULTS

As mentioned above, DiscoDNC can implement TFA, SFA, and PMOO using FIFO multiplexing or arbitrary multiplex-

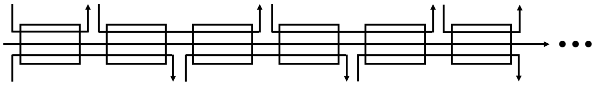


FIGURE 5. A 6-switch instance of an extendable tandem network [61].

ing. Additionally, DiscoDNC allows multiple analyses to be conducted on the same *servergraph*, which means that very little extra effort is needed to conduct all three analyses. This property makes DiscoDNC preferable to RTC Toolbox when users want to implement different analyses and multiplexing strategies. This work uses all three methods to calculate the worst delay bound for tandem networks. We derive the delay bound under both arbitrary multiplexing and FIFO multiplexing using SFA and TFA. PMOO can only be used under arbitrary multiplexing. Figure 7 (a) shows the results of all methods in DiscoDNC.

As expected, TFA gives the loosest bound, followed by SFA, while PMOO offers the tightest bound under the arbitrary multiplexing. Naturally, as the number of nodes increases, the time gap between the bounds of the different methods increases. For a ten-node tandem network, the TFA delay bound is 93.6ms, the SFA delay bound is 21.8ms, and the PMOO delay bound is as low as 16.8ms under arbitrary multiplexing. When we consider the situation that networks are under FIFO multiplexing, SFA provides tighter delay bounds than TFA. TFA under FIFO multiplexing provides smaller delay bounds than TFA under arbitrary multiplexing. SFA also offers smaller delay bounds under FIFO multiplexing than SFA under arbitrary multiplexing. Moreover, the analytical results of SFA can almost constitute linear curves. The linearity of the curve is caused by the similarity of the network structures and traffic patterns in ten networks. The delay bound will increase almost a fixed value when a new switch is added to the network.

In Figure 7 (a), we can see that SFA under FIFO multiplexing provides delay bounds smaller than those from PMOO. However, PMOO might derive smaller delay bounds in some situations. For example, when we increase the burstiness of the arrival curve to 1Mb and decrease the rate of the arrival curve to 0.67Mbps in 10 tandem networks, PMOO is able to provide smaller delay bounds than SFA under FIFO multiplexing, which is shown in Figure 6. In this case, although PMOO is under arbitrary multiplexing, its results should still be used to bound the maximum delay under FIFO multiplexing. The reason is that bits in the flow of interest do not need to wait for bits from other flows arriving later than them under FIFO multiplexing, while they need to do so under arbitrary multiplexing. Thus, delay bounds under FIFO multiplexing must be less than or equal to those under arbitrary multiplexing. PMOO can be used to bound the maximum delay under FIFO multiplexing if it can provide the smallest delay bounds among all other network calculus methods under FIFO multiplexing. The reason why PMOO can derive smaller delay bounds than SFA under FIFO multiplexing is that PMOO can provide larger leftover service curves. In our tandem networks, since all existing paths are subpath of the path of the flow of interest, PMOO does not encounter the increment of the bursts of flows when calculating leftover service curves. The way to calculate leftover service curves in a simple tandem case is shown in Equation 14. As we can see in Equation 13, SFA needs to consider the increment of the burst caused by the first switch when calculating the leftover service curve of the second switch in Figure 3. Thus, the leftover service curves calculated by SFA are smaller than those calculated by PMOO in tandem networks, which lead to larger delay bounds. Although SFA under FIFO multiplexing provides leftover service curves larger than those calculated by SFA under arbitrary multiplexing, it cannot eliminate the influence of the increment of bursts. Thus, PMOO can still derive delay bounds smaller than those from SFA under FIFO multiplexing.

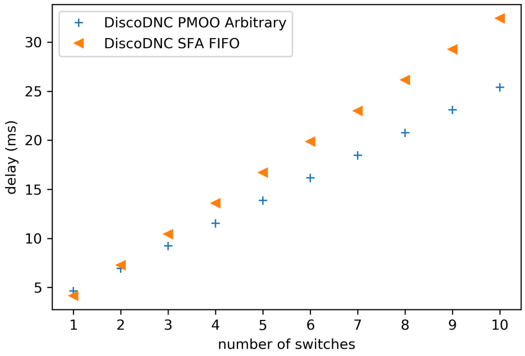


FIGURE 6. An example of when PMOO derives smaller delay bounds than SFA under FIFO multiplexing. This analysis sets all arrival curves' burstiness to 1Mb and the rate of all arrival curves to 0.67Mbps.

B. RTC TOOLBOX AND CYNC RESULTS

Recall in section III-E we mentioned CyNC is developed based on RTC toolbox, so the two tools will provide the same result when we use the same method. Thus, the two tools will have the same results in this evaluation, which is shown in Figure 7 (b). Since CyNC and RTC Toolbox do not provide functions to calculate the intersection of two curves, the RTC Toolbox is not capable of implementing TFA under arbitrary multiplexing. We want to compare the impact that the different methods have on the latency results. We can draw a similar conclusion from Figure 7 (b) to that of DiscoDNC results. PMOO derives the tightest bounds among all methods under arbitrary multiplexing. And SFA under FIFO multiplexing offers delay bounds smaller than those offered by SFA under Arbitrary multiplexing. One difference is that the delay bounds derived by SFA under FIFO multiplexing are larger than those derived by PMOO in this case. We have discussed that the network with FIFO multiplexing should have delay bounds smaller than those of the same network with arbitrary multiplexing in Section IV-A. Thus, the results of PMOO can bound the maximum delay when the network uses FIFO multiplexing instead of using the results of SFA under FIFO multiplexing when using CyNC and RTC Toolbox.

