

时间敏感网络中的动态路由与调度联合优化算法

周阳¹, 陈鸿龙², 张雷¹

(1. 南京邮电大学, 江苏 南京 210003; 2. 中国石油大学, 山东 青岛 266580)

摘要: 时间敏感网络 (TSN, time-sensitive networking) 是由 IEEE TSN 工作组制定的以太网通信标准, 致力于在以太网底层实现统一的确定性网络通信。由于 TSN 没有定义流量调度的具体实现方法, 路由与调度算法研究仍然是 TSN 研究领域的一个开放问题。面向工业应用中的 TSN 建立路由与调度联合优化问题模型, 并提出一个在线启发式算法求解动态流量的路由与调度方案。算法同时考虑传输时延和网络负载因素确定路由路径, 并利用两次剪裁操作快速确定调度时间。最后基于 OMNeT 的 NeSTiNg 框架实现 TSN 仿真环境。仿真结果表明, 所提算法与参考算法相比, 随着网络规模和网络流量增加始终可以在较短时间内完成增量流量的调度, 保证了动态变化网络环境下实时流量的传输性能。

关键词: 时间敏感网络; 802.1Qbv; 路由; 在线调度

中图分类号: TP393

文献标志码: A

doi: 10.11959/j.issn.2096-3750.2023.00318

An algorithm for joint optimization of dynamic routing and scheduling in time-sensitive networking

ZHOU Yang¹, CHEN Honglong², ZHANG Lei¹

1. Nanjing University of Posts and Telecommunications, Nanjing 210003, China

2. China University of Petroleum, Qingdao 266580, China

Abstract: Time-sensitive networking (TSN) is a set of protocols developed by the IEEE TSN task group, aiming at achieving deterministic communications over Ethernet. As the implementation method of TSN traffic scheduling is not specified in the protocols, the routing and scheduling algorithm for TSN remains an open issue. The joint optimization problem of routing and scheduling in TSN for industrial applications was modeled, and then an online heuristic algorithm was proposed to deliver the routing and scheduling solution for dynamic traffics. The routing path was determined by optimizing both the transmission delay and network load factors, and the scheduling time was quickly conducted by twice clipping operations. Finally, a simulated TSN testbed was developed with NeSTiNg framework based on OMNeT. The simulation results show that the execution time of the proposed algorithm outperforms the baseline algorithms even with large scale of network size and network traffics. It shows that the proposed algorithm guarantees the real-time performance even in dynamically changing networks.

Key words: time-sensitive networking, 802.1Qbv, routing, online scheduling

0 引言

时间敏感网络 (TSN, time-sensitive networking)

是 IEEE 正在制定的一项新的网络通信标准, 致力于在以太网底层实现统一的确定性网络通信^[1]。

TSN 通过定义一系列的实时通信协议, 为不同优先

收稿日期: 2022-09-03; 修回日期: 2022-12-28

通信作者: 张雷, lei.z@njupt.edu.cn

基金项目: 国家自然科学基金资助项目 (No.61971235, No. 52105553)

Foundation Items: The National Natural Science Foundation of China (No.61971235, No. 52105553)

级的流量提供实时服务质量 (QoS) 保障^[2]。例如, 802.1Qbv 定义了基于时间感知的整形器 (TAS, time aware shaper)^[3], 通过门控列表 (GCL, gate control list) 控制不同优先级队列的开关状态, 利用时分复用的方式实现不同优先级流量传输互不干扰; 802.1Qav 中定义了基于信用的整形器 (CBS, credit based shaper) 机制^[4], 利用信用令牌平滑流量的传输速率, 避免低优先级队列的长时间阻塞。

由于 TSN 中还没有制定流量的调度实现方法, 研究者针对 TSN 提出了很多调度算法。数据包在 TSN 中的传输时延不仅与单个端口上的整形算法有关, 也与其在网络中经过的路径有关, TSN 中的调度问题研究可以分为调度算法以及路由与调度联合优化算法两类。

第一类问题考虑传统的工业以太网通过快速生成树算法生成树状拓扑, 流量经过的网络路径固定, 然后求解流量的调度问题。针对此问题, 研究者已经提出很多算法。早期工作通常通过可满足性模数理论/优化模数理论、整数线性规划 (ILP, integer linear programming) 求得最优解, 但这些方法只适用于节点规模极其有限的小规模网络。为了适应更大的网络规模, 研究者们提出了很多启发式和元启发式算法。如 Steiner 等^[5]在调度过程中, 将空白时隙并入流量, 来均衡链路中未分配流量的时隙的位置和大小, 更好地为非时间触发流量提供传输服务。Dürr 等^[6]将 TSN 中的调度问题建模为无等待车间作业调度问题, 提出了一种禁忌搜索算法求解, 并针对调度结果进行压缩合并, 减少门控状态变化次数。Li 等^[7]提出了基于时隙窗口的调度方法, 以流量组的最大公约数为基础周期, 对于周期相同的流量, 允许错峰共用同一段时隙, 尽可能减少每个周期内的时隙数量。当网络规模变大时, 这些算法复杂度将迅速增加, 因此只适合作为离线的调度算法选择^[8]。

第二类问题同时考虑流量的调度和路由。Schweissguth 等^[9]提出一个 ILP 模型, 然后利用 ILP 求解器直接得到 TSN 流量在网络中的路由与调度方案。Falk 等^[10]分析了网络参数对 ILP 算法性能的影响, 发现流量的数量和网络拓扑复杂程度对算法运行时间有显著影响。Schweissguth 等^[11]根据流量的周期、报文大小和超周期进行演算, 并作为先验知识简化 ILP 求解中对冲突约束的计算, 提高求解速度。Hellmanns 等^[12]在输入、模型生成以及求解

器特性层面减小解搜索范围, 提高算法求解速度。流量调度问题复杂度已经被证明是 NP 难的, 因此路由与调度的联合优化问题复杂度更高, 往往不能直接得到路由与调度的有效解方案。因此也有很多工作采用两步法, 对路由与调度问题分别求解。Pahlevan 等^[13]提出一种基于遗传算法的启发式调度方法。算法中以流量组的路由选择作为基因组, 流量个体的路由作为基因, 选定路由组后, 先对流量依据其相互依赖性进行排序, 再按序进行时隙分配, 并将得到的时延作为适应度函数进行优化, 提高了资源利用率。Alnajim 等^[14]提出一种基于 QoS 的路由和调度方法, 考虑路径长度、链路剩余带宽等指标选择路由路径, 调度中允许等待, 以提高流量的可调度性。Song 等^[15]通过 K-means 算法对流量进行分类, 采用最短路径算法 (KSP, K-shortest path) 寻找路径, 然后提出改进的蚁群算法计算调度转发时间。Atallah 等^[16]通过不同流量之间路由的相似程度、报文大小和周期大小分析冲突程度, 将冲突程度最小的路由组合作为路由路径。Li 等^[17]提出了可靠路由选择的启发式算法, 以最小化路由、流量调度的成功数和链路负载作为优化目标, 为多播流量提供冗余路径。上述所提出的算法同样适用于静态的网络配置, 求解的时间和空间复杂度随着网络或流量规模增加而增加。

