

# 一种应用于大规模时间敏感网络的混合流量调度机制

李妍珏<sup>1</sup>, 韩文璇<sup>1</sup>, 朱海龙<sup>2</sup>, 尹长川<sup>1</sup>

(1. 北京邮电大学先进信息网络北京实验室, 北京 100876; 2. 北京邮电大学网络与交换技术国家重点实验室, 北京 100876)

**摘要:** 针对仅能保证设备间频率同步的大规模时间敏感网络 (TSN, time-sensitive networking) 场景, 提出一种时间感知整形 (TAS, time-aware shaper) 结合指定周期的排队和转发 (CSQF, cycle specified queuing and forwarding) 的混合流量调度机制。首先, 研究位于不同时间域内的两个相邻节点间实现周期循环映射对齐的方案; 其次, 结合段路由技术, 提出一种基于 CSQF 机制的联合路由和调度的启发式算法, 完成对大带宽流量的资源分配。实验结果表明, 相较于现有方案, 所提方案大幅度提升了系统资源利用率及调度成功率, 实现了广域网场景下的 TSN 高效流量调度。

**关键词:** 时间敏感网络; 时间感知整形; 循环排队转发; 指定周期的排队和转发; 流量调度

**中图分类号:** TP393

**文献标志码:** A

**doi:** 10.11959/j.issn.2096-3750.2023.00367

## A hybrid traffic scheduling mechanism applied to large scale time-sensitive networking

LI Yanjue<sup>1</sup>, HAN Wenxuan<sup>1</sup>, ZHU Hailong<sup>2</sup>, YIN Changchuan<sup>1</sup>

1. Beijing Laboratory of Advanced Information Networks, Beijing University of Posts and Telecommunications, Beijing 100876, China

2. State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China

**Abstract:** Aiming at large-scale time-sensitive networking (TSN) scenarios that can only ensure frequency synchronization between devices, a hybrid traffic scheduling mechanism was proposed based on time-aware shaper (TAS) combined with cycle specified queuing and forwarding (CSQF). Firstly, a scheme was investigated for achieving periodic cyclic mapping alignment between two adjacent nodes located in different time domains. Secondly, combining segment routing technology, a heuristic algorithm based on CSQF mechanism for joint routing and scheduling was proposed to complete resource allocation for large bandwidth traffic. The experimental results show that the proposed scheme significantly improves the system resource utilization and scheduling success rate compared to existing research results, achieving efficient TSN traffic scheduling in wide area network scenarios.

**Key words:** time-sensitive networking, time-aware shaper, cyclic queuing and forwarding, cycle specified queuing and forwarding, traffic scheduling

## 0 引言

以太网技术自诞生以来, 由于具有简单灵活、可扩展性强、成本较低等显著优势, 被广泛应用, 相关技术也已十分成熟, 如工业自动化网络中

的 ProfiNet<sup>[1]</sup>、Powerlink<sup>[2]</sup>、EtherCAT<sup>[3]</sup>等主流现场总线技术。但随着工业 4.0 时代的到来, 工业场景中对业务流的时间敏感度要求不断提升, 传统以太网技术越来越难以满足需求, 因此, 如何改进现有技术, 使其能够准确控制数据流的端到端时延, 成

收稿日期: 2023-05-29; 修回日期: 2023-08-29

通信作者: 李妍珏, lyj\_ycyc@163.com

基金项目: 国家重点研发计划 (No.2020YFB1805301)

**Foundation Item:** The National Key Research and Development Program of China (No.2020YFB1805301)

为学术界和工业界的研究重点。时间敏感网络(TSN, time-sensitive networking)是一种新兴的工业通信技术,前身为 IEEE 音视频桥接(AVB, audio video bridging)<sup>[4-5]</sup>技术,TSN 任务组致力于研究将以太网的实时和安全关键机制标准化<sup>[6]</sup>。TSN 可向后兼容标准以太网,从而获得低时延抖动的高鲁棒通信链路。同时,相较于标准以太网,TSN 通过时间同步机制<sup>[7]</sup>、流量整形机制、门控阵列机制、流预留等技术,支持时间敏感流与非时间敏感流、周期流与非周期流在网络中混合传输,能够为数据包提供端到端确定性、低时延、低抖动传输保障。

TSN 中的关键技术大体可分为时钟同步、流量整形与调度、网络与用户配置<sup>[8]</sup>3类,通过综合运用3类技术,保证网络的低时延、高可靠性等<sup>[9]</sup>。其中,流量调度技术会直接影响网络不同业务的服务质量(QoS, quality of service),对数据传输的实时性和确定性起着决定性作用,是目前时间敏感网络研究中的关键技术。现有研究表明,TSN 流量调度问题是 NP 完全问题<sup>[10-11]</sup>,由于实际工业场景中往往有多种不同优先级流量混合传输,采用单一调度机制很难同时兼顾不同流的流特性及传输需求。因此,现有研究开始关注混合流量调度机制。例如,文献[12]提出时间感知与循环排队(TACQ, time-aware cyclic-queuing)机制,限制时间感知整形(TAS, time-aware shaper)队列与循环排队转发(CQF, cyclic and queuing forwarding)队列不可同时开启,以实现时间敏感流的零抖动,并降低时间敏感流对其他周期流的影响,但 CQF 的双关门机制会带来缓存资源和传输带宽的浪费。文献[13]对 TAS 结合 CQF 的混合传输框架进行了探究,提出一种参数选择方法来确定网络的循环周期和时间调度单元,以减少低时间敏感度流的平均时延,但该方案只给出了高时间敏感度流的调度方案,且在确定时间调度单元时,对于 CQF 机制带给网络的约束考虑不够全面。此外,目前尚无研究关注指定周期的排队和转发(CSQF, cycle specified queuing and forwarding)机制与其他机制的结合,相较于 CQF 机制,CSQF 降低了对精准时间同步的要求,可应用范围更加广泛。

作者团队在文献[14]提出一种 TAS 结合 CQF 的域内混合流量调度机制(HSTC, hybrid traffic scheduling combining TAS and CQF),通过确定网络最小调度时隙并基于该最小时隙调整计划流量(ST,

scheduled traffic)的采样周期,以降低 ST 对发送带宽的占用,并针对流预留(SR, stream reservation)流量,通过奇偶映射的流映射策略和流偏移注入规划的算法,对 SR 流注入交换机的时间进行偏移调整,最终通过提高系统资源利用率提升网络调度成功率,实验结果证明,该混合调度机制显著提高了网络可调度性,实现了 TSN 混合流的高效调度,但该方案存在以下局限性。

1) 上下游节点需要在同时隙内完成数据包的发送和接收。为保证传输过程中每跳时延的确定性,CQF 机制要求上游节点在某时隙内发送的所有数据包,都必须在同一时隙内被下游节点接收,随着网络规模增大,传输链路增长导致时隙间隔也增大,时延抖动难以保障,且帧间间隔小于时隙间隔容易引起累积流量突发的问题。

2) CQF 机制要求上下游节点间能够保证严格精准的时间同步,从而确保所有节点的传输周期对齐,此外,CQF 机制没有容错能力,如果报文由于链路时延变化或者其他原因,在下游节点的错误周期内被接收,那全局的流调度都会因此受到影响,从而无法保证端到端的时延确定性。

针对上述现有研究存在的不足,本文提出一种 TAS 结合 CSQF 的混合流量调度机制(HSTCS, hybrid traffic scheduling combining TAS and CSQF),推导相邻节点间存在周期对齐偏差时的周期映射关系,并提出一种从预防链阻塞、均衡交换机内时隙负载、启发式算法性能优化等多方面进行综合考虑的联合路由及流量调度的算法,实验结果表明,相较于现有方案,所提方案大幅度提升了系统资源利用率及调度成功率。本文的主要贡献如下。

1) 针对域内时间同步、域间频率同步的广域网场景,提出一种 TAS 结合 CSQF 的混合流量调度机制,通过降低对精准时间同步的要求,使 TSN 相关技术可运用于大规模网络。

2) 改进文献[14]提出的确定网络最小时隙方案,确定时间调度单位,并减少 ST 的带宽占用。

3) 基于最小调度时隙,在仅能保证频率同步的条件下,推导位于不同时间域内的两个相邻节点间实现周期循环映射对齐的方案。

4) 结合段路由技术,提出一种基于 CSQF 机制的联合路由和调度的启发式算法,完成对 SR 流的资源分配。

### 1 背景知识

目前，针对 TSN 的流量调度问题，已有多种较为成熟的技术解决方案，本文主要研究将 TAS 机制与 CSQF 机制两种现有方案相结合的混合流量调度机制。

#### 1.1 TAS 机制

在实际的工业网络中，一些关键信息流（如控制流、时间同步流等）对传输的时延和抖动有极高要求，需要有特定的调度机制来保证在可预测的确定时间内到达<sup>[15]</sup>，为解决该问题，TSN 任务组提出 IEEE 802.1Qbv 标准<sup>[16]</sup>。IEEE 802.1Qbv 标准定义了 TAS 机制，如图 1 所示。在原有交换机输出端口对应的每个出口队列后添加传输门，

