Long-distance Deterministic Transmission among TSN Networks: Converging CQF and DIP

Weiqian Tan*, Binwei Wu†

*Southeast University, Nanjing, China
†Purple Mountain Laboratories, Nanjing, China
Email: t_w_q@foxmail.com, wubinwei163@163.com

Abstract—With the development of 5G, innovative applications requiring bounded transmission delays and zero packet loss emerge, e.g., AR, industrial automation, and smart grid. In this circumstance, time-sensitive networking (TSN) is proposed, which addresses the deterministic transmission in the local area networks. Nevertheless, TSN is essentially a Layer 2 technique, which cannot provide deterministic transmission on a large geographic area. To solve this problem, this paper proposes a hierarchical network for the end-to-end deterministic transmission. In the proposed network, we leverage CQF (i.e., one of the most efficient TSN mechanisms) in the access networks which aggregates the traffic from end-devices. Meanwhile, in the core network, we exploit the DIP (i.e., a well-known deterministic networking mechanism for backbone networks) for long-distance deterministic transmission. We design the cycle alignment mechanism to enable seamless and deterministic transmission among hierarchical networks. A joint schedule is also formulated, which introduces the traffic shaping at the network edge to maximize the network throughput. Experimental simulations show that the proposed network can achieve end-to-end deterministic transmission, even in the highly-load scenarios.

Index Terms—deterministic networking, long-distance end-to-end transmission, DIP, CQF

I. INTRODUCTION

With the development of the Internet and communication networks, many time-sensitive applications are emerging, such as factory automation, connected vehicles and smart grids [1]. Traditional IP networks performing best-effort transmission provide some Quality of Service (QoS) strategies (e.g., DiffServ and congestion control [2]). However, due to microburst traffic [3] existing in networks, these mechanisms can not achieve deterministic transmission, which means zero packet loss, bounded delays and jitters (i.e., delay variation) [4]. Thus, for eliminating micro-burst traffic, IEEE aims to develop a set of standards named Time-Sensitive Networking (TSN) to provide deterministic transmission in layer 2. Due to the high efficiency, Cyclic Queuing Forwarding (CQF) [5] is one of the top choices in TSN [6]. Besides, Credit-Based Shaper [7] and Time-Aware Shaper [8] are also well-known mechanisms in TSN. However, TSN cannot provide long-distance deterministic transmission because of the requests for high precision time synchronization and low propagation delays, which leads to the "information isolated island" among TSN networks. Global management of TSN applications (e.g. teleprotection, wind farm applications, etc. [9]) requests long-distance deterministic transmission among TSN networks.

For achieving large-scale deterministic transmission, the IETF Deterministic Networking (DetNet) group focus on promoting the standardization in layer 3. The standards of DetNet are still under discussion. Some drafts have been proposed, such as Cycle Specified Queuing and Forwarding (CSQF) [10] and Deterministic IP (DIP) [11]. CSQF leverages Segment Routing [12] Identifier to coordinate cyclic transmission times across large-scale networks to offer deterministic transmission. DIP is one of the effective mechanisms in DetNet. The routers performing DIP (named as DIP routers) achieve frequency synchronization. Packets are transported in a slotted fashion by carrying a cycle identifier. After investigating recently proposed deterministic forwarding mechanisms, we find that no researches have been conducted to apply DetNet to achieve long-distance deterministic transmission among TSN networks. In this paper, we propose a hierarchical network for long-distance end-to-end deterministic transmission. CQF technique is used in the access layer, and DIP supports the core layer.

Due to the differences between CQF and DIP, it is not effective to integrate these two mechanisms directly. The first problem is different time synchronization mechanisms in CQF and DIP. Time synchronization with high precision used in most TSN mechanisms is hard to achieve in long-distance deterministic transmission. Thus, DIP only requests frequency synchronization. The second problem is the duration of a cycle may be different in CQF and DIP. These two problems are the barriers to construct correspondences between cycles in CQF switches and DIP routers because cycles in CQF do not correspond to cycles in DIP one by one. However, cycle correspondences are the foundation to achieve deterministic transmission in CQF and DIP.

