Chinese Journal of Electronics
vol. 33, no. 2, pp. 549–563, March 2024
https://doi.org/10.23919/cje.2022.00.321

RESEARCH ARTICLE

Formal Modeling of Frame Selection in Asynchronous TSN Communications

Ershuai LI, Xuan ZHOU, Jinjing SUN, Huagang XIONG, and Feng HE

School of Electronic and Information Engineering, Beihang University, Beijing 100191, China

Corresponding author: Xuan ZHOU, Email: lomoo@buaa.edu.cn
Manuscript Received September 20, 2022; Accepted January 13, 2023

Abstract — The asynchronous time-sensitive networking (TSN) based on IEEE 802.1Qcr is expected to be a promising solution for the asynchronous transmissions of safety-critical flows without the support of clock synchronization. When the asynchronous traffic shaping (ATS) mechanism is adopted to meet the deadline requirements for transmissions of safety-critical flow, it is necessary to formally verify the real-time properties and corresponding network performance. However, it is still unclear how to build an efficient formal model to evaluate different frame selection methods during the ATS scheduling process, which originate from the dominations of priority or eligibility time. In this paper, we present a formal modeling framework to compare the impacts of different frame selection on transmission sequence under the ATS mechanism. According to the priority level (pATS) or eligibility time (eATS) for flows, two transmission selection methods in ATS are modeled and compared. Then, we verify the real-time properties of ATS. The result shows that the shaping-for-free property can be satisfied with the pATS method but can not be fulfilled with the eATS method. Besides, the timing analysis results illustrate that the eATS method can provide more fairness than the pATS method for the transmission of low-priority flows in TSN networks.

Keywords — Asynchronous communications, Time-sensitive networking, Frame selection, Asynchronous traffic shaping/IEEE 802.1Qcr.

Citation — Ershuai LI, Xuan ZHOU, Jinjing SUN, *et al.*, "Formal Modeling of Frame Selection in Asynchronous TSN Communications," *Chinese Journal of Electronics*, vol. 33, no. 2, pp. 549–563, 2024. doi: 10.23919/cje.2022.00.321.

I. Introduction

Time-sensitive networking (TSN) [1] is gradually considered to be the most promising solution that could provide low-latency and deterministic transmissions over standard Ethernet networks. Many shaping and scheduling mechanisms [2], [3] have been defined in TSN standards to realize applications in safety-critical environments. For instance, TSN is deemed to be a next-generation network technology targeted at industrial automation systems [4], Internet of things (IoT) [5], self-driving cars [6], and aerospace onboard communications [7]. Specifically, TSN provides fault-tolerant and deterministic forwarding services through global time-triggered scheduling, which makes the TSN more suitable for industrial communications with reliability and delay constraints compared with complex fieldbus systems. In addition, TSN can be seen as an efficient implementation of an in-vehicle functional domain network to support the coexistence of safety-critical and best-effort (BE) services simultaneously.

To overcome the scalability problem of networking caused by the complexity of time-triggered scheduling and global clock synchronization in TSN networks [8], Specht and Samii initially proposed an asynchronous traffic shaping (ATS) mechanism [9]. It can achieve deterministic network transmissions and services without global clock synchronization [10] and thus has been developed as an alternative implementation for asynchronous TSN communications [3].

In safety-critical systems, the failures of network functions can easily lead to chaos and disasters [11]. When the ATS is deployed in a safety-critical scenario, it is essential to ensure that ATS can guarantee the real-time performance under an asynchronous communication. Existing research focuses mainly on the real-time analysis of ATS [9], [12]–[16], where analytical methods such as network calculus [17] are used to evaluate the

Associate Editor: Prof. Nan ZHAO, Dalian University of Technology.

upper-bound delay of ATS. However, it is still unclear how to build an efficient formal model to evaluate the detailed frame selection process in ATS. During the implementation of ATS scheduling, we find there are two different frame selection methods. One depends on the priority class, and the another rests with the eligibility time. We name the former the priority level (pATS) method, and the latter the eligibility time (eATS) method. Both the two selection methods are consistent with the ATS mechanism, but they possibly lead to different frame transmission orders and thus affect the real-time performance of networks.

The motivation of this paper is how to model and compare the influences of different frame selection methods on transmission sequence. Previous analytical methods have never been involved in the effects of different frame selection methods on real-time performance and can only perform the overall upper-bound delay evaluation of ATS. As the formal method has a formal language with precise semantics, it can describe the ATS frame selection process and realize the formal analysis of ATS. Then, the ATS mechanism can be modeled and verified for applications to safety-critical systems. The aim of this paper focuses mainly on the verification of the ATS real-time properties using the model-checking tool UPPAAL [18]–[20], such as the starvation-free and shaping-for-free properties. Moreover, we conduct a formal timing analysis that focuses on the worst-case delay results of eATS/pATS, where the delay bound of analytical method is used as a benchmark for reference comparison.

The contributions of this paper are two folds:

1) A formal modeling framework is constructed to clarify the influences of different ATS frame selection methods on transmission sequence and corresponding real-time properties. The verification results show that the shaping-for-free property can be satisfied with the pATS method but can not be fulfilled with the eATS method.

2) The real-time performance of flows belonging to different priority classes is evaluated and compared between the two ATS frame selection methods. The end-to-end (E2E) results illustrate that the eATS method can provide more transmission opportunities than the pATS method for the lower-priority flows in asynchronous TSN networks.

II. Related Work

This section presents existing studies about the modeling of flow shaping mechanism and performance analysis for safety-critical systems, including analytical methods, simulation-based approaches, and model-checking based approaches.

Analytical methods use mathematical tools such as network calculus [17] and queuing theory [21] to analyze the real-time performance of networks. In [9], a trajectory analysis method is firstly presented to evaluate the delay bounds of ATS. The results show that the ATS does not increase the delay bounds of flows queueing with the first input first output (FIFO). The conclusion in [9] is further promoted in [13], which shows that placing a minimal interleaved regulator in any FIFO system will not increase the worst-case delay bounds. In [22], various shaping mechanisms in TSN networks are quantitatively evaluated and compared based on network calculus, which provides a reference for deploying TSN networks properly.

Unlike the analytical methods that can obtain the upper-bound delay results, the intention of simulation is to model the behaviors of shaped flows with a discrete event simulator. The simulation-based approach obtains the statistical delay results of transmission in the network. Then, we can process and analyze the recorded delay results after simulation experiments. For example, the ATS mechanism is modeled and compared with other shaping mechanisms defined in TSN networks, including the Paternoster algorithm [16], the time-aware shaper (TAS) [23], the frame preemption in TSN [24], and the credit-based shaper [25].

Although the above two methods can be applied to evaluate network performance, it is still a challenging issue to realize formal modeling of different frame selection methods in ATS. Analytical methods only derive mathematical delay bounds, but the upper-bound results are greater than the exact worst-case delay results. Therefore, the results obtained by analytical methods are usually pessimistic. For simulation-based approaches, simulation experiments evaluate network transmission performance based on the average delay results. However, it is quite difficult to obtain the worst-case delay results of flow transmissions.

Model-checking has become a well-accepted technique for the verification of safety-critical systems. In [26], Charara *et al*. firstly present and compare the E2E delay results of avionics full duplex switched ethernet (AFDX) networks using the above three methods. Based on the work in [26], an improved flow scheduling model is proposed in [27] to reduce search space with a faster exploration. Regarding the schedulability analysis, an UPPAAL-based framework for schedulability analysis of embedded systems is presented in [28], and Sun *et al.* [29] also develop an efficient analysis technique to analyze the feasibility of real-time tasks based on the timed automata models. Besides, Han *et al.* [30] propose an approach for the schedulability analysis of distributed integrated modular avionics (DIMA). The approach can be applied to multicore DIMA systems which are modeled as a set of stopwatch automata in UPPAAL. In [31], an enhanced consistent stream reservation protocol (CSRP) for resource reservation is formally verified using UPPAAL, which improves the consistency of TSN networks during the resource reservation periods. Also, Lv *et al.* [32] firstly propose a formal analysis framework for the TSN scheduler, which can be used to analyze the synchronous network transmissions based on the TAS mechanism.

This paper focuses more attention on the frame selection of the ATS mechanism. Two frame selection methods of ATS are modeled and compared to discuss the possibility of ATS-based network application in safety-critical fields. Also, we present a formal timing analysis that compares the worst-case delay results of ATS and strict priority (SP) scheduling in asynchronous TSN networks.

III. Background

This section mainly introduces the background of ATS mechanism and model-checking tool UPPAAL.

1. ATS shaping mechanism

ATS is a fundamental shaping mechanism for asynchronous flow transmissions in TSN networks. As specified in IEEE Std 802.1Qcr [3], ATS adopts a per-flow shaping or re-shaping at every hop to comply with the token-bucket model [33] and to provide deterministic transmission guarantees. At each hop, a frame is examined and then released at the earliest time when it is eligible for transmission without violating the token-bucket constraints. In a TSN network node, the functionality of ATS is to regulate the output of flows with a stable transmitting rate while still tolerating some degree of bursts. Correspondingly, two externally configured parameters, committed burst size (CBS) and committed information rate (CIR), define the maximum bursts and transmitting rate, respectively.

