

时间敏感网络流量调度综述

张 彤^{1,2} 冯佳琦^{1,2} 马延滢³ 渠思源³ 任丰原^{3,4}

¹(南京航空航天大学计算机科学与技术学院 南京 211106)

²(软件新技术与产业化协同创新中心 南京 210093)

³(兰州大学信息科学与工程学院 兰州 730000)

⁴(清华大学计算机科学与技术系 北京 100084)

(zhangt@nuaa.edu.cn)

Survey on Traffic Scheduling in Time-Sensitive Networking

Zhang Tong^{1,2}, Feng Jiaqi^{1,2}, Ma Yanying³, Qu Siyuan³, and Ren Fengyuan^{3,4}

¹(College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106)

²(Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing 210093)

³(School of Information Science and Engineering, Lanzhou University, Lanzhou 730000)

⁴(Department of Computer Science and Technology, Tsinghua University, Beijing 100084)

Abstract Many industry field applications like industrial control, avionics, in-vehicle networks, and mobile prequel networks require deterministic low-latency network transmissions. In order to achieve these transmission requirements, the IEEE 802 time-sensitive networking (TSN) working group extends standard Ethernet to TSN, which attracts continuous attentions from both academia and industry. Traffic scheduling is the core mechanism in the TSN standard system, where the scheduling algorithm determines the transmission order and time period of each data frame on egress ports of all switches to meet the respective delay and bandwidth requirements of traffic and optimize transmission performance at the same time. In this paper, we first describe the formalization of traffic scheduling problem in TSN, introduce network and traffic models, and conclude the scheduling constraints and goals. A simple example is employed to illustrate the traffic scheduling task and procedure. Then we analyze and summarize existing TSN traffic scheduling mechanisms, focusing on their solved problems, concerned traffic type, optimized performance metrics and solving algorithms. Finally, the design space and development trends of future TSN traffic scheduling are discussed, and a novel idea of joint static traffic planning and dynamic traffic scheduling is proposed in response to the existing problems of current scheduling mechanisms.

Key words time-sensitive networking; traffic scheduling; integer linear programming; heuristic algorithm; metaheuristic algorithm

摘要 工业控制、航空电子、车载网络、移动前传网络等很多行业领域应用都需要确定性低延时的网络传输。为了实现此类业务的传输需求,IEEE 802时间敏感网络(time-sensitive networking, TSN)工作组将

收稿日期:2021-03-12;修回日期:2021-07-30

基金项目:国家自然科学基金项目(62002165,61872208);江苏省自然科学基金项目(BK20200445);国家重点研发计划项目(2018YFB1700103,2018YFB1700203)

This work was supported by the National Natural Science Foundation of China (62002165, 61872208), the Natural Science Foundation of Jiangsu Province (BK20200445), and the National Key Research and Development Program of China (2018YFB1700103, 2018YFB1700203).

通信作者:任丰原(renfy@tsinghua.edu.cn)

标准以太网扩展为 TSN,受到学术界和工业界的持续关注。流量调度是 TSN 标准中的核心机制,通过调度算法在所有交换机出端口确定数据帧传输顺序和时间,满足流量各自的延时和带宽要求并同时优化传输性能。首先对 TSN 流量调度问题进行形式化描述,介绍了 TSN 网络与流量模型,并对调度约束和目标进行归纳;进而对现有 TSN 流量调度机制进行分析与总结,重点阐述每种调度机制解决的具体问题、关注的流量类型、优化的性能指标和求解算法;最后讨论了未来 TSN 流量调度的设计空间和发展趋势,并针对现有调度机制存在的问题提出了静态规划与动态调节联合的调度思路。

关键词 时间敏感网络;流量调度;整数线性规划;启发式算法;元启发式算法

中图法分类号 TP393

当前,很多行业领域都需要确定性低延时的网络连接。例如工业自动化网络要求端到端延时在几微妙到几毫秒之间^[1];汽车内的自动驾驶系统要求车载网络的端到端延时在 250 μ s 以内,车内电子控制系统更要求不超过 10 μ s^[2];航空电子全双工交换以太网(avionics full duplex switched Ethernet, AFDX)则要求 128 ms 以内的端到端延时^[3]等。除延时性能外,以上应用场景还要求几微妙的延时抖动和极低的丢包率。传统以太网致力于通过带宽复用提高资源利用率,然而缺少确定性传输机制只能提供尽力而为的传输服务,难以满足上述应用场景的传输需求。