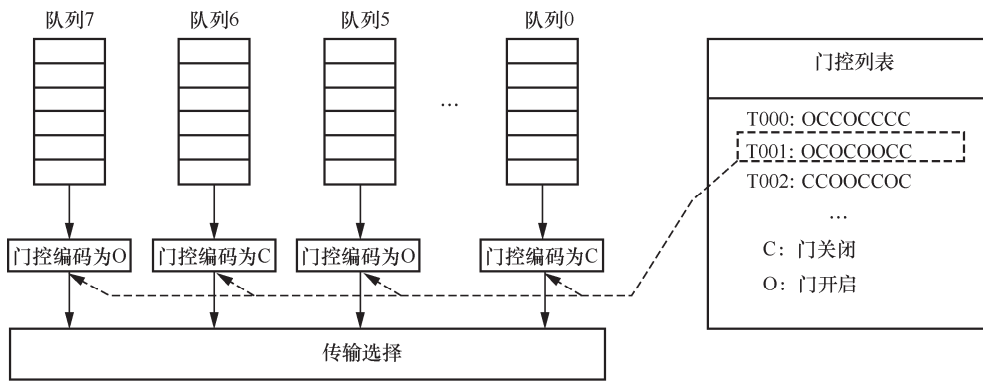


图 1 TAS 机制

#### 1.2 CSQF 机制

CSQF 机制<sup>[18]</sup>是基于 IEEE 802.1 Qch<sup>[19]</sup>中所提出的 CQF 机制的进一步改进，CSQF 机制控制输出端口前的多个队列，所控制队列分为发送队列 (SQ, send queue)、接收队列 (RQ, receive queue)、容忍队列 (TQ, tolerating queue) 3 种类型。在任意时隙内，有且仅有一个队列为 SQ，其在该时隙内向输出端口发送缓存区内数据包；同样地，任意时隙，有且仅有一个队列为 RQ，其接收在该时隙内到达的上游节点发送的数据包并进行缓存；但在同一时隙内，可以有多个 TQ，这些队列可以接收超过 RQ 缓存区大小的数据包，或未按预期到达的突发数据流，避免 RQ 缓存区溢出导致丢包，同时，TQ 也可以接收在该时隙内到达但不在下一时隙发送的数据包，实现逐跳发送时隙偏移。

CSQF 机制如图 2 所示，在第 1 个时隙内，队列 1 进行数据发送，为 SQ，队列 2 为 RQ，接收在第 2 个时隙内传输的上游数据包，队列 3 为 TQ，接收突发数据流或偏移数据流；在第 2 个时隙，队

当传输门为开启状态时，对应缓存队列内的数据帧才有可能获得传输机会，保证每次传输不会受到其他数据流的干扰<sup>[17]</sup>。每个输出端口前的一组缓存队列通过门控列表 (GCL, gate control list) 编码进行控制。门控编码为“O”时，传输门开启，可以发送数据；门控编码为“C”时，传输门关闭，不可以发送数据。多个缓存队列的传输门同时打开时，位于输出端口前的优先级选择器会根据队列优先级，由高至低传输数据包，门控列表周期性控制队列门控。其中，门控列表存储的编码值可以通过人工手动配置，或通过集中控制器进行计算后直接下发至对应交换机。门控列表基于网络的最小调度时隙进行控制，最小调度时隙的大小不固定，根据实际需要设定。

列类型交替，执行该分类下的队列功能，队列状态循环周期为 3 个网络时隙。

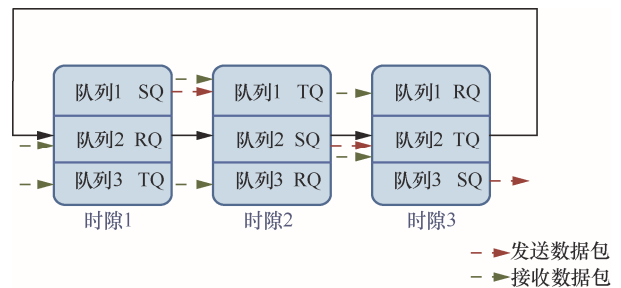


图 2 CSQF 机制

CSQF 机制将段路由标识符列表附加到分组，即将业务流在传输路径上的各个节点中所选择的时隙号以列表的形式显式存储在报文中，当流经某个转发节点时，就将报文中注明的该节点的时隙号作为入队的标识，CSQF 结合段路由机制如图 3 所示，从而对时隙的规划更加灵活，可以提供有界时延和无损分组服务传递。

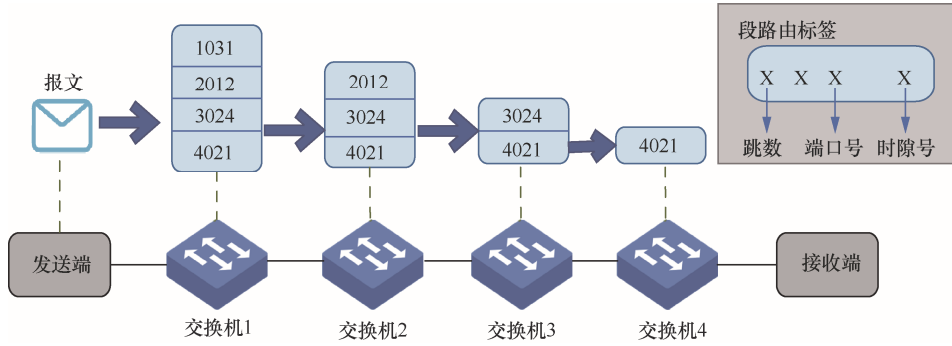


图 3 CSQF 结合段路由机制

相较于 CQF 机制，CSQF 机制调度的 SR 流的可调整参数更多。以 CSQF 机制控制 3 个出口队列为例，假设交换机单队列的最大缓存为 4 MTU，若不调整 SR 流的注入时间，源节点偏移注入示意图如图 4 所示，由于交换机 1 在 Cycle1 时隙内的缓存区已满，此时发送端的数据包到达将导致缓存溢出，数据包出现丢包，传输的确定性被破坏。而源节点若将数据包推迟一个时间单位后发送，即可被下游交换机成功缓存，因此，与 CQF 机制类似，CSQF 可通过源节点偏移，避免缓存溢出导致的调度失败。

此外，逐跳队列偏移示意图如图 5 所示，发送端的数据包成功到达交换机 1，经过缓存后正常发送至交换机 2，但此时交换机 2 的 Cycle1 时隙不足

以缓存交换机 1 发送的数据包，同样会发生缓存溢出，导致丢包。由于 CSQF 机制结合了分段路由技术，可逐跳指定时隙号，若将第二跳交换机 2 处的安全标识符 (SID, security identifier) 标签指定为进入 Cycle2 时隙对应的缓存队列，即在该跳进行队列偏移，即可缓存成功。因此，相较于 CQF 机制，CSQF 机制还可逐跳指定队列偏移，从而更加充分利用系统资源，提高网络调度成功率。

## 2 系统模型

### 2.1 交换机结构设计

本文所设计的交换机结构如图 6 所示。假设交换机共有  $w$  个输入端口，各输出端口前有一组共 8 个缓存队列。IEEE 802.1Q 标准中提出虚拟局

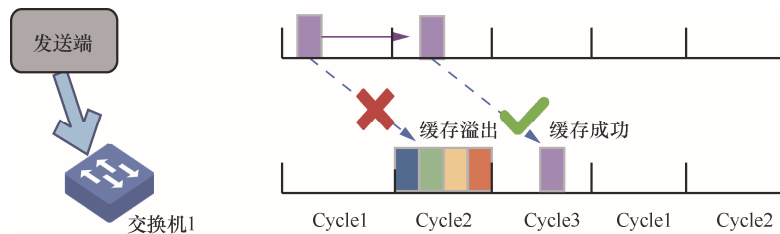


图 4 源节点偏移注入示意图

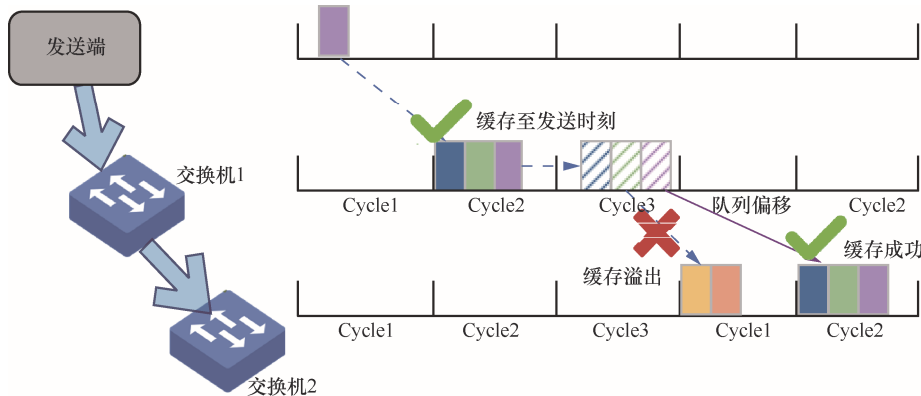


图 5 逐跳队列偏移示意图

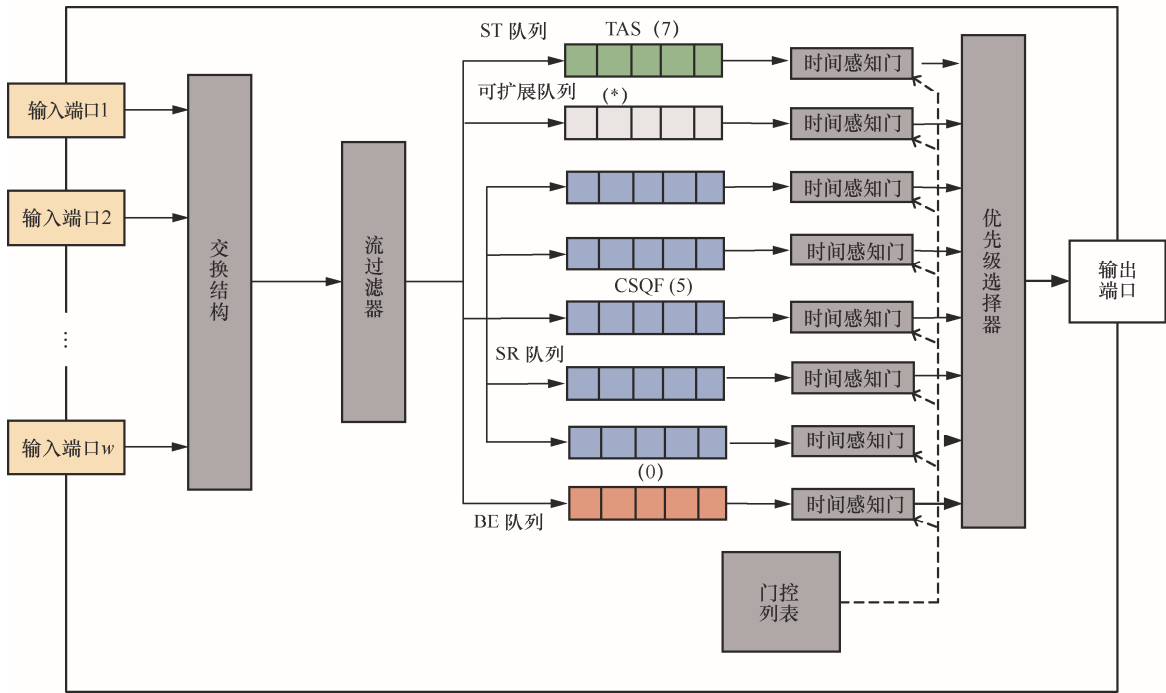


图6 交换机结构

域网（VLAN, virtual local area network）标记机制及数据流的优先级分类机制<sup>[20]</sup>，在 TSN 的 VLAN 标记机制中插入了一个 4 byte 的 VLAN 标签，用于标识帧与 TSN 相关的特征，其中，优先级标识(PCP, priority code point) 最多可标识 8 种队列优先级，流进入交换机后，流过滤器根据 PCP 字段值将数据包转发至相应的出口队列。