To tackle the problems mentioned above, this paper establishes the network model of the hierarchical network and proposes a cycle alignment mechanism to establish the correspondences between cycles in CQF and DIP. Based on the cycle alignment, we propose a traffic shaping mechanism to solve the disorder of traffic at the edge of access and core networks to achieve end-to-end joint scheduling across CQF and DIP. The joint scheduling is formulated as integer linear programming to maximize the number of acceptable time-sensitive flows. In the simulations, the scheduling is conducted in a greedy algorithm. The time-sensitive flows (TS flows) with high weight are scheduled preferentially. For each TS

flow, the scheduling chooses the transmission pattern assuming minimum resources.

The rest of this paper is organized as follows. Section II details the deterministic networking technologies used in this paper (i.e., CQF and DIP). In Section III, the network model of the hierarchical network is introduced. Section IV proposes the end-to-end joint scheduling approach. Section V shows the setup and the results of simulations. Finally, Section VI concludes this paper.

II. BACKGROUND

A. Cyclic Queuing Forwarding

IEEE 802.1Qch (CQF) standard [5] proposes a method to synchronize enqueue and dequeue operations, and schedule the transmission in a cyclic fashion. Time in CQF networks is divided into cycles with equal duration d. All CQF switches in the same CQF network are perfectly synchronized. There exist two queues ($q1$ and $q2$) in a CQF switch output port. In cycle c, queue $q1$ is open for dequeuing packets. At the same time, queue $q2$ enqueues packets from upstream. In the next cycle $c+1$, queue $q1$ enqueues packets and queue $q2$ dequeues packets. Due to the cyclic manner for transmission, the packets received in cycle c are scheduled to be retransmitted in cycle $c+1$, and the maximum possible delay experienced in a CQF switch is from the beginning of cycle c to the end of $c+1$, equal to $2d$.

The upper and lower bounds are easily calculated in CQF networks. The number of hops is defined as h. $D_{\max}$ and $D_{\min}$ represent the maximum and minimum latency respectively. The equations to calculate $D_{\max}$ and $D_{\min}$ are as follows:

$$D_{\max} = (h+1) \times d \tag{1}$$

$$D_{\min} = (h-1) \times d \tag{2}$$

B. Deterministic IP

Time in DIP is also divided into cycles with equal duration T. Because perfect time synchronization is hard to achieve in large geographical coverage, DIP only requests frequency synchronization and permits the propagation delay exceeding the cycle duration. DIP leverages correspondences between cycles to achieve deterministic transmission. Fig. 1 shows the DIP transmission among three DIP routers (DR1, DR2, and DR3). Based on the propagation delays between DR1 and DR2, and the one between DR2 and DR3, the following cycle correspondences are constructed: the cycle x in DR1 is mapped to cycle y in DR2, and the cycle y is mapped to cycle z in DR3. These cycle correspondences imply that the packets sent in cycle x will be received in DR2 not later than cycle $y-1$, and retransmitted to DR3 in cycle y in DR2. It is similar between DR2 and DR3.

The packets sent at cycle x from DR1 carry the cycle identifier x. After receiving them, DR2 will check the cycles correspondences table stored in it, and find the correspondence $x \to y$. Then DR2 enqueues these packets to the queue corresponding to cycle y. At cycle y, the queue will open and

dequeue these packets to DR3. Fig. 1 shows the maximum and minimum delay from DR1 to DR3, and the maximum jitter is less than $2T$.

Fig. 1. An example of transmission among DIP routers. The packets sent in cycle x by DR1 will be retransmitted in cycle y by DR2. The transmission cycle in DR3 is z.

III. HIERARCHICAL NETWORK

We propose a hierarchical network to achieve long-distance deterministic transmission among TSN networks, as shown in Fig. 2. The access networks perform CQF and the core network forwards packets following DIP. The following subsections will demonstrate the transmission mechanisms in the hierarchical network systematically.

Fig. 2. The hierarchical network. In the access networks, CQF guarantees deterministic transmission in the local area. In the core network, DIP provides long-distance deterministic transmission. Hosts and CQF (edge) switches perform CQF, while DIP (edge) routers perform DIP. Especially, CQF edge switches and DIP edge routers perform traffic shaping and the cycle alignment across CQF and DIP.