The proposal of ATS reduces high implementation costs of re-shaping in switches. The essential idea is the separation between per-flow queuing and per-flow state, which are considered to be different factors during the operation of the ATS mechanism [9]. The flow control based on ATS is regarded as an interleaved shaping in which various flows are enforced to satisfy shaping constraints without requiring per-flow queuing.

Before the introduction of frame selection methods in ATS, it is necessary to describe in detail how to select and output a head-of-line (HOL) frame. Overall, the processing of ATS involves two types of queuing, including a group of shaped queues for interleaved shaping and a set of shared queues for scheduling. As shown in Figure 1, we illustrate an ATS-enable switch with two input ports, and there have two priority classes for flows from different input ports. Firstly, the ATS shaper computes an eligibility time and assigns the time to the arrival frames in the shaped queues. The computation and update of eligibility time mainly depend on three parts, including the arrival time of frames and the parameters of the previous frame and shapers. Then, different flows from the shaped queues are aggregated into the shared queues according to the priority class. Finally, the scheduling mechanism selects and outputs the HOL frame from the shared queues. Note that an ATS node with P priority classes and receiving flows from N input ports requires at least $P \times N$ shaped queues.

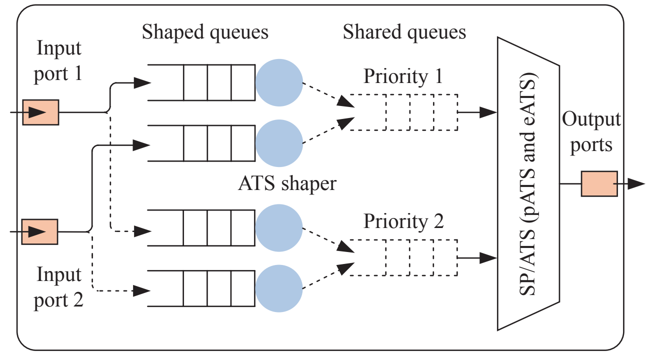


Figure 1 An ATS-enable switch with two input ports and two priority classes.

2. Model-checking tool UPPAAL

UPPAAL is a tool suite developed for modeling, simulations, and verification of systems that can be modeled as networks of timed automata [34], [35]. The tool UPPAAL provides a graphical editor that supports visual representations of timed automata and a model-checker implemented based on constraint-solving techniques.

In UPPAAL, a timed automata model is defined with a template that consists of locations (or states) and edges. To ease the modeling process, UPPAAL has been further extended with additional features. For example, UPPAAL supports a notion of committed location where no delay is allowed. Also, when the template parameters are substituted by given arguments during declaration, the template becomes a process. Besides, expressions are used over edges in timed automata to indicate the location transitions, change the value of variables, or reset clocks, including:

Select A select contains a colon separated expression as `name : type`, where `name` is a variable name and `type` is a defined type.

Guard A guard is a side-effect-free expression to decide whether a location transition occurs or not. The result of guard expression is boolean, and only clocks, integer variables, and constants are referenced.

Synchronization Synchronization is used to represent channel signals (send or receive) over edges during location transitions. The synchronization expression is side-effect-free and only refers to integers, constants, and channels.

Update Update is used to update the variable values after location transition. Hence, the update is a side-effect expression.

Invariant The invariant is used to constrain a state further. Also, it is a side-effect-free expression.

In addition, to verify the requirement specification of real-time systems with the model-checker, a machine-readable query language timed computation tree logic (TCTL) is used in UPPAAL. TCTL is a formally well-defined language, and it consists of path formulas and state formulas. Also, the path formulas can be classified into reachability, safety, and liveness, as illustrated in Figure 2.

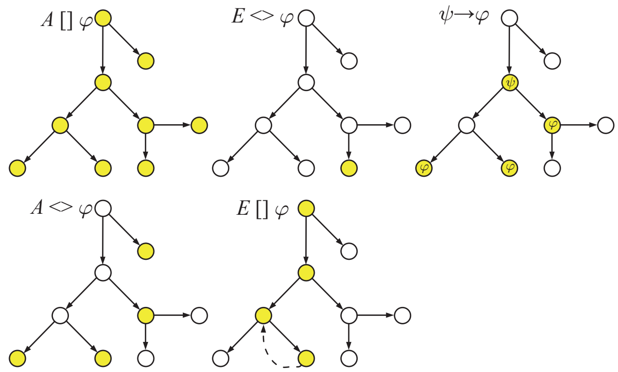


Figure 2 Path formulas supported in UPPAAL [18]. (The yellow locations are states for which a given state formula φ could hold).

The reachability properties answer whether a given state formula, φ, possibly can be satisfied by any reachable state. The second safety property is used to verify that something bad will never happen. Also, an safety property $A[]\ \varphi$ can be expressed as the possible property not $E <>$ not φ. Lastly, the liveness properties mean that something will eventually happen. In UPPAAL, we verify this property using the syntax $A <> \varphi$ and $\psi \rightarrow \varphi$. In brief, the TCTL in UPPAAL supports five types of path formulas, and they can also be reduced to two types, as illustrated in Table 1.

Table 1 Property equivalences in UPPAAL

Name	Property	Equivalent to
Possibly	$E <> \varphi$	–
Invariantly	$A[]\ \varphi$	not $E <>$ not φ
Potentially always	$E[]\ \varphi$	–
Eventually	$A <> \varphi$	not $E[]$ not φ
Leads to	$\psi \rightarrow \varphi$	$A[]\ (\psi$ imply $A <> \varphi)$

IV. Analytical Model of ATS and Problem Description

In this section, the analytical model of ATS related to performance evaluation is firstly introduced in Section IV.1. Specifically, we show the difference of eATS and pATS during ATS scheduling, and the limitation of analytical method for evaluation of these two methods will be demonstrated. Then, the frame selection problem during the implementation of ATS is presented in Section IV.2, which generates different frame transmission sequences and can even affect the real-time performance of flows belonging to different priority classes.

1. Performance evaluation of ATS

For the set of flows F transmitted in the asynchronous TSN networks, there are P priority classes. Also, each flow f can be denoted by a token-bucket model:

$$\alpha_f = r_f \times t + b_f \tag{1}$$

where t is the time variable. Also, r_f and b_f correspond to the average sustainable rate and maximum burstiness, respectively.

Then, the per-hop delay bound $d_{f,e}^{\max}$ of flow f at the hop e is expressed as [9]

$$d_{f,e}^{\max} = \max\left(\frac{\hat{b}_H^{(e)} + \hat{b}_{\rm SP}^{(e)} + l_L^{(e)}}{C_e - \hat{r}_H^{(e)}} + \frac{l_f}{C_e}\right) \tag{2}$$

where $\hat{b}_H^{(e)} = \sum\limits_{f \in F_H} b_f$ and $\hat{r}_H^{(e)} = \sum\limits_{f \in F_H} r_f$ respectively denote the aggregated bursts and flow rate generated by the set of flows F_H with higher priority level than the flow f, the $\hat{b}_{\rm SP}^{(e)} = \sum\limits_{f \in F_{\rm SP}} b_f$ is the burstiness of the set of flows $F_{\rm SP}$ with the same priority level as the flow f, $l_L^{(e)}$ is the maximum frame length for the set of flows F_L with lower priority level than the flow f, l_f is the maximum frame length of flow f, and C_e is bandwidth of physical link at the hop e.

Based on the delay bound in (2), the worst-case E2E delay bound D_f of flow f experienced from its source node s to its destination node d with a path $Path_{s,d}$ is upper-bounded as

$$D_f \leq \sum_{e \in Path_{s,d}} d_{f,e}^{\max} \tag{3}$$

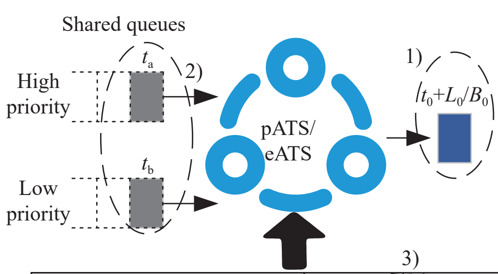
According to the above mathematical analysis, we can obtain the worst-case E2E delay bounds for flows shaped by the ATS in TSN networks. However, formula (3) can only evaluate the upper-bound delay results of ATS. Also, as described in Section IV.2, more details about frame selection cannot be presented by (3) effectively, which would have negative adverse on the performance guarantee based on ATS. For example, it is essential to choose an applicable frame selection method to configure the ATS parameters and buffer size properly. Then the buffer overflows or starvation of services for low-priority flow can be avoided in TSN networks.

Hence, it is necessary to conduct a formal analysis to formalize modeling and discuss the frame selection methods under the ATS mechanism. More details will be given in Section V.

