electronics

MDPI

Article

A Performance Analysis Framework of Time-triggered Ethernet Using Real-Time Calculus

Xiuli Yang [1], Yanhong Huang [1,2,*], Jianqi Shi [1,3] and Zongyu Cao [1]

[1] National Trusted Embedded Software Engineering Technology Research Center, East China Normal University, Shanghai 200062, China; xiuli.yang.ecnu@outlook.com (X.Y.); jqshi@sei.ecnu.edu.cn (J.S.); zongyu.cao@ntesec.ecnu.edu.cn (Z.C.)

[2] Shanghai Key Laboratory of Trustworthy Computing, Shanghai 200062, China

[3] Hardware/Software Co-Design Technology and Application Engineering Research Center, Shanghai 200062, China

[*] Correspondence: yhhuang@sei.ecnu.edu.cn

Received: 29 May 2020; Accepted: 29 June 2020; Published: 3 July 2020

Abstract: With increasing demands of deterministic and real-time communication, network performance analysis is becoming an increasingly important research topic in safety-critical areas, such as aerospace, automotive electronics and so on. Time-triggered Ethernet (TTEthernet) is a novel hybrid network protocol based on the Ethernet standard; it is deterministic, synchronized and congestion-free. TTEthernet with a time-triggered mechanism meets the real-time and reliability requirements of safety-critical applications. Time-triggered (TT) messages perform strict periodic scheduling following the offline schedule tables. Different scheduling strategies have an effect on the performance of TTEthernet. In this paper, a performance analysis framework is designed to analyze the end-to-end delay, backlog bounds and resource utilization of network by real-time calculus. This method can be used as a base for the performance evaluation of TTEthernet scheduling. In addition, this study discusses the impacts of clock synchronization and traffic integration strategies on TT traffic in the network. Finally, a case study is presented to prove the feasibility of the performance analysis framework.

Keywords: time-triggered Ethernet; performance analysis framework; delay; backlog; resource utilization; real-time calculus

1. Introduction

In recent years, as the continuous improvement of real-time demand for Ethernet in the aviation industry and other fields, over 20 forms of solutions have been continuously competing with each other to gain industry recognition. In the fierce competition, time-triggered Ethernet (TTEthernet) has gradually attracted more attention due to its characteristics of certainty, fault tolerance and reliability [1]. TTEthernet is a real-time communication protocol based on IEEE 802.3 standard Ethernet, which can fully support the timely-determinism property and mixed-criticality applications in a single network infrastructure. It is mainly applied in safety-critical industries such as aviation, automobiles, electronics, etc. [2].

TTEthernet is a promising extension of Ethernet, which introduces clock synchronization to synchronize the local clocks of all devices in the whole network. Under the control of a unified global clock, time-triggered (TT) messages are sent and received according to the offline schedule table [3]. The schedule tables are generated in an offline fashion and guarantee to be collision-free transmission of TT messages, thereby ensuring reliable and real-time transmission. Other event-triggered (ET) messages, including rate-constrained (RC) and best-effort (BE) messages [4], are transmitted during the idle periods after TT messages have been scheduled. Hence, TTEthernet is fully compatible with Avionics Full-Duplex Switched Ethernet (AFDX) and standard Ethernet.

The critical data in TTEthernet are transmitted in the form of TT traffic, so they are the core of TTEthernet to ensure the real-time and deterministic nature of TT traffic. TT traffic is scheduled based on the offline schedule table, which sets the acceptance windows of each TT frame. The schedule tables assign specific time slots for TT transmission, which is pre-designed after considering several factors such as the period of all TT traffic; the lengths of frames; and the restrictions from resources and physical links. The schedule is designed to facilitate a collision-free transmission between TT traffic, thereby minimizing a delay and jitter of TT traffic [5]. In the scenario of mixed-critical traffic, RC traffic has to be integrated with TT traffic via different scheduling strategies, which have a non-negligible impact on the performance boundary of the network.

Typically, simulation techniques, as traditional methods, are used to analyze the delay and resource utilization of TTEthernet. However, such methods usually need to expend considerable effort to build a TTEthernet model with multiple details, and it tends to take lots of time to execute the model to get simulation results. Real-time calculus (RTC) is a framework to model and analyze heterogeneous system, which can be used to analyze and evaluate the network performance based on an established abstract model.

In this paper, we present a performance analysis framework for evaluating TTEthernet and use RTC to calculate its hard upper and lower bounds. The processing capacity for critical traffic can be indicated so long as users give correlation parameters (such as event rate, message size, routing path and scheduling strategy) into a model. RTC was used to calculate delay bounds, backlog bounds and resource utilization in the transmission process from sender to receiver. This study mainly discusses the impacts of clock synchronization and traffic integration strategies for TT traffic. The main contributions of this paper are as follows:

1. This paper proposes a performance analysis framework of TTEthernet. The abstract model of TTEthernet consists of the data model, resource model and component model, and RTC is applied to analyze the feasibility of abstract model.
2. We discuss the impacts of clock synchronization and different traffic integration strategies on delay bounds of TT traffic, backlog bounds of processing nodes and resource utilization of the network.
3. Finally, the paper concludes the feasibility of the performance analysis framework through discussing a specific case that mainly focuses on different traffic integration strategies.

In the remainder of this paper, the content is organized as follows. The related work was discussed in Section 2. In Section 3, the relevant portions of TTEthernet and RTC were introduced. The performance analysis framework was described in Section 4, along with a detailed presentation for the construction of data and resource models. Section 5 showed how to obtain the performance analysis results. In Section 6, a case was applied to prove the feasibility of the performance analysis framework proposed in this paper. Lastly, Section 7 summarized the work.

2. Related Work

The deterministic and real-time nature of TT traffic in TTEthernet is mainly implemented through a predefined schedule. A well-designed schedule can ensure collision-free transmission and minimum delay in TT frames, which can guarantee the deterministic of TTEthernet. There are many scheduling algorithms that have been developed to generate schedule tables for efficient traffic transmission, such as [5–7]. The performance analysis framework proposed in this paper can be utilized to analyze whether the critical traffic can meet requirements of the system after scheduling according to these scheduling tables, so as to evaluate the merits and demerits of scheduling algorithms.

Traditional network analysis is conducted to obtain the final performance analysis result through actual simulation, such as [8]. Several details are required to form a TTEthernet model. This process is very time-consuming and requires tremendous amounts of resources. Another method, which can be used to analyze a complex performance of the network, is network calculus. For example, Zhao et al. used network calculus to analyze the worst-case end-to-end delay of RC traffic in TTEthernet [9], but it only analyzes the time delay, which is only part of the network performance. The RTC used in this paper is based on the network calculus, which can analyze the network performance at entire system level. It can analyze not just delay bounds but backlog bounds and resource utilization of network.

Performance analysis of complex systems is a common problem in embedded real-time systems. Chakraborty et al. proposed a system performance analysis framework to analyze the performance of the event flow in the system [10]. Zhang et al. devised a feasibility analysis framework to analyze the performance of time-sensitive networking [11]. In this paper, we design a performance analysis framework of TT traffic in TTEthernet and use RTC to evaluate the performances of network system.