C. NC-TANDEM-TIGHT AND DEBORAH RESULTS

With nc-tandem-tight, the analysis takes only a few seconds with a script to run from a one-node, up through a ten-node implementation. Since nc-tandem-tight only works for arbitrary multiplexing, we will compare the results of nc-tandem-tight with the results of other methods under arbitrary multiplexing in Section IV-D. Although nc-tandem-tight only works for tandem networks, it does not mean that the Linear Programming based method cannot be applied to other feed-forward networks. It just requires users to generate constraints by themselves when analyzing other topological feed-forward networks. Using Deborah, we can also derive the delay bounds of ten tandem networks. Deborah is only used for networks under FIFO multiplexing. The results of nc-tandem-tight and Deborah are shown in Figure 7 (c).

D. COMPARISON OF RESULTS FROM DIFFERENT TOOLS

Since we have derived delay bounds using different tools, we compare the results to observe their differences. We first compare the results from DiscoDNC to those from RTC Toolbox and CyNC. We can find that these three tools have the same PMOO results. However, the delay bounds derived by DiscoDNC using SFA and TFA are much tighter than those derived by RTC Toolbox and CyNC using the same methods. The reason for the discrepancy between the DiscoDNC results and the RTC Toolbox/CyNC results is that they use different ways to derive the output arrival curves and leftover service curves. In our implementation using RTC Toolbox and CyNC, we implement SFA and SFA in the way that each flow influences the leftover service curves of all other flows passing the same switch. In the DiscoDNC implementation, however, some flows are ignored when calculating the leftover service curves and output arrival curves. Thus, DiscoDNC derives larger leftover service curves, which lead to smaller delay bounds.

Figure 7 (c) shows the results of Deborah and nc-tandem tight. The results of Deborah's LUDB analysis closely track the delay bounds computed by nc-tandem-tight, and the PMOO results from DiscoDNC. For networks of size 1-8, Deborah produces tighter results. Than nc-tandem-tight and PMOO. For networks of size 9-10, the delay is slightly larger. Compared with the results of SFA under FIFO multiplexing in DiscoDNC, Deborah's results are slightly larger. There are several reasons for this. The primary contributor for the faster increase in the delay bound as the network lengthens is that the LUDB method is primarily intended for nested flows. The tandem network modeled in this experiment does not have a nested flow structure, so to use the LUDB method, the network is cut into ten subsections (in the ten node case) and these are each separately analyzed, and the resulting latency for the flow of interest is the sum of the individual latencies of the sub-network. Every cut that is performed in order to use the LUDB increases the overall end to end latency. For nested tandem networks and sink trees in general, the method finds the global minimum delay, so for smaller networks, the inefficiency of finding a global optimum for a subsection of the network does not lead to a looser delay bound. As the network grows, the effect of concatenating adjacent network delays is magnified.

When we look at the results of nc-tandem-tight, we can find that it is exactly the same as PMOO results from DiscoDNC and RTC Toolbox/CyNC. Nc-tandem-tight uses the linear programming method to derive the maximum end-to-end delay bounds. Based on [40], the constraints generated by nc-tandem-tight are constrained by the service curves, arrival curves, and the relationships between the outgoing arrival curves and the incoming arrival curves. Since linear programming problems are convex, the solution of the linear programming problem should always be the global optima if constraints are feasible. Thus, we can conclude that PMOO can also provide global optima for tandem networks under arbitrary multiplexing.

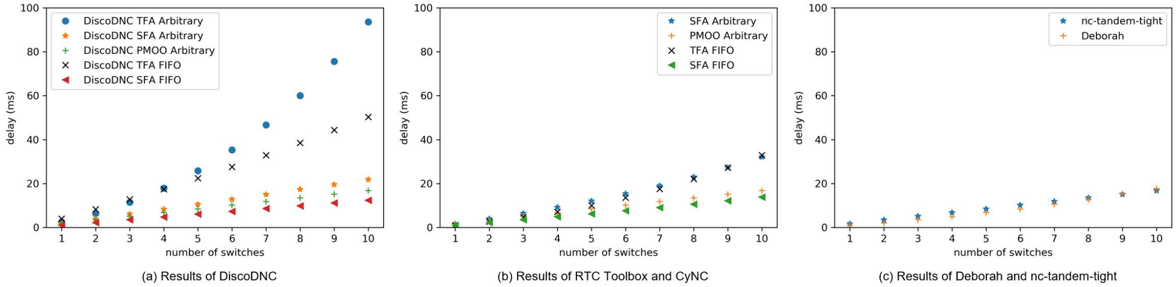
Among the results from all five tools, we can conclude that PMOO and nc-tandem-tight provide the tightest delay bounds for ten tandem networks under arbitrary multiplexing, while SFA under FIFO multiplexing provided by DiscoDNC offer the tightest delay bounds for networks under FIFO multiplexing. Thus, we will use the results of SFA under FIFO multiplexing provided by DiscoDNC in Section V.

