

基于时间感知和循环排队转发的 TSN 联合路由和混合流量调度算法

王颖, 庄智涵, 喻鹏, 邱雪松, 郭少勇
(北京邮电大学网络与交换技术全国重点实验室, 北京 100876)

摘要: 为了解决时间敏感网络 (TSN) 中多种时间敏感流混合调度的问题, 提出了一种基于时间感知 (TA) 和循环排队转发 (CQF) 的时间敏感网络联合路由和混合流量调度算法 (JR-TACQF)。结合时间感知和循环排队转发 2 种流量整形器, 设计了一种 TACQF 混合流量调度模型, 该模型通过周期选择实现时间感知整形器 (TAS) 队列和 CQF 队列同时开放, 利用 CQF 奇偶队列切换时的死区时间开放 TAS 队列, 从而实现不同流量间的软隔离, 并降低时间触发 (TT) 流和音视频桥接 (AVB) 流之间的相互冲突。根据 TACQF 混合流量调度模型设计了 JR-TACQF 混合流量调度算法, 该算法联合路由进行调度, 进行负载均衡, 从而提高带宽利用率和调度成功率。仿真实验结果表明, JR-TACQF 混合流量调度算法可以有效解决 TSN 混合流量调度问题, 提高调度成功率和带宽利用率, 并可以在较短的执行时间内生成可行的调度方案。

关键词: 时间敏感网络; 时间感知整形器; 循环排队转发; 禁忌搜索; 混合流量调度

中图分类号: TP393

文献标志码: A

DOI: 10.11959/j.issn.1000-436x.2025073

Joint routing and hybrid traffic scheduling algorithm based on time-aware and cyclic queued forwarding in TSN

WANG Ying, ZHUANG Zhihan, YU Peng, QIU Xuesong, GUO Shaoyong

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

Abstract: In order to solve the problem of hybrid scheduling of multiple time-sensitive flows in time-sensitive network (TSN), a TSN joint routing and hybrid traffic scheduling algorithm based on time-aware and cyclic queued forwarding was proposed. The TACQF hybrid traffic scheduling model was designed by combining TAS and CQF shapers. The model realized the simultaneous opening of TAS queue and CQF queue through cycle selection, and used the dead time of CQF queue to open TAS queue, so as to achieve soft isolation between different flows, and reduce conflicts between TT and AVB flows. JR-TACQF hybrid traffic scheduling algorithm was designed according to TACQF hybrid traffic scheduling model. The algorithm combined routes for scheduling and load balancing, thus improving bandwidth utilization and scheduling success rate. The simulation results show that the JR-TACQF hybrid traffic scheduling algorithm can effectively solve the TSN hybrid traffic scheduling problem, improve the scheduling success rate and bandwidth utilization rate, and generate feasible scheduling result in a low execution time.

Keywords: time-sensitive network, TAS, CQF, tabu search, hybrid traffic scheduling

收稿日期: 2024-12-31; 修回日期: 2025-04-07

通信作者: 郭少勇, syguo@bupt.edu.cn

基金项目: 国家自然科学基金资助项目 (No.62322103); 北京市自然科学基金资助项目 (No.4232009); 中央高校基本科研业务费专项资金资助项目 (No.2023ZCTH11)

Foundation Items: The National Natural Science Foundation of China (No.62322103), The Natural Science Foundation of Beijing (No.4232009), The Fundamental Research Funds for the Central Universities (No.2023ZCTH11)

0 引言

工业互联网 (IIoT, industrial Internet of things) 是将工业设备、生产系统、传感器和其他物理设备通过互联网技术进行连接和通信, 实现设备之间的数据交换、信息共享和智能化管理的一种新型工业应用模式^[1]。工业互联网通过传感器、嵌入式系统、云计算平台 and 数据分析技术, 实现对工业设备和生产过程的实时监测、远程控制和智能化决策, 以提高生产效率、降低成本、改善产品质量和实现智能制造。上述网络应用的数据上传和控制指令下发, 往往需要达到毫秒级时延和微秒级抖动^[2]。同时, 工业互联网需同时承载时间触发 (TT, time-triggered) 流和音视频桥接 (AVB, audio/video bridging) 流等多种流量。这些流的通信服务质量 (QoS, quality of service) 需求差异显著。其中, TT 流要求严格的时延确定性, AVB 流的要求则相对宽松。混合调度需确保不同流量在共享网络中互不干扰, 同时满足各自的端到端时延和抖动需求^[3]。然而, 传统以太网尽力而为的传输模式无法满足上述需求, 因此无法为工业互联网提供确定性的网络服务。

时间敏感网络 (TSN, time-sensitive networking) 增强了传统以太网, 为支持具有有限时延和低抖动的确定性通信提供了可能^[4]。TSN 的发展始于对传统以太网的改进, 以满足工业自动化领域对实时数据传输和网络可靠性的需求。TSN 是 IEEE 802.1 任务组开发的一套数据链路层协议, 用于为实时应用提供低时延和高可靠的网络通信。

TSN 由一组协议簇组成, 包括 IEEE 802.1AS^[5] 时间同步、IEEE 802.1Qbv^[6] 流量调度、IEEE 802.1Qbu^[7] 帧抢占、IEEE 802.1Qci^[8] 逐流过滤与监管和 IEEE 802.1Qcc^[9] TSN 配置, 解决了时间同步、流量调度和系统配置三大关键问题。其中, 流量调度是 TSN 标准的核心机制, 可通过时间同步、流量整形及优先级控制等技术满足业务流量端到端时延、抖动等服务质量的需求。TSN 流量调度机制主要由 IEEE 802.1Qbv^[6] 提出的时间感知整形器 (TAS, time awareness shaper) 和 IEEE 802.1Qch^[10] 提出的循环排队转发 (CQF, cycling queuing and forwarding) 等流量整形器实现。TAS 通过门控列表 (GCL, gate control list) 实现微秒级流量隔离, CQF 则以奇偶循环队列保障固定时延传输。然而,

目前时间敏感网络的 TAS 门控列表配置非常复杂, 同时 CQF 难以解决混合流量调度问题。

为了解决上述问题, 研究人员提出了一些新的 CQF 扩展方案, 如循环指定排队和转发 (CSQF, cycle specified queuing and forwarding) 和多重 CQF (Multi-CQF, multiple cycle queuing and forwarding) 等。其中, CSQF 在 CQF 原有 2 个队列的基础上引入了第 3 个队列, 用来接收未按时到达的数据帧^[11]。Multi-CQF 使用多组 CQF 队列进行调度, 每组 CQF 队列使用不同的循环周期^[12]。虽然 CSQF 和 Multi-CQF 可以实现多种周期的 AVB 流的混合调度, 但是难以解决 TT 流和 AVB 流的混合调度问题。目前, 人们大多采用结合 TAS 和 CQF 这 2 种整形器的方式实现不同类型流量的混合流量调度^[13]。例如, TAS 与 Multi-CQF (TAMCQF, TAS and Multi-CQF) 通过结合 TAS 和 Multi-CQF 进行调度, 在使用多个 CQF 队列组的基础上使用 TAS 为高时间敏感流提供更高的 QoS^[14]。通过结合 TAS 和 Multi-CQF, TAMCQF 在实现门控列表配置简便性的同时, 可以进行混合流量调度。但是, 这类结合 TAS 和 CQF 的模型也存在一些问题。首先, TT 流会影响 AVB 流的调度, 因为模型中 TT 流门控列表开启时, 需要关闭 AVB 流的门控列表, 高优先级 TT 流的集中调度会显著降低 AVB 流的调度成功率。此外, 该模型下的调度算法没有结合路由进行调度, 随着网络负载的增加, 调度成功率明显下降。

面对上述 TSN 混合流量调度中存在的问题, 本文提出了一种基于时间感知和循环排队转发的联合路由和混合流量调度算法 (JR-TACQF, joint routing and hybrid traffic scheduling algorithm based on time-aware and cyclic queued forwarding)。本文主要的研究工作如下。