上述的 TSN 路由与调度优化算法都是面向静态网络, 生成固定的网络配置。然而, 对于流量可能发生动态变化的工业应用场景, 需要在网络运行时动态生成网络配置, 持续保障工业实时数据传输服务质量。因此, 有必要面向网络动态变化的场景, 提出在线的网络配置方法。这些方法也同样分为调度算法以及路由与调度联合优化算法两类。

对于传输路径固定的网络, Nasrallah 等^[18]提出了基于 GCL 周期窗口的调度方法, 通过动态调整不同优先级流量的带宽比例, 适应网络中的流量波动。Reusch 等^[19]也提出一种基于 GCL 周期窗口的调度算法, 由估算的时延上界确定初始窗口大小, 然后对不符合时延约束的流量调整其传输时隙窗口大小。Zhou 等^[20]提出了一种动态 GCL 周期窗口调整方法, 当实际流量小于预期时, 其时隙窗口会以固定的步长缩短。

对于需要同时求解路由与调度的网络, 为了提高求解速度, 通常采用增量的方式进行动态路由与调度优化。冯泽坤等^[21]将 TSN 动态路由与调度联

合优化问题建模为 ILP 模型，对增量的每条流量同时求解路由与调度转发时间。Huang 等^[22]为每条增量流量提出在线路由与调度算法，权衡候选路径经过的跳数和流量条数选择路由路径，并利用 ILP 模型求解帧转发时间。Huang 等^[23]提出一种周期感知的路由算法，分析流量周期对链路剩余带宽的影响，选择能够容纳更多其他流量的路由，提高调度成功率。Li 等^[24]考虑工业物联网应用中的 TSN 无等待路由与调度问题，考虑传输时延以及剩余带宽确定每条增量流量的最优路由路径，为时间关键流量按照传输时延进行带宽分配，进而确定每个交换机的 GCL 周期。Syed 等^[25]面向车联网中的可靠性路由优化问题，提出 4 种路由打分指标，分别考虑流量带宽需求、链路剩余带宽、瓶颈链路负载和链路负载的标准差/平均值等影响因素。Yu 等^[26]为支持多播的 TSN 提出在线路由与调度优化方法，对流量终端节点变动引起的流量变化，基于最小距离树原理构建新的路由与调度方案。Li 等^[27]先通过 KSP 算法求解增量流量的路由路径，按照传输时延对多播流量进行分簇，在簇内利用 ILP 模型求解调度以加快求解速度。从上述工作可以看出，目前大部分动态调度算法都选择将路由问题与调度问题分别求解，路由优化考虑链路负载、链路剩余带宽等指

标进行路由路径评分，对于调度则多采用 ILP 求解流量的调度时间。

本文面向工业应用中的动态网络配置需求，建立 TSN 的路由与调度联合优化模型，提出一个启发式算法求解增量流量的路由与调度配置方案。本文主要创新点如下：1) 综合比较了 4 种路由打分指标，基于实验分析得知，综合考虑传输时延和链路负载均衡指标表现最优，选择它们作为本文路由指标。2) 提出了一个启发式调度算法解决 TSN 的无等待调度问题，利用两次剪裁快速生成调度时间。3) 通过仿真网络实验，验证了所提的路由与调度联合优化算法的有效性。

1 系统模型

1.1 网络模型

工业 TSN 由终端节点 ES、交换机节点 SW，以及全双工物理链路组成。终端节点是工业现场的控制器、传感器、执行器等。交换机节点有多个端口，通过物理链接将其与其他交换机和终端节点连接起来。假设网络中的所有节点都具备 TSN 功能，支持 802.1ASrev^[28]和 802.1Qbv^[29]标准。因此网络中所有节点的时钟都进行了时间同步，使用 TAS 调节数据帧的传输时间。TSN 交换机模型如图 1 所示，

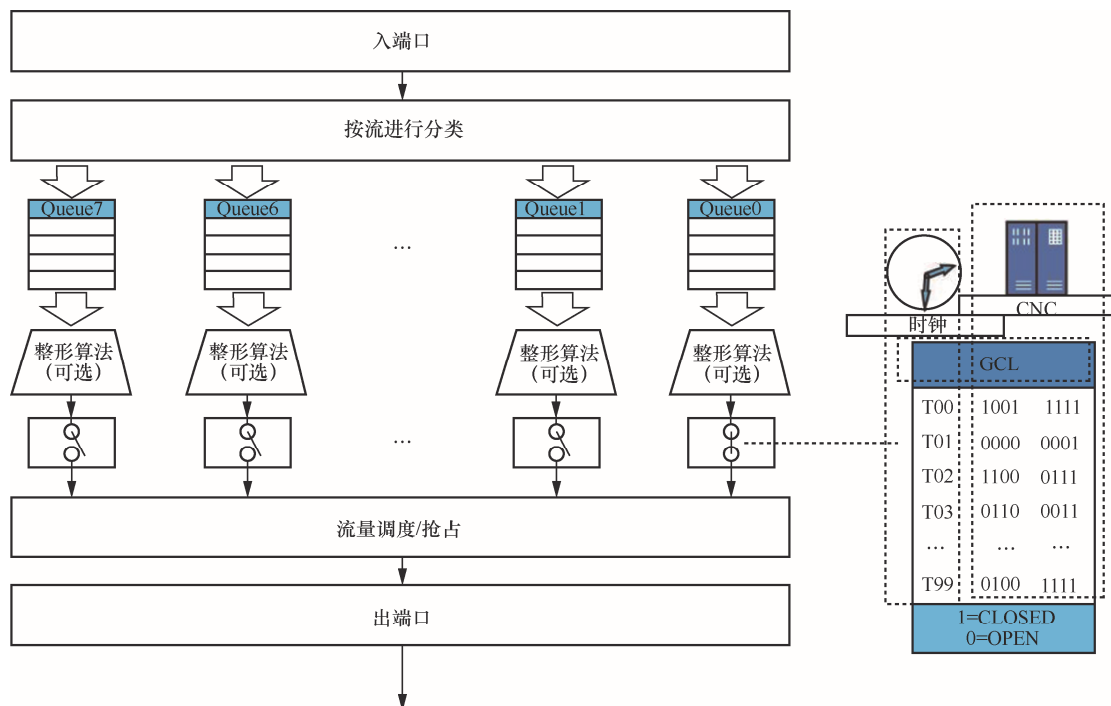


图 1 TSN 交换机模型

TSN 交换机节点的每个出端口有 8 个优先级队列, 每个优先级队列都定义了一个定时门, 在给定的时间内可以处于启用或禁用状态。当门处于打开状态时, 帧可以从各自的队列中选择, 以先入先出的方式传输到物理链路。如果门处于关闭状态, 则不会从相应队列中选择帧。因此数据帧的传输可以通过管理门控状态, 即关闭或打开来控制。门控状态通常根据预先定义的 GCL 中的条目改变。GCL 中的每个条目包含 1 个门操作, 以及该门操作的持续时间。每个门操作的键值定义了 8 个队列的门控状态。TSN 交换机通过 GCL 对各流量队列的传输时间进行集中配置。