为了实现确定性低延时传输,工业界在标准以太网的基础上提出了多种专属网络协议,如实时以太网 TTEthernet, EtherCAT, PROFINET, SERCOS III 等,称为工业以太网,逐步成为工业控制网络的主流技术。然而,互不兼容的网络技术导致了应用难兼容、互操作性差、不易移植且开发部署和运营维护成本高等问题。随着工业互联网的发展,在日益增长的互联互通和确定性网络标准化需求的驱动下,IEEE 802 将原本致力于满足带宽预留业务需求的音/视频桥接(audio video bridging, AVB)工作组升级为时间敏感网络(time-sensitive networking, TSN)工作组,提出了一系列链路层增强机制与流量策略的标准和规范,主要包括时间同步、流量调度、可靠传输和网络管理标准,将标准以太网扩展为 TSN^[4]。TSN 遵循标准的以太网协议体系,天然具有更好的互联互通优势,可以在提供确定性延时、带宽保证的同时,实现开放的 2 层转发^[5]。因此,TSN 可以对相互隔离的工业控制网络进行互联。近年来,TSN 作为新一代以太网技术,在工业互联网、移动前传、航空电子网络、车载网络、专业音视频等多个领域广泛应用,得到了学术界和工业界的持续关注。

流量调度是 TSN 标准中的核心机制。不同类别

的业务流量对网络的端到端延时和带宽具有不同的要求。流量调度通过一定的调度算法在所有交换机出端口确定每个数据帧的传输顺序和时间,保证所有帧在出口链路上依次传输而不会发生冲突,同时在全网范围联合保证每个帧能够顺利通过传输路径上的所有出端口,并满足流量各自的延时和带宽要求,使不同类别的业务流量在同一网络上得以共存^[6]。

文献[6]对 TSN 的关键协议及应用场景进行综述,阐述了时间同步协议、调度与流量整形标准、帧抢占标准、流预留协议,以及 TSN 在车载网络、工业互联网、航空电子网络和移动前传网络中的应用场景。文献[7]对车载嵌入式系统相关的 TSN 标准进行综述,包括标准的演进、基于模型的分布式软件开发、调度与可调度性分析、仿真平台、硬件演进和安全性。在调度与可调度性分析部分对现有调度机制根据性能目标进行了简单分类和描述。本文主要关注 TSN 中的流量调度,将介绍流量调度的标准框架、网络与流量模型、所有调度约束和可能的调度目标的形式化表征,对流量调度问题进行系统建模;进而按照采用的求解算法对现有调度机制进行全面的分类和详细描述,并对 TSN 流量调度的研究现状进行分析和总结;最后讨论 TSN 流量调度的设计空间和发展趋势。本文是对文献[6-7]中流量调度部分的系统化展开和扩展,补充了调度框架、问题和机制的细节。

1 TSN 流量调度标准框架

TSN 属于确定性网络,即网络中的部分流量必须遵循某种硬实时条件的网络。TSN 中实时流量的端到端延时必须小于等于 1 个确定值。此外,工业控制网络协议中使用最为广泛的 TTEthernet 也属于确定性网络。TTEthernet 是为工业控制网络中混合关键性实时应用所提出的在标准以太网之上的扩展

机制.TTEthernet 与 TSN 在很多方面存在相似之处,然而在流量调度上存在重要区别^[8].本节首先对比了 TSN 和 TTEthernet 流量调度标准框架,进而介绍 TSN 中固有的流量控制机制.TSN 中的流量调度必须在标准框架和固有机制的基础上运行.

1.1 TSN 与 TTEthernet 流量调度框架

图 1 分别展示了 TTEthernet 交换机和 TSN 交换机,每个交换机只画出了 2 个入端口和 1 个出端口.流量调度作用于网络中所有交换机的出端口.TTEthernet 和 TSN 在每个出端口均使用了 8 个优先级队列,用来标记流量的重要程度.二者的区别在于,TTEthernet 交换机在每个出端口设置 1 个共享缓冲区来存放该出端口暂未传输的帧,调度机制决定的是帧从缓冲区进入优先级队列的时间;而 TSN 交换机中共享缓冲区的位置由优先级过滤器代替,帧直接进入优先级队列进行缓存.此外,TTEthernet 的优先级队列直连出端口,一旦出端口空闲,队列中的帧可立即按优先级传输;而 TSN 的每个优先级队列通过门电路与出端口相连,只有门开放时相应队列中的帧才可按优先级传输.调度机制决定的是不同队列门的开闭时刻.