网络中的流被划分为 ST 流、SR 流、BE 流 3 类，分别采用 TAS 机制、CSQF 机制及尽力而为的方案进行调度。ST 流是指对传输时延和抖动有较高要求的数据流，如控制流、同步流等，由高精度的 TAS 机制进行控制，以保证 ST 流传输过程中尽量无等待。由文献[21-23]可知，关键控制流的采样周期通常在一定范围内可调，因此，本文定义每条 ST 流  $f_i^T$  为一个五元组： $\{T_{i,\min}^T, T_{i,\max}^T, L_i^T, D_i^T, P^T\}$ ，其中， $T_{i,\min}^T$  为流最小采样周期， $T_{i,\max}^T$  为流最大采样周期， $L_i^T$  为数据包总长度， $D_i^T$  为流截止时间， $P^T$  为流优先级。SR 流是指网络中传输非关键控制信息的大带宽周期流，相对于时间敏感的 ST 流，SR 流采样周期间隔大，而且对传输时延和抖动的要求低，但网络中的 SR 流往往数量多且包长。每条 SR 流  $f_j^R$  由一个四元组  $\{T_j^R, L_j^R, D_j^R, P_j^R\}$  定义， $T_j^R$  为流采样周期， $L_j^R$  为数据包的包长， $D_j^R$  为流传输的截止时间，

$P_j^R$  为流优先级。BE 流是指网络中尽力而为进行传输的流，无固定采样周期，本文不考虑 BE 流的具体参数。

采用 TAS 机制进行调度的 ST 队列具有最高优先级 7，并保留一个队列根据实际情况扩展使用；采用 CSQF 机制进行调度的 5 个 SR 队列具有相同优先级 5；BE 队列具有最低优先级 0。

## 2.2 网络约束

### 2.2.1 网络最小时隙约束

令  $T_u$  表示网络最小时隙，网络中所有流周期均为其整数倍，令  $T_g^R$  表示所有 SR 流周期的最大公约数，则  $T_u$  的最大取值需要小于  $T_g^R$ ， $T_u$  的取值范围为  $T_g^R$  的因数集  $L=\{l_1, l_2, \dots, l_k\}$ ，即

$$\begin{aligned} T_g^R &= \text{GCD}(T_1^R, T_2^R, \dots, T_m^R) \\ T_u^R &\leq T_g^R, T_u^R \in L \end{aligned} \tag{1}$$

由于 ST 流采样周期在有界范围内可变，因此  $T_u$  取值与 ST 流采样区间相关，即

$$\begin{aligned} T_u &\geq \max(T_{i,\min}^T) \\ T_u &\leq \min(T_{i,\max}^T) \end{aligned} \tag{2}$$

ST 流数据包需要在一个单位时隙内被全部转发，从而减少关键流的时延和抖动，即

$$T_u \geq \sum_{i=1}^n \frac{L_i^T}{U} \quad (3)$$

其中， $U$  表示交换机出口转发速率。

SR 流采用 CSQF 机制进行调度，每个缓存队列内的数据包需要能够在一个时隙内被完全发送，即

$$T_u \geq \frac{\text{BufSize}}{U} + \delta_{\max} \quad (4)$$

其中，BufSize 是出口队列的缓存区大小， $\delta_{\max}$  是网络中时间同步精度等问题造成的最大时延之和。

### 2.2.2 门控约束

对于传输 SR 流的出口队列，假设 CSQF 机制控制同一交换机内的  $x$  个出口队列，由于 CSQF 机制的本质仍是采用缓存转发，上一个  $T_u$  内门控编码为开的队列在下一个  $T_u$  内编码一定为关，因此 SR 队列门控周期  $T_{\text{TAS}}$  为  $xT_u$ ，即

$$T_{\text{CSQF}} = xT_u \quad (5)$$

对于 ST 队列，其门控周期  $T_{\text{TAS}}$  为 ST 流实际采样周期的最小公倍数，表示为

$$T_{\text{TAS}} = \text{LCM}\{T_1^T, T_2^T, \dots, T_n^T\} \quad (6)$$

门控列表循环周期  $T_{\text{Gate}}$  为 ST 队列循环周期  $T_{\text{TAS}}$  与 SR 队列门控循环周期  $xT_u$  的最小公倍数。

$$T_{\text{Gate}} = \text{LCM}\{T_{\text{CSQF}}, T_{\text{TAS}}\} \quad (7)$$

交换机的门控编码条目数为一个有限的定值，因此，假设在同一个交换机出口的门控列表循环周期内，交换机最多可存储  $\text{Gate}_{\max}$  个门控事件，因此  $T_{\text{Gate}}$  不能大于  $\text{Gate}_{\max}$ ，即

$$T_{\text{Gate}} \leq \text{Gate}_{\max} \quad (8)$$

### 2.2.3 ST 流约束

本文仍采用 TAS 机制调度 ST 流量，因此 ST 的相关约束，包括采样周期约束、总线占用率约束以及截止时间约束，都与文献[14]研究中相同。

#### 1) 采样周期约束

$$\text{令 } r_i = \left\lceil \frac{T_{i,\min}^T}{T_u} \right\rceil, \text{ 由式(2)可知 } T_u \geq \max(T_{i,\min}^T),$$

因此  $r_i = 1$ 。

$$\text{令 } k_i = \left\lceil \frac{T_{i,\max}^T}{T_u} \right\rceil, \text{ 对于任意一条 ST 流 } f_i^T, \text{ 其实}$$

际采样周期  $T_i^T$  的取值范围为  $[1, 2, \dots, k_i] \times T_u$ ，即

$$\begin{aligned} T_i^T &= v_i \times T_u \\ v_i &\leq k_i, v_i \in Z^+ \end{aligned} \quad (9)$$

#### 2) 总线占用率约束

为保证最大化网络调度成功率，首先需要保证在任意单位时隙内，ST 流对发送带宽的占用率均不大于 1，即

$$\frac{\sum_{i=1}^n \frac{L_i^T}{U}}{T_u} \leq 1 \quad (10)$$

#### 3) 截止时间约束

取  $f_i^T$  的实际采样周期  $T_i^T$  为 ST 流的传输截止时间，则  $f_i^T$  的数据包从源节点发出至目的节点接收的总时延需要小于或等于截止时间，即

$$f_i^T.\text{delay} \leq D_i^T \quad (11)$$

$f_i^T$  时延构成包括比  $f_i^T$  先进入缓存区的其他 ST 流数据包的传输时延、 $f_i^T$  本身数据包传输时延，表示为

$$f_i^T.\text{delay} = \sum_{\substack{k=0 \\ f_i^T \cdot T_u^q = 1}}^i \frac{L_k^T}{U} \quad (12)$$

其中， $f_i^T \cdot T_u^q = 1$ ，表示流  $f_i^T$  在第  $q$  个单位时隙内采样。

### 2.2.4 SR 流约束

#### 1) 转发时延约束

本文采用 CSQF 机制对 SR 流量进行调度，SR 队列的门控周期  $T_{\text{TAS}}$  为  $xT_u$ ，CSQF 机制要求进入出口队列的数据包在该时隙内保持缓存状态，然后在至多  $T_{\text{TAS}}$  个时隙内被发送，由此保证数据包单跳时延可控。因此，在 CSQF 机制控制下，SR 数据包的单跳转发时延  $f_j^R.\text{hop\_delay}$  需要小于或等于  $xT_u$ ，否则将出现传输超时或丢包。

$$f_j^R.\text{hop\_delay} \leq xT_u \quad (13)$$

$f_j^R.\text{hop\_delay}$  的实际大小取决于  $f_i^T$  发送时延、与  $f_j^R$  在同时隙进入同队列的其他 SR 流的发送时延、 $f_j^R$  数据包缓存时延  $f_j^R.\text{cache}$ 、 $f_j^R$  发送时延。CSQF 机制可在报文中通过 SID 标签，显示携带在每跳选择的注入时隙号，即在该跳选择的出口队列，当不重新指定出口队列时， $f_j^R$  数据包进入当前时隙对应的 RQ，此时其最大缓存时延为  $T_u$ ，而