2. Problem of frame selection in ATS

According to the ATS mechanism, the selection and output of HOL frames are determined by the priority classes and eligibility time. Also, a HOL frame has a ready state for transmission when the current time of the scheduler clock is not less than the assigned eligibility time of the HOL frame. Since there are two factors, priority and eligibility time, selecting a ready HOL frame appears to be two different methods. No matter what kind of methods it adopts, it should maintain the frame sequences and consider the real-time requirements of flows. This paper mainly concerns two frame selection methods, which depend on the priority class (pATS) or the latest eligibility time (eATS), respectively.

The main problem about the selection method is shown in Figure 3. 1) Suppose that a HOL frame with frame length L_0 starts to transmit at time t_0. Then the HOL frame would finish its transmission at time $t_0 + L_0/B_0$ under the non-preemption policy, where B_0 is the bandwidth of physical link. 2) As the schedule of frames is work-conserving[*1], the scheduler needs to choose and send the next ready HOL frame from the two shared queues at time $t_0 + L_0/B_0$. Also, as shown in Figure 3, the eligibility time of the next ready HOL frames belonging to the high- and low-priority in the shared queues are t_a and t_b, respectively. 3) When the pATS method is adopted, the scheduler prefers to select the frame with higher priority if both the two frames belonging to different priorities arrive at their ready state with adequate eligibility time. For example, no matter which is bigger for t_a and t_b, the pATS will always select the high HOL frame at time $t_0 + L_0/B_0$. On the other hand, when the eATS method is adopted, the select results are closely related to the eligibility time of HOL frames. Still taking the situation shown in Figure 3 as the example, if the value of t_b is less than t_a, the scheduler would select the low-priority HOL frame since its transmission requirement looks more urgent than the high-priority HOL frame. Under this condition, we can find the selecting reuslts of pATS and eATS are different.



Conditions	pATS	eATS
$t_a \le t_b \le t_0+L_0/B_0$	High	High
$t_b < t_a \le t_0+L_0/B_0$	High	Low

Figure 3 Two frame selection methods under the ATS mechanism.

Based on the above explanation for frame selection, it can be seen that the two frame selection methods possibly generate various frame transmission sequences, which could affect the real-time performance of flows belonging to different priority classes.

V. The Modeling Framework

As shown in Figure 4, a workflow of formal analysis is presented to model the ATS mechanism and then compares the frame selection methods formally. Firstly, an overall model framework of the ATS mechanism should be established to achieve a complete frame transmission process according to ATS. The ATS mechanism is formalized as timed automata models with the graphical user interface in UPPAAL. Also, the real-time networking requirements are described by a subset of the TCTL queries. Afterward, the templates of timed automata are instantiated and used as processes to construct a whole system. Finally, the model-checker of UPPAAL is used to verify whether the real-time properties of ATS possibly can be satisfied in the system.

1. Overview of model

Figure 4 presents an overall model framework of the ATS mechanism. According to the selection and output of frames in ATS, the framework mainly includes flow generation, frame processing, and transmission scheduling models.

```
system flow_model,
        frame_processing_model,
        transmission_scheduling_model;
```

The flow generation model generates flows belonging to different priority classes in a two-port ATS-enabled switch, i.e., the high- and low-priority. The frame processing model computes the eligibility time and assigns the time to the arrival frames according to the ATS mechanism. The computation and assignments of eligibility time are determined by the arrival time of frames and the parameters of the previous frame and shapers. The transmission scheduling model selects and outputs the HOL from queues belonging to different priority classes. Also, the scheduling and frame selection methods are defined in the transmission scheduling model, including the SP and ATS scheduling.

Besides, the communications between different models are closely related to the enqueue and dequeue of frames under the ATS transmissions. Hence, the clock variables and channel signals are defined to process the forwarding of frames from different queues. For example, the global clock `t_global` records the arrival time of frames and achieves the consistency of time among different flow generation instantiations. Also, a scheduler clock `t_sch` is defined in the transmission scheduling model, which runs at the same rate as the global clock `t_global`. Lastly, we define a four-channel signal `tran[Qnum]` to distinguish frames transmitted from various queues.

2. Flow generation model

UPPAAL supports complex user-defined types such as data structure, which allows to define the abstract of `Frame` and `Queue` in the flow generation model.

Definition of Frame a frame could be represented as a data structure `Frame` in UPPAAL, and the detail of it is given as follows:

```
typedef struct{
    int framePriority;
    int flowID;
```

[*1] A scheduler is work-conserving if and only if it never idles time slots when there exists at least one frame awaiting transmission in the queues.

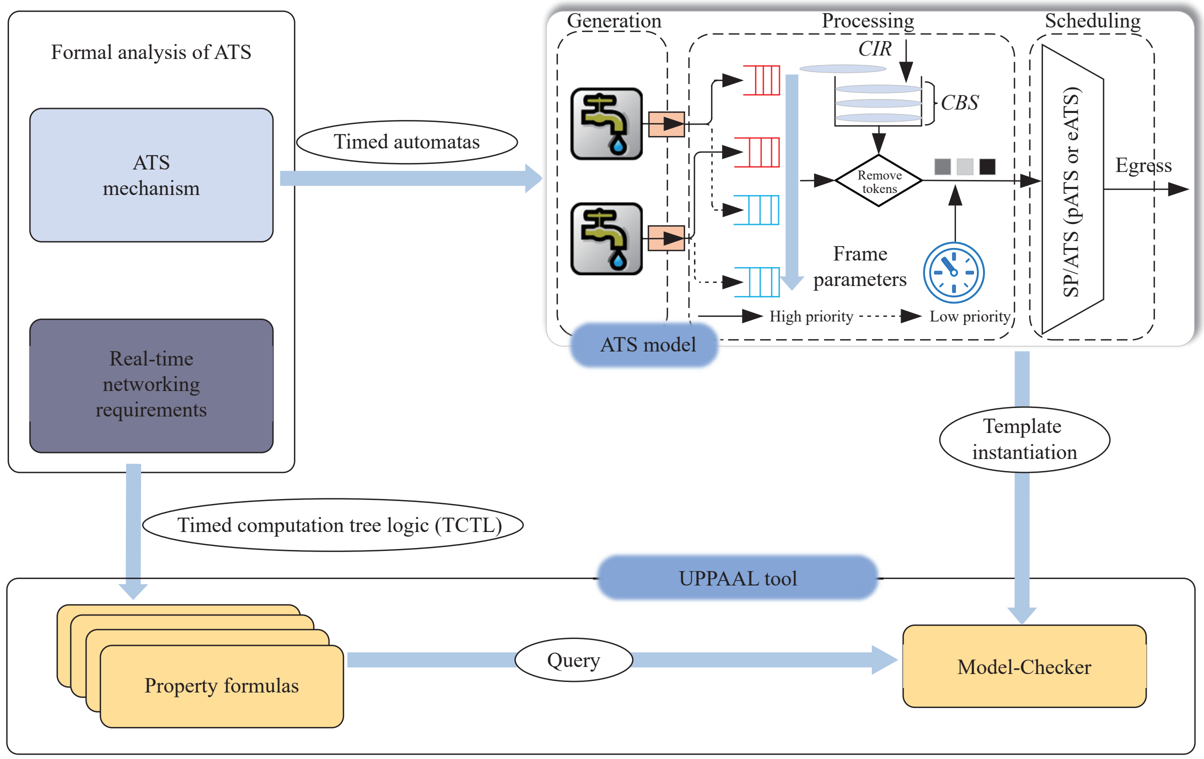


Figure 4 Workflow to model the ATS and compare the frame selection methods.

```
        int frameLength;
        int timeStamp;
        int period;
        int queueID;
        int timeEligible;
    }Frame;
```

1) `framePriority` defines the priority of a frame, such as the high or low priority;

2) `flowID` is used to identify the ID of flow;

3) `frameLength` means the frame length, which adopts the transmission time to represent the corresponding frame length. Typically, it can be obtained by the ratio of the frame length to the bandwidth of physical link;

4) `timeStamp` records the time when a frame is generated in the flow generation model;

5) `period` represents the period time to generate a new frame for flows;

6) `queueID` is used to identify the ID of the queue that stores frames during the shaping process;

7) `timeEligible` records the time when a frame is eligible for transmitting according to the ATS mechanism.

Definition of queue a queue is defined as a 4-tuple structure `Queue` which contains an array of frames `Frames[n]` with depth `n`. Then,

```
    typedef struct{
        int queuePriority;
        int front;
        int rear;
        Frame frames[n];
    }Queue;
```

1) `queuePriority` defines the priority of queue;

2) `front` and `rear` point to the first and last element of queue, respectively;

3) `frames[n]` represents an array of frames with a maximum buffering space `n`.

The flow generation model is the source of flows that creates frames periodically. Then the generated frames will be stored in queues and waiting to be shaped based on the ATS mechanism.