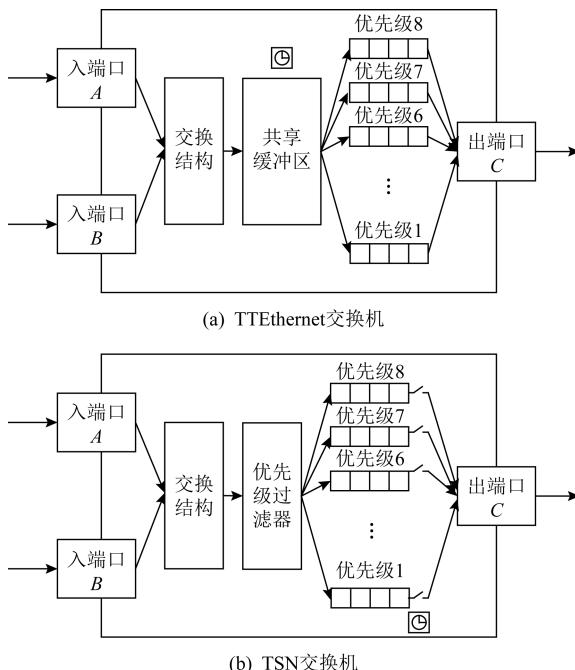


Fig. 1 Illustration of TTEthernet switch and TSN switch

图 1 TTEthernet 交换机与 TSN 交换机示意图

TTEthernet 中的每台交换机都具备根据本地实时网络调度分配数据帧的能力.本地实时网络调度可由全局传输机制推导得出,规定了每个实时数据帧在网络每一跳的发送和接收时间窗口.在交换

机内部,实时网络调度则规定了每个出端口数据帧从缓冲区进入优先级队列的时刻.不同队列之间按严格优先级传输.实时流量与非实时流量进入不同队列进行隔离.实时流量进入高优先级队列,非实时流量进入低优先级队列,以减少对实时流量的干扰,保证实时流量在交换机内的有界传输延时.TTEthernet 需要时间同步协议 SAE AS6802 提供全网的统一时钟,各交换机协同调度,使得数据帧能够顺利地依次通过路径上的各交换机,保证实时流量端到端延时也有界.

TSN 由精确时间协议(precise time protocol, PTP)或通用精确时间协议(generic PTP, gPTP)提供全网各节点的时钟同步.在统一时钟的基础上,由端口门控制列表(gate control list, GCL)在不同时刻开放和关闭各队列出口的门,决定各队列是否可以传输数据.GCL 可根据实时流量的周期和尺寸离线生成.表中每个门开放的时间段称为时间窗口,每个时间窗口可传输 1 个或多个数据帧,取决于窗口大小.实时流量进入高优先级队列,非实时流量进入低优先级队列,在门开放的所有队列之间按严格优先级传输.TSN 和 TTEthernet 在流量调度上的区别主要体现在 3 个方面:1) 流量特性不同.TTEthernet 定义的流量类型包括时间触发(time-triggered, TT)流量、速率限制(rate-constrained, RC)流量和尽力而为(best-effort, BE)流量,而 TSN 定义的流量类型包括 TT 流量、音视频桥接(AVB)流量和 BE 流量.RC 流量和 AVB 流量均为事件触发流量,但 AVB 流量主要传输音频和视频.TTEthernet 的消息只包含 1 帧,而 TSN 中每个消息可能包含多个帧.2) 调度位置不同.TTEthernet 的调度位于共享缓冲区,决策的是每个帧进入优先级队列的时刻;而 TSN 的调度位于优先级队列出口,决策的是各队列的门的开闭时刻.3) 调度粒度不同.TTEthernet 的调度粒度为单个数据帧,而 TSN 的调度粒度为队列.由于调度粒度更粗,TSN 在单个帧传输过程中存在一定程度的不确定性,提供的确定性等级低于 TTEthernet,但对调度方案的约束更少,扩大了解空间,具有更多的灵活性.