当 SID 标签所标识的注入时隙与当前到达时隙不一致, 即重新指定出口队列时, 就会导致缓存时延增加, 令  $f_j^R.queue\_offset$  表示该跳的队列偏移量, 则  $f_j^R$  的最大缓存时延  $f_j^R.cache\_max$  为

$$f_j^R.cache\_max = f_j^R.queue\_offset + T_u \quad (14)$$

令  $hp(j)$  表示优先级高于  $f_j^R$  的流即 ST 流的集合,  $sp(j)$  表示所有与  $f_j^R$  进入同一交换机相同队列的 SR 流集合, 则此时, SR 流的单跳转发时延为

$$f_j^R.hop\_delay = \sum_{\substack{\forall k \in hp(j) \\ f_k^T \cdot T_u^{p+x} = 1}} \frac{L_k^T}{U} + \sum_{\substack{\forall w \in sp(j) \\ f_w^R \cdot T_u^p = 1}} \frac{L_w^R}{U} + \quad (15)$$

$$f_j^R.cache\_max$$

其中,  $f_k^T \cdot T_u^{p+x} = 1$ , 表示流  $f_k^T$  在超周期第  $(p+x)$  个时隙内采样;  $f_w^R \cdot T_u^p = 1$ , 表示流  $f_w^R$  在超周期第  $p$  个时隙内采样。

## 2) 截止时间约束

对于任意一条 SR 流  $f_j^R$ , 其可在源节点处进行一次注入偏移; 其次, 在大规模网络中链路传播时延不可忽略。因此, SR 流在整个网络中的传输时延由源节点偏移时延、链路传播时延以及各交换机内部时延组成, 表示为

$$f_j^R.delay_h = f_j^R.source\_delay + \sum_{h=1}^{f_j^R.hop\_nums} f_j^R.hop\_delay_h + \sum_{i=1}^{f_j^R.link\_nums} \frac{f_j^R.link\_length_i}{r} \quad (16)$$

其中,  $f_j^R.hop\_nums$  表示流  $f_j^R$  传输所经过的总交换机数,  $f_j^R.link\_nums$  表示流  $f_j^R$  传输流经的链路总数,  $f_j^R.link\_length_i$  表第  $i$  条流经链路的长度,  $r$  表示电磁波的传播速率。

SR 流总时延需要小于截止时间, 表示为

$$f_j^R.delay \leq D_j^R \quad (17)$$

## 3) 源节点偏移约束

若不进行队列偏移, CSQF 机制近似退化为 CQF 机制, 数据包在每跳交换机处经历的最大转发时延为  $2T_u$ , 再由截止时间约束可推出, 源节点处每条 SR 流的偏移量  $f_j^R.source\_delay$  需要满足

$$f_j^R.source\_delay \leq D_j^R - f_j^R.hop\_nums \times 2 \times T_u \quad (18)$$

## 4) 队列偏移约束

在满足截止时间约束且不考虑周期对齐存在偏差的情况下, 在第  $h$  跳处, 每条 SR 流最多可选择偏移进入除正在发送队列外的其余  $(x-2)$  个队列之一中, 即

$$f_j^R.queue\_offset_h \leq (x-2)T_u \quad (19)$$

考虑源节点偏移、上游节点总时延、总链路传播时延、下游节点不进行队列偏移时的转发时延上限, 上述所有时延与本跳时延相加之和需要小于该流的截止时间, 即

$$f_j^R.source\_delay + \sum_{p=1}^{h-1} f_j^R.hop\_delay_p + \sum_{i=1}^{f_j^R.link\_nums} \frac{f_j^R.link\_length_i}{r} + f_j^R.hop\_delay_h + \quad (20)$$

$$(f_j^R.hop\_nums - h) \times 2 \times T_u \leq D_j^R$$

令

$$queue\_band_h = D_j^R - f_j^R.source\_delay - \sum_{p=1}^{h-1} f_j^R.hop\_delay_p - \sum_{i=1}^{f_j^R.link\_nums} \frac{f_j^R.link\_length_i}{r} - \quad (21)$$

$$2 \times T_u - (f_j^R.hop\_nums - h) \times 2 \times T_u$$

则各跳交换机处的最大队列偏移约束为

$$f_j^R.queue\_offset_h \leq \min\{(x-2)T_u, queue\_band_h\} \quad (22)$$

## 5) 最大队列缓存约束

为避免发生缓存溢出丢包, 需要保证超周期的任意时隙内, 交换机出口队列缓存数据包大小之和都小于  $BufSize$ , 表示为

$$\sum_{j=1}^m Map(f_j^R, Switch'_{a,b}) \times L_j^R \leq BufSize$$

$$\forall t \in \{0, 1, 2, \dots, T_{sup}\},$$

$$Map(f_j^R, Switch'_{a,b}) \in \{0, 1\}, \quad (23)$$

$$\forall a \in \{0, 1, 2, \dots, SwitchNums\},$$

$$\forall b \in \{0, 1, 2, \dots, QueueNums\}$$

其中,  $Map(f_j^R, Switch'_{a,b})$  表示流  $f_j^R$  与交换机  $a$  的第  $b$  个 CSQF 队列在时刻  $t$  的映射关系, 当在第  $t$  个时隙内, 流  $f_j^R$  进入  $a$  交换机的  $b$  队列进行缓存, 则  $Map(f_j^R, Switch'_{a,b})$  值为 1, 反之值为 0;  $SwitchNums$  为网络中交换机的个数;  $QueueNums$  为由 CSQF 机制控制的队列数。

本文研究场景为大规模广域网，相较于局域网，其拓扑更加复杂，流数量也大幅度增加，很难寻找到同时支持所有数据流传输的调度方案，只能尽力提升可调度流占比，但由于 ST 流数量总体较少，且传输重要信息，需要保证所有 ST 流的正常传输。因此，本文将调度成功率  $SRate$  定义为，在保证所有 ST 流成功调度的前提下，成功调度的 SR 流的数量  $m_{SR,success}$  占 SR 流总数量  $m$  的比值，表示为

$$SRate = \frac{m_{SR,success}}{m} \quad (24)$$

### 3 HSTCS

CSQF 机制消除了 CQF 机制中对时间精准同步的严格限制，交换机接收到分组之后，可根据相邻设备之间的周期映射关系，计算出下游节点的接收时隙；此外，CSQF 机制可与路由选择算法相结合，通过 SID 标签逐跳指定注入的交换机及传输时隙号，提高网络的传输调度能力。因此，本文提出的 HSTCS 主要研究解决以下两个问题。

1) 推导跨域节点间的周期循环映射对齐方案。在仅能保证频率同步的前提下，推导出当位于不同时间域内，两个相邻节点间实现周期循环映射对齐的方案，为设计跨域网络的流量调度方案奠定基础。

2) 设计联合路由和调度的流规划算法。首先，根据本文所研究 TSN 调度场景下的网络特性，选择合适的路由方案；其次，研究基于 CSQF 机制的 SR 流调度表的高效求解算法，算法性能目标为更好地实现网络负载均衡，提高网络的资源利用率，从而提升系统的调度性能。

#### 3.1 跨域节点间的周期映射

CSQF 机制基于存储转发且其循环周期为网络最小调度时隙的整数倍，因此与文献[14]中所述相同，在进行系统流调度前，首先要确定网络最小调度时间单位  $T_u$ ，并尽可能降低 ST 流的带宽占用率，为大带宽 SR 留出更多可使用的网络资源，从而提高网络的可调度性。相较于文献[14]中采用 CQF 机制进行调度，在网络约束中，最小时隙约束及 ST 流约束与文献[14]中所述完全相同，CSQF 机制仅影响门控列表约束，因此，本文确定最小调度时隙及 ST 流采样周期的步骤如下。

**步骤 1** 提取 SR 流集合中的采样周期参数  $T_g^R$ ，将其因数集  $L$  内的元素升序排列，并令  $x=1$ ， $T_u = L(x)$ 。

**步骤 2** 检查此时  $T_u$  的取值是否满足式(1)~式(4)的最小时隙约束，若满足进入步骤 3，否则令  $x = x + 1$ ， $T_u = L(x)$ ，重复步骤 2。

**步骤 3** 按照流编号，对每一条 ST 流计算  $k_i = \left\lceil \frac{T_{i,max}^T}{T_u} \right\rceil$ ，并取  $T_i^T = k_i \times T_u$ 。

**步骤 4** 检查此时各参数取值是否满足式(5)~式(8)的门控约束，若满足则流程结束，否则令  $x = x + 1$ ， $T_u = L(x)$ ，返回步骤 2。

在确定网络最小调度时隙  $T_u$  及 SR 流的实际采样周期后，即可进一步推导当交换机间周期对齐存在偏差时的映射对齐方案。

由于 ST 流采用 TAS 机制进行调度，其始终只进入下一交换机的相同 TAS 出口队列中，仅需要根据相邻交换机间的周期对齐时间差，确定下一跳门控列表的开启时间即可。而对于采用 CSQF 机制的 SR 流，当相邻交换机间存在时间差时，由于每个时隙对应的 RQ 不同，需要根据数据包达到下一跳的时隙确定进入的缓存队列，CSQF 机制周期循环对齐示意图如图 7 所示。

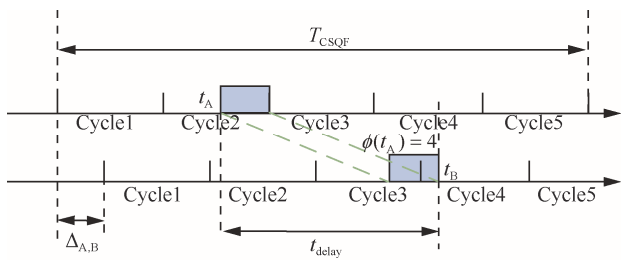


图 7 CSQF 机制周期循环对齐示意图

A、B 表示两个位于不同时间域内的相邻交换机，以 CSQF 机制的  $T_{CSQF}$  为周期，每个  $T_u$  内的 SQ、RQ 和 TQ 是固定不变的。以 A 交换机的 Cycle1 时隙起点处为周期对齐的基准点，假设 A、B 交换机间存在  $\Delta_{A,B}$  大小的时间差， $\phi(t) = z$  表示在 A 交换机  $t$  时刻发送的数据包，将在 B 交换机的第  $z$  个时隙内被完全接收，如图 7 所示情况，则  $\phi(t_A) = 4$ 。从 A 交换机开始发送至 B 交换机完全接收所经历的时延  $f_j^R.delay$  由数据包在上一跳交换机的发送时延  $f_j^R.delay\_trans$  以及链路传播时延  $f_j^R.delay\_prop$