3. Preliminaries

3.1. Time-Triggered Ethernet

Time-triggered Ethernet can realize deterministic communication by combining the mature fault-tolerance and real-time mechanisms of the time-triggered technology on the basis of standard Ethernet. A globally unified clock is established in the network system to schedule communication between terminals, and the data transmission rate can be increased by up to 1 Gb/s. To support the application of different real-time and security requirements, TTEthernet divides different traffic into three categories: time-triggered (TT) traffic, rate-constrained (RC) traffic and best-effort (BE) traffic. The three data frames adopt the standard format of the Ethernet frame, with the different value in the type field. The value of TT frame's type field is 0x88d7, RC frame's is 0x0888 and BE frame's is 0x0800 [12].

TT messages adopt a global clock synchronization to ensure real-time communication. TT scheduling follows the offline schedule table to ensure that TT messages communicate at a predefined time [13]. It is suitable for communication applications with low jitter and deterministic delay. At the same time, TT message has the highest priority; that is to say, it can get resource prior to other messages. RC messages have lower priority as compared to TT messages and are used for applications with relatively weak deterministic and real-time. RC is a rate-constrained communication, which is determined by the minimum allowed time interval and the maximum allowed frame size between consecutive frames. The minimum allowed time interval between consecutive frames is known as the bandwidth allocation gap (BAG). As different controllers can simultaneously send multiple RC messages to the same receiver, different RC messages tend to queue up in the switching machine, thereby resulting in increased jitter of transmission communication. In comparison to TT messages and RC messages, BE messages are transmitted using the remaining bandwidth of the network. They have the lowest priority and their performance cannot be guaranteed due to delays and reliability issues [14].

Figure 1 shows the internal architecture of a TTEthernet switch. The switch receives traffic from different nodes and classifies traffic, including TT traffic, RC traffic and BE traffic with different priorities [15]. TTEthernet switch can ensure that TT frames are transmitted from an internal buffer at the time specified in the schedule, and a critical traffic is scheduled according to the allocated transmission time. Protocol control frames (PCFs) are TTEthernet frames of a special kind exchanged between TTEthernet components in order to establish and maintain synchronization, and their priority should be higher than

TT traffic, thereby affecting the transmission of TT traffic. In addition, if a low-priority frame starts to be transmitted just before the scheduling time of a TT frame, it will also cause a delay for the critical traffic. The delay caused by low priority traffic is determined by the traffic integration strategy of TT and ET traffic (primarily referring to RC traffic).

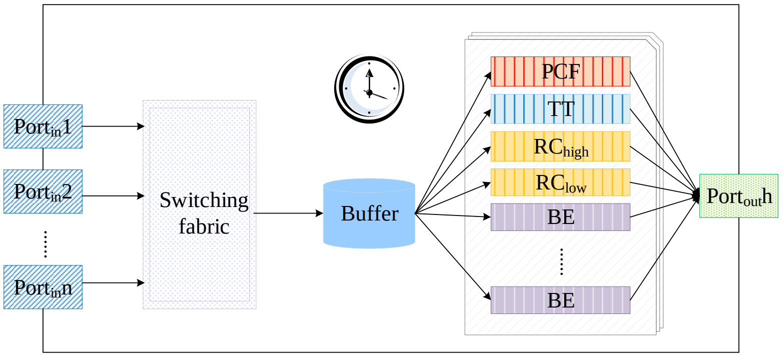

Figure 1. TTEthernet switch architecture.

The integrated transmission modes of TT and RC traffic are divided into three types: preemption, timely-block and shuffling [16], as shown in Figure 2. Preemption indicates a RC frame can be preempted and relayed after the transmission of a TT frame. In timely-block mode, before a RC frame is transmitted, it needs to predict whether there is enough idle time between the next adjacent TT frame for its complete transmission. If not, the RC frame will be delayed for transmission. Shuffling is the mode in which TT frame will be delayed until the transmission for RC is finished.

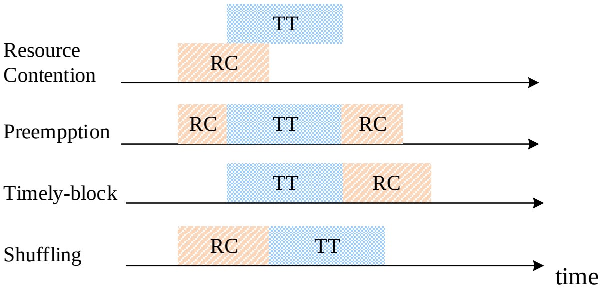

Figure 2. Integration policies of time-triggered (TT) and rate-constrained (RC) traffic.

3.2. Real-Time Calculus

Real-time calculus (RTC) is an extension of network calculus in real-time applications [17]. Network calculus [18] is a mathematical approach to modeling network behavior. RTC relies on the modeling of data flows and resources for processing components with curves called arrival curves and service curves. Arrival curves are functions of relative time to constrain the traffic that can occur in an interval of time. Service curves are used to count available resources. A component can be described with curves; then RTC gives exact delay bounds, backlog bounds and resource utilization for already-modeled components to evaluate network performance. The arrival curve and service curve are defined as follows.

Definition 1 (Arrival Curve)**.** *Given a time interval $[s, t)$, the cumulative function $R(t) \geq 0$ represents the total number of data flows that has arrived in the network component up to time t. Further, assume that the amount of* *data flows arriving within any interval of time is bounded above by a function called the upper arrival curve, denoted by α^u. Similarly, a lower bound on the number of data flows arriving is given by a lower arrival curve α^l. α^l and α^u are related by the following inequality:*

$$\alpha^l(t-s) \leq R(t) - R(s) \leq \alpha^u(t-s), \forall s < t \tag{1}$$

where α^l (0) = α^u (0) = 0.

Definition 2 (Service Curve)**.** *Given a time interval $[s,t)$, C(t) is a cumulative function that represents the available communication resources in the network component up to time t. Similarly to arrival curves, β^u and β^l denote upper and lower service curves of a resource respectively; the following inequality holds:*

$$\beta^l(t-s) \leq C(t) - C(s) \leq \beta^u(t-s), \forall s < t \tag{2}$$

where β^l (0) = β^u (0) = 0.

RTC is based on min-plus algebra and max-plus algebra. In these algebras, convolution and deconvolution are commonly used operations. These operations are defined as follows, where inf means infimum (greatest lower bound) and sup means supremum (least upper bound).

Definition 3 (Min-plus Algebra)**.** *The min-plus convolution $\otimes$ and deconvolution $\oslash$ of f and g are defined as:*

$$(f \otimes g)(t) = \inf_{0 \leq s \leq t}\{f(t-s) + g(s)\} \tag{3}$$

$$(f \oslash g)(t) = \sup_{s \geq 0}\{f(t+s) - g(s)\} \tag{4}$$

The max-plus deconvolution is similar to the min-plus convolution.