Figure 5 depicts the template of flow generation model with four locations, including the initial `idle`, `ready`, committed `sent` and committed `waiting`. The initial location `idle` represents the beginning of frame generation. When the guard expression, the global clock `t_global` equals to the initialization time `initTime`, is satisfied, the model will move to the location `ready`. In the location `ready`, the model generates a frame by an update expression, and then moves to the next location `sent` in one period of time according to an invariant expression.

In the committed location `sent`, the newly generated frame will be sent into the shaped queue, then the model will change into location `waiting`. In the `waiting` location, the frame waits for the shaping process according to ATS shaping mechanism. Also, a synchronization channel signal `tran[QueueID]!` is used during the waiting time to indicate the ID of the shaped queue which has stored the newly generated frame. Lastly, the model returns to the location `ready` to repeatedly generate the next frame.

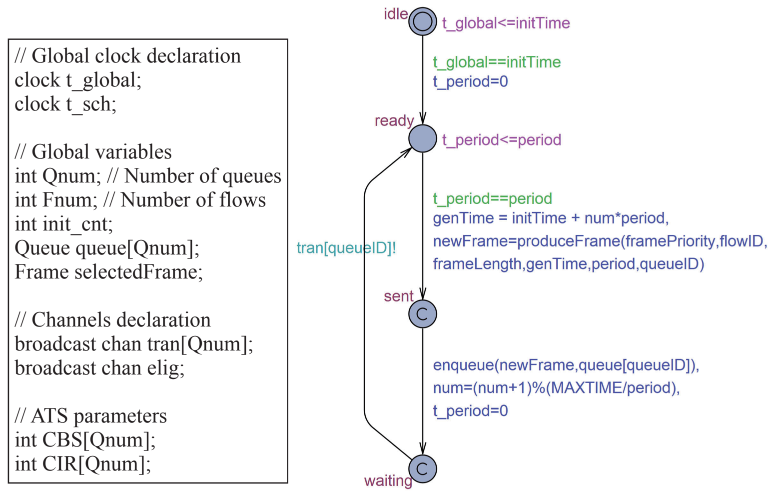


Figure 5 Flow generation model. (MAXTIME is the maximum bound of the variable `timeStamp`. For example, the value of MAXTIME is 20000 in this model).

Besides, the variables and functions defined in the flow generation model as shown in Figure 5, including:

1) `t_period` is a local clock to measure the duration time occupied by the frame transmission;

2) `genTime` is a local variable that records the time when a frame is generated, and then its value will be assigned to the timestamp of frame;

3) `produceFrame`() is a function to generate frames;

4) `enqueue`() is a function that inserts a new frame at the end of queue.

3. Frame processing model

We first explain how to compute the eligibility time for the arrival frames based on IEEE Std 802.1Qcr [3]. Then, the procedures of the frame processing model will be demonstrated in detail.

The computation of the eligibility time for a frame consists of three steps: Firstly, the time variables associated with shaping, such as the eligibility time of shaper and the bucket full time, are updated according to the arrived frame and the parameters of ATS shaper. Then, the eligibility time of frame is the maximum value of the frame arrival time and the eligibility time of shaper. Lastly, the bucket empty time is updated for the next possible frame. In this model, the computation of eligibility time for frames is defined in the function `frameProcess`() as presented in Figure 6.

As shown in Figure 6, the frame processing model is simply formalized as a timed automata template based on the frame process defined in ATS. When a synchronization channel signal `tran`[`QueueID`]? is received in the processing model, it moves from the location `initial` to the committed location `eligible`. During this period, the function `frameProcess`() is invoked to compute the eligibility time for the arrival frame. Then, the model returns to the initial location and sends a channel signal `elig`! to indicate that frame is ready to be scheduled according to different frame select functions. Note that there are four shaping queues in this paper. Hence, the

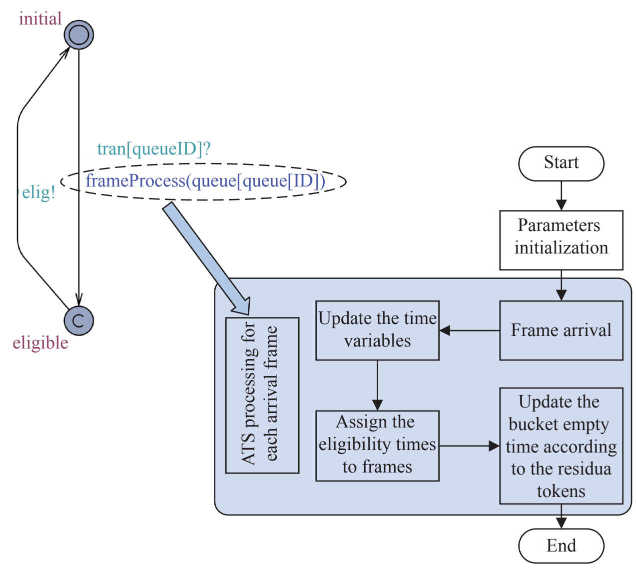


Figure 6 Frame processing model.

frame processing model should be instantiated into four different subprocesses.

4. Transmission scheduling model

As the pATS and eATS methods may lead to different frame transmission sequences, they are defined as two different methods in the transmission scheduling model. Meanwhile, we build a formal model that only uses the SP scheduling mechanism as the benchmark for delay reference. The difference between the SP and pATS is that the SP does not consider the ready state of the HOL frame as there is no eligibility time concept.

1) Model of ATS scheduling

For the frame transmission in ATS scheduling model, it is necessary to ensure that the HOL frame has a ready state. Then, a HOL frame waiting for sending is selected according to different frame selection methods. After the delivery of the selected HOL frame, a new HOL frame will be selected again for the subsequent dispatch.

We use Algorithm 1 to find a HOL frame that has the latest eligibility time or belongs to the highest priority class. The HOL frames $\{f_1, \dots, f_q, \dots, f_{\text{Qnum}}\}$ of shaped queues are given as the input parameters. Also, frame variables $\{fr_1, \dots, fr_p, \dots, fr_P\}$ are used in Algorithm 1 to record the latest eligibility time of p priority queues, respectively. Overall, our algorithm has the following three steps:

– Initialize the eligibility time of $\{fr_1, \dots, fr_p, \dots, fr_P\}$.

– Classify HOL frames belonging to priority p and search for their latest eligibility time, respectively.

– Select HOL frame according to the pATS or eATS method.

Algorithm 1 Frame selection method with pATS or eATS

Require: $\{f_1, \dots, f_q, \dots, f_{\text{Qnum}}\}$: the HOL frames of shaped queues with P priorities.

Ensure: selected frame f_m.

// Initialize the eligibility time of priority p
for $p = 1$ to P do
  fr_p.timeEligible $= 0$;
end for
// Classify the HOL frames belonging to priority p and
// search for the latest eligibility time
for $q = 1$ to Qnum do
  if f_q belongs to priority p then
    if f_q.timeEligible $>$ fr_p.timeEligible then
      fr_p.timeEligible $=$ f_q.timeEligible;
    end if
  end if
end for
// Select frame according to the pATS or eATS method
if ATS == eATS then
  for $p = 1$ to P do
    if fr_p has the latest eligibility time then
      $f_m = fr_p$;
    end if
  end for
else if ATS == pATS then
  for $p = 1$ to P do
    if fr_p belongs to the highest priority class then
      $f_m = fr_p$;
    end if
  end for
end if

Figure 7 demonstrates a timed automata model of the ATS scheduling mechanism, and there are four locations in the model, including `initial`, `ready`, `startTrans` and `transmitted`.

In the `initial` location, the model would change its location to the `ready` if a synchronization channel signal `elig`? is received. During this period, the Algorithm 1 is

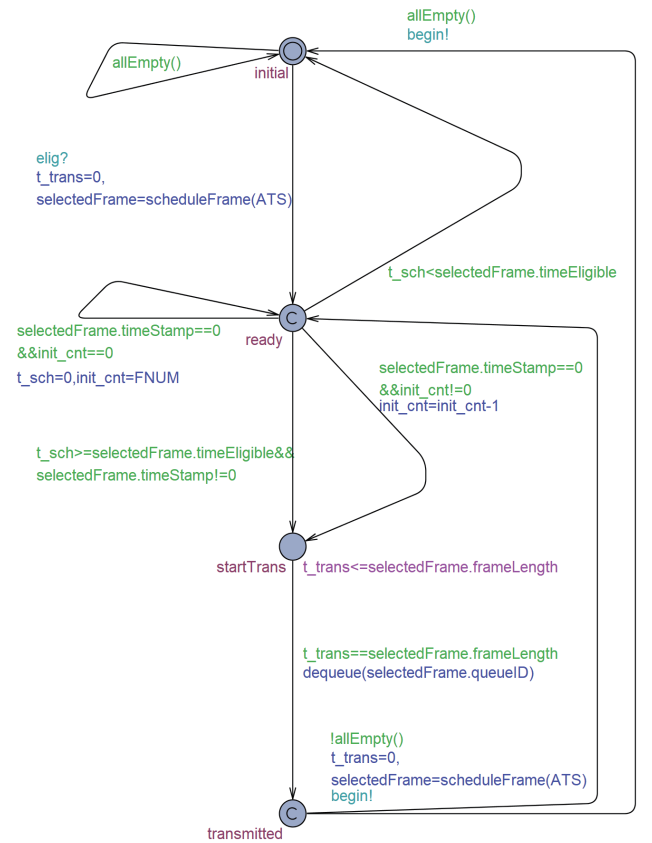


Figure 7 ATS scheduling model.