A. Network model

The whole network is represented as $G = \{V, E\}$. V is the set of network devices, and E is the set of links. Network devices contain source hosts, destination hosts, CQF switches, CQF edge switches, DIP routers, and DIP edge routers, i.e., $V = \{V_{\text{src}}, V_{\text{dest}}, V_{\text{cqf}}, V_{\text{cqf}}^{\text{edge}}, V_{\text{dip}}, V_{\text{dip}}^{\text{edge}}\}$. The processing delays in all devices are ignored.

For a link $e \in E$, source v_1 and end v_2 uniquely identify the link, i.e., $e = (v_1, v_2)$. A link e corresponds to a propagation delay Δ_e. The bandwidth of link e is denoted as BW_e.

A path from the source $v_0 \in V_{\text{src}}$ to the destination $v_{|p|} \in V_{\text{dest}}$ is denoted by $p = (v_0, v_1, \cdots, v_{|p|})$, where $|p|-1$ is the number of intermediate nodes along path p. A set P contains all possible paths from V_{src} to V_{dest}, i.e., $p \in P$.

Time-sensitive applications located in source hosts generate time-sensitive flows (TS flows). F represents the set of TS flows. Each TS flow $f \in F$ is periodic, and the period is Δ_f.

A TS flow f emits packets in size of ω_f bits during every Δ_f. Moreover, the source and destination of a flow f are deterministic. The maximum acceptable end-to-end delay of f is Δ_f^{e2e}.

B. Time models in CQF and DIP

Time models in CQF and DIP are different. We define every access and core network as a time-domain named CQF-D and DIP-D respectively, as shown in Fig. 3. For uniformly scheduling, we introduce *hypercycle*, which contains multiple cycles in CQF and DIP.

CQF-D: CQF-D contains source/destination hosts, CQF edge switches and CQF switches. Fig. 3 shows that, in CQF-D, time is divided into time cycles with the same duration Δ_{cqf}. A hypercycle contains N_{cqf} consecutive cycles. The length of a hypercycle is $\Delta_{\mathrm{hc}} = N_{\mathrm{cqf}}\Delta_{\mathrm{cqf}}$. All devices in CQF-D achieve perfect time synchronization, which means that the start time of a cycle (also a hypercycle) in all devices is the same.

DIP-D: DIP-D covers DIP routers and DIP edge routers. Similar to CQF-D, time in DIP-D is divided into time cycles with the same duration Δ_{dip}. A hypercycle in DIP-D contains N_{dip} consecutive DIP cycles. Thus, the length of a hypercycle is $\Delta_{\mathrm{hc}} = N_{\mathrm{dip}}\Delta_{\mathrm{dip}}$. As shown in Fig. 3, DIP-D only requires frequency synchronization. For a pair of adjacent devices (A, B) in DIP-D, there may exist a constant gap between the start time of cycles in A and B. The gap is defined as the offset of cycles, $\Delta_{\mathrm{co}}^{A,B}$. $\Delta_{\mathrm{co}}^{A,B}$ is the difference between the start time of cycles in A and B, and $\Delta_{\mathrm{co}}^{A,B} \geq 0$.

Hypercycle: For allocating deterministic cycles to each TS flow, the length of a hypercycle should satisfy:

$$\Delta_{\mathrm{hc}} = N_{\mathrm{cqf}}\Delta_{\mathrm{cqf}} = N_{\mathrm{dip}}\Delta\mathrm{dip} = N_F\Delta_f \quad (3)$$

where $N_{\mathrm{cqf}}, N_{\mathrm{dip}}, N_F \in \mathbb{Z}^+$, N_{cqf} and N_{dip} are greater than the number of queues in output ports of CQF switches and DIP routers respectively.

Fig. 3 illustrates hypercycles in CQF-D and DIP-D. The offset of hypercycle between a pair of adjacent devices (A, B) is defined as $\Delta_{\mathrm{hco}}^{A,B}$. $\Delta_{\mathrm{hco}}^{A,B}$ is the difference between the start time of hypercycles in A and B, and $\Delta_{\mathrm{hco}}^{A,B} \geq 0$.