Definition 4 (Max-plus Algebra)**.** *The max-plus convolution $\overline{\otimes}$ and deconvolution $\overline{\oslash}$ of f and g are defined as:*

$$(f \overline{\otimes} g)(t) = \sup_{0 \leq u \leq t}\{f(t-u) + g(u)\} \tag{5}$$

$$(f \overline{\oslash} g)(t) = \inf_{u \geq 0}\{f(t+u) - g(u)\} \tag{6}$$

4. Performance Analysis Framework

This section describes the construction of a performance analysis framework for TTEthernet. As shown in Figure 3, a data model, resource model and system model were devised and used to build the abstract model of the system required for the performance analysis of TTEthernet. The data model is mainly described by the arrival curve under the traffic configuration. The resource model is represented by the service curve under different scheduling strategies. The system model was built based on component model and system architecture. For already-modeled abstract model, RTC was applied to analyze network performance and get analysis results.

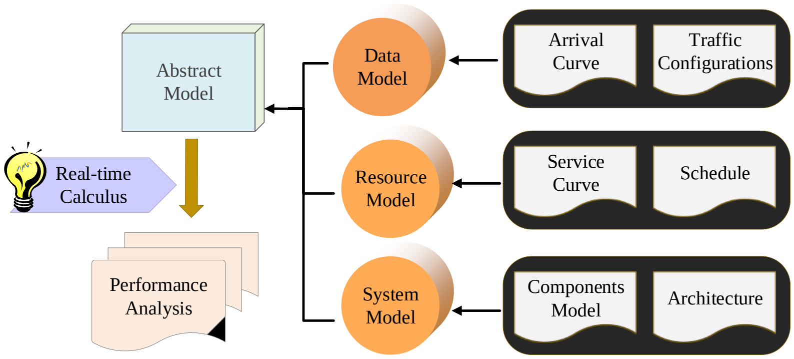

Figure 3. Performance analysis framework of TTEthernet.

4.1. Data Model

The data model is described by the arrival curve, which defines lower and upper bounds on the amount of data flows that can occur in a window of time [19]. Since TT frames can only be served in reserved time slots, the arrival curve of TT flow can be represented by a staircase function. The arrival curve of a single TT flow is given by Equation (7).

$$\alpha_{TT}(t) = \begin{cases} l_{TT} \cdot \left\lceil \frac{t+\triangle}{P_{TT}} \right\rceil, & t > 0 \\ 0, & t \leq 0 \end{cases} \tag{7}$$

where P_{TT} is the periodic of TT flow, l_{TT} represents the frame length and $\triangle$ is the delay variation.

However, multiple TT flows are often transmitted on the same port. Therefore, an aggregate arrival curve of TT flows needs to be processed. As shown in Figure 4, it is assumed that there are three types of TT flows which are transmitted at one output port, with their periods P_{TT_1} = 2 ms, P_{TT_2} = 3 ms and P_{TT_3} = 6 ms. Since all TT flows are scheduled periodically, the aggregate TT flows will be scheduled based on the least common multiple period (P_{LCM}) of these three TT flows. As shown in Figure 4, the P_{LCM} of these three flows is 6 ms.

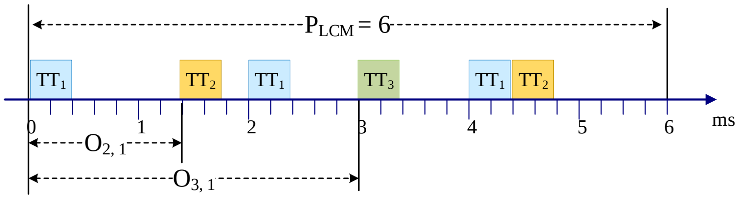

Figure 4. Period of aggregated TT flows.

To obtain the aggregated arrival curves of all TT flows, one TT flow is used as a reference to analyze the arrival curves of other flows. Thereafter, the arrival curves of all TT flows are accumulated to obtain the aggregated arrival curve of the reference one [20]. Assuming that the flow TT_i is used as a reference, the upper aggregated arrival curve $\alpha_{TT}^{i,u}(t)$ and the lower aggregated arrival curve $\alpha_{TT}^{i,l}(t)$ about TT_i are given by Equations (8) and (9).

$$\alpha_{TT}^{i,u}(t) = \sum_{k=1}^{N_{TT}} \alpha_{TT_k}^{u}(t - O_{k,i}) \tag{8}$$

$$\alpha_{TT}^{i,l}(t) = \sum_{k=1}^{N_{TT}} \alpha_{TT_k}^{l}(t - O_{k,i}) \tag{9}$$

where $O_{k,i}$ is the relative offset which defines the interval between the first frame of the reference flow TT_i and the first frame of the flow TT_k. N_{TT} refers to the number of TT traffic.

$\alpha_{TT}^{i,u}(t)$ and $\alpha_{TT}^{i,l}(t)$ can be calculated by the upper arrival curve and the lower arrival curve of a single TT flow given by

$$\alpha_{TT_k}^{u}(t) = \begin{cases} \min\{l_{TT_k} \cdot \left\lceil \frac{t+j}{P_{TT_k}} \right\rceil, \lceil \frac{t}{d} \rceil\}, & t > 0 \\ 0, & t \leq 0 \end{cases} \tag{10}$$

$$\alpha_{TT_k}^{l}(t) = \begin{cases} l_{TT_k} \cdot \left\lceil \frac{t-j}{P_{TT_k}} \right\rceil, & t > 0 \\ 0, & t \leq 0 \end{cases} \tag{11}$$

where j is the jitter of TT flows, d is the minimum interval distance of traffic and d is much smaller than P_{TT_k}.

There are three TT flows in Figure 4, and we can get three different aggregate arrival curves with these TT flows as references, as shown in Figure 5. The blue and gray dotted lines indicate upper arrival curves and lower arrival curves for each of these flows. Finally, the worst-case upper aggregate arrival curve $\alpha_{TT}^{u}(t)$ and the lower aggregate arrival curve $\alpha_{TT}^{l}(t)$ are represented by red and green solid lines, respectively. Their formula is given in Equations (12) and (13).

$$\alpha_{TT}^{u}(t) = \max_{1 \leq i \leq N_{TT}} \{\alpha_{TT}^{i,u}(t)\} \tag{12}$$

$$\alpha_{TT}^{l}(t) = \min_{1 \leq i \leq N_{TT}} \{\alpha_{TT}^{i,l}(t)\} \tag{13}$$

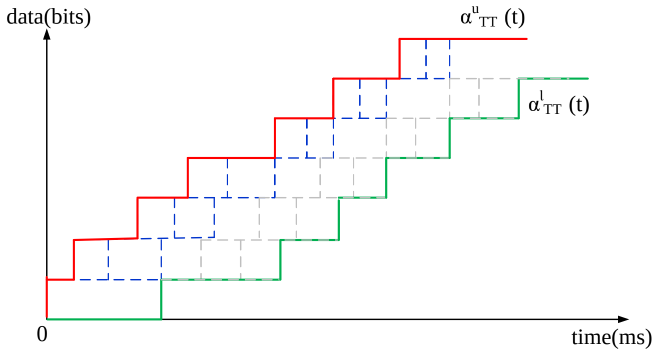

Figure 5. Aggregate arrival curve of TT flows.