used to select a HOL frame. Based on Algorithm 1, the function `selectFrame(ATS)` is defined as an update expression in the scheduling model. Mostly, the model would move to the `startTrans` location from the `ready` location when the current time of the scheduler clock `t_sch` is not less than the eligibility time of the selected HOL frame. However, some exceptional cases due to the bounded range of variables also need to be considered. In UPPAAL, the integer variables have a bounded range. For example, the `timeStamp` value of frames should be with the range [0, MAXTIME]. As shown in Figure 5, when frames are continuously generated, the value of `timeStamp` would be in a periodic cycle. Correspondingly, the eligibility time of frames calculated according to the `timeStamp` would also have a bounded range. Therefore, the scheduler clock `t_sch` needs to be reset to 0 when all of the `timeStamp` for the shaped HOL frames are zero in a new MAXTIME generation cycle. In this paper, we define an integer variable `init_cnt` to count the number of frames whose eligibility time is equal to zero, and the initial value of `init_cnt` is `Fnum`.

Then, the model stays in the location `startTrans` until the transmission of frame is completed. Simultaneously, the function `dequeue`() is used to output the selected HOL frame when the model goes to the location `transmitted`.

Lastly, the model would go to the location `ready` if there still exists a nonempty queue, and the next HOL frame that can be transmitted is selected by the function selectFrame(ATS). Otherwise, the model would return to the location initial waiting for the next possible transmission opportunity.

Besides, the necessary variables and functions of the ATS scheduling model are defined as follows:

1) t_trans is a local clock to measure the duration time occupied by frame transmission in the location startTrans;

2) init_cnt records the number of frames with eligibility time equal to 0 during frame transmissions;

3) selectFrame(ATS) is a function that chooses a HOL frame according to different frame selection methods defined in Algorithm 1. We use pATS or eATS as different input parameters in the function selectFrame(ATS).

4) selectedFrame represents the HOL frame that is ready to be transmitted;

5) dequeue() is a function that outputs frame at the head of queue.

2) Model of SP scheduling

Figure 8 illustrates the timed automata model of the SP scheduling with four locations. Compared with the ATS scheduling model, the main difference of the SP model is that it does not consider the eligibility time of frame during the frame selection process, and it is unnecessary to have a ready HOL frame to transmit. Thus, the output of frames in the SP model is only determined by the priority class.

Besides, it is straightforward to recognize that the states of the SP model are almost the same as those in the ATS model. The main difference between the two models is that the latter uses a SP scheduling mechanism to select a frame if there exist nonempty shaped queues. Correspondingly, there is no need to compare the scheduler clock with the eligibility time of frame during frame transmission. Lastly, the exceptional cases related to the bounded variable ranges in UPPAAL also need to be considered. Hence, we use the integer variable init_cnt to count the number of frames whose timestamp is equal to zero. Also, the initial value of init_cnt is Fnum in the SP scheduling model.

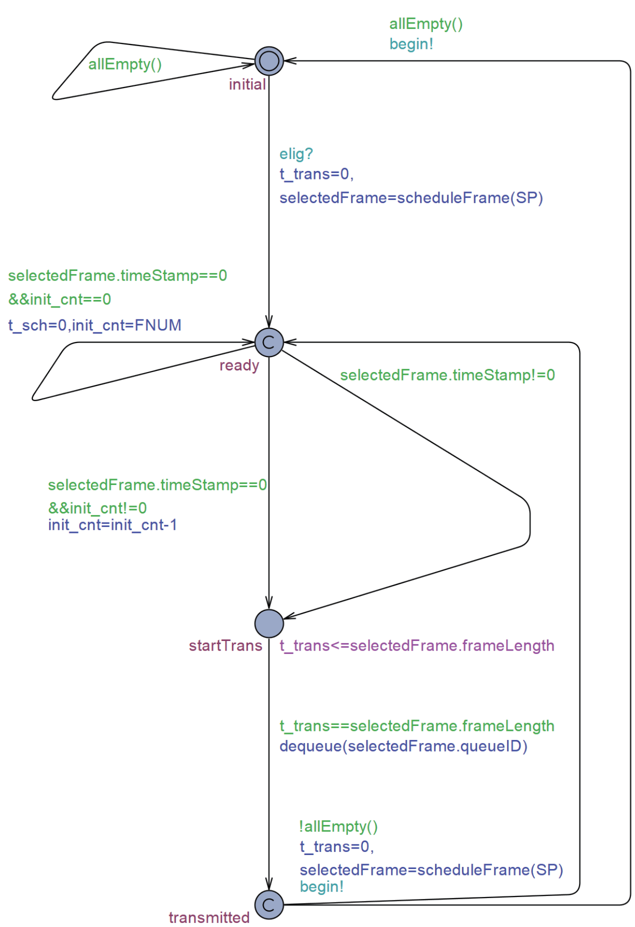


Figure 8 SP scheduling model.

5. Test automata model

In UPPAAL, a test automata can be built based on the delay-bounded feature of transmission to realize the timing analysis of ATS. The test automata template usually contains channel signals and integer variables and defines a timeout location that indicates the worst-case delay result for frame transmission. Then, the test of worst-case delay is abstracted into a reachability property. By verifying whether the timeout location is reachable, the worst-case delay results of transmission can be obtained. In this paper, the delay results of the SP scheduling mechanism are used as the delay reference. If the delays of ATS are equal to the results under the SP, it can be concluded that the shaping-for-free property can be satisfied with the ATS mechanism.

As shown in Figure 9, we define a template for the delay test of ATS, which is necessary for the timing analysis of ATS and the verification of the shaping-for-free property. Firstly, a synchronization expression begin! is used in the ATS/SP scheduling model when frame transmission has just been finished. At this moment, the test automata model will move to the location Testing when a channel signal begin? is received. Then, in the committed location Testing, the test automata would change into the location Timeout if a transmitted frame suffers the worst-case delay. Otherwise, it needs to continue testing on the next frame transmission. Besides, the integer variables PRI and maxD are defined in the model to indicate the priority class and record the worst-case delay result, respectively.

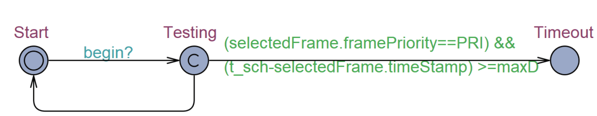


Figure 9 Test automata model for delay result.

The verification case that whether the test automaton model can be in the location Timeout is a query for reachability property. We use the model-checker tool in UPPAAL to perform a reachability check for a set of `maxD` values. If the query result of reachability changes from satisfied to unsatisfied when the value of `maxD` is increased by one unit, the current value of `maxD` is just the worst-case delay of the flow with the `PRI` priority level.

VI. Experiment and Formal Analysis

As shown in Section V, the modeling framework of ATS is presented using UPPAAL. Afterwards, it is feasible to verify and compare the two different frame selection methods of ATS. In this section, we conduct several experiments to compare the two different frame selection methods regarding real-time properties and analyze their real-time performance. The experiments are performed on the UPPAAL 4.1.24 and a laptop computer with an Intel processor.

1. Experimental configuration

As shown in Table 2, there are four shaped queues and eight flows categorized into the high- and low-priority classes. The bandwidth of physical link and depth of queues are equal to 100 Mbps and 20, respectively. Also, the *CBS* and *CIR* of the ATS shaper for the high- and low-priority classes are given in the second and third columns of Table 2, respectively. According to the above configuration for the high-priority flows, it can be calculated that the transmission time for a high-priority frame is $600 \times 8 / 100 = 48$ μs, and the rate of a high-priority flow is $600 \times 8 / 1000 = 4.8$ Mbps. Similarly, the transmission time for a low-priority frame is 40 μs, and the rate of a low-priority flow is 4 Mbps.

Table 2 Parameters of configuration

Priority	Queues			Flows		
	CBS (B)	*CIR* (Mbps)	Qnum	*L* (B)	*P* (μs)	Fnum
High	1000	32	2	600	1000	4
Low	800	16	2	500	1000	4

2. Formal verification

In the following verification cases, we use the network configurations listed in Table 2 to verify the ATS model formally. The real-time properties of ATS are modeled according to the networking specification requirements, and the query formulas are given in Table 3. The first query is used to verify the safety property guaranteeing that the system is deadlock-free. The second and fourth queries are used to verify the possible reachability properties of the ATS model. The third query is to ensure that the scheduling mechanism can complete the frame transmission eventually. The last query is defined as a safety property to verify that the shaping mechanism in ATS would not increase the extra delay or worsen flow transmission. The worst-case delay results can be obtained by this verification case with the manually configured variable maxD. Then, the delay results are compared to the delays in the SP scheduling mechanism. If the delay results in ATS are exactly the same as those of SP, it can be concluded that the shaping-for-free property is satisfied with ATS.