组成, 表示为

$$\begin{aligned} f_j^R.\text{delay} &= f_j^R.\text{delay\_trans} + f_j^R.\text{delay\_prop} \\ f_j^R.\text{delay\_trans} &= \frac{L_j^R}{U} \\ f_j^R.\text{delay\_prop} &= \frac{f_j^R.\text{link\_length}_{A,B}}{r} \end{aligned} \quad (25)$$

令数据包在 A 交换机开始发送的时间为  $t_{A,\text{send}}$ , B 交换机接收到全部数据包的时间为  $t_{B,\text{receive}}$ , 则  $t_{B,\text{receive}}$  为

$$t_{B,\text{receive}} = t_{A,\text{send}} + t_{\text{delay}} - \Delta_{A,B} \quad (26)$$

进一步可由式(27)计算得出  $t_{B,\text{receive}}$  位于 B 交换机的第  $z$  个时隙内。

$$z = \left\lceil \frac{t_{B,\text{receive}}}{T_u} \right\rceil \bmod \left( \frac{T_{\text{CSQF}}}{T_u} \right) \quad (27)$$

综上, 两个相邻且存在周期对齐偏差交换机间的周期映射对齐式  $\phi(t)$  如式(28)所示。

$$\begin{aligned} \phi(t_{A,\text{send}}) &= \\ & \left\lceil \frac{t_{A,\text{send}} + \frac{L_j^R}{U} + \frac{f_j^R.\text{link\_length}_{A,B}}{r} - \Delta_{A,B}}{T_u} \right\rceil \bmod \left( \frac{T_{\text{CSQF}}}{T_u} \right) \end{aligned} \quad (28)$$

设实际选择的队列偏移量为  $dT_u$ , 则数据包开始转发至下一跳的发送时间  $t_{B,\text{send}}$  为

$$\begin{aligned} t_{B,\text{send}} &\in \left[ \left( \phi(t_{A,\text{send}}) + dT_u \right) \bmod \left( \frac{T_{\text{CSQF}}}{T_u} \right), \right. \\ & \left. \left( \phi(t_{A,\text{send}}) + (d+1)T_u \right) \bmod \left( \frac{T_{\text{CSQF}}}{T_u} \right) \right] \end{aligned} \quad (29)$$

### 3.2 联合路由及流量调度的 SR 流规划算法

基于第 3.1 节中确定  $T_u$ 、ST 流实际采样周期及相邻交换机间的周期映射对齐方案, 本节进一步研究对占据网络大部分发送带宽的 SR 流的流规划算法。

#### 3.2.1 路由选择方案

在给定网络拓扑的前提下, 对于一条待调度 SR 流  $f_j^R$ , 可生成发送节点与接收节点间的所有无环路由集合  $\text{Route}_j^R$ , 令  $r_j^R$  为  $f_j^R$  选择的路由方案, 即  $r_j^R \in \text{Route}_j^R$ 。经典的路由选择算法如 Dijkstra 算法、Floyd 算法等, 可快速寻找到最短路径, 从而降低

传输的时延及抖动, 但该方法很容易形成瓶颈链路, 导致整个网络的数据传输受限于该链路, 而其余大量链路的带宽浪费。

受文献[24]所提出的竞争性在线路由选择算法的启发, 本文提出一种综合考虑传输时延、传输跳数及链路负载均衡的路由选择方案。

首先, 定义拓扑中任意一条边代表的传输链路  $e_i$  的阻塞率为

$$\text{ebr}(e_i) = \sum_{t=1}^{T_u^{\text{sup}}} \text{lb} \left( 1 - \min \left( \frac{\sum_{j=1}^m \text{Map}(f_j^R, e_i, t) \times L_j^R}{U \times T_u}, 1 \right) + \varepsilon \right) \quad (30)$$

其中,  $\text{Map}(f_j^R, e_i, t)$  的取值为 0 或 1, 当取值为 1 时, 表示流  $f_j^R$  在第  $t$  个时隙内, 在编号为  $e_i$  的链路上传输数据,  $U \times T_u$  为单个时隙内单条链路上的最大传输容量。

令

$$\text{bu}_i = \frac{\sum_{j=1}^m \text{Map}(f_j^R, e_i, t) \times L_j^R}{U \times T_u} \quad (31)$$

其中,  $\text{bu}_i$  值表示当前已调度流对该条链路带宽的占用率,  $\varepsilon$  是一个很小的数, 加上该值是为了防止当带宽资源耗尽,  $\text{ebr}(e_i)$  值为负无穷。在计算链路的阻塞率时, 通过对剩余带宽率取对数, 可以放大当该链路存在资源即将耗尽情况时对计算结果的影响, 从而降低该路由被选择的概率, 预防瓶颈链路的出现。

路由方案的整体阻塞率为

$$\text{br}(r_j^R) = \frac{\sum_{e_i \in r_j^R} \text{ebr}(e_i)}{f_j^R.\text{hop\_nums}} \quad (32)$$

其中,  $f_j^R.\text{hop\_nums}$  表示流  $f_j^R$  所经路由跳数, 通过求和  $r_j^R$  中链路的阻塞率再除以  $f_j^R.\text{hop\_nums}$  求平均, 可以降低长链路阻塞率之和更高的影响, 避免优先选择最短路由。

#### 3.2.2 基于改进退火算法的 SR 流规划算法

相较于 CQF 只能进行源节点偏移, CSQF 还可通过 SID 标签逐跳选择发送时隙, 此外, 本文联合路由选择进行调度, 可调整参数较多, 求解空间急剧增大, 需要引入启发式算法进行求解。传统启发式算法, 如退火算法<sup>[25]</sup>、粒子群算法、禁忌搜索算

法<sup>[26]</sup>等，本质均是通过在上一轮求解结果的邻域内进行随机搜索，寻求下一个更优解。但在解空间较大的情况下，仅通过随机搜索来确定求解方向，算法性能相较于完全随机求解的提升效果受限。文献[27]提出在启发式算法求解过程中，在确定下一步求解方向时，引入与之相关的特定领域知识(DSK, domain-specific knowledge)即先验知识，可以减少不必要的搜索工作并获得接近最佳的结果，大幅度提升算法性能。因此，联合第3.1节中所提路由选择方案，本节提出一种基于改进退火算法的SA-CSQF调度算法，SA-CSQF调度算法见算法1。

### 算法1 SA-CSQF调度算法

输入 ST流:  $F^T = \{f_i^T\}, i=1,2,\dots,n$

SR流:  $F^R = \{f_j^R\}, j=1,2,\dots,m$

$T_u$ 、 $T_{sup}$ 、BufSize、 $U$ 、CSQF队列数  $N_{queue}$

初始退火温度  $T_{now}$ 、终止退火温度  $T_{final}$ 、退火

系数  $\gamma$ 、最大循环次数 max\_cycle

输出 SR流偏移:

$off_j = \{f_j^R.src\_offset, \{f_j^R.hop\_offset_h\}\}, (j=1,2,\dots,m)$

调度成功流集合:  $F_{suc}$

调度失败流集合:  $F_{fail}$

网络调度成功率: SuccessRate

带宽利用率: BandRate

初始化 生成随机初始解  $F_{suc}$ 、 $F_{fail}$ ，对应集合中的流数目为  $N_{suc}$ 、 $N_{fail}$

**while**  $T_{now} < T_{final}$  **do**

now\_cycle = 0

**while** now\_cycle < max\_cycle **do**

move\_num =  $\min(\lfloor m \times \omega \rfloor, N_{fail} \times 2, N_{suc})$

// 计算移出流的数量，move\_num不能大于  $F_{suc}$  元素个数  $N_{suc}$

**if** rand() >  $\delta$  **do**

RW=Rand // 随机资源分配

**else do**

RW=DSK //结合先验知识资源

分配

**if** rand() >  $\delta$  **do**

SW=Rand,

remove(sortflow( $F_{suc}$ , rand), Rand)

// 随机移出流

off\_now=FFS(sortflow( $F_{fail}$ , Rand)) //

随机流排序

**else do**

SW = DSK

remove (sortflow( $F_{suc}$ ,  $ov_j$ ),  $\theta$  dsk

+ (1- $\theta$ ) Rand) // 部分随机移出流

off\_now = FFS (sortflow ( $F_{fail}$ ,  $\Delta_{aim\_func} > 0$ )) // 按调度优先级流排序

**if**  $\Delta_{aim\_func} > 0$  **do** // 更新当前结果

off=off\_now

**else if**  $e^{-\frac{|\Delta_{aim\_func}|}{T_{now}}} > \text{rand}()$  **do**

off=off\_now

**else**

resource\_reset(off)

now\_cycle ++

$T_{now} = T_{now} \times \gamma$

算法1核心思想是通过移出部分成功调度流，释放其占用网络资源，再对调度失败流进行重新调度，逐步探索更好的调度方案。

为了避免完全随机搜索导致的大量重复和无效计算，SA-CSQF调度算法结合完全随机和部分随机缩小求解空间，加快求解。分别以  $1-\alpha$  及  $\alpha$  的概率，选择资源分配方式为完全随机分配(RW=Rand)或结合先验知识分配(RW=DSK)。当RW=DSK时，流调度时按路由整体阻塞率、时隙资源利用率进行网络资源分配；反之进行随机资源分配，扩展求解空间。