Fig. 3. Time models in the hierarchical network. In CQF-D, all devices achieve perfect time synchronization, while DIP-D only requires frequency synchronization. $N_{\mathrm{cqf}} = 3$ and $N_{\mathrm{dip}} = 2$ in this figure.

C. Cycle alignment

We design a cycle alignment mechanism to construct correspondences between cycles in different devices. The cycle alignment is defined as follows: if packets sent in cycle a in a node A will all be received no later than cycle b in the downstream node B, we define a cycle alignment relationship as $\Phi_{(A,B)}(a) = b$.

Alignment in CQF-D: Inside CQF-D, packets transmitted by a node during cycle i must be received by the downstream node during the same cycle and retransmitted in cycle $i + 1$. Thus, the cycle alignment in CQF-D is $\Phi(i) = i$.

Alignment in DIP-D: Fig. 4 illustrates a cycle alignment in DIP-D. There exist adjacent DIP devices node A and B with the same cycle length Δ_{dip}. The length of a hypercycle is $\Delta_{\mathrm{hc}} = 3\Delta_{\mathrm{dip}}$ (i.e., $N_{\mathrm{dip}} = 3$). Cycle 0 in node A is mapped to cycle 2 in node B (i.e., $\Phi_{(A,B)}(0) = 2$).

Fig. 4. The cycle alignment in DIP-D. All packets sent in node A's cycle 0 will be received no later than node B's cycle 2. The cycle alignment $\Phi_{(A,B)}(0) = 2$ is established.

The cycle alignment $\Phi_{(A,B)}(0) = 1$ can be calculated based on the duration of cycles in DIP-D Δ_{dip}, the length of hypercycle Δ_{hc}, the offset of hypercycles $\Delta_{\mathrm{hco}}^{A,B}$, and the propagation delay $\Delta_{(A,B)}$. $\Phi_{(A,B)}(\cdot)$ is defined as

$$\Phi_{(A,B)}(x) = \lceil \varphi_{(A,B)}(x) - 1 \rceil \bmod N_{\mathrm{dip}} \quad (4)$$

where

$$\varphi_{(A,B)}(x) = \frac{((x+1) \bmod N_{\mathrm{dip}}) \cdot \Delta_{\mathrm{dip}} + \Delta_{(A,B)} + \Delta_{\mathrm{hco}}^{A,B}}{\Delta_{\mathrm{dip}}} \quad (5)$$

Note that the index x loops from 0 to $N_{\mathrm{dip}} - 1$.

Alignment across CQF-D & DIP-D: Fig. 5 illustrates an established cycle alignment from node C to its adjacent node D. Node C and D are located in different time domains. The length of cycles in node C and D are Δ^C and Δ^D respectively. The length of hypercycle is $\Delta_{\mathrm{hc}} = 5\Delta^C = 4\Delta^D$, and the offset of hypercycle is $\Delta_{\mathrm{hco}}^{C,D}$. Cycle 0 in node C is mapped to cycle 3 in node D, i.e., $\Phi_{(C,D)}(1) = 3$.

Fig. 5. Cycle alignment across CQF-D and DIP-D. Nodes C and D are in different time domains. Cycle 1 in node C is mapped to cycle 3 in node D.

Due to the inequality of cycle durations in CQF-D and DIP-D, $\Delta_{\mathrm{co}}^{C,D}$ is not constant. However, the offset of hypercycles

between CQF-D and DIP-D is a constant. Taking Fig. 5 as an example, the cycle alignment can be calculated based on $\Delta_{\text{hco}}^{C,D}$, Δ^C, Δ^D, Δ_{hc}, and $\Delta_{(C,D)}$. The cycle alignment across CQF-D and DIP-D, $\Phi_{(C,D)}(\cdot)$, is defined as

$$\Phi_{(C,D)}(x) = \lceil \varphi_{(C,D)}(x) - 1 \rceil \bmod \frac{\Delta_{\text{hc}}}{\Delta^D} \qquad (6)$$

where

$$\varphi_{(C,D)}(x) = \frac{((x+1) \bmod \frac{\Delta_{\text{hc}}}{\Delta^C}) \cdot \Delta^C + \Delta_{(C,D)} + \Delta_{\text{hco}}^{C,D}}{\Delta^D} \qquad (7)$$

Note that the index x loops from 0.