Table 3 Query formulas of ATS

Property	Descriptions	Query formulas
P1	Deadlock-free	A[] not deadlock
P2	Starvation-free	E <> selectedFrame.queueID == ID, ID ∈ {0, 1, 2, 3}
P3	Liveness-of-forwarding	schATS.ready → schATS.transmitted
P4	Limited-depth	E <> (queue[ID].rear-queue[ID].front) >= maxQ, ID ∈ {0, 1, 2, 3}
P5	Shaping-for-free	A[] (selectedFrame.framePriority == PRI) imply (t_sch-selectedFrame.timeStamp) <= maxD, PRI ∈ {High, Low}

1) Deadlock-free: The system should never be in a deadlock situation. Also, the deadlock-free property always has outgoing transitions to its delayed locations. We verify this property with the safety formula and keyword *not deadlock*, and it is satisfied in our model.

2) Starvation-free: All of the queues should have the opportunity for transmissions eventually. If the transmission opportunities are overly occupied by one queue during the scheduling period, the others will be starved without adequate bandwidth resources. The reachability formula checks whether the selected frame can belong to any one of the four shaped queues. The verification results show that the starvation-free property can be satisfied in the model.

3) Liveness-of-forwarding: The process of frame transmission would be completed eventually. This property ensures that the selected frame based on the ATS mechanism can finish transmission eventually. The verification effort is conducted with the liveness formula $\psi \rightarrow \varphi$, and the results show that any frame that has been selected could be forwarded eventually.

4) Limited-depth: The safety-critical system should use limited buffer size and guarantee that the queues will never be overflow. We focus on the possibly maximum amount of frames maxQ in various queues, and the maxQ should never exceed the configured queue depth. The amount of frames can be obtained by the subtraction operation between the value of rear and front for queues. We use a reachability formula to verify this property, and the maxQ is manually configured under

different cases. The results show that the maxQ will never exceed the configured queue depth. Then the limited-depth property can be satisfied in the model.

5) Shaping-for-free: Existing research [13] indicates that the delay bounds can be maintained for shaped flows regardless of whether a minimal interleaved regulator is included in an arbitrary FIFO system. The delay results of frame transmissions can be obtained by the subtraction operation between the scheduler clock time when finishing frame transmission and the timeStamp of the selected frame. We use a safety formula to check the worst-case delay results for frame transmission. Also, the maxD is the worst-case delay of the flow with the `PRI` priority class.

We formally verify the shaping-for-free property for flows belonging to different priority classes. The worst-case delay results show that the results under the pATS method are equal to the results under the SP scheduling mechanism. Therefore, the shaping-for-free property can be satisfied with the pATS method. Specifically, the maximum delay results for the high- and low-priority flows with pATS/SP are 232 μs (48 μs × 4 + 40 μs) and 352 μs (48 μs × 4 + 40 μs × 4), respectively. However, the verification results demonstrate that the maximum delay results for the high- and low-priority flows with the eATS method are 272 μs (48 μs × 4 + 40 μs × 2) and 352 μs respectively, indicating that the shaping-for-free property can not be satisfied with the eATS method. In brief, the shaping-for-free property can be satisfied with the pATS method but not with the eATS method.

Besides, the verification times and the memory usages of UPPAAL are given in Table 4 to show the complexity of the ATS model. As it can be seen in Table 4, the verification of reachability properties (e.g., the P2 and P4) usually takes less time with a lower resource consumption. On the other hand, longer verification time with an increasing memory consumption is required for the verification of the safety properties (e.g., the P1). The second important result observed from Table 4 is that more verification time and memory usage are required to verify the shaping-for-free property of ATS compared with the SP. We suppose the most possible reason is that ATS needs to compute and select a output frame based on the eligibility time in the verification case, which increases the complexity of model.

3. Timing analysis of eATS and pATS

The verification case of shaping-for-free shows the different impacts of eATS and pATS methods on the delay results for the high- and low-priority classes. However, the delay results of verification are only based on a single-node test, and more efforts are needed to compare the two frame selection methods for timing analysis. Hence, we first analyze the effects of ATS parameters, *CIR* and *CBS*, on real-time performance. By comparing the worst-case delay results of the two frame selection methods under different ATS parameter configurations, it can provide network designers with suitable networking schemes chosen for application scenarios. Also, it is necessary to confirm whether eATS and pATS can maintain consistent real-time guarantee effects under multiple priority class scenarios. Therefore, we implement a case study where three priority classes are configured including high-, medium-, and low-priority classes to illustrate the delay results of multi-priority classes for flows under the two frame selection methods. Besides, an E2E network scenario is used to compare the worst-case delay results of eATS/pATS among different network performance analysis methods. Finally, based on the above experimental results, we discuss and summarize the ad-

Table 4 Verification results

Property number	Verification results	Verification/Kernel/Elapsed time	Peak resident/Virtual memory usage
P1	Yes	9.812 s/0.282 s/10.105 s	892,356 KB/1,784,636 KB
P2 (ID=0)	Yes	0.063 s/0.062 s/0.134 s	22,104 KB/57,752 KB
P2 (ID=1)	Yes	0 s/0 s/0.005 s	22,364 KB/58,092 KB
P2 (ID=2)	Yes	0 s/0 s/0.001 s	22,368 KB/58,096 KB
P2 (ID=3)	Yes	0 s/0 s/0.001 s	22,372 KB/58,100 KB
P3	Yes	17.219 s/0.343 s/17.564 s	913,080 KB/1,837,156 KB
P4 (ID=0)	Yes	3.125 s/0 s/3.134 s	885,332 KB/1,768,816 KB
P4 (ID=1)	Yes	3.141 s/0 s/3.128 s	885,340 KB/1,768,824 KB
P4 (ID=2)	Yes	3.094 s/0 s/3.1 s	885,340 KB/1,768,824 KB
P4 (ID=3)	Yes	3.094 s/0 s/3.096 s	885,340 KB/1,768,824 KB
P5 (eATS, PRI=High)	No	5.687 s/0.125 s/5.805 s	892,860 KB/1,768,204 KB
P5 (eATS, PRI=Low)	Yes	5.688 s/0.141 s/5.83 s	892,852 KB/1,786,088 KB
P5 (pATS, PRI=High)	Yes	5.735 s/0.125 s/5.861 s	895,928 KB/1,791,072 KB
P5 (pATS, PRI=Low)	Yes	5.719 s/0.125 s/5.836 s	895,920 KB/1,791,244 KB
P5 (SP, PRI=High)	–	1.953 s/0.094 s/2.047 s	528,540 KB/1,065,736 KB
P5 (SP, PRI=Low)	–	1.984 s/0.125 s/2.111 s	528,556 KB/1,065,748 KB

vantages and disadvantages of eATS and pATS.

1) Effects of CIR/CBS configuration on the worst-case delay results

To understand the impacts of ATS parameters on real-time performance, we first adopt the flow configuration in Table 2 but adjust the CIR/CBS to analyze the worst-case delay results of different priority classes under various ATS parameter conditions. There are total eight flows that can be categorized into the high- and low-priority classes. The rate and frame length of a high-priority flow are 4.8 Mbps and 48 μs. Also, for a low-priority flow, its rate and frame length are 4 Mbps and 40 μs, respectively. Note that the frame processing of ATS should maintain the transmission without packet losses for a safety-critical system. Therefore, the value of CIR in ATS should be greater than or at least equal to the total bandwidth requirement of flows. Otherwise, the shaped queue might be overflow from the long-term transmission aspect. In addition, the value of CBS is mainly related to the maximum allowed bursts in TSN networks. Lastly, the worst-case delay results under different CIR/CBS configurations can be obtained by the test automata.

The experiment results show that the worst-case delay of pATS can always remain the same value even if the CIR/CBS configurations are adjusted in various test cases. However, for the eATS method, the worst-case delay results of the high-priority class can be greatly affected by the configuration of *CBS*. When the configuration of *CIR* is slightly larger than the total bandwidth of aggregated flows, the delay results of high-priority flows with eATS will be larger than that with pATS, and the main difference between them is a maximum frame transmission time belonging to the low-priority class. Conversely, when the value of *CIR* and *CBS* are large enough to allow more bursts in networks, the delay results of eATS will be the same as those of pATS.

2) Performance analysis of eATS/pATS under multiple priority classes

The experimental results in Section 3.1 show that the worst-case delays of the high-priority class have significant differences under various CIR/CBS configuration cases when different frame selection methods are adopted in asynchronous TSN networks. To clarify whether eATS and pATS can maintain consistent real-time performance under multiple priority class scenarios, we need to understand the influences of eATS and pATS on the transmissions of prioritized flows. Therefore, we take three priority settings as an example and compare the delay results of flows under different priority classes using formal analysis, simulation-based approach, and analytical method, respectively.