1) 结合 TAS 和 CQF 这 2 种流量整形器, 设计了一种基于时间感知和循环排队转发的 TACQF 模型。该模型允许 TAS 队列和 CQF 队列同时开放, 通过设计周期选择方案和调度算法实现不同流量间的软隔离。利用 CQF 奇偶队列切换时的死区时间开放 TAS 队列, 增加 CQF 队列的有效开放时间, 提高带宽利用率, 从而提高调度成功率。

2) 提出一种联合路由的混合流量调度方案, 通过路由选择实现负载均衡, 降低 TT 流的集中调

度, 在保证 TT 流量调度的同时, 提高了 AVB 流的调度成功率。使用禁忌搜索进行调度方案的计算, 相比求解器算法显著缩短了求解时间。

3) 基于 NS-3 仿真系统进行仿真实验, 实验结果表明, 本文提出的混合流量调度算法可以将时间敏感流的调度成功率提高约 35%, 带宽利用率上升约 30%, 并可以在有效时间内得到可行的调度方案, 高负载网络下算法执行时间相比求解器算法缩短了约 30%。

1 相关工作

流量调度作为时间敏感网络的核心技术, 通过时间同步与资源管理, 确保时间敏感流量在工业互联网中实现确定性低时延传输。其中, TAS 和 CQF 作为 2 种典型机制, 分别从不同维度解决了流量调度难题。

如图 1 所示, TAS 根据预先生成的门控列表, 通过门操作控制交换机出端口队列实现流量调度。交换机出端口对应 8 个不同的优先级队列, 每个队列可以独立开关。门控列表由一系列有序的门控操作组成, 其中, c 表示门控开启, o 表示门控关闭, 门控列表 T_m 表示时隙, Q_n 表示交换机出端口队列。TAS 根据预先生成的门控列表, 通过控制交换机出端口队列来控制帧传输的时序, 进而实现流量调度, 为时间敏感流提供确定性服务。

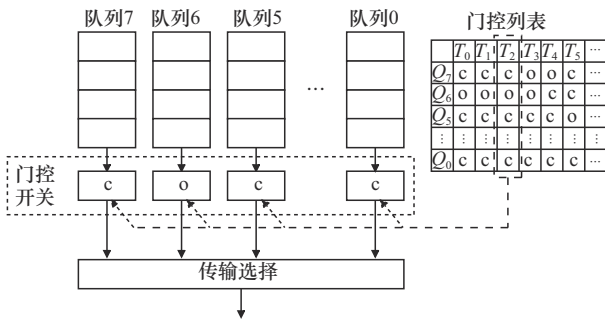


图1 TAS原理框架

在 TAS 调度方面, 文献[15]提出了一种使用可满足性模理论计算时间敏感网络静态门控列表的算法。文献[16]将 TT 流量调度问题建模为无等待数据包调度问题, 该问题映射为无等待作业车间调度问题。文献[17]提出了一种集群整数线性规划 (cluster-ILP, cluster integer linear programming) 算法来解决多播 TT 流的路由和调度联合优化问题。

文献[18]提出了一种基于学习的可扩展调度和路由协同设计架构来解决 TSN 调度和路由问题。TAS 可以为时间敏感流提供确定性的服务, 然而, TAS 的门控机制需要配置网络中所有交换机上每个出端口队列的门控列表^[19]。在大规模网络中, 这样的配置非常复杂。

为了简化 TSN 交换机门控列表的设计, IEEE 802.1Qch 在 IEEE 802.1Qbv 中门控列表的基础上提出了循环排队转发模型^[10]。CQF 通过使用奇偶队列来提供确定的、可预测的时延。如图 2 所示, CQF 使用 2 个队列和一个循环计数器。在奇周期内, 队列 1 接收从接收端口收到的数据帧而不发送, 队列 2 发送上一周期接收到的数据帧而不接收任何数据; 在偶周期内, 队列 2 接收从接收端口收到的数据帧而不发送, 队列 1 发送上一周期接收到的数据帧而不接收任何数据。网络中的交换机需要时间同步并使用相同的周期。在选择了合适的 CQF 周期 T_{CQF} 后, CQF 可以确保在一个周期内从上游节点发送数据包, 并在同一周期内的下游节点接收数据包, 且在下一周期将数据包发送出去。因此, 端到端时延仅取决于 CQF 周期大小 T_{CQF} 和路径跳数 H 。其中, 最大时延界限为 $(H + 1)T_{CQF}$, 最小时延界限为 $(H - 1)T_{CQF}$, 端到端抖动最大为 $2T_{CQF}$ 。但是, 在混合流量调度时, 非周期流会导致 CQF 无法选择合适的周期, 大的周期会导致端到端时延进一步增加, 小的周期会导致带宽增加^[14]。同时, 对不同周期的流使用相同的周期进行调度, 可能出现高时间敏感流无法调度的情况。因此, CQF 本身难以支持混合流量调度的场景。

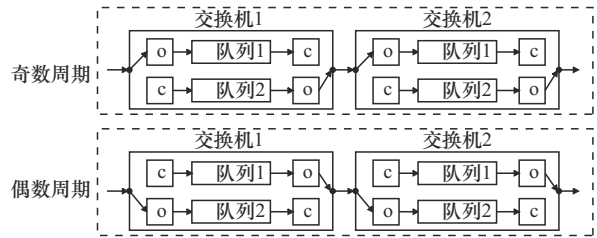


图2 CQF原理框架

在 CQF 调度方面, 文献[20]首先提出注入时间规划 (ITP, injection time planning) 解决 CQF 的调度问题。基于 ITP 机制, 文献[20]提出了一种名为 Tabu-ITP 的新型启发式调度算法。为了减轻基于求解器的算法的计算强度, CQF 中的流量调度大多采

用增量调度的思想。增量调度方案包括流排序、偏移搜索和资源判断^[21]。文献[22]提出了一种基于整数理论的流排序分析算法,给出调度灵活度指标并推导出一种高效的时隙占用判定定理。文献[23]提出的基于分数映射的调度(MSS, mapping score-based scheduling)算法结合流排序和偏移搜索进行调度,从而提高了调度性能。文献[24]提出了一种联合路由和负载均衡的调度优化(JRS-LB, joint routing and scheduling optimization with load balance)算法,以提高调度成功率和网络资源利用率。该算法联合考虑时间敏感流量的注入时隙偏移、调度和路由,虽然CQF降低了门控列表配置的复杂度,但其难以解决混合流量调度问题。

在CQF扩展方案方面,文献[25]为Multi-CQF设计了规划约束,以及基于模拟退火的元启发式解决方案,并对CQF的变体(如CSQF、Multi-CQF)进行综合评估,将它们与TAS时间感知整形器进行对比。结果显示,随着网络规模的增加,CQF和Multi-CQF整形器将导致更大的时延。在这种情况下,需要结合TAS来调度需求更严格的流。

综上,TSN技术可以确保网络传输达到最小时延和最小抖动,增强了工业网络的稳定性和可靠性,在工业界的控制网络和以太网的融合中发挥了重要的作用。然而,工业网络中存在控制流量和音视频监控流量等多种流量,这要求网络除了保证通信服务质量外,还要具备混合流量调度的能力,但上述算法难以支持。

在混合流量调度方面,文献[26]提出了一种结合TAS和CQF的混合调度机制,实现了时间敏感网络的时延可控和零抖动。但是该模型中不同类型的流量通过门控列表的开关进行物理隔离,当TAS队列开启时,CQF队列必须关闭。这使队列切换时的死区时间无法利用,影响了带宽利用率和调度成功率。文献[27]在结合TAS和CQF的同时,通过调整预定流量的采样周期和设计流偏移规划,提高混合流量的可调度性。但是,该算法未结合路由进行调度。集中的TT流量调度会使TAS预定流量大幅增长,进而挤压AVB流的调度资源,严重影响AVB流的调度成功率^[28]。文献[14]提出了一种TAMCQF模型来实现确定性混合流量调度,设计并使用混合整数线性规划(MILP, mixed integer linear programming)求解器对其进行求解。与文献[27]类似,

TAMCQF模型中集中的高优先级流量调度也会显著影响低优先级流量调度。同时,TAMCQF-ILP算法使用求解器进行求解,在大规模网络下的求解时间长。此外,上述算法在调度混合流量时均未结合路由规划,随着流量的增加,难以实现负载均衡,调度成功率会明显下降。