分别以  $1-\delta$  及  $\delta$  的概率，选择移出成功流和调度失败流的顺序为完全随机(SW=Rand)或结合先验知识的部分随机(SW=DSK)。当SW=Rand时，随机选中  $F_{suc}$  中的 move\_num 条流，释放分配的系统资源，并将该部分流加入  $F_{fail}$  集合中，之后按随机顺序调度  $F_{fail}$  中流；当SW=DSK时， $F_{suc}$  按调度优先级的权值  $ov_j$  进行升序排序， $ov_j$  计算式如式(33)所示，顺序移出  $\theta \times \text{move\_num}$  条流，随机移出  $(1-\theta) \times \text{move\_num}$  条流，并按  $ov_j$  降序调度  $F_{fail}$  中的流。引入小部分随机，是为了防止若完全按  $ov_j$  权值顺序移出，当接近最优方案后，会进行多次无效重复计算，引入小部分随机可不断试探邻域内有无更优解。

$$\begin{aligned} \text{ov}_j &= x_1 \times (1 - T_j^R / T_{\max}^R) + x_2 \times (1 - D_j^R / D_{\max}^R) + \\ &x_3 \times L_j^R / L_{\max}^R \\ x_1 + x_2 + x_3 &= 1, \\ j &= 1, 2, \dots, m \end{aligned} \quad (33)$$

最后, 根据目标函数计算值, 更新最优解。现有使用启发式算法对 TSN 调度方案进行求解的目标函数, 大多是最大化可调度流的数目, 例如文献[28-29], 本文目标函数为加权网络调度成功率及系统带宽利用率, 如式(34)所示。根据实验结果, 其能更好地实现提高网络调度成功率与系统带宽利用率的折中, 具体对比分析结果在仿真结果中进行说明。

$$\begin{aligned} \text{aim\_func} &= \\ \omega_1 \times \text{SuccessRate} + \omega_2 \times \text{BandRate} \end{aligned} \quad (34)$$

在确定系统资源分配方式及流调度顺序后, 对  $F_{\text{fail}}$  流集中的 SR 流采用失败流调度 (FFS, fail flow schedule) 算法进行重新调度, FFS 调度算法见算法 2。

#### 算法 2 FFS 调度算法

输入 move\_num、 $F_{\text{suc}}$ 、 $F_{\text{fail}}$ 、 $T_u$ 、RW、SW

输出 OFF、 $F_{\text{suc}}$ 、 $F_{\text{fail}}$

```

for  $f_j^R$  in  $F_{\text{fail}}$  do
    satisfy = 0 // 记录流是否调度成功的标志位, 初始为 0
    Route $_j^R$  = calculate_all_path( $f_j^R$ )
    // 根据网络拓扑信息生成所有可能的无环路由方案的集合
    SrcDelay $_j^R$  = calculate_src_delay( $f_j^R$ )
    SrcDelay $_j^R$  // 根据第 2.2 节中的源节点偏移约束, 确定可选择的源节点偏移量集合
    if RW=rand do
        SortRoute(Route $_j^R$ , Rand) // 随机排序路由方案
        SortSrcDelay(SrcDelay $_j^R$ , Rand) // 随机排序源节点偏移量
    else do
        SortRoute(Route $_j^R$ , br( $r_j^R$ )) // 按路由方案整体阻塞率排序
        SortSrcDelay(SrcDelay $_j^R$ , ocr_src(src_delay $_i$ )) // 按时隙资源占用率排序
         $r_j^R$  = Route $_j^R$ [0] // 选择路由方案

```

$f_j^R$ .hop\_nums=size( $r_j^R$ ) // 计算该方案下的交换机跳数

soruce calculate:

```
for src_delay $_i$  in SrcDelay $_j^R$  do
```

```
satisfy = 0
```

hop calculate:

```
for hop in 1:1:  $f_j^R$ .hop_nums do
```

```
QueueDelay $_{\text{hop}}$  = calculate_queue_delay( $f_j^R$ ,  $r_j^R$ [hop])
```

```
// 计算该跳可选的交换机队列偏移量 Route $_j^R$ 
```

```
if RW=rand do
```

```
SortQueueDelay(QueueDelay $_{\text{hop}}$ , Rand)
```

```
// 随机排序队列偏移量
```

```
else do
```

```
SortQueueDelay(QueueDelay $_{\text{hop}}$ , ocr_queue_delay $_i$ )
```

```
// 按资源占用率排序队列偏移量
```

```
for queue_delay $_i$  in QueueDelay $_{\text{hop}}$  do
```

```
if Check_meet_constraint(off $_j$ ) = True
```

```
do // 判断当前调度方案是否满足网络约束
```

```
satisfy += 1
```

```
goto hop calculate
```

```
if satisfy != hop do
```

```
// 在某跳交换机处未找到可行调度方案
```

```
goto soruce calculate
```

```
if satisfy ==  $f_j^R$ .hop_nums do
```

```
// 调度成功, 更新结果
```

```
 $f_j^R$  →  $F_{\text{suc}}$ 
```

```
off $_j$  → OFF
```

```
goto begin
```

```
else do
```

```
// 调度失败, 释放当前调度方案的占用资源
```

```
resource_reset( $f_j^R$ )
```

```
源节点偏移时隙的占用率 ocr_src(src_delay $_i$ )
```

计算式如式(35)所示。

ocr\_src(src\_delay $_i$ ) =

$$\frac{1}{s} \sum_{h=0}^{s-1} Q_{\text{band}(A)}(\text{src\_delay}_i + 1 + h \times T_j^R) s = \frac{T_{\text{sup}}}{T_j^R} \quad (35)$$

其中， $Q_{band(A)}(t)$  记录一个超周期内，已调度流在第  $t$  个时隙内，对 A 交换机输出端口发送带宽的占用率。

依据相邻节点间的周期映射关系以及最大队列偏移约束，计算得每跳处队列偏移量集合  $QueueDelay_{hop}$ 。假设  $f_j^R$  数据包在 A、B 两个交换机间传输，A 交换机传输的数据包在 B 交换机的第  $p$  个时隙内被全部接收，并在集合  $QueueDelay_{hop}$  中选择大小为  $queue\_delay_i$  队列偏移量后进入队列  $y$ ，则其将在 B 交换机的第  $p$  至  $p+queue\_delay_i$  个时隙内被缓存，并在  $p+queue\_delay_i+1$  个时隙内发送。因此，缓存期间该队列缓存不能溢出，且在发送时隙内的带宽利用率小于 1。设  $Q_{cache(B,r)}(t)$  表示 B 交换机的第  $r$  个时隙对应的接收队列在第  $t$  个时隙内的缓存占用率。则

$$Q_{band(B)}(p + queue\_delay_i + 1) \leq 1 \quad (36)$$

$$\begin{aligned} Q_{cache(B,p+queue\_delay_i)}(t) &\leq 1, \\ \forall t \in [p, p + queue\_delay_i] \end{aligned} \quad (37)$$

当采用随机资源分配时，打乱该集合内元素顺序；反之计算集合内各元素的资源占用率  $ocr\_que(queue\_delay_i)$  大小后，升序排列集合元素， $ocr\_que(queue\_delay_i)$  计算式如式(38)所示，优先选择平均综合占用率小的队列偏移量，实现网络负载均衡。

$$\begin{aligned} ocr\_que(queue\_delay_i) &= \frac{1}{s} \times \frac{1}{queue\_delay_i + 1} \times \\ &\sum_{h=0}^{s-1} \left\{ \left[ \sum_{d=p}^{p+queue\_delay_i} (Q_{cache(B,p+queue\_delay_i)}(d + h \times T_j^R)) \right] + \right. \\ &\left. (p + queue\_delay_i + 1 + h \times T_j^R) \right\} \quad (38) \end{aligned}$$

## 4 实验和评估

为验证本文所提出的将 TAS 与 CSQF 结合的混合流量调度机制 HSTCS 的有效性，本节将从退火算法目标函数选择、改进启发式算法的性能提升、与现有 CSQF 调度机制的性能对比等多个角度，将所提方案与现有研究方案进行对比验证。

在仿真中，网络的最大帧长 MTU 设为 1500 byte，交换机单个输出队列的最大缓存值 BufSize 为 6 MTU，交换机输出速率  $U$  为 1000 Mbit/s， $\delta_{max}$  取  $1 \mu s$ 。退火起始退火温度为

1000，终止温度为 0.01，退火系数为 0.95，控制随机概率的 3 个参数中  $\vartheta$ 、 $\delta$  取值均为 0.6， $\theta$  取值为 0.7，同温度下的最大循环次数为 50。每条 ST 流的最大采样周期和包长分别从集合 {1, 2, 4} ms、{0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1} KB 中随机选取，ST 流的最小采样周期均设置为 0.1 ms，截止时间为各 ST 流的实际采样周期。SR 流采样周期和包长分别从集合 {8, 12, 24, 48} ms、{2, 2.5, 3, 3.5, 4, 4.5, 5} KB 中随机选取，SR 流的截止时间随机取  $[0.5 \times T_j^R, T_j^R]$  范围内一个整数。两个相邻交换机间的周期对齐差值的绝对值取  $[0.5 \times T_u, T_u]$  范围内的随机整数。仿真拓扑如图 8 所示，ST 流及 SR 流随机选择其发送的源节点。

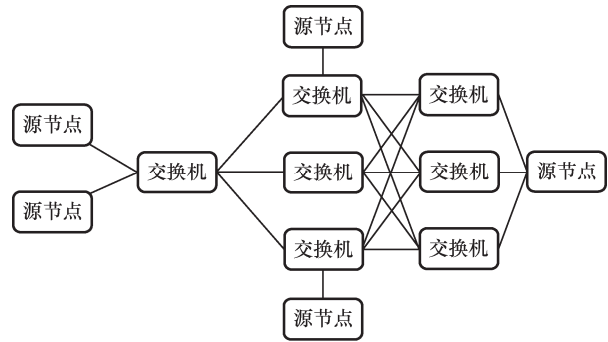


图 8 仿真拓扑

### 4.1 退火算法目标函数选择

退火算法需要根据每次迭代后目标函数结果的变化值来确定是否接受当前解，从而决定下一次求解的邻域范围，因此目标函数的取值对于算法性能至关重要。为验证本文所选取的加权带宽利用率及网络中流的调度成功率作为目标函数的有效性，仿真加权系数为 0.5，网络中存在 20 条 ST 流、3000 条 SR 流的情况下，系统的带宽利用率和整体调度成功率，不同目标函数下 HSTCS 的性能如图 9 所示。