As shown in Table 5, there are 6 shaped queues and 12 flows classified as high-, medium-, and low-priority classes. According to the network parameters given in Table 5 for the high-priority flows, it can be calculated that the transmission time for a high-priority frame is $400 \times 8 / 100 = 32$ μs, and the rate of a high-priority flow is $400 \times 8 / 1000 = 3.2$ Mbps. Similarly, the transmission time for the medium- and low-priority frame are 64 μs and 80 μs, and the rate of the medium- and low-priority flows are 6.4 Mbps and 8 Mbps, respectively.

Table 5 Network parameters with three priority classes

Priority	Queues			Flows		
	CBS (B)	*CIR* (Mbps)	Qnum	*L* (B)	*P* (μs)	Fnum
High	600	16	2	400	1000	4
Medium	1200	16	2	800	1000	4
Low	1600	16	2	1000	1000	4

Based on the formal analysis, simulation approach and analytical method [9], the delay results under three-priority classes are presented in Figure 10, and the priority class is used as the x-axis. Overall, the lower the priority of flow is, the larger its transmission delay becomes. Also, the red and blue histograms respectively represent the worst-case delay results of the three priority flows under the formal analysis method. For the flows belonging to the low-priority class, the delay results of both eATS and pATS are 704 μs (32 μs × 4 + 64 μs × 4 + 80 μs × 4). However, the delay results of the medium- and high-priority flows under the eATS are larger than those under the pATS. A possible reason is that lower-priority flows may have more opportunities to transmit when they arrive earlier than higher-priority flows under the eATS method. Then, the higher-priority flows under the eATS have to wait more time in shaped queues and resulting in a larger transmission delay. For this reason, it can be concluded that eATS provides more transmission opportunities for the lower-priority class at the expense of real-time performance for higher-priority classes. Besides, according to the delays of flows obtained from simulation experiments in Figure 10, it can also be seen that the average delay results of the low-priority flows under eATS has been significantly improved.

3) Delay results in an E2E network scenario

In the E2E experiment case, we intend to compare the worst-case E2E delay results to discuss the applicability of different frame selection methods under an E2E scenario. As shown in Figure 11, the network topology of the E2E case consists of four end systems (ES) and two bridges connected by a full-duplex link with 100 Mbps. Also, there are a total of eight flows $\{f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8\}$ in the network, of which the red color belongs to the high-priority class and the black color is the low-priority class. All flows start from the Talkers (ES1/ES2) and go through 3-hop forwarding before finally reaching the Listeners (ES3/ES4). The flow parameters in Table 2, including the frame length and period, are adopted to configure the high- and low-priority flows, and we also use the CIR/CBS parameters in Table 2 to configure the

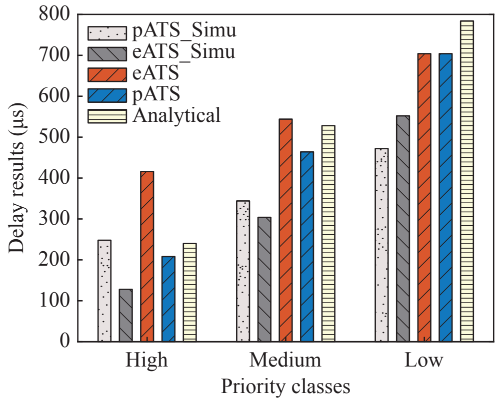


Figure 10 Delay results under three-priority classes.

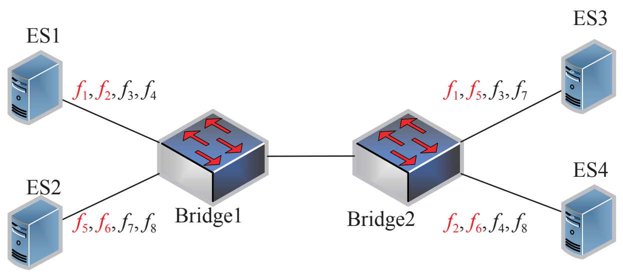


Figure 11 Network topology.

ATS shapers in networks.

Besides, we also evaluate the E2E worst-case delay bounds of eATS/pATS with the analytical results [9] and experimental simulations as benchmarks for reference comparison. Figure 12 shows the E2E delay results under formal analysis, simulation-based approach and analytical method. For each priority class, five different E2E delay results are demonstrated, including the worst-case delays of eATS/pATS method, the average simulation results in UPPAAL and the analytical calculation results with (3). As shown in Figure 12, for the delay results of low-priority class in formal analysis, the eATS and pATS have the same E2E worst-case delay results, 704 μs. However, the worst-case delay for the high-priority flows in eATS (544 μs) is larger than the delay in pATS (504 μs). Meanwhile, the average delay results for high- and low-priority flows are calculated based on the simulation experiments in UPPAAL. As shown in Figure 12, for the average delay results under the simulation experiments, the average delay of high-priority flows in eATS is larger than that in pATS. On the contrary, the average delay of low-priority flows in eATS is smaller than that in pATS. Therefore, the fairness of transmission for lower priority flows based on the eATS method can also be effectively verified in an E2E network scenario.

Besides, as it can be seen from Figure 12, the delay results for all priority classes with the analytical method are much larger than the results of eATS/pATS. Hence, the worst-case E2E delay results obtained by the analytical method can only show the upper-bounded delay of different priority classes, but cannot effectively distinguish the details of the frame selection process.

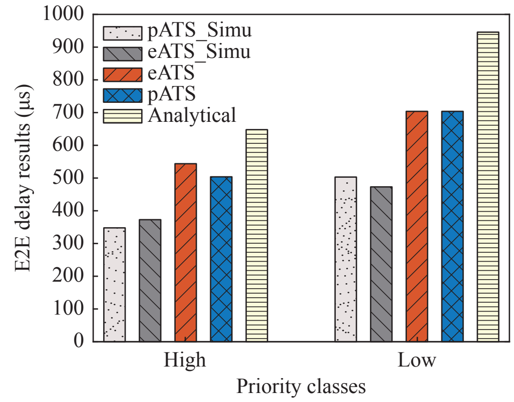


Figure 12 Comparison of delay results in an E2E network scenario.

4) Discussion

According to the comparison results of the above different cases, we can discuss and summarize advantages and disadvantages of the two frame selection methods.

Firstly, the pATS can always satisfy the shaping-for-free property under different *CIR/CBS* configurations, so it ensures the real-time performance of higher-priority flows during network transmissions. Secondly, the transmission of higher-priority flows with the eATS method might be delayed by lower-priority flows due to the factor of eligibility time. Hence, the lower-priority flows could have more transmission chances thanks to the possibly earlier eligibility time. Based on the above reason, it can be concluded that the eATS can provide more fairness for transmitting low-priority flows in networks, and the improvement for lower-priority class is mainly at the cost of a slight increase in worst-case delay for high-priority class. Lastly, in most cases we just take two-priority classes as an example to analyze the real-time performance of flows in eATS/pATS. However, the conclusions from this experiment are still valid for more priority scenarios according to the performance analysis of eATS/pATS under multiple priority classes, and the detailed implementation schemes of networking need to be evaluated and optimized further by the network designers to meet the deadline requirements.

VII. Conclusion

In this paper, we present a formal modeling framework of ATS to compare the influences of frame selection methods on real-time properties and corresponding network performance. Based on the formal model, the deadlock-free, starvation-free, and shaping-for-free properties have been verified for possible safety-critical applications. The verification results show that the shaping-for-free property is satisfied with the pATS method, but it is not fulfilled with the eATS method. Besides, we conduct a timing analysis for the two transmission selection methods. The E2E worst-case delay bounds under the two frame selection methods are compared and analyzed. The results illustrate that the eATS can provide more fairness for the transmission of low-priority flows in asynchronous TSN networks.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (Grant No. 62071023) and the Technology Development Fund of Shenzhen (Grant No. 2021SzVUP082).

References

[1] N. Finn, "Introduction to time-sensitive networking," *IEEE Communications Standards Magazine*, vol. 2, no. 2, pp. 22–28, 2018.

[2] IEEE, "802.1Qbv - Enhancements for Scheduled Traffic," Available at: http://www.ieee802.org/1/pages/802.1bv.html, 2016.

[3] IEEE, "IEEE Standard for Local and Metropolitan Area Networks - Bridges and Bridged Networks-Amendment 34: Asynchronous Traffic Shaping," Available at: https://standards.ieee.org/standard/802_1Qcr-2020.html, 2020.

[4] M. Wollschlaeger, T. Sauter, and J. Jasperneite, "The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0," *IEEE Industrial Electronics Magazine*, vol. 11, no. 1, pp. 17–27, 2017.

[5] W. W. Lu, S. L. Gong, Y. H. Zhu, "Timely data delivery for energy-harvesting IoT devices," *Chinese Journal of Electronics*, vol. 31, no. 2, pp. 322–336, 2022.