将 TSN 建模为有向图 $G = \langle V, E \rangle$, 其中 $V = ES \cup SW$ 表示网络中所有设备的集合。为了方便问题描述, 本文假设所有的交换机节点具有相同的属性, 如带宽、端口数等。 E 表示网络中所有物理链路的集合, 支持全双工数据传输。每条全双工物理链路由二元组 (v_i, v_j) 和 (v_j, v_i) 表示, 其中第 1 个元素为发送设备, 第 2 个元素定义为接收设备。TSN 拓扑示意图如图 2 所示, 包含了 3 个 SW 节点和 8 个 ES 节点。实线代表了节点之间的物理链路, 以 ES1 和 SW1 之间的物理链路为例, 它们之间有两个独立的有向链路, 分别用 $[ES1, SW1]$ 和 $[SW1, ES1]$ 表示。

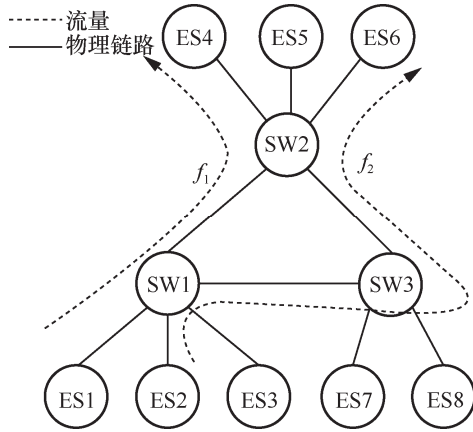


图2 TSN 拓扑示意图

1.2 流量模型

TSN 将工业应用中具有严格时间约束的流量定义为控制流 (CT, control traffic)。此类流量具有最高优先级, 以保证其网络转发时间的确定性和低时延。假设输入为待调度的一组控制流量集合为 F , 用一个五元组描述其中一条控制流量为 $f_i = (v_{src}^i, v_{dst}^i, T_i, l_i, D_i)$, 其中 v_{src}^i 为流量 f_i 的发送终

端, v_{dst}^i 为流量 f_i 的目的终端, T_i 为流量 f_i 的传输周期, l_i 为流量 f_i 的报文长度, D_i 为流量 f_i 的最大允许传输时延。时延约束确保每个控制流量能够在最大允许时延内到达目的终端。

假设流量 f_i 在网络中经过的所有可能路径有 s 条, 路由集合记为 γ_i 。 r_{ij} ($j \in [1, \dots, s]$) 为路由集合 γ_i 中的一条路径, 从发送终端 v_{src}^i 经过 k 个中间节点 $v_1^i, v_2^i, \dots, v_k^i$ 到达终端节点 v_{dst}^i , 则路由 r_{ij} 可表示为 $((v_{src}^i, v_1^i), (v_1^i, v_2^i), \dots, (v_k^i, v_{dst}^i))$ 。用 $t_i^{p,q}$ 表示流量 f_i 在一个超周期内第一帧数据在链路 (v_p^i, v_q^i) 上的转发时间, 则所有链路上的转发时间集合 $\theta_i = \{t_i^{src,1}, t_i^{1,2}, \dots, t_i^{k,dst}\}$ 称为流量 f_i 的调度表。

流量 f_i 在网络中的传输时延 d_i 由 4 个部分组成, 分别为发送时延、传输时延、处理时延和排队时延。发送时延 t_i^{sd} 主要与报文长度有关; 传输时延 t_i^{tr} 主要与链路长度有关; 处理时延 t_i^{pr} 是交换机转发一帧数据的时间, 主要与交换机性能有关; 排队时延 t_i^{qw} 是流量在网络中的排队等待时间, 与网络中的转发路径和路径上每一跳的转发时间有关。端到端时延 d_i 为

$$d_i = \text{len}(r_{ij}) \cdot (t_i^{sd} + t_i^{tr} + t_i^{pr}) - t_i^{pr} + t_i^{qw} \quad (1)$$

2 问题模型

TSN 设计的首要目标是保障工业应用中具有严格时间约束的控制流量的实时性能。对于给定的网络和流量, t_i^{sd} 、 t_i^{tr} 、 t_i^{pr} 均为固定常数, 端到端时延主要取决于流量在网络中的排队时延。如果一个有严格时间要求的数据帧在交换机的出端口队列中等待, 将导致一个非确定的排队时延。如果该数据帧的端到端时延 d 超过其最大允许时延 D , 则无法满足工业应用要求。对此, TSN 的方法是利用流量调度, 完全消除从源设备到目的设备路径上的交换机的非确定性排队时延, 控制流量的数据包到达交换机时间, 以零排队时延被转发。

TSN 设计的另一个挑战是 TSN 动态应用的出现, 如工厂中机器的动态插拔, 需要在网络配置发生变化时, 在保证调度性能的前提下, 迅速生成路由和调度时间表。对此, 一般做法是尽可能地重用原来的时间表, 以减少网络变化时的计算, 而不是完全更新它。

因此,本文要解决的问题可以描述为:对于给定的 TSN 拓扑结构 G ,原有流量集合 U ,新增一组待调度的控制流量 F 。在不改变网络中原有调度的前提下,为待调度的流量 F 快速寻找一个合适的路由与调度结果,消除由意外传输冲突引起的控制流量的非确定性排队时延,使所有控制流量的端到端时延要求都能得到满足。

3 动态路由与调度联合优化算法

TSN 中的路由求解问题已经被证实是 NP 难问题,流量调度问题也被证明为 NP 难问题,所以同时求解路由与调度问题也同样是 NP 难问题。采用 ILP 算法虽然可以给出问题的最优解,但是随着流量条数增加,求解的时间会以指数增长,需要花几个小时甚至更多时间得到优化的解决方案。而已有的启发式算法对全局网络流量求解,也会耗费较长的时间。这意味着一旦网络发生改变,调度算法需要较长的执行时间重新调度流量使其满足实时性需求,因此不适合网络中流量动态变化的场景。为了达到合理的运行性能,本文将路由与调度联合优化问题解耦为路由与调度两个子问题,并分别求解。

3.1 算法流程

动态路由与调度联合优化算法见算法 1。首先找到受网络变化影响的增量流量集合,并按照紧急程度对流量排序。规定周期越短的流量优先级越高;对于周期相同的流量,帧长度越大的流量优先级越高。对每一条待调度流量,根据网络拓扑寻找其候选路由集。然后从候选路由中选择一条路径,对流量进行调度。若调度成功,则保存当前路由路径及调度结果,并对已有的路由与调度计划进行更新。若调度不成功,选择下一条路由继续进行调度。如果候选路由中已经没有其他路由,则该流量调度失败,并对下一条动态流量进行路由与调度求解,直到所有流量都完成路由与调度求解。

算法 1 动态路由与调度联合优化算法

输入 TSN 拓扑 G 、待调度的流量集合 F 、原有流量集合 U 、对应的路由配置 R_0 和调度配置 θ_0

输出 待调度流量 F 的路由配置 R 和调度配置 θ
 $F \leftarrow \text{FlowSort}(F)$

$\theta \leftarrow \emptyset, R \leftarrow \emptyset$

```

for  $f_i$  in  $F$  do
   $\gamma_i \leftarrow \text{Routing}(f_i, G, U, R_0)$ 
  for  $r_{ij} \in \gamma_i$  do
     $\theta_i \leftarrow \text{Scheduling}(r_{ij}, f_i, \theta_0, R_0)$ 
  if  $\theta_i \neq \emptyset$  do
     $R_i \leftarrow r_{ij}$ 
     $\theta_0 \leftarrow \theta_i, R_0 \leftarrow R_i$ 
    break
  end if
end for
if  $\theta_i = \emptyset$  do
  流量  $f_i$  调度失败
end if
end for