因此,针对上述TSN混合流量调度中存在的问题,本文设计并实现了基于时间感知和循环排队转发的TSN联合路由和混合流量调度算法。相比于其他混合流量调度算法,本文算法允许TAS队列和CQF队列同时开放,通过软隔离方式实现混合流量调度,同时联合路由进行调度,实现负载均衡,在调度成功率和带宽利用率等方面均有显著提高。

2 系统模型

2.1 网络模型

网络拓扑抽象为有向图 $G(V,E)$,其中, V 是节点集, E 是边集。节点集 $V=(S\cup H)$,其中, S 和 H 分别是交换机和终端主机的集合。边集 $E=\{(V_i,V_j)|V_i,V_j\in V,i\neq j\}$ 表示网络中链路的元组集合,集合中的每个元素表示一条单向链路。TSN采用全双工以太网链路,节点 V_i 和 V_j 之间的物理链路对应2条有向链路,用 $[V_i,V_j]$ 和 $[V_j,V_i]$ 表示。

2.2 网络流量模型

流量分为TT流和AVB流。TT流对应于有严格时间敏感需求的业务,具有严格的端到端时延和抖动要求。相比于TT流,AVB流的端到端时延和抖动需求更加宽松,可以进一步分为A类和B类。相比于A类AVB流,B类AVB流的发送间隔更大。本文为数据流建立了数据模型。

TT流 f_i 被定义为 $[f_i.src,f_i.dst,f_i.ddl,f_i.period,f_i.J,f_i.size,f_i.pathList]$,其中,src和dst分别表示流的源站和目的站,用于算法生成对应的路由,所有可行路由存储在路径集pathList中,ddl为最大端到端时延,period为TT流的发送周期, J 为最大抖动,size为一个周期内发送的总数据量,pathList使用红黑树存储,根据路由权重进行排序,路径path数组表示为 $[V_{src},\dots,V_{dst}]$ 。AVB流的模型定义与TT流类似,其端到端时延和抖动的要求更加宽松。

本文JR-TACQF流量调度算法的目标是为网络

中的 TT 流和 AVB 流生成调度方案 $f_i.schePlan = [slot, path]$, 其中, slot 表示流开始传输的注入时间, path 为最终选择的路由方案。

2.3 TACQF 流量调度模型

本文提出的 TACQF 混合流量调度模型由队列 Q_7 使用 TAS 整形器传输 TT 流, 队列 Q_6 和 Q_5 组成 CQF 整形器的奇偶队列传输 AVB 流。

TACQF 交换机 $s_a \in S$ 表示为 $[Q_{size_{ab}}, Q_{occu_{ab}}, GCI_{ab}]$,

其中, $Q_{size_{ab}}$ 表示交换机 s_a 到 s_b 的出端口队列容量, 若交换机 s_a 和 s_b 之间没有链路, 则该值为 0; $Q_{occu_{ab}}$ 表示交换机 s_a 到交换机 s_b 的出端口已被占用的队列容量, 每一列表示同一时隙下不同队列的占用情况; GCI_{ab} 表示门控列表, 每一列表示同一时隙下不同队列的门控, 每一行表示一个队列在不同时隙下的门控。

在 CQF 模型中, CQF 奇偶队列交替开关。如图 3 所示, 考虑网络中交换机 s_a 向交换机 s_b 发送数据帧的情况。当一个 CQF 奇偶周期结束由奇周期队列向偶周期队列切换时, 将开启对偶队列数据帧的接收, 并关闭对奇队列数据帧的接收。这时, 交换机 s_a 在奇周期末尾发送的数据帧将无法被交换机 s_b 的奇队列接收, 这将导致数据帧的丢失。因此, 在 CQF 奇偶周期切换时, 会有一段死区时间无法进行数据帧的传输。此外, 传统混合流量调度模型一般采用抢占式传输来保证 TT 流的低时延和零抖动。这也将导致队列切换, 并造成死区时间。

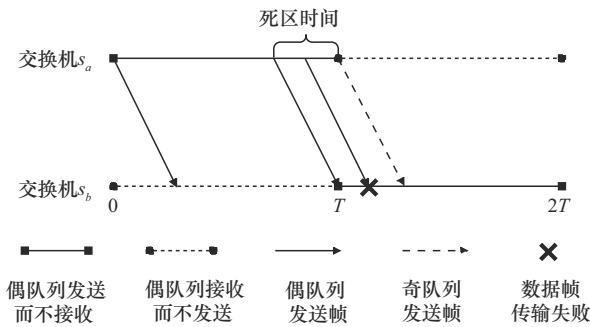


图3 CQF 时序图

死区时间主要包括发送时延和传播时延, 通过增加 CQF 周期时长、降低奇偶队列的切换频率, 可以降低死区时间的占比。但是 CQF 整形器下数据流的时延与 CQF 周期时长成正比, 过长的 CQF

周期将导致数据流超时。因此在大多数 CQF 周期下, 死区时间会占用 5%~10% 的时间。

在 TAMCQF 等结合 TAS 和 CQF 的模型中, TAS 和 CQF 队列同时只能开启一个。如图 4 所示, 由于死区时间的影响, CQF 周期的最后一段时间中虽然 CQF 队列开启了, 但是无法传输数据帧, 造成了网络资源的浪费。死区时间大小主要受数据帧的发送时延和传播时延影响。由于 AVB 流的数据帧往往远大于 TT 流的数据帧, 因此一个 CQF 的死区时间一般可以传输十几个 TT 流的数据帧。在本文 TACQF 模型使用的周期选择方案中, 优先在 CQF 死区时间开放 TAS 队列传输 TT 流, 降低了死区时间的影响, 并提高了信道利用率。同时, 在相同的循环周期下, CQF 队列可以有更长的开放时间, 增加了 AVB 流的调度成功率。

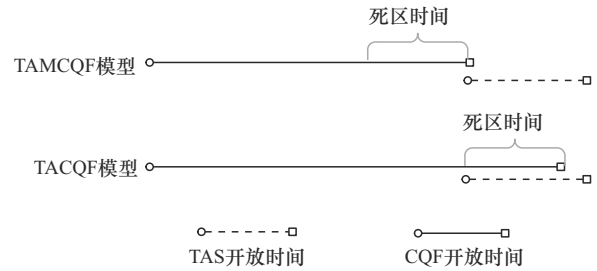


图4 TACQF 时序模型

2.4 问题模型

1) CQF 周期 T_{CQF} 约束。 T_{CQF} 为一次 CQF 队列开启到下一次 CQF 队列开启的时间长度, 由式(1)和式(2)共同约束。 T_{CQF} 应当能被所有的 CQF 流的周期整除, 即

$$kT_{CQF} = \text{GCD}(f_i.\text{period}) \quad (1)$$

其中, GCD 表示最大公约数, $f_i.\text{period}$ 表示流 f_i 的发送周期, k 为正整数常系数。

同时, 交换机出端口队列里的全部数据帧需要在一个 CQF 周期内从上一个节点传输到下一个节点。若数据帧未能在指定 T_{CQF} 内发送, 则会在下一个周期之后发送, 这会使数据帧时延混乱, 造成数据流抖动增大; 若数据帧发送后未能在指定 T_{CQF} 内到达下一个节点, 奇偶队列切换后, 会导致该数据帧被丢弃。因此, CQF 周期需要大于或等于全部数据帧的发送时延和传播时延之和, 即

$$T_{CQF} \geq \frac{Q_{size} \cdot \text{MTU}}{b} + t_{prop} \quad (2)$$

其中, MTU表示最大传输单元, b 表示链路带宽, Q_{size} 为队列容量, t_{prop} 为传播时延。

2) TAS队列开放时间 T_{TAS} 约束。 T_{TAS} 为一次TAS队列开启到关闭的时间长度, 并且应当小于或等于 T_{CQF} 并大于或等于CQF队列切换时的死区时间, 即

$$T_{\text{deadtime}} \leq T_{\text{TAS}} \leq T_{\text{CQF}} \quad (3)$$

其中, T_{deadtime} 表示CQF死区时间。死区时间小于或等于一个数据帧从上一个节点开始发送到下一个节点完成接收的时长, 即

$$T_{\text{deadtime}} \leq \frac{\text{MTU}}{b} + t_{\text{prop}} \quad (4)$$

同时, TAS队列内的全部数据帧需要在一个开放时间内传输完毕, 即

$$T_{\text{TAS}} \geq \frac{Q_{\text{TASsize}} \text{MTU}}{b} \quad (5)$$

其中, Q_{TASsize} 表示TAS队列容量。

3) 交换机队列资源约束。交换机 s_a 在时隙 T_i 所有队列发送到交换机 s_b 的数据帧数之和为 $Q_{\text{sum}_{ab}}^{T_i}$, 即

$$Q_{\text{sum}_{ab}}^{T_i} = Q_{\text{occp}_{ab}}^{T_i} \mathbf{GCL}_{ab}^{T_i \top} \quad (6)$$

