Demonstration of Latency Control Label-Based Bounded-Jitter Scheduling in a Bridged Network for Industrial Internet

Jiahao Ma[(1)], Jiawei Zhang[(1)]*, Jim Zou[(2)], Hao Yu[(3)], Tarik Taleb[(3)], Yaojia Dong[(1)] and Yuefeng Ji[(1)]

[(1)] State Key Lab of Information Photonics and Optical Communications, Beijing University of Posts and Telecommunications (BUPT), Beijing, China. *Corresponding author*: *zjw@bupt.edu.cn.
[(2)] ADVA Optical Networking SE, Maerzenquelle 1-3, Meiningen, Germany.
[(3)] Aalto University, Espoo, Finland.

Abstract *We propose and experimentally demonstrate a latency control label (LCL)-based bounded-jitter scheduling for Industrial Internet applications in an asynchronous bridged network. The demonstration results show that our proposed scheme can achieve a deterministic packet delay variation regardless of the number of hops.*

Introduction

Emerging use cases in Industrial Internet such as collaborative manufacturing and precision motion control[1], require stringent end-to-end latency performance guarantees for precise operation. These requirements focus on a bounded end-to-end latency as low as hundreds of microseconds, and packet delay variation (PDV) (a.k.a. jitter) that is one order of magnitude lower[2]. Fig. 1 shows a typical collaborative manufacturing scenario for Industrial Internet. To enable terminals (e.g., robotic arms and machine tools) in different factories in the same park to cooperate with each other accurately, bounded jitter transmission is required between the production command center and the terminals. To meet this requirement, many factories build their dedicated industrial networks[3].

A well-known solution is the time-sensitive networking (TSN), where low PDV is achieved by synchronization-based scheduling mechanisms, such as time-aware shaper (TAS) in 802.1Qbv[4]. However, the cost of establishing a network with all nodes synchronized is high[5]. For medium and small enterprises, an asynchronous-based scheme is more cost-effective.

There are many asynchronous strategies proposed to achieve deterministic transmission. Frame preemption proposed in 802.1Qbu allows high-priority frames to interrupt the transmission of low-priority frames. Paternoster algorithm based on cyclic queuing and forwarding (CQF)[6] provides a bounded delay but removes the dependence on synchronous timing. However, the PDV achieved by the above schemes will increase with the number of hops along the network path, which limits the scalability of industrial networks[7],[8].

In this paper, we propose and experimentally demonstrate a latency control label-based scheduling scheme to achieve bounded-jitter transmission for time-critical Industrial Internet applications in an asynchronous bridged network. The latency control label (LCL) is a key enabler to realise dynamic traffic scheduling in a switching node. Experimental results show that the proposed scheme can achieve a deterministic PDV regardless of the number of hops.

LCL-based Traffic Scheduling

The transmission latency variations mainly result from the uncertain queuing delay in the switching node, and it will be accumulated by nodes along the path. To address this problem, we implement an LCL-based scheduling (LCLS) scheme, where

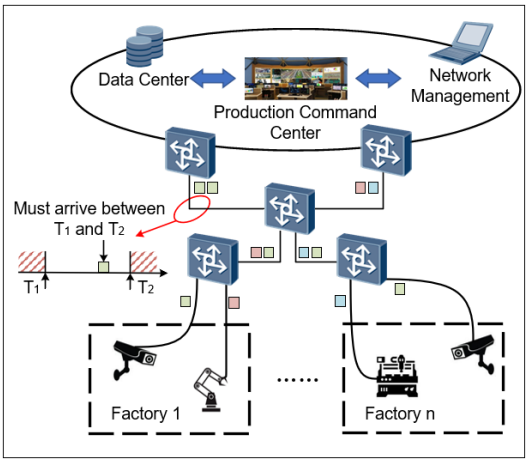

Fig. 1 Collaborative manufacturing scenario.

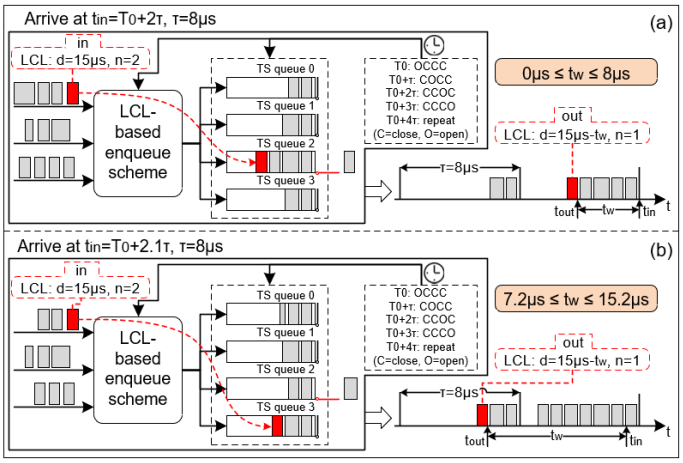

Fig. 2 Examples of LCL-based scheduling.