```

3.2 路由算法

本节介绍候选路由选择的过程,路由算法见算法 2。首先,筛选出流量在拓扑中所有可行的候选路由;然后,对每条候选路由进行评分;最后,依据每条候选路由的评分对所有的候选路由进行排序。

算法 2 路由算法

输入 TSN 拓扑 G 、待调度的流量 f_i 、原有流量集合 U 、对应的路由配置 R_0

输出 待调度流量 f_i 的候选路由 γ_i

$\gamma_i \leftarrow \text{FindRouts}(f_i, G)$

$\gamma_i \leftarrow \text{Filter}(\gamma_i)$

for r_{ij} in γ_i do

$M_i \leftarrow \text{Score}(r_{ij}, R_0, U)$

end for

$\gamma_i \leftarrow \text{SortM}(r_i, M_i)$

寻找候选路由模块 FindRouts() 首先通过广度优先算法遍历找出从 f_i 发送终端至目的终端的所有路由。

筛选路由模块 Filter() 按时延约束删除不符合条件的候选路由。如第 1.2 节所述,流量 f_i 要求能够在最大允许时延 D_i 之前到达目的节点,即

$$d_i < D_i \quad (2)$$

由于控制流量在 TSN 中实现无等待转发,式(1)可以简写为

$$d_i = \text{len}(r_{ij}) \cdot (t_i^{\text{sd}} + t^{\text{tr}} + t^{\text{pr}}) - t^{\text{pr}} \quad (3)$$

将式(3)代入式(2),得到时延约束限制下路由的

最大节点跳数。最后以此条件删除不符合时延约束的候选路由。

$$\text{len}(r_{ij}) < \frac{D_i + t^{\text{pr}}}{(t_i^{\text{sd}} + t^{\text{tr}} + t^{\text{pr}})} \quad (4)$$

得到所有满足延时约束的路由集合后, 评分模块 $\text{Score}()$ 用于给所有路由评分, 评分函数有多种形式。传统基于最短路径的路由算法将导致大量流量导入瓶颈链路, 可能带来严重的局部网络拥塞问题, 造成网络资源浪费。对流量进行恰当的路由分流能够很好地避免此问题。因此本文在最短路径的基础上, 考虑不同的路由分流方式, 提出3种不同评分函数。

考虑路径长度和网络负载均衡性的评分函数为

$$\text{Score}_1(r_{ij}) = w \cdot \text{len}(r_{ij}) + (1-w) \cdot \sigma(B) \quad (5)$$

其中, w 为权重系数, $\text{len}(r_{ij}) = k$ 是选择的路由路径 r_{ij} 上经过的节点数, 利用流量经过网络的路径长度表征流量经过网络的端到端时延。用 $b_p^{p,q}$ 表示链路 (v_p^i, v_q^i) 的负载, B 为拓扑中所有链路负载的集合。 $\sigma(B)$ 为网络中所有链路的负载标准差, 用于衡量不同链路对网络负载分布的影响。在求标准差之前需要对相应链路负载进行更新, 新增一条流量 f_i , 负载由其报文长度与周期计算得到, 流量在其经过链路的负载更新为 $b_p^{p,q} = b_p^{p,q} + l_i / T_i, \forall (v_p^i, v_q^i) \in r_{ij}$ 。

考虑路径长度和所选路由的剩余带宽的评分函数为

$$\text{Score}_2(r_{ij}) = w \cdot \text{len}(r_{ij}) + (1-w) \cdot \frac{1}{\min(B_0 - B_{ij})} \quad (6)$$

其中, B_0 为链路的负载能力, B_{ij} 为新增流量 f_i 后链路的当前负载。

考虑路径长度和网络链路的最大负载的评分函数为

$$\text{Score}_3(r_{ij}) = w \cdot \text{len}(r_{ij}) + (1-w) \cdot \max(B) \quad (7)$$

3.3 调度算法

本节介绍调度表的生成, 调度算法见算法3。首先网络中所有流量发送周期的最小公倍数定义为超周期 L 。假设 T_F 是网络中所有流量发送周期的集合, 则 $L = \text{LCM}(T_F)$ 。流量的传输以一个超周期为单位重复进行, 因此我们只需要在第一个超周期 L 中进行流量调度。

本算法将流量跨度的最小化作为调度算法的优化目标。流量跨度指在一个超周期的范围内, 从超周期的开始时间计算, 直到所有流量中最晚到达目的节点的流量传输完毕所经过的时间跨度^[30]。优点是生成相对紧凑的调度, 使流量不会分散在超周期内, 而是尽可能地被压缩到周期的开始。这使得同一优先级的不同流量合并时隙的可能性大大增加, 减少门控状态变化次数, 则相应的保护带宽需求也变少, 可以减少带宽浪费。流量跨度的压缩也为其他优先级的流量提供了更大的连续的时间, 为发送超长带宽的其他优先级流量提供更多可能性。

对于调度对象流量 f_i , 首先, 基于原有调度表, 寻找路由经过的所有链路上尚未被占用的时间片; 然后, 对这些时间片进行两次裁剪操作, 使其满足相关约束条件, 包括无等待约束和互不冲突约束, 如式(8)和式(9)所示。若剪裁后的时间片长度不小于流量 f_i 的传输时间, 则选择时间片的下界作为流量 f_i 的帧偏移发送时间, 并对已有调度进行更新; 否则流量 f_i 调度失败, 调度结果为空。

算法3 调度算法

输入 待调度的流量 f_i 、进行调度的路由 r_{ij} 、原有流量对应的路由配置 R_0 和调度配置 θ_0

输出 待调度流量 f_i 的调度结果 θ_i

$\theta_i \leftarrow \text{FindTimeSlot}(r_{ij}, \theta_0, R_0)$

$\theta_i \leftarrow \text{Cut}(\theta_i)$

if $\theta_i \neq \emptyset$ do

$\theta_i \leftarrow \min(\theta_i)$

else do

$\theta_i \leftarrow \emptyset$

end if

无等待约束: 控制流量到达交换机后, 不经过任何等待直接进行转发, 如式(8)所示。

$$t_i^{\alpha, \alpha+1} - t_i^{\alpha-1, \alpha} = t^{\text{tr}} + t^{\text{pr}} + t_i^{\text{sd}}, \forall \alpha \in \{2, \dots, k\} \quad (8)$$