其中, $Q_{\text{occp}_{ab}}^{T_i}$ 为交换机 s_a 到交换机 s_b 的出端口队列在时隙 T_i 上的占用向量, $\mathbf{GCL}_{ab}^{T_i \top}$ 为交换机 s_a 到交换机 s_b 的出端口队列在时隙 T_i 上的门控列表向量。

$Q_{\text{sum}_{ab}}^{T_i}$ 应小于或等于交换机出端口队列容量 $Q_{\text{size}_{ab}}$, 即

$$Q_{\text{sum}_{ab}}^{T_i} \leq Q_{\text{size}_{ab}} \quad (7)$$

4) 时延约束。数据流传输时延 $f_i.\text{delay}$ 应当满足端到端时延需求 $f_i.\text{ddl}$ 的限制, 即

$$f_i.\text{delay} \leq f_i.\text{ddl} \quad (8)$$

其中, 流 f_i 的时延受交换机跳数和注入时间 $f_i.t$ 影响。由于约束式(1)和约束式(2), 每个数据帧从上一个节点到下一个节点的用时都为 T_{CQF} 。另外, 数据帧的初始发送时间会向后推迟 $f_i.t$ 。因此, 流 f_i 的时延为路径跳数与 T_{CQF} 的积与注入时间之和, 即

$$f_i.\text{delay} = (f_i.\text{path.len} - 1)T_{\text{CQF}} + f_i.t \quad (9)$$

其中, $f_i.\text{path.len}$ 为路径长度, $f_i.t$ 为 f_i 的注入时间。

3 算法设计

3.1 JR-TACQF混合流量调度算法

本文设计的JR-TACQF混合流量调度算法主要分为3个部分: 周期选择、TT流量调度和AVB流量调度。

首先, 需要选择合适的CQF周期 T_{CQF} 。CQF调度中, 数据流的时延与CQF周期 T_{CQF} 长度成正比, 过长的周期会使流的传输超时, 过短的周期会使交换机无法在一个周期内完成队列中全部数据帧的传输而增大抖动。因此, 需要根据约束式(1)和约束式(4)来确定 T_{CQF} 的范围, 并选择范围中的最小值作为 T_{CQF} 。

确定 T_{CQF} 后, 即可根据选定的 T_{CQF} 以及约束式(2)和约束式(3)并使用3.2节中联合路由的TT流调度算法进行TT流的调度。完成TT流调度后, 根据网络中交换机出端口队列的占用情况和交换机中TAS队列的最长开放时间计算的 T_{TAS} 最小值, 并结合约束式(2)确定最终的 T_{TAS} 。当 T_{CQF} 和 T_{TAS} 确定后, 即可确定TACQF模型下的门控列表。

最后, 根据 T_{CQF} 减去 T_{TAS} 得到每个CQF周期中的CQF实际可发送时间。使用3.3节中基于禁忌搜索的联合路由调度算法调度AVB流, 并生成最终的混合流量调度方案。

3.2 联合路由的TT流量调度算法

当TT流较多且CQF死区时间不足以满足TT流量调度需求时, 本文算法会进一步缩短CQF开放时间来延长TAS开放的时间。同时, 整个网络的交换机之间的CQF队列的门控列表需要同步开关, 否则将造成数据帧的丢失。因此, 当网络中的TT流需要集中通过某一个交换机时, 即使大多数交换机中并不需要开放长时间的TAS队列进行TT流量调度, 也会导致全网交换机的TAS周期增加, CQF周期减少。集中的高优先级TT流量调度会显著降低AVB流的调度成功率, 甚至出现整个网络负载不高, 但是AVB流仍无法调度的情况。

因此, 在混合流量的调度中, 通过路由选择进行负载均衡从而避免TT流量过于集中十分重要。本文设计了一种联合路由的TT流量调度算法, 如算法1所示。该算法在CQF循环周期的基础上, 根据TT流的注入时间规划和路由选择来生成调度方

案, 并根据交换机出端口队列负载情况计算式(10)中的负载权重, 选择权重最低的调度方案。

算法 1 联合路由的 TT 流量调度算法

输入 网络拓扑 $G = (V, E)$, 流集 $F = F_{TT}$, 循环次数上限为 Setting.times

输出 调度方案 $F.schePlan$, TAS 队列开放时间 T_{TAS}

- 1) 初始化成功调度 $F_{sche} = \emptyset$ 和未成功调度 $F_{unsche} = F$;
- 2) 计算交换机门控列表时隙;
- 3) 为 F 生成所有可行路径 $F.pathList$;
- 4) 定义 $n = 0$;
- 5) 循环
- 6) $f_i = \text{randomFlow}(F_{unsche})$;
- 7) $f_i.schePlan = \text{minCostPlan}(f_i.pathList)$;
- 8) $F_{unsche} \leftarrow F_{unsche} - f_i$;
- 9) $F_{sche} \leftarrow F_{sche} + f_i$;
- 10) 更新调度方案 $F.schePlan$;
- 11) until $F_{unsche} = \emptyset$ 或 $n \geq \text{Setting.times}$;
- 12) $T_{TAS} = \text{calTASTime}()$;
- 13) return $F.schePlan, T_{TAS}$;

负载权重 loadCost 由当前网络中交换机出端口队列的时隙占用情况 Q_{occ}^T 和路径长度 path.len 确定, 即

$$\text{loadCost} = \frac{\sum_{a=1}^{\text{path.len}-1} Q_{occ}^{T_i}}{\text{path.len}} \quad (10)$$

算法 1 中, 步骤 1) 设定初始状态为已成功调度流集 F_{sche} 为空集, 未成功调度流集 F_{unsche} 为全部 TT 流。步骤 2) 根据周期选择阶段确定的 T_{CQF} 计算每个交换机的门控列表的时隙。步骤 3) 为每个数据流生成全部可行的从源站到目的站的无环交换机路径。由于属性相似的数据流在路由选择时容易选择相同或相似的路径, 因此若相似的数据流连续进行调度, 则该路径的负载会不断增大, 并使后续相似的数据流发生“绕远”, 即选择负载更低但是路径更长的路由, 这会使更多交换机的负载增大和 TAS 队列的开放时间延长, 导致调度成功率下降。因此, 步骤 5)~步骤 11) 从所有未调度的流中随机选择一个流, 降低相邻选择的数据流的相似性, 从而更容易选择负载低且路径短的路由。然后, 计算负载权

重 loadCost 最小的调度方案, 并加入最终的调度方案列表。循环结束后, 返回调度方案列表。步骤 12) 计算网络中最长的 TAS 队列开放时间作为 T_{TAS} 范围的最小值, 并与约束式(2)确定的 T_{TAS} 范围相交, 选择最小值作为 T_{TAS} 。

如图 5 所示, 确定 T_{CQF} 和 T_{TAS} 后即可计算门控列表。图 5 中, CQF slot_1 和 CQF slot_2 分别为 CQF 奇偶队列 Q_6 和 Q_5 的门控列表开放时隙, TAS slot 为 TAS 队列 Q_7 的门控列表开放时隙。一个 T_{CQF} 可以表示一个 CQF 奇周期或偶周期, 因此门控列表的超周期为 $2T_{CQF}$ 。在一个奇周期内, TAS 队列 Q_7 初始为关闭状态, 并在周期的最后根据 T_{TAS} 时长开放 Q_7 ; CQF 队列 Q_6 初始为开放状态, Q_5 始终保持关闭状态。若 T_{TAS} 时长大于死区时间 $T_{deadtime}$, 则在奇周期的最后关闭 Q_6 , 关闭时长为 $T_{TAS} - T_{deadtime}$ 。偶周期与奇周期门控列表设置方式类似, 但是 Q_6 和 Q_5 的设置方式相反。