[6] S. Samii and H. Zinner, "Level 5 by layer 2: Time-sensitive networking for autonomous vehicles," *IEEE Communications Standards Magazine*, vol. 2, no. 2, pp. 62–68, 2018.

[7] A. Jabbar. "Establishing a joint project between SAE AS-1A2 and IEEE 802 to develop TSN profile for aerospace," available at: https://www.ieee802.org/1/files/public/docs 2020/new-Jabbar-TSN-for-Aerospace-0720-v01.pdf, 2020.

[8] W. Steiner, S. S. Craciunas, and R. S. Oliver, "Traffic planning for time-sensitive communication," *IEEE Communications Standards Magazine*, vol. 2, no. 2, pp. 42–47, 2018.

[9] J. Specht and S. Samii, "Urgency-based scheduler for time-sensitive switched Ethernet networks," in *Proceedings of the 2016 28th Euromicro Conference on Real-Time Systems (ECRTS)*, Toulouse, France, pp.75–85, 2016.

[10] J. Prados-Garzon, T. Taleb, and M. Bagaa, "Optimization of flow allocation in asynchronous deterministic 5G transport networks by leveraging data analytics," *IEEE Transactions on Mobile Computing*, vol. 22, no. 3, pp. 1672–1687, 2023.

[11] J. C. Knight, "Safety critical systems: Challenges and directions," in *Proceedings of the 24th International Conference on Software Engineering*, Orlando, FL, USA, pp.547–550, 2002.

[12] J. Prados-Garzon and T. Taleb, "Asynchronous time-sensitive networking for 5G backhauling," *IEEE Network*, vol. 35, no. 2, pp. 144–151, 2021.

[13] J. Y. Le Boudec, "A theory of traffic regulators for deterministic networks with application to interleaved regulators," *IEEE/ACM Transactions on Networking*, vol. 26, no. 6, pp. 2721–2733, 2018.

[14] E. Mohammadpour and J. Y. Le Boudec, "Analysis of dampers in time-sensitive networks with non-ideal clocks," *IEEE/ACM Transactions on Networking*, vol. 30, no. 4, pp. 1780–1794, 2022.

[15] H. Hu, Q. Li, H. G. Xiong, *et al.*, "The delay bound analysis based on network calculus for asynchronous traffic shaping under parameter inconsistency," in *Proceedings of the 2020 IEEE 20th International Conference on Communication Technology (ICCT)*, Nanning, China, pp.908–915, 2020.

[16] Z. F. Zhou, Y. Yan, M. Berger, *et al.*, "Analysis and modeling of asynchronous traffic shaping in time sensitive networks," in *Proceedings of the 2018 14th IEEE International Workshop on Factory Communication Systems (WFCS)*, Imperia, Italy, pp.1–4, 2018.

[17] J. Y. Le Boudec and P. Thiran, *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet*. Springer, Berlin, 2001,doi: 10.1007/3-540-45318-0.

[18] G. Behrmann, A. David, and K. G. Larsen, "*A tutorial on uppaal 4.0*," Dept. Comput. Sci., Aalborg University, 2006.

[19] R. Alur and D. L. Dill, "A theory of timed automata," *Theoretical Computer Science*, vol. 126, no. 2, pp. 183–235, 1994.

[20] R. Alur, C. Courcoubetis, and D. Dill, "Model-checking for real-time systems," in *Proceedings of the Fifth Annual IEEE Symposium on Logic in Computer Science*, Philadelphia, PA, USA, pp.414–425, 1990.

[21] Y. M. Jiang, "Network calculus and queueing theory: Two sides of one coin: invited paper," in *Proceedings of the Fourth International ICST Conference on Performance Evaluation Methodologies and Tools*, Pisa, Italy, article no.37, 2009.

[22] L. X. Zhao, P. Pop, and S. Steinhorst, "Quantitative performance comparison of various traffic shapers in time-sensitive networking," *IEEE Transactions on Network and Service Management*, vol. 19, no. 3, pp. 2899–2928, 2022.

[23] A. Nasrallah, A. S. Thyagaturu, Z. Alharbi, *et al.*, "Performance comparison of IEEE 802.1 TSN time aware shaper (TAS) and asynchronous traffic shaper (ATS)," *IEEE Access*, vol. 7, pp. 44165–44181, 2019.

[24] A. Grigorjew, F. Metzger, T. Hoßfeld, *et al.*, "A simulation of asynchronous traffic shapers in switched Ethernet networks," in *Proceedings of the 2019 International Conference on Networked Systems (NetSys)*, Munich, Germany, pp.1–6, 2019.

[25] B. W. Fang, Q. Li, Z. J. Gong, *et al.*, "Simulative assessments of credit-based shaping and asynchronous traffic shaping in time-sensitive networking," in *Proceedings of the 2020 12th International Conference on Advanced Infocomm Technology (ICAIT)*, Macao, China, pp.111–118, 2020.

[26] H. Charara, J. L. Scharbarg, J. Ermont, *et al.*, "Methods for bounding end-to-end delays on an AFDX network," in *Proceedings of the 18th Euromicro Conference on Real-Time Systems*, Dresden, Germany, pp.191–202, 2006.

[27] M. Adnan, J. L. Scharbarg, J. Ermont, *et al.*, "An improved timed automata approach for computing exact worst-case delays of AFDX sporadic flows," in *Proceedings of the 2012 IEEE 17th International Conference on Emerging Technologies & Factory Automation (ETFA 2012)*, Krakow, Poland, pp.1–8, 2012.

[28] A. David, J. Illum, K. G. Larsen, *et al.*, "Model-based framework for schedulability analysis using UPPAAL 4.1," in *Model-Based Design for Embedded Systems*, G. Nicolescu and P. J. Mosterman, Eds. CRC Press, Boca Raton, FL, USA, pp.117–144, 2018.

[29] J. H. Sun, N. Guan, R. X. Shi, *et al.*, "Schedulability analysis for timed automata with tasks," *ACM Transactions on Embedded Computing Systems*, vol. 20, no. 5s, article no. 89, 2021.

[30] P. J. Han, Z. J. Zhai, B. Nielsen, *et al.*, "Schedulability analysis of distributed multicore avionics systems with UPPAAL," *Journal of Aerospace Information Systems*, vol. 16, no. 11, pp. 473–499, 2019.

[31] D. Bujosa, I. Álvarez, and J. Proenza, "CSRP: An enhanced protocol for consistent reservation of resources in AVB/TSN," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 5, pp. 3640–3650, 2021.

[32] J. Lv, Y. X. Zhao, X. Wu, *et al.*, "Formal analysis of TSN scheduler for real-time communications," *IEEE Transactions on Reliability*, vol. 70, no. 3, pp. 1286–1294, 2021.

[33] P. P. Tang and T. Y. Tai, "Network traffic characterization

using token bucket model," in *Proceedings of the Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies: The Future is Now*, New York, NY, USA, pp.51–62, 1999.

[34] J. Bengtsson, K. Larsen, F. Larsson, *et al.*, "UPPAAL - a tool suite for automatic verification of real-time systems," in *Proceedings of the International Hybrid Systems Workshop*, Berlin, Germany, pp.232–243, 1995.

[35] Y. Jiang, H. Liu, H. B. Song, *et al.*, "Safety-assured model-driven design of the multifunction vehicle bus controller," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 10, pp. 3320–3333, 2018.

Ershuai LI was born in 1993. He received the B.S. degree from the Shandong University. He is currently pursuing the Ph.D. degree in communication and information systems with the School of Electronic Information Engineering, Beihang University, China. His current research interests include avionics systems and real-time networks.
(Email: ershuai_li@buaa.edu.cn)

Xuan ZHOU was born in 1994. She received the Ph.D. degree in communication and information system from Beihang University in 2021. She is now a Postdoctoral Researcher in Beihang University, and her current research covers scheduling design and performance evaluation for real-time communication systems.
(Email: lomoo@buaa.edu.cn)

Jinjing SUN was born in 1993. She received the Ph.D. degree in communication and information system from Beihang University in 2022. Her research interests include 5G networks and multicarrier communication.
(Email: sunjinjing2015@buaa.edu.cn)

Huagang XIONG received the Ph.D. degree in communication and information systems from the School of Electronic Information Engineering, Beihang University, China, in 1998. He is the Head of the Avionics and Bus Communications Research Team (ABC), School of Electronic Information Engineering, Beihang University. His research interests include communication network theory and technology, avionics information integration, airborne networks, and standards.
(Email: hgxiong@buaa.edu.cn)

Feng HE received the Ph.D. degree in communication and information systems from Beihang University, Beijing, China, in 2009. He is an Associate Professor with the School of Electronics and Information Engineering, Beihang University. His research interests include real-time networking and scheduling, and avionics systems. He is a Member of the Avionics and Air Traffic Control Branch of China Society of Aeronautics and Astronautics.
(Email: robinleo@buaa.edu.cn)