978-1-6654-3868-1/21/$31.00 ©2021 IEEE

each frame records the number of nodes it has passed through, and the queuing delay experienced in these nodes. Therefore, the traffic scheduling can be adjusted in the subsequent nodes to lower the end-to-end latency variations. The LCL consists of two parts: d and n, where d denotes the remaining available queuing delay and n denotes the number of subsequent nodes along the path. Before entering the network, a time-sensitive (TS) frame will initialize d as an acceptable queuing delay for an end-to-end path according to the service requirements, and n as the total number of hops based on the pre-calculated route. It should be noted that we just pay attention to the queuing delay in this paper, thus the routing of TS frames and the propagation latency via fiber are fixed. Once a frame passes through a node, LCL will be updated accordingly.

Fig. 2 illustrates two examples, where a TS frame with the same ingress LCL comes into the same node but arrives at different time. Typically, when a TS frame enters a node, it will be put into one of four TS queues in the output port under the LCL-based enqueue scheme. These TS queues transmit in turn according to a local clock, which does not need synchronization. Let τ denote the transmission time of each cyclic TS queue, $d_{ph} = {}^{d}/_{n}$ denotes the per-hop queuing delay derived from LCL, t_w denotes the real queuing delay experienced in a node. In Fig. 2(a), a TS frame with LCL ($d = 15\mu s, n = 2$) arrives at the node at $T_0 + 2\tau$ ($\tau = 8\mu s$) when TS queue 2 is just beginning to transmit. Comparing d_{ph} with the remaining transmission time of TS queue 2, TS frame is put into TS queue 2 so that it can be transmitted at an appropriate time. According to the load of TS queue 2, t_w can vary from 0 to $8\mu s$. In Fig. 2(b), TS frame arrives at $T_0 + 2.1\tau$. At this time, TS queue 2 has $7.2\ \mu s$ transmission time left, which is less than d_{ph} ($7.5\ \mu s$). Thus, TS frame is put into TS queue 3. In this case, t_w can vary from $7.2\mu s$ to $15.2\mu s$. Accordingly, if the target queue is not full, putting TS frame in an appropriate queue can make t_w meet the formula: $d_{ph} - \tau \le t_w \le d_{ph} + \tau$. Otherwise, TS frame will be put into the next non-full TS queue, and LCL will record the real queuing delay so that the scheduling in the subsequent nodes can modify it. For each frame that is ready to leave, the switching node will update its LCL according to the t_w, as shown in Fig.2. Based on the LCLS, the end-to-end latency d_{e2e} is bounded within $[d_0 - \tau,\ d_0 + \tau]$, where d_0 is the initial value of d. With this, LCLS can guarantee a bounded jitter transmission (i.e., $PDV = 2\tau$) of TS frames.

Hardware Design of Switching Node

Fig. 3 shows the hardware functions design of a LCLS enabled switching node structure. The output port of a node contains five First-In First-Out (FIFO) queues, four of which are used for TS frames, and one is used for best-effort (BE) frames. The input BE frames are directly put into BE queue while the input TS frames are put into TS queues according to the LCL-based enqueue scheme explained in previous section. Among TS queues, four queues transmit in turn with a cycle of τ. The LCL will be tagged in the LCL update module. To reduce the impact of BE frames on the delay of TS frames, frame preemption is implemented between the TS queues and the BE queue.

Experimental Setup and Results

To demonstrate the performance of LCLS, we setup an experimental testbed shown in Fig. 4(a). We use FPGA Xilinx Virtex UltraScale+ HBM VCU128 board with four 100Gbps Ethernet interfaces to realize the switching nodes. Considering each 100GbE interface can be separated into four independent 25Gbps Ethernet interfaces, we implement three switching nodes with LCLS scheme on a board with two 25GbE input ports and one 25GbE output port for each node, and each 25GbE interface can support preemption 802.3br feature. Fig. 4(b) shows the overview of the testbed and our implementation. There are three LCLS enabled Switches (SWs) and one commercial Ethernet SW #4 (without LCLS) for data flow feedback. Four TS traffic generators with analyzers are used to evaluate the PDV and end-to-end latency of TS traffic. The onboard functions of FPGA are used to generate constant bit rate (CBR) BE flows as background flows, each of them is connected to an LCLS-enabled SW. As shown in Fig. 4 (b), flow 1 generated by TS traffic generator #1 returns to the analyzer