4.2. Resource Model

The resource model is described by a service curve, which represents the amount of data that a component can process. Here we suppose that the scheduling strategy is known, to analyze the data processing ability of nodes under this scheduling strategy.

In a time-triggered system, the schedule defines the "receiving point" (the expected time of transmission) of TT traffic. Due to clock drifts and jitter, the receiving point is not exactly known and thus acceptance windows are given [21]. The acceptance windows is a predefined duration interval around the expected "receiving point." If a transmission occurs within the defined acceptance window, the transmission is "scheduled"; otherwise it is "unscheduled" and such unscheduled transmission is considered to be invalid. The size of acceptance window for TT flow is predefined, but under the influence of clock synchronization and traffic integration strategies, the actual open window is smaller than a predetermined window.

As is shown in Figure 6, we give an example to illustrate the influence of clock synchronization and traffic integration strategy on the acceptance windows of TT flows. In this example, it is assumed that P_{LCM} of TT flow is 6ms and consists of three TT flows. W_{TT} represents a predefined acceptance window of TT flow, and W^{*}_{TT} represents an actual open acceptance window after considering the effects of clock synchronization and traffic integration strategies. Suppose that the i-th time slot of TT transmission is used as the reference point for next calculation. $O^{j,i}_{TT}$ represents the time interval between the i-th open window and the j-th open window. I^{i}_{TT} represents the longest waiting time before the i-th open window.

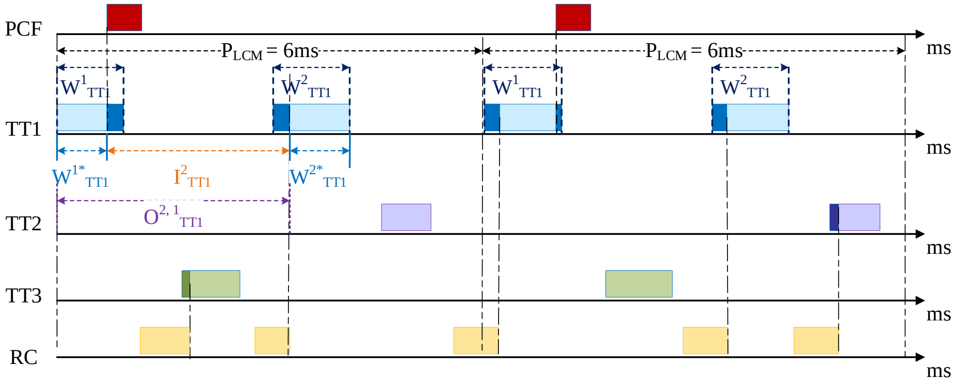

Figure 6. Acceptance windows for TT.

It is vital to ensure clock synchronization for TT frame transmission. If clock synchronization cannot be maintained, the transmission will be congested and blocked, thereby impacting real-time of critical traffic. TTEthernet defines dedicated synchronization messages as the PCFs, whose priority should be higher than that of TT frames, so that the transmission of PCFs will have an impact on acceptance windows of TT frames. There are three types of PCF: cold-start (CS) frame, cold-start acknowledge (CA) frame and integration (IN) frame. The first two are executed to reach an initial synchronization of the local clocks in the system. The latter is executed during normal operation mode to keep the local clocks synchronized to each other and used for components to join an already synchronized system. Figure 6 shows the effect of IN frames on the TT acceptance window. The time slot of the PCFs indicates that clock synchronization will be performed during this period, so PCFs will preempt TT traffic during this period, thereby affecting the size of the TT acceptance window. As shown in Figure 6, the expected acceptance window W^{1}_{TT1} in first P_{LCM} will be reduced under the influence of PCFs. After being preempted by PCFs, the actual open window of TT1 becomes W^{1*}_{TT1}. Suppose $t^{o,i}_{TT_{pcf}}$ and $t^{c,i}_{TT_{pcf}}$ represent the opening and closing times of the TT acceptance window under the influence of PCFs respectively. $t^{o,i}_{TT_{pcf}}$ and $t^{c,i}_{TT_{pcf}}$ are calculated by Equations (14) and (15).

$$t^{o,i}_{TT_{pcf}} = t^{o,i}_{TT} + d^{o,i}_{TT_{pcf}} \tag{14}$$

$$t^{c,i}_{TT_{pcf}} = t^{c,i}_{TT} - d^{c,i}_{TT_{pcf}} \tag{15}$$

where $t^{o,i}_{TT}$ and $t^{c,i}_{TT}$ represent the opening and closing times of the expected TT window respectively. $d^{o,i}_{TT_{pcf}}$ and $d^{c,i}_{TT_{pcf}}$ are open delay and closed delay of TT acceptance window under the influence of PCFs respectively and given by

$$d^{o,i}_{TT_{pcf}} = \begin{cases} \min\{\frac{l^{max}_{pcf}}{C}, t^{c,i}_{PCF} - t^{o,i}_{TT}\}, & t^{o,i}_{TT} < t^{c,i}_{PCF} \\ 0, & t^{o,i}_{TT} \geq t^{c,i}_{PCF} \end{cases} \tag{16}$$

$$d^{c,i}_{TT_{pcf}} = \begin{cases} t^{c,i}_{TT} - t^{o,i}_{PCF}, & t^{c,i}_{TT} > t^{o,i}_{PCF} \\ 0, & t^{c,i}_{TT} \leq t^{o,i}_{PCF} \end{cases} \tag{17}$$

where l^{max}_{pcf} is the maximum length of PCFs and C is the physical link rate. Similarly, $t^{o,i}_{PCF}$ and $t^{c,i}_{PCF}$ are the opening and closing times of PCF transmission.

There exists competition in TTEthernet when a RC frame is already for transmission while encountering a TT acceptance window. Three integration strategies of TT and RC, i.e., preemption, timely-block and shuffling, are illustrated in Figure 2. In preemption and timely-block strategies, RC frame will not cause the delay for TT frame's transmission. Thus, a TT acceptance window has no effect under these two strategies. Under the shuffling integration strategy, if a TT frame is scheduled while an RC frame is already being transmitted, then the TT frame will be delayed until the transmission of RC is finished. Therefore, the transmission of RC frame has an influence on acceptance window of TT in a shuffling strategy. As shown in Figure 6, for the acceptance window W^2_{TT1} in the first P_{LCM}, the actual acceptance window is W^{2*}_{TT1} under the influence of a shuffling integration strategy. Suppose $t^{o,i}_{TT_{rc}}$ represents the opening time of TT acceptance window affected by integration strategies. $t^{o,i}_{TT_{rc}}$ is given by Equation (18).

$$t^{o,i}_{TT_{rc}} = t^{o,i}_{TT} + d^{o,i}_{TT_{rc}} \tag{18}$$

where $d^{o,i}_{TT_{rc}}$ indicates the delay of TT acceptance window under the influence of integration strategy. Note that $d^{o,i}_{TT_{rc}}$ is caused by the shuffling integration strategy and expressed as follows:

$$d^{o,i}_{TT_{rc}} = \begin{cases} \min\{\frac{l^{max}_{RC}}{C}, t^{c,i}_{RC} - t^{o,i}_{TT}\}, & t^{o,i}_{TT} < t^{c,i}_{RC} \\ 0, & t^{o,i}_{TT} \geq t^{c,i}_{RC} \end{cases} \tag{19}$$

where l^{max}_{RC} represents the maximum length of the RC frames and $t^{c,i}_{RC}$ is the end time of RC transmission.

