Research on the e2e Latency Calculation Considering the Gate Mechanism in Time-Sensitive Networking

Zhipeng Cao*, Qinrang Liu and Dongpei Liu
PLA Strategic Support Force Information Engineering University, Zhengzhou, China

*E-mail: 704634515@qq.com

Abstract. Time-Sensitive Networking (TSN) is an emerging technology that solves the problem of Ethernet deterministic transmission. As its main idea, TSN is compatible with traditional Ethernet Best Effort transmission, while enabling transmission with bounded delay, low jitter, and low packet loss rate for Time-Sensitive traffic. However, existing methods of solving the problem of delay calculation are either too complex or too impractical. This paper proposed a Time-aware calculation method that can calculate the transmission delay in TSN. Firstly, we analyze the 802.1Qbv Time-Aware shaper for specifying its function; Secondly, the network topology, the network traffic, and especially the function of the switch are modeled. We divide the switch into 3 time-windows. Finally, a Time-aware calculation method is proposed, and the transmission cost is calculated considering the frame size, bandwidth, and the percentage of protected window. The experiments show that the method can calculate the end-to-end translation with different senarios and parameters with low complexity.

1. Introduction
Nowadays, with the development of new technologies like 5G, IoT, ADAS, and Industry 4.0 has raised a huge requirement on the deterministic transmission of data packets. A lot of application scenarios in industrial automation and automotive networks domain ask for a better QoS with bounded low latency, low jitters, and fault tolerance. Although traditional solutions like FlexRay[1], ARINC664 p7[2], PROFINET[3], or EtherCAT[4] have solved the problems to some extent, but they still have their shortcomings. As technology improves, there’s an expectation that various sensors, processors, actuators, and other equipment need to be compatible with each other, so that they can work together. On the one hand, existing methods are all based on Fieldbus or Ethernet technology, and the various solutions are different from software to hardware, even some technologies are protected proprietary technologies. On the other hand, the integration of IT technology and OT technology is the future development trend[5]. People hope to merge the manufacturing system in the industrial field and the operation management system in the office together. On a unified information platform, we can achieve integration in all aspects of production management, operational decision-making, and manufacturing execution. Thus, the work efficiency is comprehensively improved.

Time-Sensitive Networking is an emerging technology that solves the problem of Ethernet deterministic transmission. The IEEE has proposed a series of standards to support transmission on the Ethernet with bounded delay, low jitter, and low packet loss rate for Time-Sensitive traffic, while together with traditional Best Effort traffic on the same data link. TSN mainly works in the Data Link Layer and tries to set up a general standard of deterministic transmission. In 2006, the AVB Task Group was established, aiming at providing time synchronization, low latency, and bandwidth
reservation for professional audio and video (A/V). They came up with some special ideas, which then became the basis of TSN called 802.1BA[6] standard. But, due to the simple use scenarios and limited functions, AVB can’t provide more mechanisms to satisfy different transmission requirements. In 2012, the AVB Workgroup upgraded to the TSN Task Group. The TSN Workgroup is going to provide services through IEEE 802 networks with guaranteed packet transmission, which have bounded latency, low packet delay variation, and low packet loss. TSN has broadened the use cases of AVB, including industrial control, automobile networks, and even 5G fronthaul networks. The newest standard is 802.1Qcp[7], which specified the YANG data model in TSN.

2. 802.1Qbv mechanism
As mentioned above, the most critical aspect of TSN is the converged transmission, which allows time-sensitive traffic and the general Best Effort to be delivered and received on the same link. On the one hand, we need a global clock reference with bounded precision. On the other hand, the isolation and scheduling of different traffic formed the foundation of converged transmission. We mainly focus on the latter one. Indeed, the 802.1Qbv[8] has proposed a mechanism working as a TDMA fashion to guarantee the time-triggered operation.

![Figure 1. A typical TSN switch supporting 802.1Qbv](image)

The 802.1Qbv standard formalized a mechanism called the Time-Aware Shaper (TAS). TAS is a method of scheduling based on a synchronized clock. As shown in figure 1, when frames arrive at the switch from the ingress port, it is forwarded to the corresponding output port and assigned to its respective queue. There are at most eight queues for different priority. Each queue corresponds to a time-aware gate, which decides whether the data in the queue can be transmitted. When the corresponding gate is open, the data can be transmitted; when the corresponding gate closes, the data can’t be transmitted. The control status of gates can then form a schedule list called the Gate Control List (GCL). The GCL defines the exact time when a gate is open or not. Hence, the GCL guarantees the isolation of different traffic and the time-triggered operation of time-sensitive data flows.