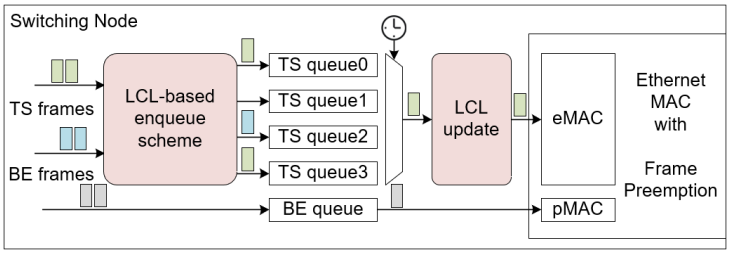

Fig. 3 Hardware structure of switching node.

Tab. 1 TS and BE traffic settings

	Traffic Generator Setting				
TS	Type	Average Bandwidth	Burst Bandwidth	Burst Time	Frame Length
	Burst	5.78125 Gbps	25Gbps	8μs	128 Bytes
BE	Type	Bandwidth	Frame Length		
	Constant	5Gbps	64~1518Bytes		

	Latency Control Label
TS1	$d_0 = 10\mu s, n_0 = 1$
TS2	$d_0 = 10\mu s, n_0 = 2$
TS3	$d_0 = 15\mu s, n_0 = 3$
TS4	$d_0 = 20\mu s, n_0 = 3$

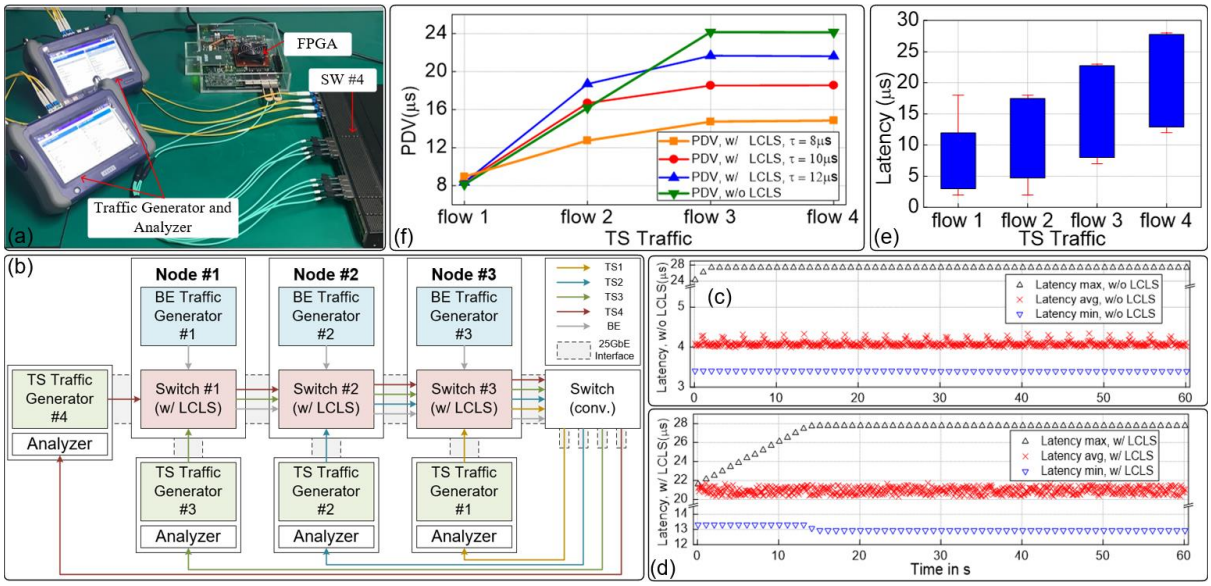

Fig. 4: (a) Experimental setup; (b) Overview of testbed and our implementation; (c) Latency of flow 4 w/o LCLS; (d) Latency of flow 4 w/ LCLS. (e) Bounded latency by LCLS; (f) PDV reduction by LCLS with different τ.

after one hop; flow 2 returns after two hops; flow 3 and flow 4 return after three hops. The BE frames exceeding the upper limit of bandwidth of an output port (25Gbps in experiment) will be discarded. The details of traffic generator settings and initial labels of TS traffic are shown in Tab. 1, where d_0 and n_0 denote the initial value of d and n. Considering the length of most time-sensitive control frames for Industrial Internet are in the range of [40, 250] bytes (i.e., short frame), we set the length of TS frame to 128 bytes in the experiment.