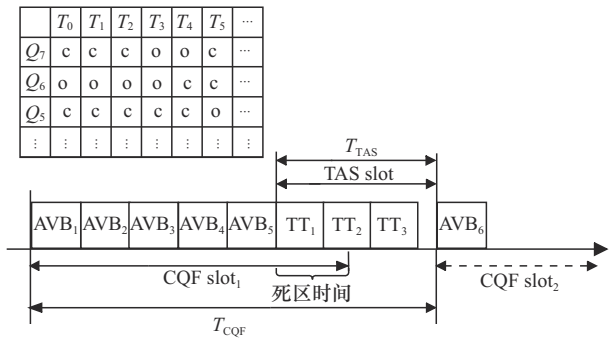


图 5 TACQF 门控列表

3.3 基于禁忌搜索的联合路由调度算法

在完成周期选择和 TT 流预调度后, 需要调度 AVB 流, 基于此, 本文设计了一种基于禁忌搜索的联合路由调度算法, 如算法 2 所示。该算法基于禁忌搜索联合路由求解调度方案, 同时考虑路由选择和注入时间规划。

算法 2 基于禁忌搜索的联合路由调度算法

输入 网络拓扑 $G = (V, E)$, 流集 $F = F_{AVB}$, 循环次数上限为 Setting.times

输出 AVB 流量调度方案 $F_{AVB.schePlan}$

- 1) 初始化成功调度 $F_{sche} = \emptyset$ 和未成功调度 $F_{unsche} = F$;
- 2) 为 F 生成所有可行路径 $F.pathList$;
- 3) 定义 $i = 0$;

- 4) 循环
- 5) $f_i = \text{random}(F_{\text{unsche}})$;
- 6) $f_i.\text{schePlanList} = \text{邻域移动策略}(f_i)$;
- 7) $f_i.\text{schePlan} = \text{邻域选择策略}(f_i.\text{schePlanList})$;
- 8) if $f_i \in F_{\text{unsche}}$ and $\text{random}()$ then
- 9) 根据调度方案预分配网络资源;
- 10) end if
- 11) else if $f_i \in F_{\text{sche}}$ then
- 12) 释放 f_i 当前调度方案的网络资源;
- 13) end if
- 14) $\text{costFunc}(\text{next.schePlan})$;
- 15) $\text{tabuUpdateCurrent}(\text{next.schePlan})$;
- 16) $\text{tabuUpdateBest}(\text{next.schePlan})$;
- 17) until $i \geq \text{Setting.times}$;
- 18) $F.\text{schePlan} = \text{best}$;
- 19) return $F.\text{schePlan}$;

算法2将AVB流集分为已调度流集 F_{sche} 和未调度流集 F_{unsche} ,并将流 f_i 的调度方案定义为注入时间和路由的二元组,即 $f_i.\text{schePlan} = \{\text{slot}, \text{path}\}$ 。步骤1)设定初始状态为已成功调度流集为空集,未成功调度流集为全部AVB流。步骤2)为AVB流计算所有的无环可行路径。与其他只选择最短路径的路由算法不同,算法2放宽了路径长度的限制,所有满足约束式(4)的无环路径都将被加入 pathList 路径列表,这扩大了可选择的路由空间,并通过约束式(4)确保路径不会导致超时。该路由生成和选择算法可以提高负载均衡性能,进而提高带宽利用率和调度成功率。步骤4)~步骤17)的每次循环中,通过步骤6)邻域移动策略随机选择一个流,根据约束式(3)的资源限制尝试生成当前网络状态下该流所有可行的调度方案。然后,步骤7)邻域选择策略选择 f_i 调度方案,更新网络资源占用情况并将该调度方案加入禁忌表。步骤16)根据每次循环生成调度方案的目标函数值,更新最优解。

步骤6)邻域移动策略会随机选择一个流来生成解集,当选择的流是未调度流时,则根据当前网络状态集约束式(3),生成该流所有可以被调度的注入时间和路由选择的解集,并根据目标函数大小进行排序。步骤7)邻域选择策略将在解集中选择目标函数值最低、不在禁忌列表中的解加入当前调度方案,并在步骤9)更新网络资源占用情况。当选

择的流是已调度流时,除了生成可调度的解外,还将生成一种取消该流量调度的特殊解。当连续多次循环中没有新的流被成功调度时,将选择特殊解。当邻域选择策略选择特殊解加入调度方案时,步骤12)将使该流变成未调度状态,并释放对应的时隙占用。

流 f_i 调度方案的目标函数值综合考虑路由跳数对时延的影响和负载均衡。因此,流 f_i 调度方案的目标函数值 W_{f_i} 表示为

$$W_{f_i} = \alpha_1 C_1 + \alpha_2 C_2 \quad (11)$$

其中, α_1 和 α_2 为常数; C_1 表示实际端到端时延 $f_i.\text{delay}$ 与时延要求 $f_i.\text{ddl}$ 的相对关系, C_2 表示 f_i 的调度方案对所选链路的网络资源占用,分别如式(12)和式(13)所示。

$$C_1 = \frac{f_i.\text{delay}}{f_i.\text{ddl}} \quad (12)$$

$$C_2 = \sum_{a=1}^{\text{path.len}-1} Q_{\text{size}_{a(a+1)}} - Q_{\text{sum}_{a(a+1)}}^{(T_i+a)\%T} \quad (13)$$

其中, path.len 表示路径长度, $Q_{\text{size}_{a(a+1)}}$ 表示交换机 s_a 到下一交换机出端口队列的容量, $Q_{\text{sum}_{a(a+1)}}^{(T_i+a)\%T}$ 表示在时隙 $(T_i+a)\%T$ 交换机 s_a 到下一交换机的队列占用量。

基于禁忌搜索的联合路由调度算法的目标是通过调度方案的配置使网络中尽可能多的AVB流可以调度,并且在满足流的服务质量要求的前提下,尽可能降低端到端时延和抖动。因此,禁忌搜索在流集 F 下的目标函数 W 为

$$W = \sum_{f_i \in F} f_i.\text{isSche} W_{f_i} + \alpha(1 - f_i.\text{isSche}) \quad (14)$$

其中, α 为常数, $f_i.\text{isSche}$ 表示流 f_i 是否被成功调度,即

$$f_i.\text{isSche} = \begin{cases} 0, & f_i \text{未成功调度} \\ 1, & f_i \text{成功调度} \end{cases} \quad (15)$$

当网络中未被成功调度的流越多或流的时延越高时,目标函数值 W 越大。

3.4 复杂度分析

设网络中节点数量为 $|E|$,链路数量为 $|V|$,流数量为 i ,交换机一个超周期的时隙数为 t ,则为流

集生成所有可行路径的算法复杂度为 $O(i(|E| + |V|))$ 。算法 1 中计算流的最小负载权重的复杂度为 $O(n(|E| + |V|)t)$ ，因此算法 1 整体时间复杂度为 $O(i_{TT}(n(|E| + |V|)t + (|E| + |V|)))$ 。设禁忌搜索最大迭代数为 N ，禁忌列表长度为 T ，则邻域移动策略时间复杂度为 $O((|E| + |V|)t)$ ，邻域选择策略时间复杂度为 $O(i((|E| + |V|)t)^2)$ ，因此算法 2 整体时间复杂度为 $O(N((|E| + |V|)t + i_{AVB}((|E| + |V|)t)^2) + T)$ 。

4 仿真实验与性能评估

4.1 仿真环境

本程序在 Windows 环境下，基于 AMD Ryzen9 4 GHz 处理器和 16 GB 内存平台运行。使用 Java 实现本文提出的基于时间感知和循环排队转发的 TSN 流量调度算法，并使用 NS-3 设置 TSN 仿真环境。