3. Problem formalization
3.1 Network topology
The IEEE 802 network includes end systems, switches, and physical links. End systems can consist of different senders and receivers. We model the network topology as $G = \{V, E\}$, where $v_i \in V$ stands for the vertices in the network, including the end systems and the switches, $(v_i, v_j), (v_j, v_i) \in E$ stands for the edges. As shown in figure 2, A feasible path can be $R_1 = \{e_0, v_0, v_5, v_7, e_7\}$ or $R_2 = \{e_0, v_0, v_1, v_3, v_6, v_7, e_7\}$. 
3.2 Traffic
There are three kinds of traffic in the TSN environment, which can be described as follows.

Time-triggered traffic (TT): Strict requirements, with bounded latency and low jitter.

Audio Video Bridging traffic (AVB): General requirements, with bounded latency.

Best Effort traffic (BE): No special requirements.

The 802.1Q frame header can recognize the traffic type with a VLAN tag like the priority code point (PCP), which consists of 3 bits to represent eight different priorities. There can be at most 2 AVB queues in each egress port (AVB class A and AVB class B), while TT traffic and BE traffic share the rest queues.

3.3 Switch function
We assume that all the devices in our discussion are capable of 802.1Qbv; that is to say, they support the isolation of different traffic. According to the operating principle of GCL, we can divide it into three windows, which are Protected Window, Unprotected Window, and Guard Band, as shown in figure 3. The GCL runs in a cyclic way that these windows will always repeat until the configuration has been changed.

Protected Window: It is used for time-triggered and critical traffic only; no other traffic can be transmitted in this time window. During this window, the gate of the corresponding queue is set to open, and the other gates are set close.

Unprotected Window: It is used for unprotected traffic like AVB and BE traffic. If multiple queues are opening, they may compete on the egress port base on their priority.

Guard Band: In order to ensure that the remaining unprotected traffic cannot affect the transmission of protected traffic, a guard band is created. It is necessary to stop the transmission of unprotected traffic far in advance of the protected time-slot.

4. Calculation of TAS-aware latency

4.1 The latency formalism
We mainly focus on the transmission delay and the switching delay. Transmission delay refers to the time it takes for a node to send a data block from the node to the transmission medium, i.e., the total time required for a node from the start of sending the frame to the completion of it. The transmission
delay of a specific link is related to its bandwidth and the size of the data. And it can be calculated by (1).

\[ t_{(i,j)} = \frac{\text{size}}{b_{(i,j)}} \] (1)

The switching delay is mainly caused by the 802.1Qbv TAS mechanism, and we ignore the processing delay and the queuing delay. According to the TAS, when the gate of TT traffic is open, it can be transmitted on the link; When the gate is set close, the TT traffic cannot transmit and have to wait for the next time slot for protected window. The GCL runs in a cyclic way that whether the frame has to wait is decided by the time when it arrives. We use the function \( f(t) \) to represent whether the frame is in the protected window, as formula (2).

\[
f(t) = \begin{cases} 
0, & (t \% T_{cycle}) \in [0, T_{protected}] \\
1, & (t \% T_{cycle}) \in [T_{protected}, T_{cycle}]
\end{cases}
\] (2)

So, the dispatching time of the corresponding switch can be then formalized as (3).

\[ t_{k+1} = t_k + t_{\text{trans}} + t_{\text{sw}} \] (3)

According to (1):

\[ t_{\text{trans}} = \frac{\text{size}}{b_{k,k+1}} \] (4)

According to (2):

\[ t_{\text{sw}} = f(t_k + t_{\text{trans}}) \cdot (T_{cycle} - (t_k + t_{\text{trans}}) \% T_{cycle}) \] (5)

### 4.2 TAS-aware calculation

Our propose is to find a proper method to calculate the end-to-end latency. The end-to-end latency can be formalized as (6).

\[ t_{e2e} = t_{\text{last}_\text{sw}} + t_{(\text{last}_\text{sw}, \text{rev})} - t_{\text{start}} \] (6)

Here, we use the TAS-aware calculation method. The basic idea of TRC method is stated as follow:

**Step1.** Initialize the nodes and the links between them;

**Step2.** Set the starting node and the ending node, calculate all the feasible paths between them;

**Step3.** Calculate the latency in every path, considering the effect of TAS.

We define the cost of transmission that equals to the time spent on the link and the switch. Whenever a frame arrives at the egress port of a switch, the dispatching time of each possible link can be seen as the cost. In the calculation of all the feasible paths between the starting node and the ending node, we use an algorithm of recursive in Algorithm 1. The variable path represents one feasible path from beginning to end, and the variable paths represent all the paths. The path from starting node to the ending node can be seen as a plus between the path from the starting node to its neighbors and the path from the neighbors to the ending node. For each calculation, we add the new result to the variable paths.