Considering the influence of clock synchronization and integration strategy, the start time $t^{s,i}_{TT}$ and end time $t^{e,i}_{TT}$ of actual acceptance window for TT are as follows:

$$t^{s,i}_{TT} = \max\{t^{o,i}_{TT}, t^{o,i}_{TT_{pcf}}, t^{o,i}_{TT_{rc}}\} \tag{20}$$

$$t^{e,i}_{TT} = \min\{t^{c,i}_{TT}, t^{c,i}_{TT_{pcf}}\} \tag{21}$$

Therefore, an actual acceptance window for TT is

$$W_{TT}^{i*} = \begin{cases} \max\{t_{TT}^{e,i} - t_{TT}^{s,i}, \frac{l_{TT}^{min}}{C}\}, & t_{TT}^{s,i} < t_{TT}^{e,i} \\ 0, & t_{TT}^{s,i} \geq t_{TT}^{e,i} \end{cases} \tag{22}$$

Assuming the i-th acceptance window is used as a baseline, the upper service curve $\beta_{TT}^{i,u}$ and the lower service curve $\beta_{TT}^{i,l}$ for TT traffic in a P_{LCM} are given by

$$\beta_{TT}^{i,u}(t) = \sum_{j=i}^{i+N_{TT}-1} \beta_{TT}^{j,i,u}(t) \tag{23}$$

$$\beta_{TT}^{i,l}(t) = \sum_{j=i}^{i+N_{TT}-1} \beta_{TT}^{j,i,l}(t) \tag{24}$$

where N_{TT} is the amount of TT traffic. $\beta_{TT}^{j,i,u}$ and $\beta_{TT}^{j,i,l}$ are as follows:

$$\beta_{TT}^{j,i,u} = \beta_{P_{LCM},W_{TT}^{j*}}(t + P_{LCM} - W_{TT}^{j*} - O_{TT}^{j,i}) \tag{25}$$

$$\beta_{TT}^{j,i,l} = \beta_{P_{LCM},W_{TT}^{j*}}(t + P_{LCM} - W_{TT}^{j*} - I_{TT}^{i} - O_{TT}^{j,i}) \tag{26}$$

$O_{TT}^{j,i}$ is the time interval between the j-th open window and the i-th open window and can be obtained by the following formula:

$$O_{TT}^{j,i} = (j - i) \cdot P_{TT} + O_{TT}^{j*} - O_{TT}^{i*} \tag{27}$$

where

$$O_{TT}^{i*} = t_{TT}^{s,i} - t_{TT}^{o,i} \tag{28}$$

I_{TT}^{i} represents the longest waiting time before the i-th open window.

$$I_{TT}^{i} = t_{TT}^{s,i} - t_{TT}^{e,i-1} \tag{29}$$

The service curve $\beta_{P_{LCM},W_{TT}^{j*}}(t)$ is calculated by

$$\beta_{T,L}(t) = C \cdot max(\left\lfloor \frac{t}{L} \right\rfloor L, t - \left\lceil \frac{t}{L} \right\rceil (T - L)) \tag{30}$$

By considering the different baselines, the final upper service curve of $\beta_{TT}^{u}(t)$ and the final lower service curve $\beta_{TT}^{l}(t)$ of TT flows are given by Equations (31) and (32).

$$\beta_{TT}^{u}(t) = \max_{1 \leq i \leq N_{TT}} \{\beta_{TT}^{i,u}(t)\} \tag{31}$$

$$\beta_{TT}^{l}(t) = \min_{1 \leq i \leq N_{TT}} \{\beta_{TT}^{i,l}(t)\} \tag{32}$$

Figure 7 shows the worst-case service curve of $TT1$ traffic for the example above.

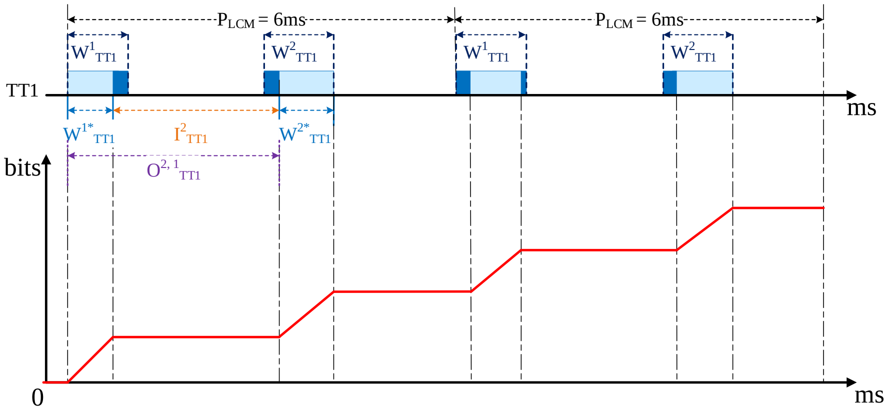

Figure 7. Worst-case service curve of TT1.

4.3. Abstract Model

The abstract model consists of all component resources and data flows of the system. Firstly, to build an abstract model of the system, analysis was undertaken on how a resource processes the flows through it. The arrival curve $\alpha(t)$ of the data model gets processed by the service curve $\beta(t)$ provided by the resource model, thereby generating an outgoing arrival curve $\alpha'(t)$ and a remaining service curve $\beta'(t)$. The outgoing data flows might enter another resource for further processing. The remaining resources might provide services to other data flows through the component. Given a data model which is specified by its arrival curves $\alpha^u(t)$ and $\alpha^l(t)$ and resources which processes these data flows, its processing capability is specified by its service curves $\beta^u(t)$ and $\beta^l(t)$. Let $\alpha^{u'}(t)$ and $\alpha^{l'}(t)$ denote the outgoing upper and lower arrival curves, respectively. Furthermore, the remaining upper and lower service curves of the component are denoted by $\beta^{u'}(t)$ and $\beta^{l'}(t)$, respectively [10]. Then these curves are related by the following expressions:

$$\alpha^{u'} = min\{(\alpha^u \otimes \beta^u) \oslash \beta^l, \beta^u\} \tag{33}$$

$$\alpha^{l'} = min\{(\alpha^l \oslash \beta^u) \otimes \beta^l, \beta^l\} \tag{34}$$

$$\beta^{u'} = (\beta^u - \alpha^l)\overline{\oslash}0 \tag{35}$$

$$\beta^{l'} = (\beta^l - \beta^u)\overline{\otimes}0 \tag{36}$$

In this discussion, we consider two combinations of components as shown in Figures 8 and 9.