D. Traffic shaping

The hierarchical network leverages cycle shifting to achieve traffic shaping. A cycle shift represents the additional cycles for retransmission based on the result of $\Phi(\cdot)$. Cycle shifts in sources, CQF switches/DIP routers, and CQF edge switches/DIP edge routers are different respectively.

Cycle shifting in sources: A cycle shift r_f^A is corresponding to the TS flow f and the source A. If packets of f arrive at node A in cycle a, and a hypercycle contains N cycles, these packets will be retransmitted in cycle $\left((a + r_f^A) \bmod N \right)$, where $r_f^A \in [0, N-1]$. In Fig. 6, packets of TS flows f_2 all arrive at cycle 0 from upper layer. Due to $r_{f_2}^A = 3$, f_2 will be sent out in cycle $((0 + 3) \bmod 5)$.

Fig. 6. Cycle shifting in sources. When too many TS flows (f_1, f_2, and f_3) arrive in a cycle (cycle 0), cycle shifts are introduced for traffic shaping. $r_{f_1}^A = 0$ represents that f_1 will be retransmitted in cycle 0. $r_{f_2}^A = 3$ indicates the retransmitting cycle of f_2 is $(0+3) \bmod 5 = 3$. f_3 will be retransmitted in cycle $((0 + 2) \bmod 5 = 2)$.

Cycle shifting in CQF switches/DIP routers: In the device $v \in V_{\text{cqf}} \cup V_{\text{dip}}$, the cycle shift is always 1. Assume that there is a pair of intermediate nodes (A, B) with $\Phi_{(A,B)}(a) = b$. TS flow f is sent in node A's cycle a. A hypercycle contains N cycles in node B. Because $r_f^B = 1$, flow f is retransmitted in cycle $(b+1) \bmod N$.

Cycle shifting in edge switches/routers: The cycle shifting in edge devices is illustrated in Fig. 7. Node C and node D are in different time domains. A hypercycle contains $N = 4$ cycles of node D. There exists $\Phi_{(C,D)}(1) = 3$. The cycle shift of TS flow f_1 in node D is $r_{f_1}^D = 2$, so packets of f_1 will be retransmitted in cycle $(\Phi_{(C,D)}(1) + r_{f_1}^D) \bmod N = 1$. The range of $r_{f_1}^D$ is $[0, N-1]$.

IV. JOINT SCHEDULING

A. Decision variables

There are a series of decision variables to schedule TS flows in F. These variables can be classified into three types: admission control, path selection, and cycle shifting.

Fig. 7. Cycle shifting in edge switches or routers. TS flow f_1 is sent in cycle 1 in node C, and cycle 0 is mapped to cycle 3 in node D (i.e., $\Phi_{(C,D)}(0) = 3$). A hypercycle contains 4 cycles in node D. Since $r_{f_1}^D = 2$, f_1 will be retransmitted in cycle $(3 + 2) \bmod 4 = 1$ in node D.

Admission control: For a TS flow $f \in F$, x^f is defined to describe whether f should be accepted ($x^f = 1$) or not ($x^f = 0$). Let $\mathbf{x} = \{x^f | f \in F\}$.

Path selection: $p^f = (v_0, v_1, \cdots, v_{|p^f|})$ is defined as the scheduled path for TS flow f. v_0 is the source and $v_{|p^f|}$ is the destination. Let $P = \{p^f | f \in F\}$.

Cycle shifting: The cycle shifting for TS flow f is defined as $\mathbf{r}_{p^f} = (r_f^{v_0}, r_f^{v_1}, \cdots, r_f^{v_{|p^f|}})$, where $r_f^{v_i}$ is the cycle shifts for f at node v_i. Let $\mathbf{r} = \{\mathbf{r}_{p^f} | p^f \in P\}$.