1.2 TSN 固有流量控制机制

所有流量调度机制都需要在 TSN 固有流量控制机制的基础上运行.因此本节对 TSN 中定义的固有流量控制机制进行简单介绍,主要包括保护带机制、帧抢占机制和流量整形机制.

1.2.1 保护带机制

标准以太网未定义抢占机制,如果1个数据帧已经开始在端口传输,即使更高优先级的帧到达该端口也必须先等待当前帧传输完成,这称为优先级反转.这种现象使得实时帧可能在预设的时间窗口内无法完成传输,只能等待下个窗口,又会导致下个窗口内预设的其他帧无法完成传输.这种连锁反应会造成该出端口及路径上后续所有出端口的帧传输错乱,大幅增加端到端延时,导致不确定性.为缓解这种现象,TSN提出了保护带机制,主要有2种实现方法:1)在每个实时队列的时间窗口前设置一段所有门全部关闭的时间作为保护带,长度为最大帧传输时长.保护带内不允许任何新的帧开始传输,但之前已经在传输的帧可在保护带内继续完成传输.这样在实时队列门开放时一定不会发生优先级反转.2)不显式设置保护带,而是在每次实时队列的门开放前进行判断,阻止会干扰实时帧传输的非实时帧,防止优先级反转的发生.保护带机制能够避免非实时流量对实时流量的干扰,但会同时引入带宽浪费.

1.2.2 帧抢占机制

IEEE 802.3br^[9]和 IEEE 802.1Qbu^[10]标准共同定义了帧抢占机制以解决优先级反转问题. IEEE 802.3br 定义了 MAC 合并子层, 以及帧抢占的切片操作、切片还原和验证等核心功能. IEEE 802.1Qbu 则提供了抢占接口及模块级别的定义. 若端口上高优先级帧到达时低优先级帧正在传输,首先判断低优先级帧已传输的部分和剩余部分是否均大于等于以太网的最小帧长(60 B),若满足则允许切片操作:中断低优先级帧的传输,并将低优先级帧已传输的部分补上校验和组装成完整的帧,然后在1个帧间隙后传输高优先级帧,待高优先级帧传输完成后低优先级帧剩余部分补全为完整的帧继续传输.与保护带机制相比,帧抢占机制减少了带宽浪费,但是会引入不可避免的操作开销.此外,抢占机制需要专用的硬件结构支持,增加了硬件复杂度;不满足切片条件的情况下无法实施抢占操作,不能完全消除优先级反转问题.

1.2.3 流量整形机制

流量整形的主要功能是限制网络中的突发流量,将突发流量以较为均匀的速率向外发送可以减少网络拥塞并降低丢包率.TSN固有的流量整形机制主要包括时间感知整形器(time-aware shaper, TAS)、基于信用值的整形器(credit-based shaper, CBS)、

循环排队和转发(cyclic queuing and forwarding, CQF)和异步流量整形器(asynchronous traffic shaping, ATS).

TAS由IEEE 802.1Qbv标准定义^[11],在GCL的时间窗口中调度不同优先级队列的流量传输.在全网时钟同步的条件下,TAS中GCL周期性地控制各队列出口门的开闭,并且每一时刻在所有门开放的队列中按照严格的优先级进行传输.TAS的基本思想是时分多址(time division multiple access, TDMA)技术,为不同队列分配不重叠的传输时隙,使流量传输速率匹配出口链路带宽的同时对不同优先级的流量进行隔离,减少低优先级流量对高优先级流量的干扰.由于需要时间同步操作且门控列表的配置较为复杂,TAS在扩展性方面存在挑战.

CBS由IEEE 802.1Qav标准定义^[12],是一种常用的网络流量整形算法,在TSN中作用于AVB流量.CBS赋予每个AVB队列1个信用值,初始为0.当队列中AVB帧传输时,信用值以发送率为斜率线性降低;当队列中AVB帧等待时,信用值以空闲率为斜率线性增长;当AVB队列出口的门为关闭状态时,信用值保持不变;当AVB队列为空而信用值大于0时,将信用值重置为0.只有当队列的信用值非负时,AVB帧才可开始传输.CBS通常与TAS联合使用,因此AVB帧开始传输需要满足3个条件:1)队列出口门开放;2)队列信用值非负;3)门开放的更高优先级队列中无数据传输.CBS在TAS的基础上进一步对AVB流量的传输进行整形,降低了AVB流量的突发并缓解了更低优先级队列的饥饿.