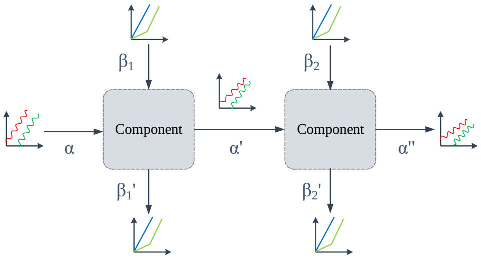

Figure 8. A model consisting of two components.

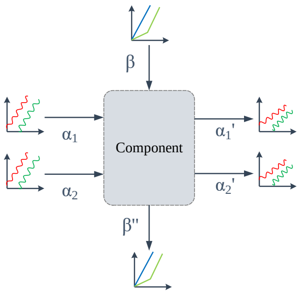

Figure 9. A model consisting of two flows.

In the first case, we consider the impact of a single data flow through different components on the arrival curve. As shown in Figure 8, suppose a TT flow constrained by arrival curve α passes through two components. After the first component processing, the outgoing arrival curve of data flow becomes α'. The data flow continues to enter into the second component with its arrival curve α'. Finally, an outgoing arrival curves α'' is obtained.

In another combination, there are two different flows passing through the same component at the same time. The component provides services to process different flows in order of priority. As shown in Figure 9, the resource model will first serve the high-priority flow α_1 with service curve β. Thereafter, the remaining resource is used to process the low-priority flow α_2. With α_1 and α_2 done, the remaining service curve is known as β''. These remaining resources might be used to process other lower priority flows. In the worst case, the available resources are exhausted when processing α_1, thereby preventing the remaining resources from satisfying α_2, which in turn will block the low-priority flows until the next slot.

Based on a system architecture and the combination of the above two modes, an abstract model of the system can be constructed. The abstract model includes all the necessary information for network performance analysis for the next calculation.

5. Performance Analysis

In the actual analysis network, it is easily to get the arrival curve of TT flows in the initial node according to TT period, size and other information. However, as flows are transmitted, the arrival curve at each node gets changed with delay and jitter. Simultaneously, according to the transmission capacity, the service curve in each node can be obtained. But after processing incoming data flows, transmission resources are consumed, and the service curve changes accordingly. In this paper, we use RTC framework to analyze the change process of arrival curves and service curves. The analyzable performance metrics mainly include delay bounds, backlog bounds and resource utilization. In this section we formalize these notions and state the formulas for deriving these performance metrics.

Delay is an important performance characteristic in computer networks, which reflects to the time required for data flows to be transmitted from source to destination through the network. Typically, network delay consists of processing, queuing, transmission and propagation delays. In this study, delay mainly refers to processing and queuing delay. Suppose that transmission rate is determined; then transmission delay can be calculated by the relative calculating formula. Propagation delay is generally of the order of nanoseconds and negligible in the ideal situation. Delay bound can be expressed by the following inequality:

$$delay \leq \sup_{t \geq 0}\{\inf\{\tau \geq 0 : \alpha^u(t) \leq \beta^l(t+\tau)\}\} \tag{37}$$

Backlog is another performance metric, which is often related to the determination of worst-case buffer fill levels. When high volumes of traffic reach a processing node, if the resources are occupied, the traffic first enters a buffer and waits for scheduling. Thus, the processing node needs to reserve a buffer space; that is, the maximum backlog needs to be considered. The backlog satisfies

$$backlog \leq \sup_{t \geq 0} \{\alpha^u(t) - \beta^l(t)\} \tag{38}$$

As depicted in Figure 10, the maximum delay corresponds to the maximal horizontal distance between α^u and β^l, represented by d_{max}. The backlog is bounded by the vertical deviation between the arrival and service curves, signified by b_{max}.

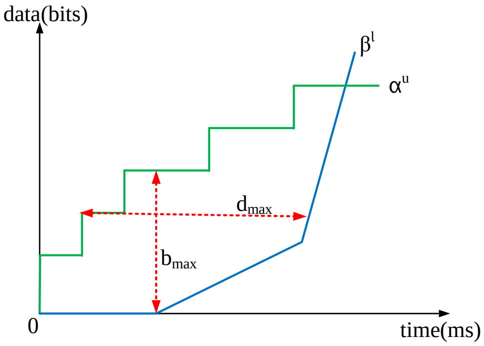

Figure 10. Abstract representation of delay and backlog.

Finally, the resources of each node are limited in TTEthernet. The network system is needed to not only satisfy real-time and reliability requirements for critical traffic, but also improve resource utilization as far as possible. Let β^u and $\beta^{l'}$ be the initial upper service curve and the final lower (remaining) service curve of a resource; then its long term maximum utilization can be given by

$$utilization = \lim_{\Delta \to \infty} \frac{\beta^u(\Delta) - \beta^{l'}(\Delta)}{\beta^u(\Delta)} \tag{39}$$

Generally, non-critical traffic (such as BE) use the remaining bandwidth of the network. For critical traffic, the fewer resources that are overhead while barely impacting the delay, the more resources that can be utilized for non-critical traffic. Thus, resources exploited in an optimal way facilitate higher resource utilization of the whole network system.

Through the calculation of delay bounds, backlog bounds and resource utilization, a comprehensive performance analysis over TTEthernet can be undertaken. Meanwhile, it also can be used to evaluate potential bottlenecks that exist in a network system.

6. Case Study

In this section, a case study is demonstrated to evaluate the feasibility of the proposed performance analysis framework. The TTEthernet system has a topology of two TTE switches and six end systems, running six TT flows and eight RC flows in Figure 11. The transmission rate of this network is 100 Mbps. The details of the TT and RC flows are presented in Table 1. The flow name, route, period (or BAG) and size are presented in columns 1, 2, 3 and 4, respectively. The scheduled time slots for each TT flow are displayed in Table 2.

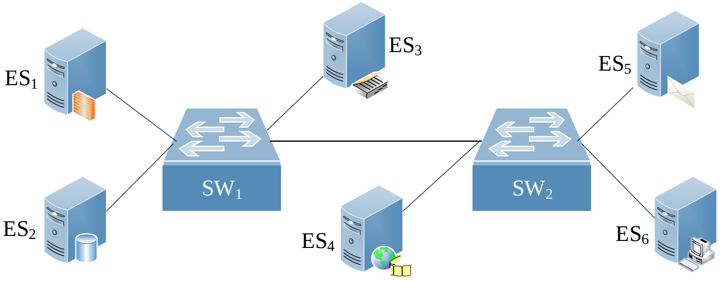

Figure 11. TTEthernet topology.

Table 1. The details of TT and RC flows.