本文使用乘员探索车 (CEV, crew exploration vehicle) 网络作为实验拓扑^[29]，结构如图 6 所示，包括 31 个终端和 14 个交换机。链路带宽为 1 000 Mbit/s。由于原 CEV 网络链路利用率较低^[30]，本文对数据流进行了扩展。数据流根据标准^[31]生成，包括 50 个 TT 流和 100~1 000 个 AVB 流，AVB-A 流和 AVB-B 流的数量相等，TT 流和 AVB 流设置如表 1 所示。其中，TT 流最大端到端时延与周期相等，AVB 流最大端到端时延为 1.5 倍周期。根据 3.1 节中提出的周期选择方案，设置 CQF 循环周期 T_{CQF} 为 125 μ s。

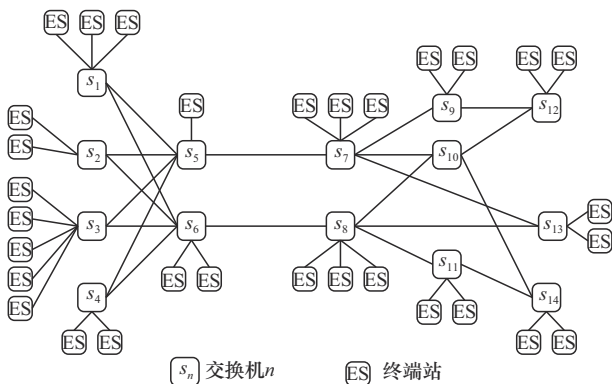


图 6 CEV 拓扑结构

数据流种类	数据包长度/B	周期/ μ s
TT	100	1 000
AVB-A	1 500	1 000
AVB-B	1 500	2 000

4.2 对比算法

仿真实验通过将本文提出的 JR-TACQF 算法与 TAMCQF-ILP^[12]、JRS-LB^[22]和 TACQ^[24]算法进行比较，来评估本文算法的性能。此外，实验还设置了本文算法的最短路由版本 TACQF 作为对照。5 种算法的详细介绍如下。

1) JRS-LB: 基于 CQF 模型的调度算法。该算法联合考虑了流量的注入时隙规划和路由。

2) TAMCQF-ILP: 基于 TAMCQF 模型的 MILP 求解器算法，能够生成全局最优调度结果，具有较高的调度能力。

3) TACQ: 结合了时间感知整形器和循环排队转发的算法，未利用 CQF 死区时间提高带宽利用率，未考虑负载均衡。

4) JR-TACQF: 本文提出的基于 TACQF 的联合路由调度算法，支持 TT 流和 CQF 流的混合流量调度，利用 CQF 死区时间提高带宽利用率。

5) TACQF: 本文提出的 JR-TACQF 使用最短路由的版本。

本文从以下 4 个方面对算法性能进行评估。

1) 调度成功率: 满足时延要求的成功调度流的数量与流总数的比率。

2) 总带宽利用率: 所有成功调度的流占用的带宽占总带宽的比例，总带宽只包括交换机端口的带宽，不包括终端设备的端口带宽。

3) 链路带宽利用率: 该链路上业务流占用的带宽占链路总带宽的比例。

4) 算法执行时间: 算法运行并得出调度结果所花费的时间。

4.3 仿真结果

1) 收敛性分析

禁忌搜索中的禁忌列表长度选择会对算法的性能产生显著影响。本节在 50 个 TT 流和 300 个 AVB 流的场景下，基于不同的禁忌列表长度，对本文算法的收敛性进行分析，结果如图 7 所示。

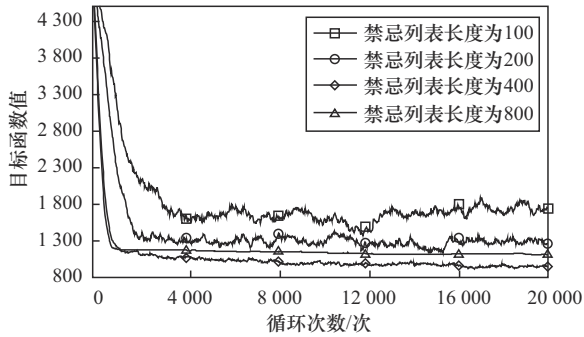


图7 循环次数与目标函数值的关系

当禁忌列表长度为100或200时,禁忌列表长度过小,搜索范围过大,削弱了全局探测能力,出现了频繁的解的更改和波动,导致搜索过程不稳定,增加了找到优质解的难度,出现无法收敛的情况。当禁忌列表长度为800时,禁忌列表长度过大,搜索范围过小,导致算法在一个狭隘的解空间中徘徊,影响全局搜索的效果,算法难以找到目标函数值更低的解。当禁忌列表长度为400时,算法能够在可行循环次数内取得最优解。因此,本文选择400作为禁忌列表长度。

2) TT流排序

联合路由的TT流量调度算法中,逐个对TT流进行调度,不同的TT流量调度顺序会对调度结果产生影响。因此,本文比较了3种排序算法调度后网络中最大的TAS队列开放时间。3种排序算法如下。

S1: 按流的源站和目的站进行基数排序。

S2: JRS-LB中的流排序算法。该算法综合考虑最大端到端时延、周期和数据包大小,并根据这些参数加权计算得到权重进行排序。

S3: 随机选择数据流。

本实验中,TT流的数据包长度在50~200 B范围内随机生成,周期在500 μs 、625 μs 、750 μs 、875 μs 和1 000 μs 的范围内随机生成,设置CQF循环周期 T_{CQF} 为125 μs 。10次实验的平均结果如图8所示。

排序算法S1中相同源站和目的站的数据流会进行相邻调度,这会导致相同源站和目的站中后进行调度的数据流更容易选择负载低但路径长的路由,从而使整个网络负载增大,因此S1算法下 T_{TAS} 最大。排序算法S2优先调度负载更低的数据流,这会导致后调度的负载更大的数据流更容易

选择负载低但路径长的路由。在JRS-LB算法中排序后的路由选择只会在最短路径中进行选择,这规避了上述问题,但也减小了路由空间。因此,在本文路由选择算法下,排序算法S2也会使最大TAS队列开放时间增大。排序算法S3随机选择数据流进行调度,避免了相似的数据流连续进行调度,降低了选择负载低但路径长的路由的可能性,这也使S3的最大TAS队列开放时间低于其他2种排序算法。因此,本文联合路由的TT流量调度算法中通过S3进行调度。

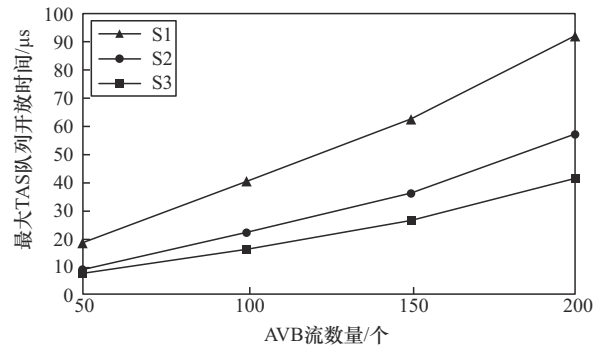


图8 3种排序算法的最大TAS队列开放时间

3) 调度成功率

本节比较了5种算法在50个TT流和100~1 000个AVB流场景下的调度成功率。

在TT流量调度方面,TAMCQF-ILP、TACQ和JR-TACQF这3种混合流量调度算法都可以调度全部的TT流,调度成功率为100%。另外,由于JRS-LB算法不支持TAS调度,因此使用CQF调度TT流,JRS-LB的排序算法S2会使TT流优先进行调度,所以TT流量调度成功率也为100%。

如图9所示,随着AVB流数量的增加,各调度算法的成功率都出现了下降。其中,本文提出的调度算法JR-TACQF在100~300个流时可以调度全部的AVB流,在500个流时成功率下降至90.2%,在1 000个流时下降至61.5%,高于其他混合流量调度算法约35%。其他的调度算法只有在100个流时可以成功调度全部的AVB流,在1 000个流时调度成功率下降至约50%。