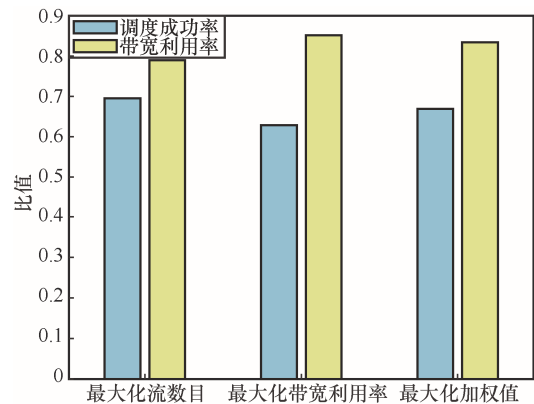


图 9 不同目标函数下 HSTCS 的性能

由仿真结果可知，最大化可调度流数量为目标函数时的带宽利用率最低，这是由于当无法找到可调度网络中全部流的调度方案时，启发式算法的求解方向逐渐变为优先选择调度资源占用小的流，如包长小、周期大、截止时间长的流，这将导致占用资源较多的流更难分配到传输资源；同理，当将目标函数设定为最大化带宽利用率时，其调度成功率为三者中最低；而将目标函数设置为最大化加权值时，既考虑了优先调度高资源占用的流保证带宽利用率，也可尽量将低资源占用率的流穿插其中，实现调度效果的良好折中。

### 4.2 进行源节点偏移及队列偏移的影响

CSQF 机制相较于 CQF 机制的主要改进为，可以通过结合 SID 标签实现分段路由，即逐跳指定队列偏移，不进行队列偏移时，CSQF 机制近似退化为 CQF 机制，为验证队列偏移对提升网络可调度性的影响，仿真当网络中存在 20 条 ST 流、3 000 条 SR 流的情况下，不进行任何负载均衡，流在源节点产生即注入下一跳交换机的原生 (native) 方法与只进行队列偏移、只进行源节点偏移、结合进行队列偏移和源节点偏移的 HSTCS，在系统的带宽利用率和调度成功率上的性能对比，不同偏移机制性能对比如图 10 所示。

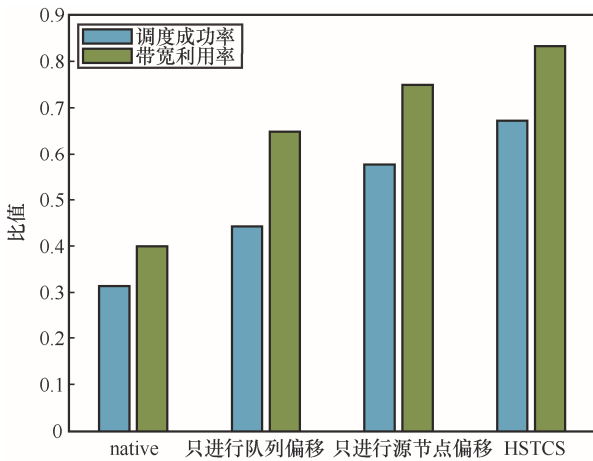


图 10 不同偏移机制性能对比

在拓扑固定的情况下，当 SR 流数目达到 3 000 时，整个网络达到高负载状态，由仿真结果可知，进行源节点偏移和队列偏移均对提升网络调度性能有显著作用，而综合了源节点偏移及队列偏移的 HSTCS 取得最佳性能。其中，源节点处的偏移注入对网络调度成功率和带宽利用率影响最大，这是因为

当拓扑固定，随着流数目增加，首跳交换机最容易出现溢出，导致后续队列也无法进行偏移调整。

### 4.3 基于改进退火算法的 SR 流规划算法性能验证

为求解 TSN 跨域场景下的流调度表，在本文中提出一种基于改进退火算法的 SA-CSQF 调度算法，为验证其相对于传统启发式求解方案以及现有研究方案的性能提升，在拓扑固定情况下，仿真对比了网络中存在 20 条 ST 流、3 000 条 SR 流情况下，在求解 SR 流调度表时，系统分别采用随机选择注入偏移及队列偏移的随机逐流调度，传统模拟退火 (Pure SA, pure simulated annealing) 算法，基于退火算法的注入时间规划-交换 (SA-ITP-Exchange, simulated annealing - injection time planning-exchange) [28] 机制，以及采用 SA-CSQF 调度算法的 HSTCS 时的调度成功率、带宽利用率及求解时长，不同方案的性能及求解时长对比分别如图 11、图 12 所示。

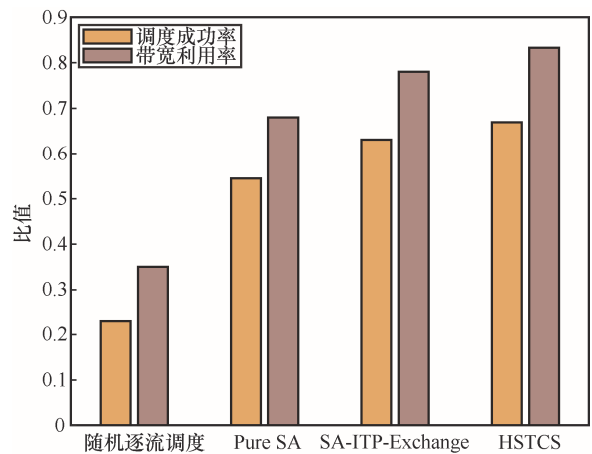


图 11 不同方案性能对比

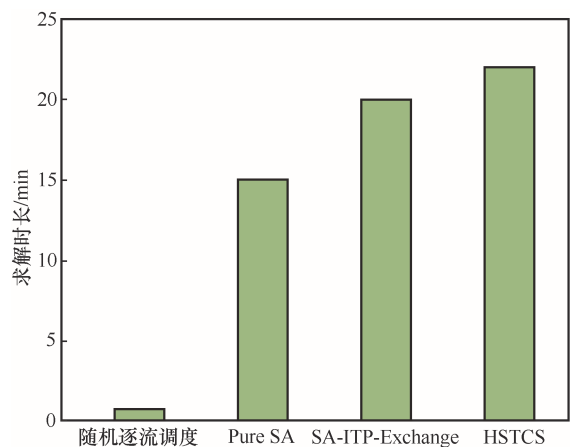


图 12 不同方案的求解时长对比

由仿真结果可知, 3种启发式算法相对于随机逐流调度方案的性能均有显著提升。文献[28]提出的 SA-ITP-Exchange 机制和本文提出的 HSTCS 均相对于 Pure SA 算法有较大性能提升, 这是由于在两种机制中均结合相关先验信息, 引导启发式算法的求解方向, 减小了 Pure SA 算法中完全随机求解带来的不确定性。而本文提出的 HSTCS 相对于 SA-ITP-Exchange 机制的性能又有进一步提升, 这是因为在 HSTCS 中引入了预防链路阻塞的路由选择方案, 并在结合先验信息确定移出流时, 采用了按序移出结合小部分随机移出的方案, 以实现接近最优方案时, 每次计算仍通过移入小部分随机来扩展求解空间, 试探有无更优解, 进一步提升算法求解性能。在求解时长上, 启发式方案由于采用迭代求解, 时长均远大于随机逐流调度方案, 其中, Pure SA 算法由于采用完全随机求解, 较容易持续陷于局部最优解而停止迭代, 因此其求解时长虽短但调度成功率低。本文所提的 HSTCS 相较于 SA-ITP-Exchange 机制在求解时长上有所上升, 这是由于 SA-CSQF 调度算法在计算时采用按序移出结合小部分随机移出的方法, 从而实现在较优解邻域内持续试探有无更优解, 因此扩大了求解空间, 求解时长增加, 但调度成功率也随之提升, 在离线调度场景下, 通过增加少量求解时间换取更高的网络调度成功率是有意义的。

#### 4.4 综合性能对比

为综合验证本文提出的 HSTCS 的有效性, 将 HSTCS 与随机逐流调度、文献[29]所提出的流偏移及周期移动 (FO-CS, flow offset and cycle shift) 机制进行性能对比。FO-CS 是基于逐流调度思想的 CSQF 调度求解机制, 其根据固定的 SR 流调度顺序进行调度表的求解计算, 针对每条流, 在其最大源节点偏移范围内, 按偏移量由小至大依次选择偏移值, 然后按照路由顺序, 在每跳交换机处, 根据队列偏移值的大小, 顺序寻找可分配的网络资源, 逐跳完成调度。仿真对比了网络中存在 20 条 ST 流、1 000~3 000 条 SR 流情况下, 不同方案下的网络调度成功率及系统带宽利用率, 不同方案的调度成功率对比如图 13 所示, 不同方案的带宽利用率对比如图 14 所示。

由仿真结果可知, 相较于随机逐流调度, 其余两种方案均有明显的性能提升, 且本文提出的 HSTCS 具有最佳性能, 相较于 FO-CS 机制, 其至多可提升

20%的调度成功率及 18%的带宽利用率, 这是由于 FO-CS 机制采用一次性的逐流调度, 相较于启发式算法, 其求解的搜索空间很小; 其次 FO-CS 机制没有考虑路由选择及交换机内部时隙分配的负载均衡问题, 因此更容易出现瓶颈路由及缓存溢出、发送溢出的情况, 影响调度性能。综上, 相较于现有研究所提方案, 本文所提出的 HSTCS 具有更好性能。