V. SIMULATION RESULTS

We have derived the delay bounds for tandem networks in Section IV. However, delay bounds from Network Calculus are analytical bounds. Simulation results can be good supplements to analytical results. Although simulation results cannot verify the correctness of analytical results, they can show the pessimism of Network Calculus. If the maximum delay in the simulation is larger than the worst-case delay bound calculated by Network Calculus, the analytical results might be inaccurate.

Some toolboxes, such as DIMTOOL, RTaW-Pegase, and SINETPLAN, have simulation functions bundled with tools. Using toolboxes like these, users can omit the extra simulation steps. Moreover, DIMTOOL provides a worst-case simulation that has a higher probability of achieving the maximum delay during the simulation. However, these toolboxes are all private. They may not be affordable for everyone. Thus, the auxiliary network simulators are necessary for public tools.

In this section, we use OMNeT++ to simulate the FIFO tandem networks that are used in Section IV. OMNeT++



(a) Results of DiscoDNC (b) Results of RTC Toolbox and CyNC (c) Results of Deborah and nc-tandem-tight

FIGURE 7. Analytical results using different methods and different tools. (a) shows the results of DiscoDNC. (b) shows the results of RTC Toolbox and CyNC. (c) shows the results of nc-tandem tight and Deborah. The template for this tandem network is seen in Figure 5.

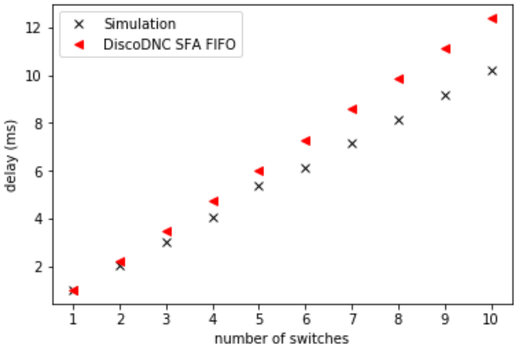


FIGURE 8. The figure shows the analytical results and simulation results of delay bounds of ten tandem networks. The network is FIFO, and the analytical results are derived from SFA using DiscoDNC.

is a well known object-oriented simulator written in C++. It can be used for traffic modeling and protocol modeling. In the simulation, we use the same parameters of the arrival curve and the service curve, and the same tandem networks as those used in Section IV. Among all results from Section IV, DiscoDNC SFA FIFO provides the tightest delay bounds. Since switches in OMNeT++ are FIFO, we compare the maximum delay in each simulation with the delay bounds derived by SFA under FIFO multiplexing.

Both analytical results and simulation results of worst-case delay bounds are shown in Figure 8. The analytical delay bound is 13.7ms when there are ten switches in the network, while the simulation result is 10.2ms in the same network. As shown in 8, analytical results are always larger than simulation results. This agrees with our expectations from the analytical results. When the number of switches increases, the gap between the simulation results and the analytical results becomes larger. It means that Network Calculus provides more conservative results when the network becomes larger. However, the simulation results in OMNeT++ might not provide the worst-case scenario, which has been discussed in Section III-H.

VI. RECOMMENDATIONS

There are many factors, such as languages and scheduling policies, influencing the choice of Network Calculus toolboxes. When users want to look for a toolbox for Network Calculus, they should make choices based on the usage scenarios and their skills. After considering the overall information, one can select a suitable tool to use. In this section, we discuss how to select tools based on different factors, which influence the choice of tools. Figure 9 illustrates the Network Calculus tools surveyed in this paper and provides the main factors for tool selection consideration.

A. SCALABILITY OF TOOLS

The scalability of the tool is very important. When the network is large, and the traffic pattern is complicated, it is hard to use some tools because they are really time-consuming. In some tools like RTC Toolbox and CyNC, implementing analyses for complex networks are troublesome because users must spend much effort to calculate leftover service curves and construct network topologies by themselves. DiscoDNC provides a class called *servergraph* to build the network topology and traffic pattern, and it has embedded functions to calculate the analytical results. However, when the complexity of the network increases, it still costs a lot of time to construct the *servergraph*.

The scalability of private tools is better than public tools. DIMTOOL can use CSV files to describe the network. SINETPLAN, WOPANets, RTaW-Pegase, and DelayLyzer all have GUIs to configure the network. Thus, when the system is complicated, users can choose tools providing interfaces that are used to construct the network topology to save effort.