B. Constraints

There exist two constraints in joint scheduling: constraints on end-to-end delays and resource capacity.

Constraints on end-to-end delay: The end-to-end delay of a TS flow f depends on the scheduled path p^f and the cycle shifting $\mathbf{r}_{p^f}$. Thus, the end-to-end delay of f is described as $\Delta(p^f, \mathbf{r}_{p^f})$, and satisfies $\Delta(p^f, \mathbf{r}_{p^f}) \leq \Delta_f^{\text{e2e}}$ (Δ_f^{e2e} is the maximum acceptable end-to-end delay of f).

For a link $e_i = (v_i, v_{i+1})$ in path p^f, we define the delay on e_i as $\Delta_{e_i}^f$, which contains the maximum time staying at v_i and the propagation delay on e_i (i.e., Δ_{e_i}). Thus, the upper bound of $\Delta_{e_i}^f$ can be defined as

$$U(\Delta_{e_i}^f) = (r_f^{v_i} + 1)\Delta^{v_i} + \Delta_{e_i} \qquad (8)$$

where Δ^{v_i} is the duration of a cycle in node v_i.

The upper bound of path p^f's end-to-end delay can be calculated by

$$U(\Delta(p^f, \mathbf{r}_{p^f})) = \sum_{i=0}^{|p^f|-1} U(\Delta_{e_i}^f) \qquad (9)$$

For every TS flow $f \in F$, the constraint of end-to-end delay can be described as

$$U(\Delta(p^f, \mathbf{r}_{p^f})) \leq \Delta_f^{\text{e2e}} \qquad (10)$$

Constraints on resource capacity: The resource in the hierarchical network is cycles in every node. For a cycle c in node v_i on link $e = (v_i, v_{i+1})$, if a set of TS flows F_c is assigned to it, the total bits of F_c should not exceed the transmission capacity of c (i.e., formula (11)).

$$\sum_{f \in F_c} \omega_f^c \leq \Delta^{v_i} \cdot BW_e \qquad (11)$$

where ω_f^c is the total bits of f transmitted in cycle c, Δ^{v_i} is the duration of a cycle in node v_i, and BW_e is the bandwidth of link e.

C. Objective function

The target of the joint scheduling is to accept as many $f \in F$ as possible. The priority of every TS flow may be different. A weight value $v^f \in (0, 1]$ represents the priority of TS flow f. The problem can be formulated as an integer programming (IP):

$$\max_{\mathbf{x}, P, \mathbf{r}} \sum_{f \in F} v^f x^f \tag{12a}$$

$$s.t. \quad v^f \in (0, 1], x^f \in \{0, 1\} \tag{12b}$$

$$(10), (11) \tag{12c}$$

V. SIMULATION

The hierarchical network shown in Fig. 2 is constructed in the simulation. The topology of the core network containing 15 DIP routers (with 10 μs as Δ_{dip}) is established based on the network model Atlanta [13]. The lengths of links in the core network are 30 km to simulate long-distance transmission. The bandwidth of ports on DIP routers is 10 Gbps. We deploy 10 access networks, and every access network contains 2 CQF switches (with 25 μs as Δ_{cqf}). The bandwidth of ports on CQF switches is 1 Gbps [14]. TS flows are connected to CQF switches. Packets of TS flows have to pass through the access network containing the source hosts, the core network, and the access network connected by the destination hosts.

Each access network contains 200 TS flows. The data rate of each TS flow is 500 kbps. Thus, the size of packets is 500 bit, and the interval of packets is 1ms. Besides, the maximum acceptable end-to-end delay of these TS flows is 1 ms. The schedule of packets is conducted in a greedy algorithm. In the simulation, the best-effort interference flows with a packet size of 1500 Byte are injected into the converged network. For verifying the effectiveness of the proposed joint scheduling in this paper, two experiments are conducted: **(1)** In different levels of interference flows, we compare the end-to-end delays of jointly scheduled flows and best-effort transmitted flows to verify the capability to providing bounded delays and jitters; **(2)** To prove the effectiveness of traffic shaping and path selection, the comparison of flows admission in three strategies (i.e., with traffic shaping and path selection, without traffic shaping, and without path selection) is performed.