互不冲突约束: 同一时刻, 属于不同流量的两个数据帧不允许在同一条链路上进行传输, 否则会发生碰撞。假设链路 (v_p, v_q) 上的两条周期不同的两条流量 f_i, f_r , 以超周期的起始时刻作为0点, 其发送时刻为 $t_i^{p,q}$ 和 $t_r^{p,q}$, 不考虑发送时延的情况下, 传输时延为 t_i^{sd} 和 t_r^{sd} , 设 g 为两条流量周期的最大公因数, 当两个流量在其周期的最大公因数内互不冲突, 那么它们在各自的任意周期内也互不冲突, 如式(9)所示。

$$\begin{aligned}
 t_i^{sd} &\leq (t_i^{p,q} - t_i^{p',q'}) \bmod g \leq g - t_i^{sd}, \\
 \forall f_i, f_{i'} \in F, \exists (v_p^i, v_q^i) \in r_{ij}, \\
 (v_{p'}^{i'}, v_{q'}^{i'}) \in r_{i'j'}, (v_p^i, v_q^i) &= (v_{p'}^{i'}, v_{q'}^{i'})
 \end{aligned}
 \tag{9}$$

Cut() 模块中裁剪分两步进行。对于周期小于超周期的流量，在一个超周期内会发送多次，每个发送帧被称为帧实例。一个超周期内，待调度流量 f_i 的帧实例个数表示为 $N_i = \frac{L}{T_i}$ 。

首先基于流量 f_i 在网络中经过的每条链路进行第一次裁剪，选择一个超周期内的 N_i 个帧实例对应的候选时间片取交集，确保流量 f_i 的每个帧实例都能以相同的帧偏移时间进行传输。然后基于流量 f_i 在网络中经过的所有链路进行第二次剪裁，确保流量 f_i 经过网络时为无等待转发，同一帧实例在相邻链路上的转发时间满足式(8)约束条件。

以图 2 拓扑中的流量 f_1 为例，在网络中的传输路径经过了链路[ES1, SW1]、[SW1, SW2]和[SW2, ES4]。首先针对每条链路，对流量 f_1 在一个超周期内的所有候选时间片进行偏移，偏移量为所在发送周期的起始时间，然后对偏移后的所有候选时间片取交集，得到流量 f_1 在该链路的候选时间片。同一链路上的候选时间片裁剪操作如图 3 所示。以此类推，得到传输路径中其余链路的候选时间片。剪裁后每个传输周期的发送时间片保持一致，这意味着如果一个发送周期内的调度满足约束条件，则整个超周期内的调度也满足约束条件。随后对一个传输周期内流量 f_1 经过的所有链路上的候选时间片取

交集完成无等待转发的约束条件。相邻链路上的候选时间片裁剪操作如图 4 所示，此处的虚线表示经过发送、传输、处理后，相邻节点所对应的的时间片。经过两次裁剪操作，最终得到流量 f_1 的可选时间片，选择时间片的最小时间作为一个发送周期内的帧偏移发送时间。

3.4 算法复杂度分析

假设网络中原有流量 m 条，新增待调度的流量 n 条。上述算法实现中，对流量进行优先级排序的时间复杂度为 $O(n^2)$ ，每条流量找到 s 条路由并进行排序的时间复杂度为 $O(s^2)$ ，调度算法的主要操作是寻找多个时间片求取交集，而每次调度所能找到的候选时间片数量与网络中原有流量条数有关，因此调度表生成的时间复杂度为 $O(m)$ 。因此，本文提出的动态流量路由与调度联合优化算法的时间复杂度为 $O(n^2 + n(s^2 + m))$ 。

4 实验结果

NeSTiNg 是基于流行的 OMNeT++/INET 离散事件模拟框架，专门用于搭建 TSN 仿真平台以及相关的实验验证的仿真框架^[31]。本文基于 OMNeT++ 的开源仿真框架 NeSTiNg，搭建一个仿真的 TSN 平台，对算法进行性能评估。

4.1 参数设置

算法的运行环境为 Python3.7，操作系统为 Windows11，计算机处理器为 Intel i7 8750H 6 核，主频为 2.20 GHz，内存 8 GB。

选择如图 3 所示的网络作为实验拓扑，网络中

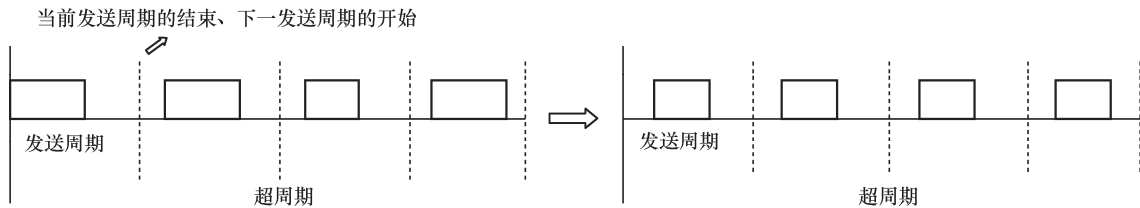


图 3 同一链路上的候选时间片裁剪操作

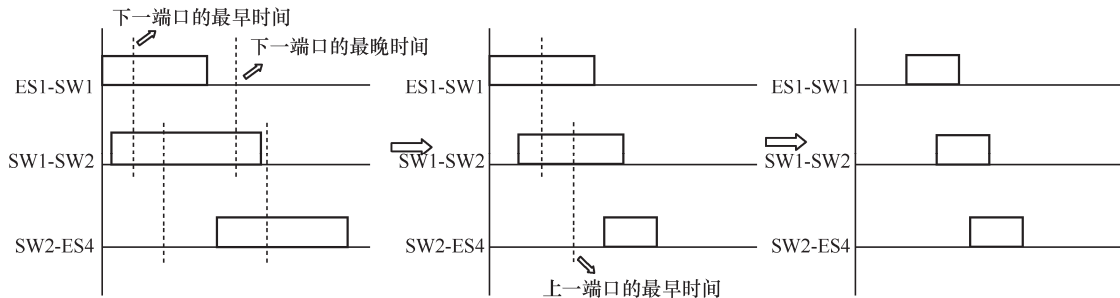


图 4 相邻链路上的候选时间片裁剪操作

包含3台交换机,8个终端和11条全双工物理链路。假设交换机的服务能力相同,带宽均为1 Gbit/s。交换机处理时延设置为5 μs,全双工物理链路传输时延设置为0.1 μs。

网络中的流量随机生成,发送周期为{1 ms, 2 ms, 4 ms},帧大小范围为[64 byte, 1 512 byte],最大允许时延为0.2 ms。

4.2 数值结果分析

首先,对比不同路由优化指标对算法性能的影响。在图2的拓扑上,随机生成300、600、900、1 200条流量,以第3.2节所提的3种不同路由优化指标以及最短路径算法对其进行调度。路由优化指标对调度性能的影响比较如图5所示,可以看出,当流量数量较少时,任何路由指标都能在一次完成每条流量的调度。随着流量数量增加,最短路径算法需要更多的调度次数来寻找可行路径完成调度。当流量的条数增加到接近链路负载1 Gbit/s时,以网络最大负载和链路剩余带宽为指标的优化方案相比最短路径算法稍有下降,而网络负载均衡指标表现出更少的平均调度次数,并且调度成功率更高。可见网络负载均衡指标通过综合考虑路由路径长度和网络链路负载的标准差,尽可能达到网络中的负载分布均衡,避免了个别节点上的负载拥塞对新增流量调度的影响,且没有因过度追求负载均衡而导致路由复杂,进而导致网络拥塞。