B. TIGHTNESS OF BOUNDS

One critical thing to be considered is the tightness of the bounds that one needs: PMOO and SFA give a significantly tighter bound than TFA. However, most tools cannot support all methods. In order to get a tight bound, it is reasonable if the tool can support SFA and PMOO simultaneously, and then users can take the minimum. In the previous function, we discuss that PMOO can obtain global optima because

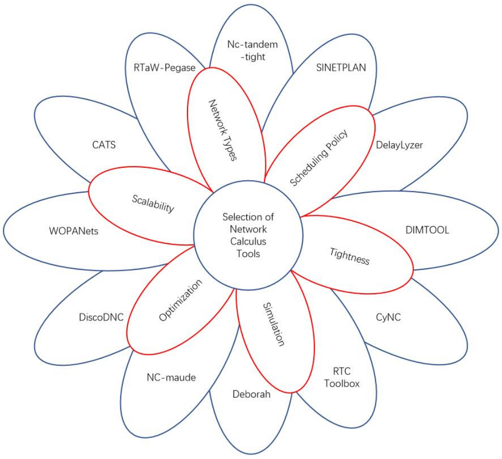


FIGURE 9. The figure shows the main factors for tool selection consideration in the inner layer and the Network Calculus tools surveyed in this paper in the outer layer.

it has the same result as nc-tandem-tight. Thus, in most cases, PMOO can provide the tightest bound under arbitrary multiplexing. However, since results from PMOO can also bound the worst-case delay under FIFO multiplexing, it is important to compare SFA results and PMOO results under FIFO multiplexing. Thus, if users want to get a tighter bound or compare the results derived from different methods, DIMTOOL and DiscoDNC are suitable options. If users want to derive results using LUDB, which can provide tight worst-case delay bounds for FIFO multiplexing, then Deborah is the only candidate.

C. SCHEDULING POLICIES

Scheduling policies that users want to implement conclusively influence the choice of tools. Fidler et al. [23] conduct a survey on how to calculate the leftover service curve for a particular scheduling policy. RTC Toolbox does not have a function for finding the intersection between the arrival curve and service curve as shown in Figure 1. Thus, there is no convenient way to calculate delays under arbitrary multiplexing using TFA in RTC Toolbox. However, RTC Toolbox does allow users to implement varieties of leftover service curve computations to derive leftover service curves in different scheduling policies. RTC allows users to specify the scheduling policies at specific nodes explicitly. The calculation of leftover service curves in DiscoDNC works cannot provide tight delay bounds for networks, which are not best-effort networks. In toolboxes like CyNC, WOPANets, RTaW-Pegase, and DIMTOOL, there are embedded functions for different scheduling policies. RTaW-Pegase even contains unique scheduling algorithms: TAS and CBS, which are not provided by other tools. When users aim to analyze the network with special scheduling policies, using toolboxes supporting these policies can save effort and provide accurate results.

D. TYPES OF NETWORKS

The type of networks is one of the most critical factors that we need to take into consideration when selecting the tool. Several popular types of networks, such as AFDX, TSN, and PROFINET, attract much attention from researchers. Different types of networks require different protocols. When analyzing a specific type of network, choosing a proper toolbox is necessary.

TSN networks are designed for time-critical flows [65], which have hard requirements for the maximum delay. Since traditional best-effort networks cannot provide guaranteed service to time-critical flows, TSN is developed using TAS, CBS, and Strict Priority (SP) to provide guaranteed delay bounds for time-critical flows. However, most toolboxes cannot provide automated computation for TSN. Thus, when users want to analyze TSN networks, they can only use RTaW-Pegase since RTaW-Pegase provides all protocols needed by TSN. However, if users want to use public tools, they may use RTC Toolbox because of its flexibility, while users need to spend much more time developing the analysis than when using RTaW-Pegase.

AFDX is an enhancement for Ethernet in avionic systems [66]. Some additional features, such as virtual links and redundancy, are appended to provide guaranteed service for flows in AFDX. WOPANets and RTaW-Pegase provide functions for AFDX networks. If AFDX networks are the targets, then these two toolboxes are the right candidates. Since AFDX is developed based on the Ethernet, other tools can still perform analyses for AFDX, although they do not contain built-in functions for AFDX.

PROFINET is a standard for data transmission over industrial Ethernet. It is designed to collect data and control facilities. Similar to TSN, data in PROFINET has tight time constraints. There are IO-Controllers, IO-Device, and IO-Supervisor in the network. SINETPLAN is designed for PROFINET. It provides both Network Calculus analyses and simulations for the system. Moreover, it has built-in models of industrial equipment so that users can easily configure PROFINET networks. SINETPLAN is the best tool for PROFINET.

NoC [67] is used to address global communication in large scale SoC (Systems-on-Chip). NoC is built based on computer networking theories. It provides larger bandwidth and full-duplex links compared with traditional buses. Since the number of processors is increasing in one chip, NoC plays an important role in enabling communication between different processors in SoC. Among tools introduced in this paper, RTaW-Pegase has a embeded function for NoC.

In summary, the selection of tools is heavily dependent on the technology types of networks, since various protocols are used in different types of networks. A suitable tool can provide accurate results and save effort.

E. SIMULATION

Toolboxes that support simulations should be considered if users want to derive and validate the analytical results simultaneously. DIMTOOL, RTaW-Pegase, and SINETPLAN are three toolboxes with built-in simulation functions. DIMTOOL even provides a worst-case simulation, which is unique in all toolboxes. Since Network Calculus assumes the worst-case scenarios of arrival curves and service curves, delay bounds from simulations are usually smaller than the analytical results. Thus, simulation bounds in the toolboxes are good supplements of Network Calculus analyses. However, simulations can be implemented using network simulators. Since DIMTOOL, RTaW-Pegase, and SINETPLAN are all private toolboxes, a combination of free toolboxes and public network simulators can be used to replace the private tools. For example, in Section V, we used OMNeT++ to get the simulation results.