Flow	Route	Period (ms)	Size (B)
TT_1	ES_1-SW_1-SW_2-ES_5	2	880
TT_2	ES_1-SW_1-ES_3	1	730
TT_3	ES_1-SW_1-SW_2-ES_6	2	340
TT_4	ES_2-SW_1-SW_2-ES_6	2	560
TT_5	ES_2-SW_1-ES_3	2	1300
TT_6	ES_4-SW_2-ES_5	2	900
Flow	**Route**	**BAG (ms)**	**Size (B)**
RC_1	ES_1-SW_1-SW_2-ES_6	2	1320
RC_2	ES_2-SW_1-SW_2-ES_5	2	890
RC_3	ES_2-SW_1-ES_3	4	1250
RC_4	ES_4-SW_2-ES_5	2	760
RC_5	ES_2-SW_1-SW_2-ES_6	8	900
RC_6	ES_1-SW_1-SW_2-ES_5	2	1050
RC_7	ES_1-SW_1-SW_2-ES_5	2	1200
RC_8	ES_1-SW_1-ES_3	2	980

Table 2. TT schedule.

Link	VL ID	Time Slot (ms)
$[ES_1, SW_1]$	1	[0.65,0.90]
	2	[0.35,0.60]
	3	[1.05,1.30]
$[ES_2, SW_1]$	4	[1.70,1.95]
	5	[1.05,1.30]
$[SW_1, SW_2]$	1	[1.40,1.65]
	3	[0.10,0.35]
	4	[0.75,1.00]
$[SW_1, ES_3]$	2	[0.45,0.70]
	5	[0.80,1.05]
$[ES_4, SW_2]$	6	[1.35,1.60]
$[SW_2, ES_5]$	1	[0.65,0.90]
	6	[1.00,1.25]
$[SW_2, ES_6]$	3	[0.80,1.05]
	4	[1.70,1.95]

During nonsynchronous operations such as cold start and restart in TTE networks, end systems dispatch PCFs (such as CS frame and CA frame) to TTE switches. Then, TTE switches generate a new PCF and dispatch it to all end systems. The nonsynchronous operation process has no effect on TT flows. During clock synchronization, end systems plan to dispatch IN frames in each integration cycle duration, which might impact on TT transmission. Each PCF frame is a fixed size of 64 bytes. In this case study, it is assumed that six end systems respectively dispatch PCF frames to the TTE switch at the beginning of the integration cycle duration. In this experiment, we adopt a pessimistic approach to calculate the delay under the impact of clock synchronization; that is, PCFs preempt TT traffic in the worst case scenario.

In accordance to the previous analysis, the data model based on the period, rate and other relevant information can be built, so as to get the arrival curve of TT flows. The resource model for each TT flow is built according to the TT schedule, so as to get the service curve. After building the data model and resource model, an abstract model of the system is generated based on the component model and the TTEthernet architecture in Figure 11. In the next part, we mainly discuss the impacts of three types of traffic integration strategies. Under the preemption and timely-block strategies, it is observed that RC flows have no effect on the transmission of TT flows. Therefore, the two strategies can be regarded as one case to analyze.

The worst-case delay of TT traffic can be obtained by the maximum horizontal distance between the modeled arrival curve and the modeled service curve, and concrete results are shown in Figure 12. In contrast to the preemption/timely block and shuffling integration mode, it is not difficult to find that under the shuffling integration strategy, the worst-case delay is higher than that under the other two integration strategies. This is consistent with the previous observations that the transmission of RC flows affect the transmission of TT flows.

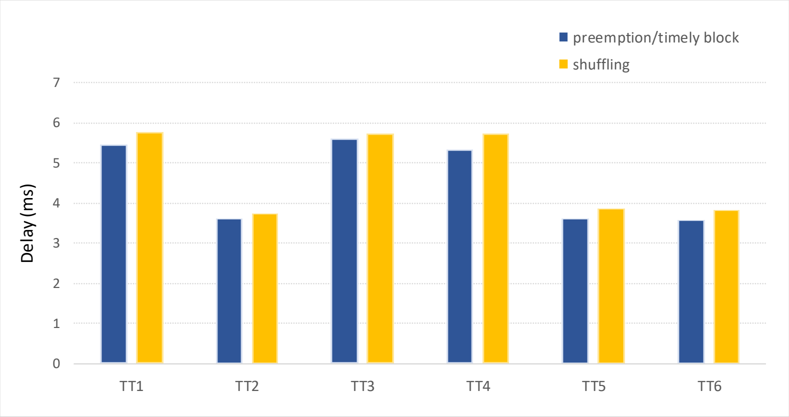

Figure 12. Worst-case delay under different traffic integration strategies.

Through the previous analysis, it became evident that the worst-case backlog of processing node is the maximum vertical distance between the arrival curve and the service curve. Since the arrival curve in this case uses a staircase model, the influence of different integration strategies on the nodes can be ignored. The concrete buffer size in this case study is displayed in Table 3.

Table 3. Worst-case backlog of processing node.

Node	WCB (bits)
ES_1	24,160
ES_2	14,880
SW_1	78,080
SW_2	65,280
ES_4	7200

Finally, we consider the resource utilization of each link. The resource utilization is obtained according to the upper service curve and the final lower (remaining) service curve of the link, and the calculation process is given by Equation (39).

In this case, we take the resource utilization on the ES1–SW1 link as an example to calculate the resource utilization of the transmission of TT1, TT2 and TT3 flows under three integration strategies, and the results are shown in Figure 13. It can be observed that resource utilization under shuffling strategy is higher in comparison to the other two strategies.

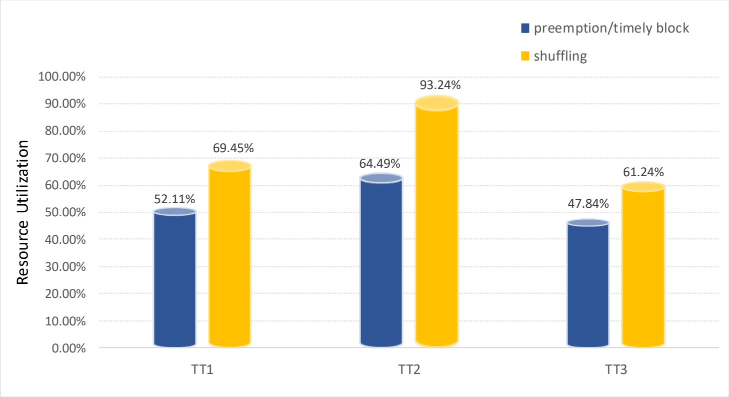

Figure 13. Resource utilization under different traffic integration strategies.

In summary, under the preemption/timely block integration strategy, the worst-case delay of TT traffic is lower, and the system has higher real-time performance. Under the shuffling integration strategy, though the delay of TT flows increases slightly, the delay of RC flows can be guaranteed. From the aspect of resource utilization, the performance under the shuffling strategy gets better. Which integration strategy to choose should be combined with the real-time requirements of the system. The experimental results show that the proposed performance analysis framework is feasible in evaluating the performance of TTEthernet. Moreover, the analysis results can assist in selecting the appropriate scheduling algorithm and traffic integration strategy for the system.