然后,比较了不同调度算法对算法性能的影响。作为动态调度算法,算法的执行时间至关重要,将算法执行时间作为主要的性能指标。仍然以图2为网络拓扑,随机生成10、20、30和40条流量,选取添加流量和链路故障两种典型测试场景。将基于ILP的动态流量均衡调度算法^[21]作为对比算法,算法执行时间比较如图6所示。从结果上来看,不论是新增一条流量还是一条链路故障的情况,所提算法执行时间都远小于对比算法。

从算法的时间复杂度来分析,对于n条待调度流量来说,在拓扑较为简单的情况下,所提算法的时间复杂度可简化为 $O(n^2)$ 。而对于基于ILP的动态流量均衡调度算法来说,其时间复杂度通常为多项式级的,多项式级的时间复杂度往往会因为算法的发散而导致其复杂度呈指数级增长,因此,在相同规模的网络环境中,基于ILP的动态流量均衡调度算法的时间复杂度往往都大于所提算法。

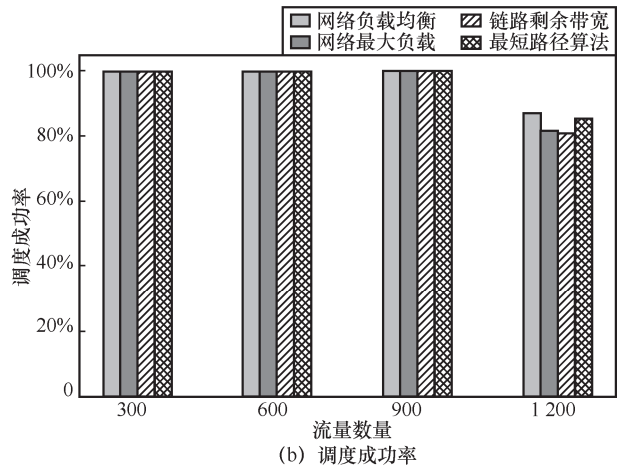
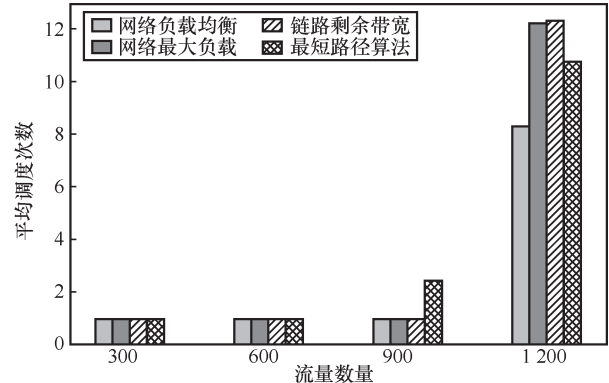


图5 路由优化指标对调度性能的影响比较

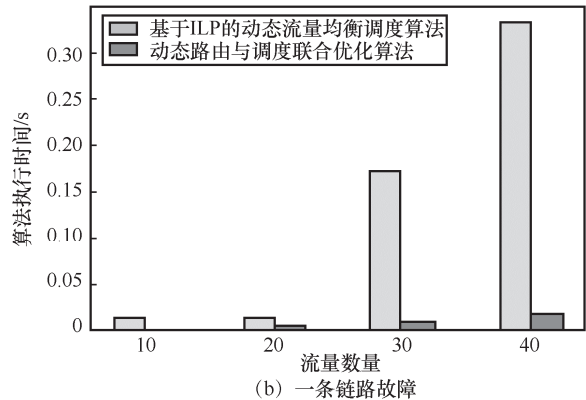
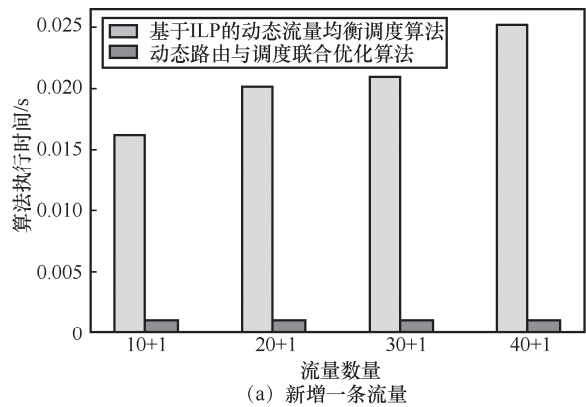


图6 算法执行时间比较

为了更好地分析所提算法的性能,不同流量和拓扑规模对算法性能的影响比较如图 7 所示。在图 2 拓扑下,图 7(a)展示了不同的流量数量下,平均每条流量的调度时间的最小值、平均值和最大值。最小值为优先调度的流量调度时间,值接近于 0 并且保持不变。平均值随着流量的增加缓慢增加,但当流量数量达到 1 200 条时,平均调度时间仍然低于 0.01 s。最大值则是最后调度的流量,调度时间随着链路负载增加而增加,从 0.01 s 增至 0.05 s。

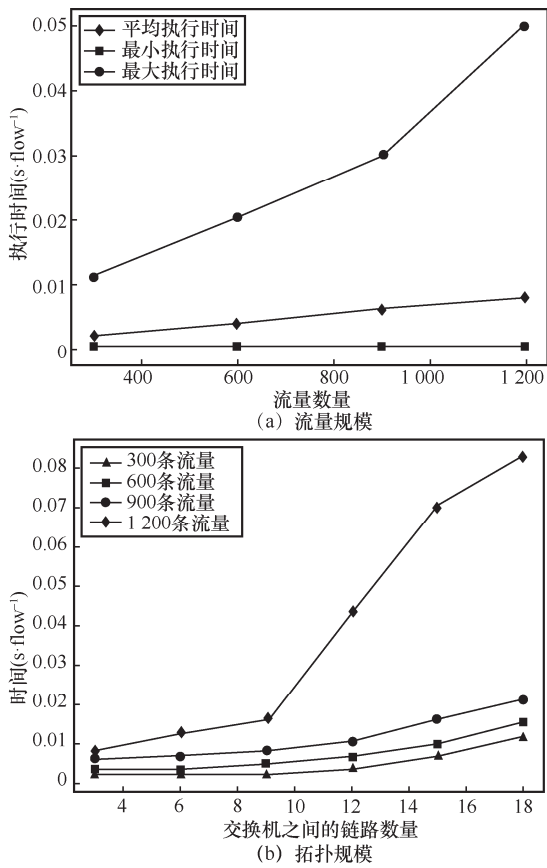


图 7 不同流量和拓扑规模对算法性能的影响比较

图 7(b)展示了网络拓扑中的链路数量对每条流量的平均调度时间的影响。网络拓扑依据 ER 模型^[32]随机生成具有 3、6、9、12、18 条交换机链路的拓扑,可以看出,随着拓扑链路数量的增加,算法执行时间也随之增加。从 300 条到 900 条流量的情况,执行时间从 0.002 s 增加到 0.021 s,增长趋势比较平缓。原因是网络中的 900 条流量尚未超出链路的负载能力,大多数流量在第一条路径就能调度成功。而当流量的条数达到 1 200 条,执行时间从 0.007 8 s 增加到 0.082 s,时间增加较为明显。因为此时,流量数量接近链路的最大负载能力,很多流量需要遍历