F. OPTIMIZATION

Sometimes users need to optimize the network based on the analytical result. Three toolboxes in this paper, which are RTaW-Pegase, WOPANets, and SINETPLAN, have embedded optimization functions. Optimization functions in three toolboxes can both minimize the delay and maximize the utilization of networks. SINETPLAN has a particular optimization function, which is the cost-optimization. Since SINETPLAN is designed for industrial systems, it can also reduce the wire length to reduce the cost of systems. Optimization functions in RTaW-Pegase can design the location and number of switches. Thus, it can reduce the wire length by setting the location of switches. Furthermore, it can decrease the cost by reducing the number of switches needed in the network. In summary, both RTaW-Pegase and SINETPLAN provide powerful optimization functions to reduce the cost and increase the performance of the network.

G. LANGUAGE

Language is an essential factor in choosing a toolbox. However, since most private tools provide a GUI and are proprietary, the language is not vital in these private tools. However, language is important to open-source public tools. Users can extend the open-source public tools to implement functionalities, which are not supported by original tools. If users are familiar with Java, users can understand the implementation and extend DiscoDNC. If Matlab and Simulink are preferable to users, CyNC and RTC Toolbox can be suitable tools.

VII. CONCLUSION

Network Calculus is a theory used to estimate the network performance of real-time systems. In this paper, we have introduced twelve different tools which implement Network Calculus analyses, and discussed their advantages and disadvantages. Moreover, we have compared the results of five public tools in terms of the tightness of the delay bound. We also provide recommendations on tool selections based on different factors.

While the benefits of Network Calculus are recognized, there are some limitations of the tools used to implement Network Calculus analyses of which one should be aware. One limitation comes from how the arrival and service curves are modeled. There are no toolboxes having the ability to automatically model service curves and arrival curves currently. Most toolboxes ask for user inputs to set service-curve and arrival-curve parameters. In the future, toolboxes may be able to characterize service curves and arrival curves based on the arrival process of switches following specific protocols mentioned in [1], [68], and [69].

Another limitation of Network Calculus tools for real-time network analyses is that it only works for feed-forward networks. Many real-time systems have control loops, which necessitates bi-directional communication, and they cannot be analyzed without making additional assumptions. Providing deterministic bounds would be very useful for these applications because latency and jitter can heavily impact controllability, depending on the loop cycle time. Existing research [43] [70] provides ideas addressing this issue. However, existing toolboxes have not yet provided functionalities that can convert non-feed-forward networks to feed-forward networks. This may be future work of Network Calculus tool development.

VIII. ACKNOWLEDGEMENT

This work is supported by NSF Award No. 1646458. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the author(s) and do not necessarily reflect the views of the sponsors of the research.

REFERENCES

[1] H. Yang, L. Cheng, and X. Ma, "Combining Measurements and Network Calculus in Worst-Case Delay Analyses for Networked Cyber-Physical Systems," in IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 1065–1066.

[2] L. Zhao, F. He, E. Li, and J. Lu, "Comparison of time sensitive networking (tsn) and ttethernet," in 2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC), pp. 1–7.

[3] E. Wandeler and L. Thiele, "Real-Time Calculus (RTC) Toolbox," `http://www.mpa.ethz.ch/Rtctoolbox`, 2006, visited 2020-02-02. [Online]. Available: http://www.mpa.ethz.ch/Rtctoolbox

[4] E. Wandeler, Modular performance analysis and interface-based design for embedded real-time systems. ETH Zurich, 2006.

[5] E. Wandeler, L. Thiele, M. Verhoef, and P. Lieverse, "System architecture evaluation using modular performance analysis: a case study," International Journal on Software Tools for Technology Transfer, vol. 8, no. 6, pp. 649–667, 2006.

[6] L. Thiele and N. Stoimenov, "Modular performance analysis of cyclic dataflow graphs," in Proceedings of the seventh ACM international conference on Embedded software, 2009, pp. 127–136.

[7] L. Zhao, P. Pop, Z. Zheng, and Q. Li, "Timing Analysis of AVB Traffic in TSN Networks Using Network Calculus," in 2018 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), April 2018, pp. 25–36.

[8] S. Chakraborty, S. Kunzli, and L. Thiele, "A general framework for analysing system properties in platform-based embedded system designs," in 2003 Design, Automation and Test in Europe Conference and Exhibition, pp. 190–195.

[9] L. Bisti, L. Lenzini, E. Mingozzi, and G. Stea, "Deborah: A tool for worst-case analysis of FIFO tandems," in International Symposium On Leveraging Applications of Formal Methods, Verification and Validation. Springer, 2010, pp. 152–168.

[10] A. Bouillard and G. Stea, "Exact worst-case delay in FIFO-multiplexing feed-forward networks," IEEE/ACM Transactions on Networking, vol. 23, no. 5, pp. 1387–1400, 2014.

[11] L. Bisti, L. Lenzini, E. Mingozzi, and G. Stea, "Numerical analysis of worst-case end-to-end delay bounds in FIFO tandem networks," Real-Time Systems, vol. 48, no. 5, pp. 527–569, 2012.