CQF由IEEE 802.1Qch标准定义^[13],主要解决帧传输的有界延时问题.桥接网络基础标准802.1Q^[3]指定的延时敏感流的转发和排队机制中最坏情况的端到端延时与网络拓扑结构密切相关,难以实现延时敏感流量所要求的有界延时.为解决此问题,CQF以循环的方式协调交换机的入队和出队操作.CQF将时间分为等长的周期,且所有交换机的周期对齐.每个流量等级使用2个队列,偶数周期内队列1接收新到达的帧而不传输帧,队列2传输上一周期到达的帧而不接收新的帧;奇数周期内,队列2接收新到达的帧而不传输帧,队列1传输上一周期到达的帧而不接收新的帧.因此在第*i*周期到达的帧将在第*i*+1周期内传输,且必须在该周期内被下一跳交换机接收.需要精确设置周期长度,既保证周期内所有帧能顺利传输至下一跳,又不会造成过长的延时.通过CQF,最坏情况的端到端延时只与

周期长度和网络跳数有关,与拓扑结构无关,上下界均很容易计算.

ATS 由 IEEE 802.1Qcr 标准定义^[14],是基于紧迫度调度器(urgency-based scheduler, UBS)^[15]的异步整形机制,适用于满足漏桶速率限制的所有周期或非周期流量,即任意时间间隔 d 内,流 s_i 的累计数据量 $w_i(d) \leq b_i + r_i d$, b_i 为突发程度, r_i 为漏桶速率. ATS 在每个出端口设置 2 层队列:整形队列和优先队列. 每个整形队列设单独的整形器,能够对队列中的多条流进行交叉整形;优先队列决定输出流量的优先级,队列与优先级一一对应. 到达出端口的流量先进入整形队列,整形器为队中的每条流单独维护一个时间戳,记录该流最近一次传输帧的时刻,并采用漏桶算法或令牌桶算法确定下一帧的传输时刻,以此限制每条流的传输速率. 每条优先队列从所有整形队列的出口收集相应优先级的流量,队列间按严格优先级传输. ATS 中每个交换机根据本地时间对流量进行整形和优先级调度,不需要全局时钟同步,因此在扩展性上优于 TAS, CBS, CQF.

2 TSN 流量调度问题

本节主要介绍 TSN 流量调度的形式化问题,包括 TSN 网络模型、流量模型、调度约束和调度目标. TSN 流量调度问题有 2 类形式化方法:1)以帧为调度对象,计算每一帧在经过的所有出端口的队列分配及传输开始时刻;2)以时间窗口为调度对象,计算每一帧在所有出端口上的时间窗口分配以及所有时间窗口的开放和关闭时刻. 可以看出,第 1 类问题求解的是 TTEthernet 的调度对象,第 2 类问题求解的是 TSN 的调度对象. 然而,第 1 类问题的调度结果可直接转化为第 2 类问题的结果,反之则不成立. 通过队列中所有帧的传输开始时刻和帧大小可推导出该队列传输帧的所有时段,即时间窗口;但已知时间窗口的开闭时刻并不能确定窗口内每一帧的传输时间. 因此第 1 类形式化问题更加精确,并且同时适用于 TTEthernet 和 TSN. 可见尽管 TTEthernet 与 TSN 的实际调度框架存在显著区别,但在形式化的调度问题是一致的. 本文即介绍此类形式化调度问题.

2.1 网络与流量模型

将网络抽象为有向图 $G = (V, E)$, 如图 2 所示. V 为点集, 表示网络中的交换机(switch, SW)和端

系统(end system, ES). 图 2 包括 2 个交换机和 4 个端系统, 每个交换机有多个入端口和出端口, 负责根据目的地址将入端口的帧转发至对应出端口; 端系统可以是传感器、执行器、电子控制单元, 通常包含 CPU、内存和网卡. $E \subseteq V \times V$ 为边集, 其中每个元素代表 1 条单向链路. TSN 建立在全双工以太网之上, 节点 v_a 与 v_b 间的物理链路对应 2 条单向链路 $[v_a, v_b]$ 和 $[v_b, v_a]$. 每条单向链路 $[v_a, v_b]$ 由三元组 $\langle c_{ab}, d_{ab}, n_{ab} \rangle$ 定义, 分别表示该链路的带宽容量、传播延时以及相连的出端口队列数. 在 TSN 标准中, 端系统出口链路 $n_{ab} = 1$, 而交换机出口链路 $n_{ab} = 8$.