更多的候选路由才能成功调度。同时复杂的拓扑带来候选路由的数量增加,在提高调度成功率的同时,也增加了调度时间。

总而言之,可以看出随着网络的规模不断变大,网络中流量负载不断增加,所提算法仍然表现出优秀的性能,平均每条流量的调度时间始终保持在毫秒级,可以很好地适应网络动态变化的应用场景。

4.3 仿真案例分析

最后,在仿真 TSN 平台上进行一个具体的案例分析。网络拓扑如图 2 所示,网络中控制流量的数量为 10 条。通过静态调度算法对其进行调度,并配置相应的 GCL。然后在此基础上随机新增一条控制流量,调度前后的网络时延分布如图 8 所示。

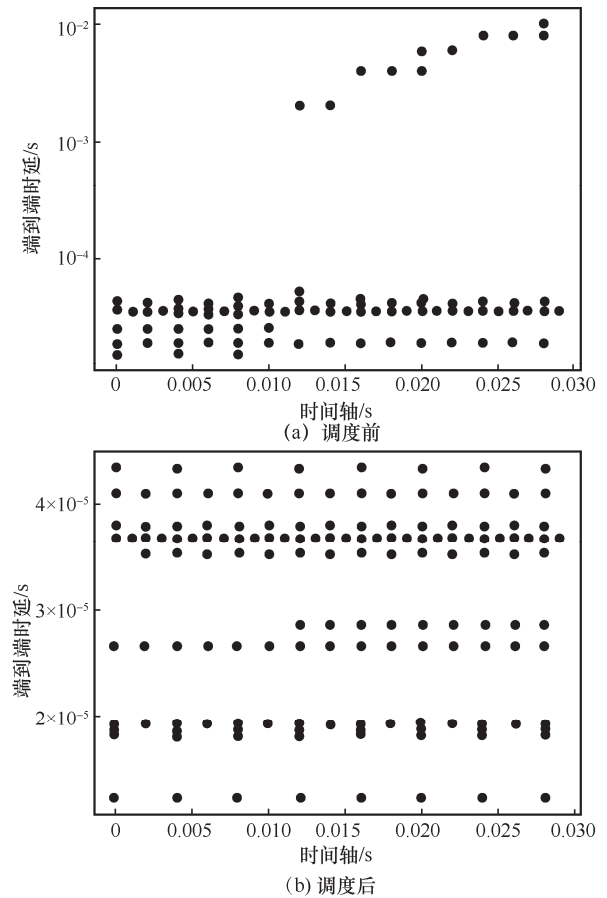


图 8 调度前后的网络时延分布

从图 8 可以看出,未调度前新增流量会造成原有部分流量的端到端时延增加。而调度后所有流量,包括新增流量,都可以实现无等待转发。调度前后网络主要链路的甘特图如图 9 所示。由图 9(a)可知,对于新增流量 f_{i1} ,调度前调度表中并未对其

分配时隙, 因此该流量只能在能够供它传输的流量 f_7 、 f_{10} 的合并时隙进行传输。由于交换机采用的是不可抢占式报文转发, 因此在流量 f_{11} 完成传输之前, 其他流量到达该端口后只能在队列中排队等待, 导致这些流量无法在被分配的传输时间片进行传输, 进而导致该链路流量的时延大大增加。在图 9(b)中, 利用所提算法对流量 f_{11} 进行调度后, 在链路 SW2-SW1 上分配了专用时隙, 并调度了该流量的到达时间, 因此该流量与链路上的其他流量的冲突解除, 原来冲突导致的时延增加也恢复正常。

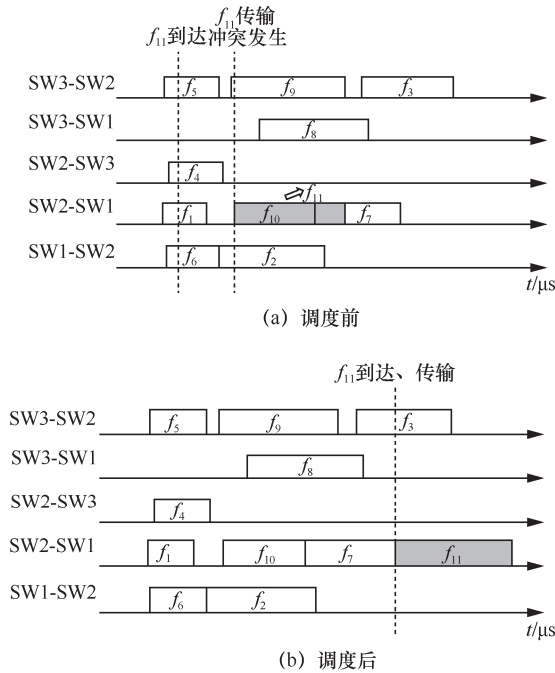


图9 调度前后网络主要链路的甘特图

5 结束语

本文针对流量动态变化的场景, 提出 TSN 的动态路由与调度联合优化算法。算法考虑传输时延和链路负载, 提出路由优化指标, 然后对候选时间片进行两次剪裁, 快速生成动态流量的调度表。实验结果表明所提算法在调度成功率和算法执行速度上均显著优于基于 ILP 的动态路由与调度联合优化算法。随着网络规模增大, 网络流量负载增加, 所提算法仍然表现出优秀的性能, 每条流量的平均计算时间始终保持在毫秒级, 可以很好地适应网络动态配置的应用需求。

工业应用中不仅有周期性的实时流量, 也有非周期的实时流量。未来工作将考虑利用在线路由与

调度机制, 探索利用异步流量整形器 (ATS, asynchronous traffic shaper) 机制解决 TSN 中非周期实时流量的传输保障问题。

参考文献:

- [1] 蔡岳平, 姚宗辰, 李天驰. 时间敏感网络标准与研究综述[J]. 计算机学报, 2021, 44(7): 1378-1397.
CAI Y P, YAO Z C, LI T C. A survey on time-sensitive networking: standards and state-of-the-art[J]. Chinese Journal of Computers, 2021, 44(7): 1378-1397.
- [2] FINN N. Introduction to time-sensitive networking[J]. IEEE Communications Standards Magazine, 2018, 2(2): 22-28.
- [3] LOBELLO L, STEINER W. A perspective on IEEE time-sensitive networking for industrial communication and automation systems[J]. Proceedings of the IEEE, 2019, 107(6): 1094-1120.
- [4] ZHAO L, HE F, LI E S, et al. Comparison of time sensitive networking (TSN) and TTEthernet[C]//Proceedings of 2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC). Piscataway: IEEE Press, 2018: 1-7.
- [5] STEINER W. Synthesis of static communication schedules for mixed-criticality systems[C]//Proceedings of 2011 14th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops. Piscataway: IEEE Press, 2011: 11-18.
- [6] DÜRR F, NAYAK N G. No-wait packet scheduling for IEEE time-sensitive networks (TSN)[C]//Proceedings of RTNS '16: Proceedings of the 24th International Conference on Real-Time Networks and Systems. [S.l.:s.n.], 2016: 203-212.
- [7] LI Q, LI D, JIN X, et al. A simple and efficient time-sensitive networking traffic scheduling method for industrial scenarios[J]. Electronics, 2020, 9(12): 2131.
- [8] 张彤, 冯佳琦, 马延滢, 等. 时间敏感网络流量调度综述[J]. 计算机研究与发展, 2022, 59(4): 747-764.
ZHANG T, FENG J Q, MA Y Y, et al. Survey on traffic scheduling in time-sensitive networking[J]. Journal of Computer Research and Development, 2022, 59(4): 747-764.
- [9] SCHWEISSGUTH E, DANIELIS P, TIMMERMANN D, et al. ILP-based joint routing and scheduling for time-triggered networks[C]//Proceedings of RTNS '17: Proceedings of the 25th International Conference on Real-Time Networks and Systems. [S.l.:s.n.], 2017: 8-17.
- [10] FALK J, DÜRR F, ROTHERMEL K. Exploring practical limitations of joint routing and scheduling for TSN with ILP[C]//Proceedings of 2018 IEEE 24th International Conference on Embedded and Real-Time Computing Systems and Applications. Piscataway: IEEE Press, 2018: 136-146.
- [11] SCHWEISSGUTH E, TIMMERMANN D, PARZYJEGLA H, et al. ILP-based routing and scheduling of multicast realtime traffic in time-sensitive networks[C]//Proceedings of 2020 IEEE 26th International Conference on Embedded and Real-Time Computing Systems and Applications. Piscataway: IEEE Press, 2020: 1-11.
- [12] HELLMANN D, HAUG L, HILDEBRAND M, et al. How to optimize joint routing and scheduling models for TSN using integer linear programming[C]//Proceedings of RTNS'2021: 29th International Confer-