[12] A. Bouillard and G. Stea, "Exact worst-case delay for FIFO-multiplexing tandems," in 6th International ICST Conference on Performance Evaluation Methodologies and Tools. IEEE, 2012, pp. 158–167.

[13] L. Grillmayer, "Determinism for Ethernet flows in industrial networks," Network, vol. 73, 2014.

[14] S. Bondorf and J. B. Schmitt, "Boosting sensor network calculus by thoroughly bounding cross-traffic," in 2015 IEEE Conference on Computer Communications (INFOCOM), pp. 235–243.

[15] S. Bondorf and J. Schmitt, "Calculating accurate end-to-end delay bounds-you better know your cross-traffic," in Proceedings of the 9th EAI International Conference on Performance Evaluation Methodologies and Tools. ICST (Institute for Computer Sciences, Social-Informatics and . . . , 2016, pp. 17–24.

[16] S. Bondorf and J. B. Schmitt, "The DiscoDNC v2: a comprehensive tool for deterministic network calculus," in Proceedings of the 8th International Conference on Performance Evaluation Methodologies and Tools. ICST (Institute for Computer Sciences, Social-Informatics and . . . , 2014, pp. 44–49.

[17] J. B. Schmitt and F. A. Zdarsky, "The disco network calculator: a toolbox for worst case analysis," in Proceedings of the 1st international conference on Performance evaluation methodologies and tools. ACM, 2006, p. 8.

[18] B. Cattelan and S. Bondorf, "Iterative design space exploration for networks requiring performance guarantees," in 2017 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC), pp. 1–10.

[19] P. Schon and S. Bondorf, "Towards Unified Tool Support for Real-time Calculus & Deterministic Network Calculus," in Proceedings of the 29th Euromicro Conference on Real-Time Systems (ECRTS 2017), Dubrovnik, Croatia, pp. 28–30.

[20] M. Boyer, "NC-maude: a rewriting tool to play with network calculus," in International Symposium On Leveraging Applications of Formal Methods, Verification and Validation. Springer, 2010, pp. 137–151.

[21] M. Boyer, N. Navet, and M. Fumey, "Experimental assessment of timing verification techniques for AFDX," in ERTS2012 - Embedded Real Time Software and Systems, Toulouse, France. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02189869

[22] H. Schioler, H. P. Schwefel, and M. B. Hansen, "Cync: a matlab/simulink toolbox for network calculus," in Proceedings of the 2nd international conference on Performance evaluation methodologies and tools. ICST (Institute for Computer Sciences, Social-Informatics and . . . , 2007, p. 60.

[23] M. Fidler, "Survey of deterministic and stochastic service curve models in the network calculus," IEEE Communications surveys & tutorials, vol. 12, no. 1, pp. 59–86, 2010.

[24] H. Schioler, J. D. Nielsen, K. G. Larsen, and J. Jessen, "CyNC: A method for real time analysis of systems with cyclic data flows," Journal of Embedded Computing, vol. 2, no. 3, 4, pp. 347–360, 2006.

[25] H. Schioler, J. Jessen, J. D. Nielsen, and K. G. Larsen, "CyNC-towards a General Tool for Performance Analysis of Complex Distributed Real Time Systems," in Session of the 17 th Euromicro Conference on Real-Time Systems, 2005.

[26] M. Moy and K. Altisen, "Arrival curves for real-time calculus: the causality problem and its solutions," in International Conference on Tools and Algorithms for the Construction and Analysis of Systems. Springer, 2010, pp. 358–372.

[27] A. Mifdaoui and H. Ayed, "WOPANets: a tool for WOrst case Performance Analysis of embedded Networks," in 2010 15th IEEE International Workshop on Computer Aided Modeling, Analysis and Design of Communication Links and Networks (CAMAD), pp. 91–95.

[28] R. A. Vingerhoeds, A. Mifdaoui, and P. d. Saqui-Sannes, "Educational Challenges For Cyber-Physical Systems Modelling," 2018.

[29] H. Daigmorte, P. de Saqui-Sannes, and R. Vingerhoeds, "A SysML Method with Network Dimensioning," in 2019 International Symposium on Systems Engineering (ISSE). IEEE, pp. 1–8.

[30] H. Ayed, A. Mifdaoui, and C. Fraboul, "Frame packing strategy within gateways for multi-cluster avionics embedded networks," in Proceedings of 2012 IEEE 17th International Conference on Emerging Technologies & Factory Automation (ETFA 2012), pp. 1–8.

[31] H. Ayed, "Analysis and optimiozation of heterogeneous avionics networks," Ph.D. dissertation, 2014.

[32] E. Heidinger, S. Burger, S. Schneele, A. Klein, and G. Carle, "DIMTOOL: A platform for determining worst case latencies in switched queuing networks," in 6th International ICST Conference on Performance Evaluation Methodologies and Tools. IEEE, 2012, pp. 45–53.

[33] E. Heidinger, S. Schneele, A. Klein, and G. Carle, "Modeling Aeronautic Networks for Internet Scenarios."

[34] M. Schmidt, S. Veith, M. Menth, and S. Kehrer, "DelayLyzer: a tool for analyzing delay bounds in industrial Ethernet networks," in Measurement, Modelling, and Evaluation of Computing Systems and Dependability and Fault Tolerance. Springer, 2014, pp. 260–263.