Tabel I and Fig. 8 show the result of experiment 1, and prove the effectiveness of guaranteeing bounded end-to-end delays and jitters. We randomly choose a TS flow in an access network to observe. The TS flow is transmitted in best-effort and the proposed joint scheduling. The total throughput of all 200 TS flows in the access network is 100 Mbps. With the increase of intensity (i.e., throughput) of interference flows, the end-to-end delays using best-effort (BE) grow significantly. In the light-load scenario injecting interference flows with throughput 229 Mbps, the maximum

delay is 980 μs, and the jitter is 375 μs. In the medium-load scenario with 534 Mbps interference flows, the average delays are higher compared with light-load scenarios. The maximum delay is up to 1173 μs, and the jitter is up to 567 μs. Besides, there exist 26 TS flows having end-to-end delays beyond the maximum acceptable delay (1 ms). The delays using the joint scheduling is a constant 801 μs with zero jitters. Delays are not affected by the intensity of interference flows. In an extremely light-load scenario (i.e., the throughput of interference flows is 131Mbps), the maximum delay of best-effort transmission is less than the one in deterministic transmission. This phenomenon results from the extra delays introduced from cyclic transmission fashion in deterministic networking and cycle shifting. For guaranteeing bounded end-to-end delays and jitters, the extra delays are acceptable, and the delays in deterministic transmission are significantly less than the maximum delay in light-load, medium-load, and high-load scenarios.

Fig. 8. The cumulative distribution of end-to-end delays in different intensities of interference flows. With the increase of interference flows intensity, the delays and jitters of the TS flow transported in best-effort (BE) increase sharply. However, the joint scheduling can provide a constant end-to-end delay in 801 μs, and zero jitters.

Fig. 9 illustrates the result of experiment 2. For verifying the proposed joint scheduling can raise the throughput of TS flows, we deploy a different number of TS flows that are waiting for scheduling in every access network. Obviously, the total throughput of these TS flows is less than the capacity of links. Using the joint scheduling with traffic shaping and path selection, the number of TS flows rejected for deterministic transmission starts increasing when every access network contains 1725 TS flows. Without path selection (i.e., all flows choose the path with a minimum number of hops), the number of rejected flows starts increasing when 1700 TS flows wait for scheduling in an access network. Moreover, with the same number of TS flows, the number of rejected flows is higher than the one using joint scheduling. Without traffic shaping, the scheduling starts to reject TS flows when there exist 1200 TS flows. The number of rejected flows starts to increase sharply when an access network contains 1500 TS flows. The rejection is severer than the joint scheduling and the scheduling without path selection. The result shows the traffic shaping and path selection can improve the throughput of TS flows, and

TABLE I
THE STATISTICS OF END-TO-END DELAYS IN BE AND JOINT SCHEDULING

Intensity of interference flows (Mbps)	Best-Effort				Joint scheduling			
	mean(μs)	max(μs)	jitter(μs)	beyond deadline	mean(μs)	max(μs)	jitter(μs)	beyond deadline
130.848	612	750	144	0	801	801	0	0
228.984	621	980	375	0	801	801	0	0
534.296	692	1173	567	26	801	801	0	0
697.856	847	2031	1426	402	801	801	0	0

the traffic shaping is the more important factor to improve the throughput. Due to traffic shaping can deploy flows to different transmission cycles, the conflict for transmission resources is reduced. However, because of the tree topology in access networks, all flows in the same access network have to pass through common CQF switches, which attenuate the influence of path selection. The increase of these three curves is not linear, because the newly added TS flows may have a high priority to be scheduled preferentially, and this will make the TS flows accepted formerly being rejected. Thus, the increase is not linear.

Fig. 9. The number of rejected TS flows in different intensities of TS flows. Compared with scheduling without traffic shaping or path selection, the proposed joint scheduling can decrease the number of rejected TS flows. Especially compared with scheduling without traffic shaping, the proposed scheduling can increase network throughput.