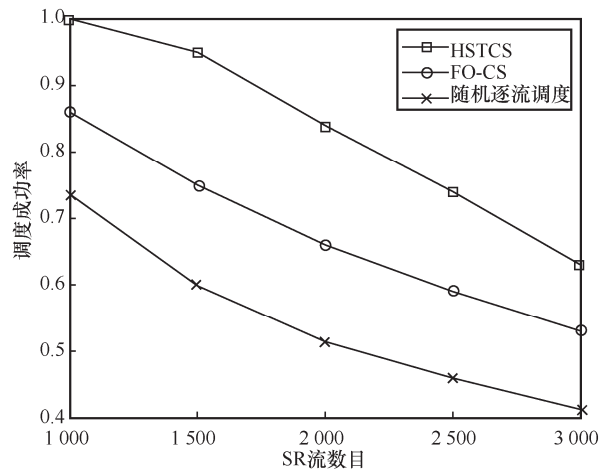


图 13 不同方案的调度成功率对比

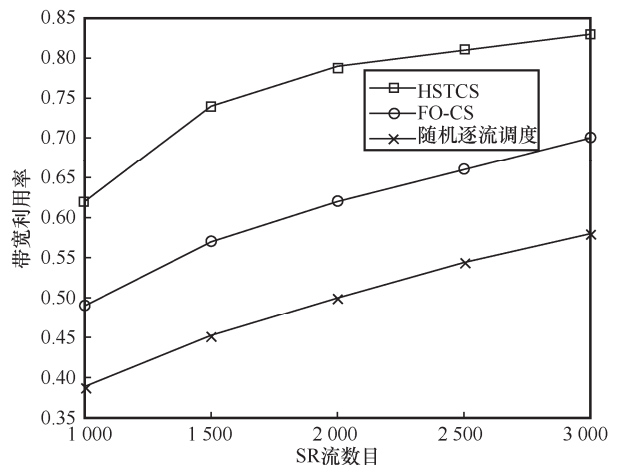


图 14 不同方案的带宽利用率对比

## 5 结束语

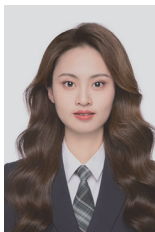
在网络设备仅能保证频率同步的广域网场景下, 本文提出一种将 TAS 调度算法与 CSQF 调度算法相结合的混合流量调度机制 HSTCS。首先对系统模型进行介绍, 给出交换机的结构设计, 对主要问题及遵循的网络约束进行阐述; 然后对 HSTCS 中的周期映射方案及 SR 流规划算法进行详细介绍; 最后, 搭建实验平台, 对本文所提方案的性能进行评估验证, 证明了方案的有效性。需要说明的是,

本文的调度方式为静态调度，但在实际网络场景中，动态调度可以更好地适应网络结构调整或突发流的处理，未来可以在该方向做进一步研究。

### 参考文献：

- [1] FELD J. PROFINET - scalable factory communication for all applications[C]//Proceedings of IEEE International Workshop on Factory Communication Systems. Piscataway: IEEE Press, 2005: 33-38.
- [2] CENA G, SENO L, VALENZANO A. Performance analysis of Ethernet powerlink networks for distributed control and automation systems[J]. *Computer Standards & Interfaces*, 2009, 31(3): 566-572.
- [3] JANSEN D, BUTTNER H. Real-time Ethernet: the EtherCAT solution[J]. *Computing and Control Engineering*, 2004, 15(1): 16-21.
- [4] FU S S, ZHANG H S, CHEN J H. Time sensitive networking technology overview and performance analysis[J]. *ZTE Communications*, 2018, 16(4): 57-64.
- [5] LIM H T, HERRSCHER D, WALTL M J, et al. Performance analysis of the IEEE 802.1 Ethernet audio/video bridging standard[C]//Proceedings of the 5th International ICST Conference on Simulation Tools and Techniques. New York: ACM Press, 2012: 27-36.
- [6] 黄韬, 汪硕, 黄玉栋, 等. 确定性网络研究综述[J]. *通信学报*, 2019, 40(6): 160-176.  
HUANG T, WANG S, HUANG Y D, et al. Survey of the deterministic network[J]. *Journal on Communications*, 2019, 40(6): 160-176.
- [7] IEEE. IEEE standard for local and metropolitan area networks - timing and synchronization for time-sensitive applications in bridged local area networks: IEEE Std 802.1AS-2011[S]. 2011.
- [8] IEEE. IEEE standard for local and metropolitan area networks--bridges and bridged networks -- amendment 31: stream reservation protocol (SRP) enhancements and performance improvements: IEEE 802.1Qcc-2018[S]. 2018.
- [9] 中国电子技术标准化研究院. 时间敏感网络白皮书[R]. 2020. China Electronics Standardization Institute. Time sensitive network white paper[R]. 2020.
- [10] GAVRILUT V, POP P. Traffic-type assignment for TSN-based mixed-criticality cyber-physical systems[J]. *ACM Transactions on Cyber-Physical Systems*, 2020, 4(2): 1-27.
- [11] PAHLEVAN M, OBERMAISSER R. Genetic algorithm for scheduling time-triggered traffic in time-sensitive networks[C]//Proceedings of 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA). Piscataway: IEEE Press, 2018: 337-344.
- [12] HUANG Y D, WANG S, WU B W, et al. TACQ: enabling zero-jitter for cyclic-queuing and forwarding in time-sensitive networks[C]//Proceedings of ICC 2021 - IEEE International Conference on Communications. Piscataway: IEEE Press, 2021: 1-6.
- [13] ZHANG J L, XU Q M, LU X Z, et al. Coordinated data transmission in time-sensitive networking for mixed time-sensitive applications[C]//Proceedings of IECON 2020, the 46th Annual Conference of the IEEE Industrial Electronics Society. Piscataway: IEEE Press, 2020: 3805-3810.
- [14] 尹长川, 李妍珏, 朱海龙, 等. HSTC: TSN 中的混合流量调度机制[J]. *通信学报*, 2022, 43(6): 119-132.  
YIN C C, LI Y J, ZHU H L, et al. HSTC: hybrid traffic scheduling mechanism in time-sensitive networking[J]. *Journal on Communications*, 2022, 43(6): 119-132.
- [15] STEINER W, PEÓN P G, GUTIÉRREZ M, et al. Next generation real-time networks based on IT technologies[C]//Proceedings of 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA). Piscataway: IEEE Press, 2016: 1-8.
- [16] IEEE. 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)[S]. 2015.
- [17] IEEE. IEEE standard for local and metropolitan area networks bridges and bridged networks amendment 26: frame preemption: IEEE Std 802.1Qbu-2016[S]. 2016.
- [18] CHEN M, GENG X, LI Z. Segment routing (SR) based bounded latency[EB]. 2018.
- [19] IEEE. IEEE Standard for local and metropolitan area networks -bridges and bridged networks--amendment 29: cyclic queuing and forwarding: IEEE 802.1Qch-2017[S]. 2017.
- [20] 李德水. 基于 IEEE 802.1Q 帧标记的 VLAN 实现原理[J]. *信息技术*, 2006, 30(10): 68-70.  
LI D S. Study on the principle of realizing the VLAN based on IEEE 802.1Q frame tag[J]. *Information Technology*, 2006, 30(10): 68-70.
- [21] ARZEN K E, CERVIN A, EKER J, et al. An introduction to control and scheduling co-design[C]//Proceedings of the 39th IEEE Conference on Decision and Control (Cat. No.00CH37187). Piscataway: IEEE Press, 2002: 4865-4870.
- [22] DAI X T, BURNS A. Period adaptation of real-time control tasks with fixed-priority scheduling in cyber-physical systems[J]. *Journal of Systems Architecture*, 2020(103): 101691.
- [23] DAI X T, CHANG W L, ZHAO S, et al. A dual-mode strategy for performance-maximisation and resource-efficient CPS design[J]. *ACM Transactions on Embedded Computing Systems*, 2019, 18(5s): 1-20.
- [24] JONATAN K, SÉBASTIEN M, PAOLO M, et al. Joint routing and scheduling for large-scale deterministic IP networks[J]. *Computer Communications*, 2021, 165: 33-42.
- [25] KENNEDY J, EBERHART R. Particle swarm optimization[C]//Proceedings of ICNN'95 - International Conference on Neural Networks. Piscataway: IEEE Press, 2002: 1942-1948.
- [26] GLOVER F. Future paths for integer programming and links to artificial intelligence[J]. *Computers & Operations Research*, 1986, 13(5): 533-549.
- [27] YAN J L, QUAN W, JIANG X Y, et al. Injection time planning: making CQF practical in time-sensitive networking[C]//Proceedings of IEEE INFOCOM 2020 - IEEE Conference on Computer Communications. Piscataway: IEEE Press, 2020: 616-625.
- [28] HUANG Y D, WANG S, FENG T, et al. Towards network-wide scheduling for cyclic traffic in IP-based deterministic networks[C]//Proceedings of 2021 4th International Conference on Hot Information-Centric Networking (HotICN). Piscataway: IEEE Press, 2022: 117-122.
- [29] PENG G Y, WANG S, HUANG Y D, et al. Traffic shaping at the edge: enabling bounded latency for large-scale deterministic networks[C]//Proceedings of 2021 IEEE International Conference on Communications Workshops (ICC Workshops). Piscataway: IEEE Press, 2021: 1-6.

## [作者简介]



**李妍珏**（1997- ），女，北京邮电大学先进信息网络北京实验室硕士生，主要研究方向为工业互联网、确定性网络、时间敏感网络等。



**朱海龙**（1987- ），男，博士，北京邮电大学讲师，主要研究方向为工业互联网、确定性网络、工业以太网、软件定义网络、时间敏感网络和车载网络等。



**韩文璇**（1998- ），女，北京邮电大学先进信息网络北京实验室硕士生，主要研究方向为工业互联网、时间敏感网络和车载网络等。



**尹长川**（1968- ），男，博士，北京邮电大学教授、博士生导师，主要研究方向为无线通信系统与网络理论、统计信号处理、机器学习及其在无线通信中的应用、物联网技术等。