[35] M. Schmidt, T. Schneck, and M. Menth, "Multicast Extension for Network Calculus."

[36] S. Kerschbaum, K.-S. Hielscher, and R. German, "The need for shaping non-time-critical data in PROFINET networks," in 2016 IEEE 14th International Conference on Industrial Informatics (INDIN), pp. 160–165.

[37] F. Bruns, W. Nebel, J. Walter, and K. Gruttner, "Work-in-Progress: Modeling of real-time communication for industrial distributed automation systems," in 2020 16th IEEE International Conference on Factory Communication Systems (WFCS), pp. 1–4.

[38] M. Boyer, N. Navet, X. Olive, and E. Thierry, "The pegase project: Precise and scalable temporal analysis for aerospace communication systems with network calculus," in International Symposium On Leveraging Applications of Formal Methods, Verification and Validation. Springer, 2010, pp. 122–136.

[39] J. Migge, J. Villanueva, N. Navet, and M. Boyer, "Insights on the Performance and Configuration of AVB and TSN in Automotive Ethernet Networks," Proc. Embedded Real-Time Software and Systems (ERTS 2018), 2018.

[40] A. Bouillard, L. Jouhet, and E. Thierry, "Tight performance bounds in the worst-case analysis of feed-forward networks," in 2010 Proceedings IEEE INFOCOM, pp. 1–9.

[41] A. Bose, X. Jiang, B. Liu, and G. Li, "Analysis of manufacturing blocking systems with network calculus," Performance Evaluation, vol. 63, no. 12, pp. 1216–1234, 2006.

[42] X. Jiang, G. Parker, and E. Shittu, "Envelope modeling of renewable resource variability and capacity," Computers & Operations Research, vol. 66, pp. 272–283, 2016.

[43] H. Yang, L. Cheng, and X. Ma, "Analyzing Worst-Case Delay Performance of IEC 61850-9-2 Process Bus Networks Using Measurements and Network Calculus," in Proceedings of the Eighth International Conference on Future Energy Systems. ACM, 2017, pp. 12–22.

[44] A. Van Bemten and W. Kellerer, "Network calculus: A comprehensive guide," Tech. Rep. 201603, Oct. 2016.

[45] J.-Y. Le Boudec and P. Thiran, Network calculus: a theory of deterministic queuing systems for the internet. Springer Science & Business Media, 2001, vol. 2050.

[46] S. Bondorf and F. Geyer, "Generalizing Network Calculus Analysis to Derive Performance Guarantees for Multicast Flows." in VALUETOOLS, 2016.

[47] J. B. Schmitt, F. A. Zdarsky, and M. Fidler, "Delay bounds under arbitrary multiplexing: When network calculus leaves you in the lurch..." in IEEE INFOCOM 2008-The 27th Conference on Computer Communications, pp. 1669–1677.

[48] 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.

[49] L. Lenzini, E. Mingozzi, and G. Stea, "A methodology for computing end-to-end delay bounds in FIFO-multiplexing tandems," Performance Evaluation, vol. 65, no. 11-12, pp. 922–943, 2008.

[50] S. Bondorf, P. Nikolaus, and J. B. Schmitt, "Quality and Cost of Deterministic Network Calculus: Design and Evaluation of an Accurate and Fast Analysis," Proc. ACM Meas. Anal. Comput. Syst., vol. 1, no. 1, Jun. 2017. [Online]. Available: https://doi.org/10.1145/3084453

[51] C. E. ETH Zürich and N. Laboratory, "RTC Toolbox Tutorial", 2005-10-10. [Online]. Available: https://www.mpa.ethz.ch/static/Tutorial.html

[52] M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martı-Oliet, J. Meseguer, and J. F. Quesada, "Maude: Specification and programming in rewriting logic," Theoretical Computer Science, vol. 285, no. 2, pp. 187–243, 2002.

[53] U. University, "CATS Tutorial", 2007. [Online]. Available: http://www.it.uu.se/research/group/darts/times/cats/tutorial.html

[54] T. Ferrandiz, F. Frances, and C. Fraboul, "Using network calculus to compute end-to-end delays in spacewire networks," ACM SIGBED Review, vol. 8, no. 3, pp. 44–47, 2011.

[55] S. Parkes and P. Armbruster, "SpaceWire: a spacecraft onboard network for real-time communications," in 14th IEEE-NPSS Real Time Conference, 2005. IEEE, 2005, pp. 6–10.

[56] V. Klee and G. J. Minty, "How good is the simplex algorithm," Inequalities, vol. 3, no. 3, pp. 159–175, 1972.

[57] E. Heidinger, "Rare events in network simulation using MIP," in 2011 23rd International Teletraffic Congress (ITC). IEEE, pp. 314–315.

[58] A. Varga, "OMNeT++," in Modeling and tools for network simulation. Springer, 2010, pp. 35–59.

[59] X. Chang, "Network simulations with OPNET," in WSC'99. 1999 Winter Simulation Conference Proceedings.'Simulation-A Bridge to the Future'(Cat. No. 99CH37038), vol. 1. IEEE, pp. 307–314.