VI. CONCLUSION

To empower the end-to-end deterministic transmission, this paper proposes a hierarchical network that consists of the access networks and the core networks. In the access network, we use the CQF technique to ensure the deterministic transmission during the aggregation of the traffic from hosts. In the core network, DIP is exploited for the long-distance deterministic transmission over the backbone links. The end-to-end deterministic transmissions are realized by the cycle alignment and traffic shaping at the network edge. Moreover, a joint scheduling approach is also formulated to improve the network throughput. Simulations results based on the real-world networks from sndlib show that the proposed network can achieve a deterministic transmission even in the high-load scenarios.

REFERENCES

[1] E. Grossman, "Deterministic Networking Use Cases," RFC 8578, May 2019, [Online]. Available: https://rfc-editor.org/rfc/rfc8578.txt

[2] L. Chettri and R. Bera, "A Comprehensive Survey on Internet of Things (IOT) toward 5G Wireless Systems," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 16–32, 2019.

[3] D. Shan, F. Ren, P. Cheng, R. Shu, and C. Guo, "Observing and mitigating micro-burst traffic in data center networks," *IEEE/ACM Transactions on Networking*, vol. 28, no. 1, pp. 98–111, IEEE, 2019.

[4] A. Charny and J.-Y. Le Boudec, "Delay Bounds in a Network with Aggregate Scheduling," in *International Workshop on Quality of Future Internet Services*, pp. 1–13, Springer, 2000.

[5] "IEEE Standard for Local and metropolitan area networks–Bridges and Bridged Networks–Amendment 29: Cyclic Queuing and Forwarding," *IEEE 802.1Qch-2017 (Amendment to IEEE Std 802.1Q-2014 as amended by IEEE Std 802.1Qca-2015, IEEE Std 802.1Qcd(TM)-2015, IEEE Std 802.1Q-2014/Cor 1-2015, IEEE Std 802.1Qbv-2015, IEEE Std 802.1Qbu-2016, IEEE Std 802.1Qbz-2016, and IEEE Std 802.1Qci-2017)*, pp. 1–30, Jun. 2017.

[6] A. Nasrallah, V. Balasubramanian, A. Thyagaturu, M. Reisslein, and H. ElBakoury, "TSN Algorithms for Large Scale Networks: A Survey and Conceptual Comparison," *arXiv preprint arXiv:1905.08478*, 2019.

[7] "IEEE Standard for Local and Metropolitan Area Networks – Virtual Bridged Local Area Networks Amendment 12 Forwarding and Queuing Enhancements for Time-Sensitive Streams," *IEEE Std 802.1Qav-2009 (Amendment to IEEE Std 802.1Q-2005)*, pp. C1–72, Jan. 2009.

[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- as amended by IEEE Std 802.1Qca-2015, IEEE Std 802.1Qcd-2015, and IEEE Std 802.1Q-/Cor 1-2015)*, pp. 1–57, Mar. 2016.

[9] R. Salazar, T. Godfrey, N. Finn, C. Powell, B. Rolfe, and M. Seewald, "White Paper-Utility Applications of Time Sensitive Networking White Paper," IEEE, 2019.

[10] M. Chen, X. Geng, and Z. Li, "Segment Routing (SR)-based Bounded Latency," *Internet Engineering Task Force, Internet-Draft draft-chendetnet-sr-based-bounded-latency-00*, 2018.

[11] L. Qiang, B. Liu, D. Yu, and C. Wang, "Large-scale deterministic network forwarding technology," *Telecommunications Science*, vol. 35, no. 9, pp. 12–19, 2019.

[12] C. Filsfils, S. Previdi, L. Ginsberg, B. Decraene, S. Litkowski, and R. Shakir, "Segment Routing Architecture." RFC 8402, Jul. 2018. [Online]. Available: https://rfc-editor.org/rfc/rfc8402.txt.

[13] S. Orlowski, M. Pióro, A. Tomaszewski, and R. Wessäly, "SNDlib 1.0–Survivable Network Design Library," *Networks: An International Journal*, vol. 55, no. 3, pp. 276–286, 2010.

[14] J. Yan, W. Quan, X. Jiang, and Z. Sun, "Injection Time Planning: Making CQF Practical in Time-Sensitive Networking," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, pp. 616–625, IEEE, 2020.