### Algorithm 1 FindPath(graph, start, end, path[])

```
0: init path
1: if (start == end)
2:  return [path]
3: end if
4: init paths[]
5: for node in graph[start] do
6:  if node not in path
```
7: \text{newpaths} = \text{FindAllPath}(\text{graph, node, end, path})
8: \text{for newpath in newpaths do}
9: \quad \text{paths.add(newpath)}
10: \quad \text{end for}
11: \text{end if}
12: \text{end for}
13: \text{return paths}

In the calculation of transmission cost, our main idea is based on the formula (3). From the result of Algorithm 1, we can know every feasible path between the starting node and the ending node. We can then calculate the end-to-end delay of each path in a recursion way in algorithm 2.

Algorithm 2 TRA(graph, paths[])
0: init i
1: for path in paths do
2: \quad init k
3: \quad while (k < length(path) – 2)
4: \quad \quad calculate t_trans
5: \quad \quad calculate t_sw
6: \quad \quad t_{next} = tk + t_{trans} + t_{sw}
7: \quad k++
8: \quad end while
9: \quad t_{all}[i] = t[k+1] + t_{trans,77}
10: \quad i++
11: \quad end for

5. Experiment and Analysis

5.1 Experiment setup
We set the link bandwidth and the size of frames as table 1.

| item           | data |
|----------------|------|
| size of frames / Byte |      |
| 300            | 900  |
| 600            | 1200 |
| 900            | 1500 |
| link bandwidth / Gbps |    |
| 2.5            | 5    |

5.2 Discussion
From the result of Algorithm 1, we can know that from the sender to receiver under our discussion, and there can be multiple transmission paths from at most 6 hops to at least 3 hops in the switch topology.

The path of 6 hops can be:
\{v_0, v_1, v_3, v_6, v_5, v_7\} and \{v_0, v_3, v_1, v_5, v_6, v_7\}

The path of 5 hops can be:
\{v_0, v_3, v_6, v_7\}, \{v_0, v_3, v_1, v_5, v_7\} and \{v_0, v_3, v_5, v_6, v_7\}
The path of 4 hops can be:
\[ \{v_0, v_1, v_5, v_7\}, \{v_0, v_2, v_4, v_7\} \text{ and } \{v_0, v_3, v_6, v_7\} \]

The path of 3 hops can be:
\[ \{v_0, v_4, v_7\} \text{ and } \{v_0, v_5, v_7\} \]

Figure 4. The effect of bandwidth

Figure 5. The effect of frame size

Figure 6. The effect of protected window

a) The effect of bandwidth

We choose the size of 900 Bytes, the cycle time of 100us, and the 50% of protected window. The bandwidth of links varies from 1Gbps to 5Gbps, as in table 2. The end-to-end latency is reduced as the bandwidth increases. As shown in figure 4, there is a bigger decrement for 1Gbps to 2.5Gbps than 2.5Gbps to 5Gbps, which can be at most 71.8% and 50%. As for each bandwidth, the higher the bandwidth, the lower the latency variation of different hops.

Table 2. E2e latency with varying bandwidth

| Items    | 3 Hops | 4 Hops | 5 Hops | 6 Hops |
|----------|--------|--------|--------|--------|
| 1Gbps    | 28.8   | 57.2   | 64.4   | 71.6   |
| 2.5Gbps  | 11.52  | 14.4   | 17.28  | 20.16  |
| 5Gbps    | 5.76   | 7.2    | 8.64   | 10.08  |

b) The effect of frame size

We choose bandwidth of link 1Gbps, the cycle time of 100us, and the 50% of protected window. The size of frames varies from 300 Bytes to 1500 Bytes, which is the maximum of the Ethernet transmission unit, as shown in table 3. As the frame becomes larger, the end-to-end latency increases...
in figure 5. The rate of latency growth reduces when the size of the frames is relatively large. The latency is similar at the beginning nodes for 3, 4 and 5 hops situation and then varies for the 5 hops situation.