[60] M. Menth, M. Schmidt, S. Veith, S. Kehrer, and A. Dreher, "Comparison of delay bounds for Ethernet networks based on simple Network Calculus algorithms," in 2014 IEEE Symposium on Computers and Communications (ISCC). IEEE, 2014, pp. 1–7.

[61] A. Bouillard and É. Thierry, "Tight performance bounds in the worst-case analysis of feed-forward networks," Discrete Event Dynamic Systems, vol. 26, no. 3, pp. 383–411, 2016.

[62] P. J. Argibay-Losada, Y. Yoshida, A. Maruta, and K.-i. Kitayama, "Optical versus electronic packet switching in delay-sensitive 5G networks: myths versus advantages," Journal of Optical Communications and Networking, vol. 8, no. 11, pp. B43–B54, 2016.

[63] A. Soni, X. Li, J.-L. Scharbarg, and C. Fraboul, "WCTT analysis of avionics Switched Ethernet Network with WRR Scheduling," in Proceedings of the 26th International Conference on Real-Time Networks and Systems, 2018, pp. 213–222.

[64] O. Hotescu, K. Jaffrès-Runser, J.-L. Scharbarg, and C. Fraboul, "Impact of source scheduling on end-to-end latencies in a QoS-aware avionics network," in Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, 2019, pp. 643–646.

[65] A. Nasrallah, A. Thyagaturu, Z. Alharbi, C. Wang, X. Shao, M. Reisslein, and H. ElBakoury, "Ultra-low latency (ULL) networks: A comprehensive survey covering the IEEE TSN standard and related ULL research," arXiv preprint arXiv: 1803.07673, 2018.

[66] H. Bauer, J.-L. Scharbarg, and C. Fraboul, "Improving the worst-case delay analysis of an AFDX network using an optimized trajectory approach," IEEE Transactions on Industrial informatics, vol. 6, no. 4, pp. 521–533, 2010.

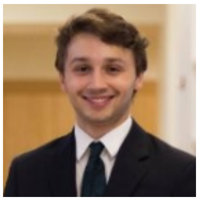
[67] T. Bjerregaard and S. Mahadevan, "A survey of research and practices of network-on-chip," ACM Computing Surveys (CSUR), vol. 38, no. 1, pp. 1–es, 2006.

[68] S. Pettie and V. Ramachandran, "An optimal minimum spanning tree algorithm," Journal of the ACM (JACM), vol. 49, no. 1, pp. 16–34, 2002.

[69] H. Yang, L. Cheng, and X. Ma, "Bounding network-induced delays for time-critical services in avionic systems using measurements and network calculus," in Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems, 2019, pp. 338–339.

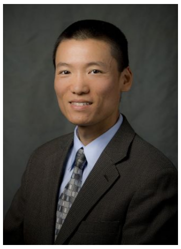
[70] D. Starobinski, M. Karpovsky, and L. A. Zakrevski, "Application of network calculus to general topologies using turn-prohibition," IEEE/ACM Transactions on Networking, vol. 11, no. 3, pp. 411–421, 2003.

BOYANG ZHOU [M](boz319@lehigh.edu) is a first year PhD student in Lehigh University Advised by Prof. Liang Cheng. His research area is using Network Calculus to design and analyze networks in Cyber-Physical Systems. He got his bachelor's degree in Electronic Science and Technology in Dalian University of Technology and Master's degree in Electrical Engineering in Washington University in St. Louis. He was a software engineer in Robert Bosch from September 2018 to June 2019. Boyang was responsible for Adaptive Cruise Control in Bosch. Then he decided to pursue a doctoral degree and enrolled in the computer engineering PhD program in Lehigh University.



ISAAC HOWENSTINE [M] (ish219@lehigh.edu) is a first year PhD student at Lehigh University advised by Prof. Liang Cheng. He got his bachelor's degree in Civil Engineering from the University of Illinois at Urbana-Champaign. His interest in computer science, transportation systems and infrastructure networks led him to pursue a PhD focusing on Network Science at Lehigh University.

SIRAPHOB LIMPRAPAIPONG (sil320@lehigh.edu) is an undergraduate student at Lehigh University advised by Prof. Liang Cheng. He is currently studying on the research paper about Optimization-based Network Calculus and focusing on implementing the optimization method to find the worst-case delay bound for feedforward network. He received a scholarship from Thailand's government to study in Computer Science at the United States from undergraduate to Ph.D.. After finishing prep school in 2016, Siraphob decided to pursue a dual degree program in Computer Science and Mathematics at Lehigh University.



LIANG CHENG [SM] (cheng@lehigh.edu) is an associate professor of Computer Science and Engineering at Lehigh University, directing the Learning and Optimization on Networks and Graphs Laboratory (LONG LAB). His research focuses on CPS/IoT (Cyber-Physical Systems/Internet of Things) and is geared toward enabling intelligent infrastructure based on the convergence of real-time sensing, model-driven data analytics, and machine learning through interdisciplinary projects. He has directed 7 Ph.D. students to their graduation, supervised 2 postdocs, advised 25 Master's degree theses, and co-authored more than 100 papers. His research group has been supported by funding agencies such as NSF, DOT, DOE, DARPA, PITA (Pennsylvania Infrastructure Technology Alliance), Christian R. & Mary F. Lindback Foundation, and industry companies. More information is available at http://liangcheng.info.

• • •