Fig. 4(c-d) show the measured end-to-end latency of flow 4, when using or without using LCLS under the condition of $\tau = 8\mu s$. The maximum and minimum latency reflect the bounds of packet delay during one experimental cycle (60 seconds). The average latency reflects the mean packet delay in a short time period (0.1 second). We can observe that the minimum and average end-to-end latency of LCLS are much higher than those when LCLS is not applied. This is because, with LCLS, the queuing delay of TS frame is bounded within $[d_{ph} - \tau, d_{ph} + \tau]$ in a SW, even if this SW has enough bandwidth to transmit it at an early time. That is why the minimum latency and average latency are higher, while the PDV is lower when LCLS is used.

As shown in Fig. 4(e), the blue rectangles depict the end-to-end latency range of four flows under the condition of $\tau = 8\mu s$, which are lower than $d_0 + \tau$ (red upper-bounded line) and higher than $d_0 - \tau$ (red lower-bounded line). This indicates that all the TS flows achieve bounded end-to-end latency as discussed above, which also means that bounded jitter is realized.

As shown in Fig. 4(f), PDV increases linearly with the number of hops without using LCLS (green line). If the hop count reaches a certain level, the PDV of TS frames may not meet the low jitter requirement of Industrial Internet. On the contrary, when LCLS is used (lines in orange, red and blue for $\tau = 8\mu s, 10\mu s$ and $12\mu s$ respectively), as the hop count n increases, d_{ph} decreases accordingly, so the traffic scheduling in SWs will be adjusted to make PDV lower than 2τ. As the number of hops increases, the proposed scheme shows much better performance in comparison to when it is not used. In addition, the transmission time of cyclic TS queue (τ) can be designed properly to satisfy diverse PDV requirements. It should be emphasized that τ can be set to be a small value to achieve a lower PDV (e.g., less than $1\mu s$). However, a small τ will lead to a small buffer memory of the TS queue, which can cause packet loss when a large burst occurs. This can be solved by arranging more FIFO TS queues in the output ports, but it also brings additional costs.

Conclusions

We proposed and experimentally demonstrated a latency control label-based bounded-jitter scheduling for time-critical industrial applications in an asynchronous bridged network. We demonstrated that the scheme can achieve deterministic jitter regardless of the number of hops. In addition, the LCL-based scheduling can easily adapt to different jitter requirements by modifying the transmission time of TS queues.

Acknowledgements

This work was supported by the National Nature Science Foundation of China Project No. 61971055, the Beijing Natural Science Foundation No. 4192039, and the Academy of Finland 6Genesis project No. 318927.

References

[1] 3GPP TS 22.104 V18.0.0 (2021-03), Service requirements for cyber-physical control applications in vertical domains (Release 18).

[2] M. Szczerban, A. -E. Kasbari, A. Ouslimani, S. Bigo and N. Benzaoui, "On the hardware cost of end-to-end latency variation control for time-slotted optical networks," 2020 European Conference on Optical Communications (ECOC), 2020, pp. 1-4.

[3] N. Shibata et al., "First Demonstration of Autonomous TSN-based Beyond-Best-Effort Networking for 5G NR Fronthauls and 1,000+ Massive IoT Traffic," 2020 European Conference on Optical Communications (ECOC), 2020, pp. 1-4.

[4] IEEE 802.1Qbv-2015, "IEEE Standard for Local and metropolitan area networks -- Bridges and Bridged Networks - Amendment 25: Enhancements for Scheduled Traffic", March 2016.

[5] J. Zou et al., "Field-Trial Evaluation of Low-Latency and Timing-Accurate 100G Ethernet Aggregator for Converged Mobile X-haul," 2018 European Conference on Optical Communication (ECOC), 2018, pp. 1-3.

[6] IEEE 802.1Qch-2017, "IEEE Standard for Local and metropolitan area networks--Bridges and Bridged Networks--Amendment 29: Cyclic Queuing and Forwarding", June 2017.

[7] J. Prados-Garzon, T. Taleb and M. Bagaa, "LEARNET: Reinforcement Learning Based Flow Scheduling for Asynchronous Deterministic Networks," ICC 2020 - 2020 IEEE International Conference on Communications (ICC), 2020, pp. 1-6.

[8] J. Prados-Garzon and T. Taleb, "Asynchronous Time-Sensitive Networking for 5G Backhauling," in *IEEE Network*, vol. 35, no. 2, pp. 144-151, March/April 2021.