**Table 3.** E2e latency with varying frame size

| Items      | 3 Hops | 4 Hops | 5 Hops | 6 Hops |
|------------|--------|--------|--------|--------|
| 300 Bytes  | 9.6    | 12     | 14.4   | 16.8   |
| 600 Bytes  | 19.2   | 24     | 28.8   | 54.8   |
| 900 Bytes  | 28.8   | 57.2   | 64.4   | 71.6   |
| 1200 Bytes | 59.6   | 69.2   | 78.8   | 109.6  |
| 1500 Bytes | 62     | 74     | 86     | 112    |

**Table 4.** E2e latency with varying protected window

| Items | 3 Hops | 4 Hops | 5 Hops | 6 Hops |
|-------|--------|--------|--------|--------|
| 0.1   | 157.2  | 207.2  | 257.2  | 307.2  |
| 0.3   | 57.2   | 64.4   | 71.6   | 107.2  |
| 0.5   | 28.8   | 57.2   | 64.4   | 71.6   |
| 0.7   | 28.8   | 36     | 57.2   | 64.4   |
| 0.9   | 28.8   | 36     | 43.2   | 50.4   |

c) The effect of protected window

We choose bandwidth of link 1Gbps, the size of frames 900 Bytes, the cycle time of 100us. The percentage of protected window varies from 0.1 to 0.9, as shown in table 4. The TAS cycle time and the length of protected window are the core parameters of the 802.1Qbv gate mechanism. Here we assume that the cycle time is fixed and change the percentage of protected window. In figure 6, the result shows that as the percentage grows, the overall end-to-end latency reduces, and there is a significant decrement of 0.1 to 0.3. From 0.5 to 0.9, the latency changes a little.

**Table 4.** E2e latency with varying protected window

6. Conclusion

In this work, we mainly address the problem of the calculation of end-to-end latency, considering the gate mechanism in Time-Sensitive Networking. A time-aware calculation method is proposed: all the paths from sender to receiver are found in a recursive way. The transmission cost is calculated based on the formula (6), where there is in a recursion fashion. For each path, the end-to-end latency is then analyzed: the key parameters that affect the behavior of frames are size, the link bandwidth, cycle time, percentage of protected window, and the hops of corresponding paths.

From the calculation of different situation, it shows that a higher bandwidth can effectively reduce the end-to-end latency. The larger the frame size is, the longer the latency, while the growth rate of latency is slowed down when the size is bigger enough. As for the effect of protected window, there will be an obvious decrease when the percentage grows to 0.3 and keeps a slow downturn trend after that.

**References**

[1] Makowitz, R. and C. Temple. *FlexRay-a communication network for automotive control systems.* in 2006 IEEE International Workshop on Factory Communication Systems. 2006. IEEE.

[2] Committee, A., *ARINC specification 664P7: Aircraft Data Network, part 7: Avionics Full-Duplex Switched Ethernet (AFDX) network.* Aeronautical Radio Inc, 2005.

[3] Feld, J., *PROFINET - Scalable factory communication for all applications.* Wfcs 2004: Ieee International Workshop on Factory Communication Systems, Proceedings, 2004: p. 33-38.
[4] Jansen, D. and H. Buttner, *Real-time Ethernet: the EtherCAT solution*. Computing and Control Engineering, 2004. 15(1): p. 16-21.

[5] Lo Bello, L. and W. Steiner, *A Perspective on IEEE Time-Sensitive Networking for Industrial Communication and Automation Systems*. Proceedings of the Ieee, 2019. 107(6): p. 1094-1120.

[6] *IEEE Standard for Local and metropolitan area networks--Audio Video Bridging (AVB) Systems*. IEEE Std 802.1BA-2011, 2011: p. 1-45.

[7] *IEEE Standard for Local and metropolitan area networks--Bridges and Bridged Networks-Amendment 30: YANG Data Model*. IEEE Std 802.1Qcp-2018 (Amendment to IEEE Std 802.1Q-2018), 2018: p. 1-93.

[8] *IEEE Standard for Local and metropolitan area networks -- Bridges and Bridged Networks - Amendment 25: Enhancements for Scheduled Traffic*. IEEE Std 802.1Qbv-2015 (Amendment to IEEE Std 802.1Q-2014 as amended by IEEE Std 802.1Qca-2015, IEEE Std 802.1Qcd-2015, and IEEE Std 802.1Q-2014/Cor 1-2015), 2016: p. 1-57.