- ence on Real-Time Networks and Systems. [S.l.:s.n.], 2021: 100-111.
- [13] 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. Piscataway: IEEE Press, 2018: 337-344.
- [14] ALNAJIM A, SALEHI S, SHEN C C. Incremental path-selection and scheduling for time-sensitive networks[C]//Proceedings of 2019 IEEE Global Communications Conference. Piscataway: IEEE Press, 2019: 1-6.
- [15] SONG Y, GUO C Y, XU P F, et al. Research on routing and scheduling algorithms for the simultaneous transmission of diverse data streaming services on the industrial Internet[J]. Scientific Reports, 2021 (11): 18351.
- [16] ATALLAH A A, HAMAD G B, MOHAMED O A. Routing and scheduling of time-triggered traffic in time-sensitive networks[J]. IEEE Transactions on Industrial Informatics, 2020, 16(7): 4525-4534.
- [17] LI H T, CHENG H, YANG L. Reliable routing and scheduling in time-sensitive networks[C]//Proceedings of 2021 17th International Conference on Mobility, Sensing and Networking (MSN). Piscataway: IEEE Press, 2021: 806-811.
- [18] NASRALLAH A, THYAGATURU A S, ALHARBI Z, et al. Performance comparison of IEEE 802.1 TSN time aware shaper (TAS) and asynchronous traffic shaper (ATS)[J]. IEEE Access, 2019(7): 44165-44181.
- [19] REUSCH N, ZHAO L X, CRACIUNAS S S, et al. Window-based schedule synthesis for industrial IEEE 802.1Qbv TSN networks[C]//Proceedings of 2020 16th IEEE International Conference on Factory Communication Systems. Piscataway: IEEE Press, 2020: 1-4.
- [20] 周小明, 仇梓鑫, 朱瑾瑜, 等. 一种时间敏感网络 GCL 自适应调整算法[J]. 北京邮电大学学报, 2022, 45(1): 26-32.
ZHOU X M, QIU Z X, ZHU J Y, et al. GCL adaptive adjustment algorithm in time sensitive networks[J]. Journal of Beijing University of Posts and Telecommunications, 2022, 45(1): 26-32.
- [21] 冯泽坤, 龚龙庆, 徐丹妮, 等. 时间敏感网络中基于 ILP 的动态流量均衡调度算法[J]. 微电子学与计算机, 2021, 38(6): 33-37.
FENG Z K, GONG L Q, XU D N, et al. ILP-based dynamic flow balancing scheduling algorithm in time sensitive network[J]. Microelectronics & Computer, 2021, 38(6): 33-37.
- [22] HUANG Y D, WANG S, HUANG T, et al. Online routing and scheduling for time-sensitive networks[C]//Proceedings of 2021 IEEE 41st International Conference on Distributed Computing Systems. Piscataway: IEEE Press, 2021: 272-281.
- [23] HUANG K, WU J K, JIANG X W, et al. A period-aware routing method for IEEE 802.1Qbv TSN networks[J]. Electronics, 2020, 10(1): 58.
- [24] LI Y T, JIANG J H, HONG S H. Joint traffic routing and scheduling algorithm eliminating the nondeterministic interruption for TSN networks used in IIoT[J]. IEEE Internet of Things Journal, 2022, 9(19): 18663-18680.
- [25] SYED A A, AYAZ S, LEINMÜLLER T, et al. Fault-tolerant dynamic scheduling and routing for TSN based in-vehicle networks[C]//Proceedings of 2021 IEEE Vehicular Networking Conference. Piscataway: IEEE Press, 2021: 72-75.
- [26] YU Q H, WAN H, ZHAO X B, et al. Online scheduling for dynamic VM migration in multicast time-sensitive networks[J]. IEEE Transactions on Industrial Informatics, 2020, 16(6): 3778-3788.
- [27] LI C, ZHANG C Y, ZHENG W, et al. Joint routing and scheduling for dynamic applications in multicast time-sensitive networks[C]//Proceedings of 2021 IEEE International Conference on Communications Workshops. Piscataway: IEEE Press, 2021: 1-6.
- [28] IEEE standard for local and metropolitan area networks - timing and synchronization for time-sensitive applications in bridged local area networks[J]. IEEE Std 802.1AS-2011, 2011: 1-292.
- [29] IEEE standard for local and metropolitan area networks - bridges and bridged networks - amendment 25: enhancements for scheduled traffic[J]. IEEE Std. 802.1Qbv-2015, 2015.
- [30] HELLMANNNS D, GLAVACKIJ A, FALK J, et al. Scaling TSN scheduling for factory automation networks[C]//Proceedings of 2020 16th IEEE International Conference on Factory Communication Systems. Piscataway: IEEE Press, 2020: 1-8.
- [31] FALK J, HELLMANNNS D, CARABELLI B, et al. NeSTiNg: simulating IEEE time-sensitive networking (TSN) in OMNeT++[C]//Proceedings of 2019 International Conference on Networked Systems (NetSys). Piscataway: IEEE Press, 2019: 1-8.
- [32] NAYAK N G, DÜRR F, ROTHERMEL K. Incremental flow scheduling and routing in time-sensitive software-defined networks[J]. IEEE Transactions on Industrial Informatics, 2018, 14(5): 2066-2075.

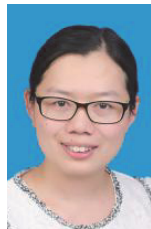
[作者简介]



周阳 (1999-), 男, 南京邮电大学在读, 主要研究方向为时间敏感网络的路由与调度优化。



陈鸿龙 (1984-), 男, 博士, 中国石油大学教授, 中国计算机学会高级会员, 主要研究方向为智能物联网、边缘计算和人工智能安全。



张雷 (1983-), 女, 博士, 南京邮电大学讲师, 主要研究方向为实时通信和网络、边缘计算。