本文算法和JRS-LB算法都联合路由进行了调度,但是本文算法通过结合TAS和CQF进行混合流量调度,TT流和AVB流之间相互隔离,降低了不同流量调度之间的相互影响,尽管TT流量调度

成功率都为 100%，但本文的 JR-TACQF 有更高的 AVB 流量调度成功率。

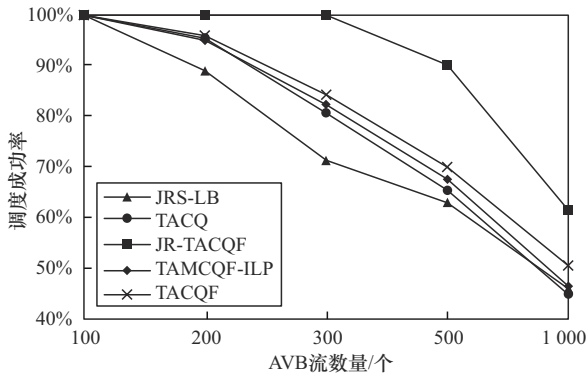


图 9 5 种算法在不同 AVB 流数量下的 AVB 流量调度成功率

在 200 和 300 个 AVB 流时，尽管网络负载较低，但 TAMCQF-ILP 和 TACQ 的调度成功率仍出现了明显下降。这是由于这 2 种算法不考虑负载均衡，尽管成功调度的全部的 TT 流，但是出现了 TT 流过于集中的情况，使 CQF 的实际开放时间减少，导致 AVB 流的调度成功率的降低。本文算法结合路由进行负载均衡，避免 TT 流过于集中，从而提高 AVB 流的调度成功率。因此，在 1 000 个 AVB 流时，相比于 TAMCQF-ILP 和 TACQ，本文算法的调度成功率有约 30% 的提升。

另外，在表 1 的数据流设置下， T_{CQF} 为 125 μ s，死区时间约为 12 μ s，在死区时间内可以进行最多 14 个 TT 流数据帧的传输。本文算法最终生成的 T_{TAS} 为 9.6 μ s，小于死区时间长度，可以在 CQF 死区时间内完成全部 TT 流的调度。通过对比 TACQF、TAMCQF-ILP 和 TACQ 可以看出，在路由策略相同的情况下，TACQF 通过优化周期选择，利用 CQF 无法调度的死区时间进行 TAS 调度，增加了 CQF 队列的实际开放时间，从而增加了 AVB 流的调度成功率。在 1 000 个 AVB 流时，相比于 TAMCQF-ILP 和 TACQ，TACQF 的调度成功率提升了约 8%。

4) 带宽利用率

本节比较了在 50 个 TT 流和 100~1 000 个 AVB 流场景下 5 种调度算法的带宽利用率，如图 10 所示。随着 AVB 流数量的增加，各调度算法的带宽利用率也不断增加，本文提出的 JR-TACQF 算法带宽利用率整体高于其他调度算法。在 1 000 个 AVB 流时，JR-TACQF 的带宽利用率相比于 TACQ 和

TAMCQF-ILP 提高了约 35%。一个原因是 JR-TACQF 算法的调度成功率高于其他算法。另一个原因是 JR-TACQF 算法联合路由进行调度，可选择的路由范围是满足 AVB 流截止时间的全部路由。因此，JR-TACQF 算法可能会选择更长的路由来进行调度，即使调度相同数量的流也可能使用更多的网络带宽。

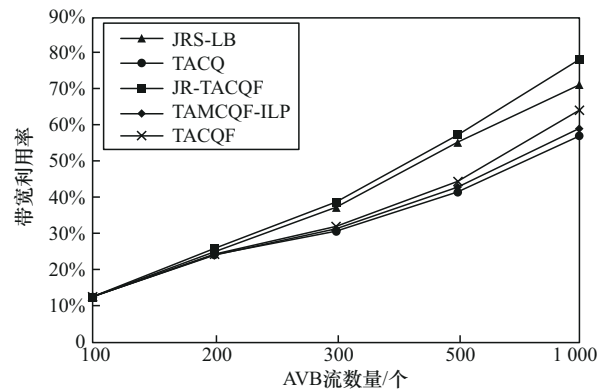


图 10 5 种算法在不同 AVB 流数量下的带宽利用率

另外，通过对比 TACQF、TAMCQF-ILP 和 TACQ 这 3 种最短路由的算法可以看出，由于对 CQF 奇偶队列切换时死区时间的利用，在相同的 T_{CQF} 下，TACQF 的带宽利用率更高。这种优势在高负载下更为明显，在 1 000 个 AVB 流时，相比于 TAMCQF-ILP 和 TACQ，TACQF 的带宽利用率提升了约 10%。

图 11 为 5 种算法在 500 个 AVB 流下交换机 s_9-s_{12} 和交换机 $s_{10}-s_{12}$ 的链路带宽利用率对比。在通往 s_{12} 的链路中， s_9 只需要承担来自 s_7 的流量，而 s_{10} 则需要承担来自 s_7 、 s_8 、 s_{14} 等的流量。

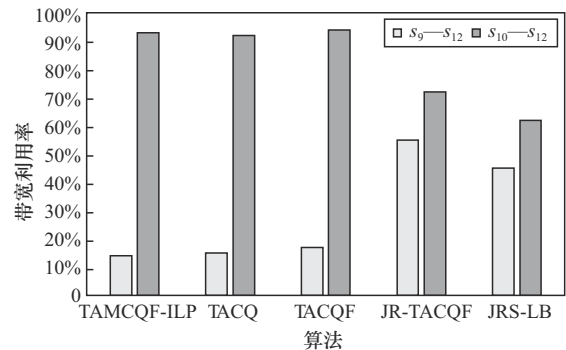


图 11 5 种算法在 500 个 AVB 流下的部分链路的带宽利用率对比

TACQF、TAMCQF-ILP 和 TACQ 算法在 s_9-s_{12} 和 $s_{10}-s_{12}$ 的带宽利用率相差约 80%。这说明由于

这3种算法都使用最短路由,没有结合路由进行调度,因此出现了部分链路负载过高、部分链路仍然空闲的情况。这使得这3种算法在总带宽利用率不高的情况下,出现了调度成功率明显下降的情况。JR-TACQF和JRS-LB算法联合路由调度进行了负载均衡,没有出现链路负载相差过大的情况,相关链路的带宽利用率之差从80%下降至20%。

5) 算法执行时间

5种算法在不同AVB流数量下的执行时间如图12所示。与JRS-LB算法相比,本文的JR-TACQF算法额外支持混合流量调度,但这导致了执行的增加。与TACQ算法相比, JR-TACQF在AVB流量调度中使用了更大的路由空间,导致时间增加,在1000个AVB流时,执行时间比TACQ高约20s。但是相比这2种算法,本文算法可以在可行时间范围内完成调度,同时在调度成功率和带宽利用率等方面均有约30%~40%的提升。TACQF算法的执行时间更短,这说明若简化对路由选择的支持,则本文算法可以缩短约40%的执行时间。与同样为混合流量调度的TAMCQF-ILP求解器算法相比,本文的JR-TACQF算法使用元启发式算法执行时间更短,尽管联合路由调度,但是在高负载网络下,仍节省了约30%的时间。

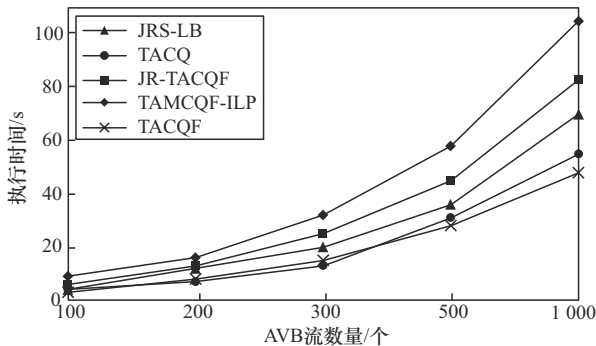


图12 5种算法在不同AVB流数量下的执行时间

5 结束语

本文对时间敏感网络混合流量调度进行研究,结合TAS和CQF这2种流量整形器,设计了TACQF流量调度模型;设计了TACQF周期选择方案,利用CQF循环周期的死区时间进行TT流的调度,降低了不同类型流量之间的影响,提高了带宽利用率;设计了一种JR-TACQF混合流量调度算法,通过联合路由的TT流量调度算法和基于禁忌