7. Conclusions

In this paper, we proposed a TTEthernet performance analysis framework, and conducted a feasibility analysis for the scheduling of critical traffic in the network. We presented the data model and resource model, and built the abstract model with system architecture and component mode. Then RTC was used to calculate the worst-case end-to-end delay of TT traffic in the network, the cache size of nodes and the utilization of network resources, so as to evaluate the performance of the system. During the modeling process, the influences of clock synchronization and traffic integration strategies on the acceptance window for the critical traffic were analyzed to ensure the reliability of this study's results. Finally, the feasibility of this study's proposed performance analysis framework was verified through a specific case study. According to the experimental results, different integration strategies can be chosen for designing the network architecture which meets the different levels of real-time requirements. This is conducive to saving money and improving the utilization of system resources.

Author Contributions: Conceptualization, Y.H. and J.S.; methodology, X.Y. and Z.C.; resources, J.S. and Y.H.; performance analysis, X.Y.; investigation, X.Y. and Z.C.; writing—original draft preparation, X.Y. and Z.C.; writing—review and editing, X.Y. and Y.H.; supervision, Y.H.; project administration, J.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research is partially supported by Shanghai Municipal Ecnomic and Infomatization Commission Project (2018-GYHLW-02012), the Shanghai Science and Technology Committee Rising-Star Program (number 18QB1402000) and Science and Technology Commission of Shanghai Municipality Project (number 18ZR1411600).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

TTEthernet	time-triggered Ethernet
RTC	real-time calculus
TT	time-triggered
ET	event-triggered
RC	rate-constrained
BE	best-effort
AFDX	Avionics Full-Duplex Switched Ethernet
BAG	bandwidth allocation gap
PCF	protocol control frames
CS	cold-start
CA	cold-start acknowledge
IN	integration

References

1. Steiner, W.; Bauer, G. TTEthernet: Time-triggered services for Ethernet networks. In Proceedings of the 2009 IEEE/AIAA 28th Digital Avionics Systems Conference, Orlando, FL, USA, 23–29 October 2009; pp. 1.B.4–1–1.B.4–1.
2. Steiner, W.; Bauer, G.; Hall, B.; Paulitsch, M. Time-triggered ethernet. In *Time-Triggered Communication;* CRC Press: Boca Raton, FL, USA, 2018; pp. 209–248.
3. Kopetz, H.; Ademaj, A.; Grillinger, P.; Steinhammer, K. The time-triggered ethernet (TTE) design. In Proceedings of the Eighth IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC'05), Seattle, WA, USA, 18–20 May 2005; pp. 22–33.
4. Steiner, W.; Bauer, G.; Hall, B.; Paulitsch, M.; Varadarajan, S. TTEthernet dataflow concept. In Proceedings of the 2009 Eighth IEEE International Symposium on Network Computing and Applications, Cambridge, MA, USA, 9–11 July 2009; pp. 319–322.
5. Suethanuwong, E. Scheduling time-triggered traffic in TTEthernet systems. In Proceedings of 2012 IEEE 17th International Conference on Emerging Technologies & Factory Automation (ETFA 2012), Krakow, Poland, 17–21 September 2012; pp. 1–4.
6. Craciunas, S.S.; Oliver, R.S.; Ecker, V. Optimal static scheduling of real-time tasks on distributed time-triggered networked systems. In Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA), Barcelona, Spain, 16–19 September 2014; pp. 1–8.
7. Zhang, Y.; He, F.; Lu, G.; Xiong, H. An imporosity message scheduling based on modified genetic algorithm for time-triggered Ethernet. *Sci. China Inf. Sci.* **2018**, *61*, 019102.

8. Abuteir, M.; Obermaisser, R. Scheduling of rate-constrained and time-triggered traffic in multi-cluster ttethernet systems. In Proceedings of the 2015 IEEE 13th International Conference on Industrial Informatics (INDIN), Cambridge, UK, 22–24 July 2015; pp. 239–245.
9. Zhao, L.; Pop, P.; Li, Q.; Chen, J.; Xiong, H. Timing analysis of rate-constrained traffic in TTEthernet using network calculus. *Real-Time Syst.* **2017**, *53*, 254–287.
10. Chakraborty, S.; Künzli, S.; Thiele, L. A General Framework for Analysing System Properties in Platform-Based Embedded System Designs. In Proceedings of the conference on Design, Automation and Test in Europe, Munich, Germany, 7 March 2003; Volume 3, pp. 10190–10195.
11. Zhang, P.; Liu, Y.; Shi, J.; Huang, Y.; Zhao, Y. A Feasibility Analysis Framework of Time-Sensitive Networking Using Real-Time Calculus. *IEEE Access* **2019**, *7*, 90069–90081.
12. Gavriluţ, V.; Pop, P. Traffic class assignment for mixed-criticality frames in TTEthernet. *ACM Sigbed Rev.* **2016**, *13*, 31–36.
13. Steiner, W.; Dutertre, B. Automated formal verification of the TTEthernet synchronization quality. In Proceedings of the NASA Formal Methods Symposium, Pasadena, CA, USA, 18–20 April 2011; Springer: Berlin, Germany, 2011; pp. 375–390.
14. Tamas-Selicean, D.; Pop, P.; Steiner, W. Synthesis of communication schedules for TTEthernet-based mixed-criticality systems. In Proceedings of the eighth IEEE/ACM/IFIP International Conference on Hardware/ Software Codesign and System Synthesis, Tampere, Finland, 7–12 October 2012; pp. 473–482.
15. Finzi, A.; Craciunas, S.S. Integration of SMT-based scheduling with RC network calculus analysis in TTEthernet networks. In Proceedings of the 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Zaragoza, Spain, 10–13 September 2019; pp. 192–199.
16. Kermia, O. Schedulability analysis and efficient scheduling of rate constrained messages in the ttethernet protocol. *Softw. Pract. Exp.* **2017**, *47*, 1485–1499.
17. Thiele, L.; Chakraborty, S.; Naedele, M. Real-time calculus for scheduling hard real-time systems. In Proceedings of the 2000 IEEE International Symposium on Circuits and Systems (ISCAS), Geneva, Switzerland, 28–31 May 2000; Volume 4, pp. 101–104.
18. Le Boudec, J.Y.; Thiran, P. *Network Calculus: A theory of Deterministic Queuing Systems for the Internet*; Springer Science & Business Media: Berlin, Germany, 2001; Volume 2050.
19. Le Boudec, J.Y. Application of network calculus to guaranteed service networks. *IEEE Trans. Inf. Theory* **1998**, *44*, 1087–1096.
20. Moy, M.; Altisen, K. Arrival curves for real-time calculus: The causality problem and its solutions. In Proceedings of the International Conference on Tools and Algorithms for the Construction and Analysis of Systems, Paphos, Cyprus, 20–29 March 2010; Springer: Berlin, Germany, 2010; pp. 358–372.
21. Perathoner, S.; Wandeler, E.; Thiele, L.; Hamann, A.; Schliecker, S.; Henia, R.; Racu, R.; Ernst, R.; Harbour, M.G. Influence of different system abstractions on the performance analysis of distributed real-time systems. In Proceedings of the 7th ACM & IEEE International Conference on Embedded Software, Salzburg, Austria, 30 September–3 October 2007; pp. 193–202.