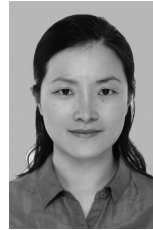
搜索的联合路由调度算法分别实现TT流和AVB流的混合调度。同时,该流量调度算法联合路由进行负载均衡,避免流量过于集中导致低优先级的AVB流量调度成功率下降。实验表明,本文提出的混合流量调度算法可以在较短的执行时间内生成可行的调度方案,并将带宽利用率和调度成功率提高约30%和35%。

参考文献:

- [1] 张祥武,韩丹涛,公彦杰.基于TSN的工业网络发展展望[J].中国仪器仪表,2023(3):23-26.
ZHANG X W, HAN D T, GONG Y J. Forecast of the development of OT network based on TSN[J]. China Instrumentation, 2023(3): 23-26.
- [2] 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.
- [3] ZHOU X, HE F, ZHAO L X, et al. Hybrid scheduling of tasks and messages for TSN-based avionics systems[J]. IEEE Transactions on Industrial Informatics, 2024, 20(2): 1081-1092.
- [4] CHAHED H, KASSLER A. TSN network scheduling: challenges and approaches[J]. Network, 2023, 3(4): 585-624.
- [5] IEEE. IEEE standard for local and metropolitan area networks—timing and synchronization for time-sensitive applications—corrigendum 1: IEEE Std 802.1AS-2020/Cor 1-2021[S]. 2022.
- [6] IEEE. IEEE standard for local and metropolitan area networks—bridges and bridged networks—amendment 25: enhancements for scheduled traffic: IEEE Std 802.1Qbv-2015[S]. 2015.
- [7] IEEE. IEEE standard for local and metropolitan area networks—bridges and bridged networks—amendment 26: frame preemption: IEEE Std 802.1Qbu-2016[S]. 2016.
- [8] IEEE. IEEE standard for local and metropolitan area networks—bridges and bridged networks—amendment 28: per-stream filtering and policing: IEEE Std 802.1Qci-2017[S]. 2017.
- [9] IEEE. IEEE standard for local and metropolitan area networks—bridges and bridged networks—amendment 31: stream reservation protocol (SRP) enhancements: IEEE Std 802.1Qcc-2018[S]. 2018.
- [10] IEEE. IEEE standard for local and metropolitan area networks—bridges and bridged networks—amendment 29: cyclic queuing and forwarding: IEEE Std 802.1Qch-2017[S]. 2017.
- [11] CHEN S, LEGUAY J, MARTIN S, et al. Load balancing for deterministic networks[C]//Proceedings of the 2020 IFIP Networking Conference. Geneva: IFIP Newsletter, 2020: 785-790.
- [12] POP P, ALEXANDRIS K, WANG T. Configuration of multi-shaper Time-Sensitive Networking for industrial applications[J]. IET Networks, 2024, 13(5-6): 434-454.
- [13] WU Y X, PAN W Q, CHEN Z X, et al. Optimized mapping and hybrid transmission integrating TAS and CQF in TSN[C]//Proceedings of the 2024 8th International Conference on Communication and Information Systems (ICCIS). Piscataway: IEEE Press, 2024: 82-88.

- [14] NIE H R, LI S S, LIU Y. TAMCQF: hybrid traffic scheduling mechanism integrating TAS and multi-CQF in TSN[C]//Proceedings of the 2023 IEEE Symposium on Computers and Communications (ISCC). Piscataway: IEEE Press, 2023: 329-334.
- [15] CRACIUNAS S S, OLIVER R S, CHMELÍK M, et al. Scheduling real-time communication in IEEE 802.1Qbv time sensitive networks[C]//Proceedings of the 24th International Conference on Real-Time Networks and Systems. New York: ACM Press, 2016: 183-192.
- [16] DÜRR F, NAYAK N G. No-wait packet scheduling for IEEE time-sensitive networks (TSN) [C]//Proceedings of the 24th International Conference on Real-Time Networks and Systems. New York: ACM Press, 2016: 203-212.
- [17] LI C, ZHANG C Y, ZHENG W, et al. Joint routing and scheduling for dynamic applications in multicast time-sensitive networks[C]//Proceedings of the 2021 IEEE International Conference on Communications Workshops (ICC Workshops). Piscataway: IEEE Press, 2021: 1-6.
- [18] XU L, XU Q M, TU J Z, et al. Learning-based scalable scheduling and routing co-design with stream similarity partitioning for time-sensitive networking[J]. IEEE Internet of Things Journal, 2022, 9(15): 13353-13363.
- [19] VLK M, HANZÁLEK Z, BREJCHOVÁ K, et al. Enhancing schedulability and throughput of time-triggered traffic in IEEE 802.1Qbv time-sensitive networks[J]. IEEE Transactions on Communications, 2020, 68(11): 7023-7038.
- [20] YAN J L, QUAN W, JIANG X Y, et al. Injection time planning: making CQF practical in time-sensitive networking[C]//Proceedings of the IEEE INFOCOM 2020 - IEEE Conference on Computer Communications. Piscataway: IEEE Press, 2020: 616-625.
- [21] 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.
- [22] ZHANG Y Z, XU Q M, XU L, et al. Efficient flow scheduling for industrial time-sensitive networking: a divisibility theory-based method[J]. IEEE Transactions on Industrial Informatics, 2022, 18(12): 9312-9323.
- [23] GUO M, GU C J, HE S B, et al. MSS: exploiting mapping score for CQF start time planning in time-sensitive networking[J]. IEEE Transactions on Industrial Informatics, 2023, 19(2): 2140-2150.
- [24] WANG X L, YAO H P, MAI T L, et al. Joint routing and scheduling with cyclic queuing and forwarding for time-sensitive networks[J]. IEEE Transactions on Vehicular Technology, 2023, 72(3): 3793-3804.
- [25] ALEXANDRIS K, POP P, WANG T T. Configuration and evaluation of multi-CQF shapers in IEEE 802.1 time-sensitive networking (TSN)[J]. IEEE Access, 2022, 10: 109068-109081.
- [26] 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 the ICC 2021 - IEEE International Conference on Communications. Piscataway: IEEE Press, 2021: 1-6.
- [27] 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.
- [28] TABOUCHE A, DJAMAA B, SENOUCI M R, et al. TLR: traffic-aware load-balanced routing for industrial IoT[J]. Internet of things, 2024, 25: 101093.
- [29] PAULITSCH M, SCHMIDT E, GSTÖTTENBAUER B, et al. Time-triggered communication (industrial applications) [C]//Time-Triggered Communication. Berlin: Springer, 2011: 121-152.
- [30] BERISA A, ZHAO L X, CRACIUNAS S S, et al. AVB-aware routing and scheduling for critical traffic in time-sensitive networks with preemption[C]//Proceedings of the 30th International Conference on Real-Time Networks and Systems. New York: ACM Press, 2022: 207-218.
- [31] IEEE Computer Society C/LAN/MAN - LAN/MAN Standards Committee. IEEE/IEC P60802: time-sensitive networking profile for industrial automation[S]. 2023.

作者简介



王颖 (1976-), 女, 内蒙古临河人, 博士, 北京邮电大学副教授、博士生导师, 主要研究方向为网络管理与通信软件、软件化网络、算力网络、确定性网络等。



庄智涵 (1998-), 男, 河南郑州人, 北京邮电大学硕士生, 主要研究方向为确定性网络、时间敏感网络等。



喻鹏 (1986-), 男, 湖北随州人, 博士, 北京邮电大学副教授、博士生导师, 主要研究方向为 5G/6G 网络智能管控。



邱雪松 (1973-), 男, 江西上饶人, 博士, 北京邮电大学教授、博士生导师, 主要研究方向为网络与业务管理、物联网与区块链。



郭少勇 (1985-), 男, 河北邢台人, 博士, 北京邮电大学教授, 主要研究方向为物联网与区块链。