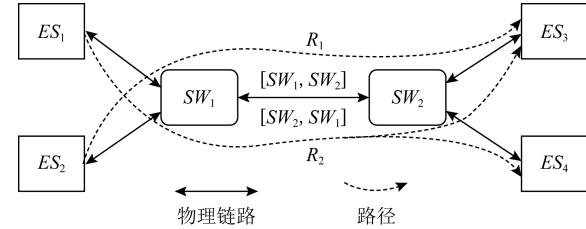


Fig. 2 TSN network model

图 2 TSN 网络模型

TSN 中的流量包括 TT 流、AVB 流和 BE 流, 传输优先级依次降低. TT 流量用于具有严格时间约束的实时应用, 需要确定性低延时保证; AVB 流量用于软实时应用, 提供有界的最坏情况端到端延时, 但比 TT 流量的延时约束宽松; BE 流量为传统的以太网流量, 无任何延时保证. 其中 TT 流为周期性数据传输, 周期长度和数据大小由应用预先定义, 为已知量; 而 AVB 流和 BE 流为非周期性随机数据传输, 到达间隔与数据大小均为未知量. 由于 TSN 的流量调度为离线操作, 无法处理未知的 AVB 和 BE 流量, 因此只对 TT 流进行形式化定义. 从发送端 v_{i1} 经过中间节点 $v_{i2}, \dots, v_{in_i-1}$ 到达接收端 v_{in_i} 的 TT 流 s_i , 其路径 R_i 可表示为 $[[v_{i1}, v_{i2}], [[v_{i2}, v_{i3}], \dots, [v_{in_i-1}, v_{in_i}]]$. 每条 s_i 可采用四元组 $\langle L_i, J_i, S_i, T_i \rangle$ 定义, 分别表示该流所能容忍的最大端到端延时、最大延时抖动、每周期发送的数据量和周期长度. 定义 $\mathcal{F}^{[v_a, v_b]}$ 表示通过链路 $[v_a, v_b]$ 的流集合, 则链路 $[v_a, v_b]$ 的超周期 $h^{[v_a, v_b]} = \text{lcm}\{T_i, s_i \in \mathcal{F}^{[v_a, v_b]}\}$ 为通过该链路所有 TT 流周期的最小公倍数. 如图 3 所示, 流 s_1 的周期为 $100 \mu\text{s}$, 流 s_2 的周期为 $150 \mu\text{s}$, 则 2 条流共同链路的超周期为 $300 \mu\text{s}$. 调度机制只需确定 1 个超周期的 GCL 并在多个超周期间循环执行. $p_i^{[v_a, v_b]}$ 则表示 s_i 在与链路 $[v_a, v_b]$ 相连的出端口上的队列分配, 因此有 $p_i^{[v_a, v_b]} \leq n_{ab}$.

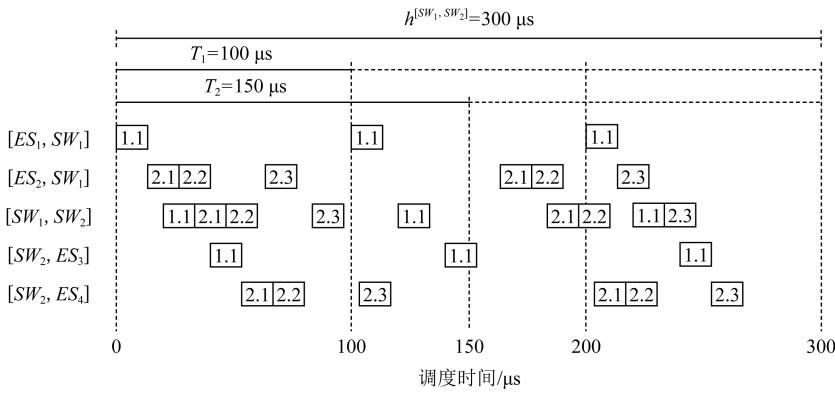


Fig. 3 TT schedule synthesis example

图 3 TT 流调度实例

所有流在 TSN 中均以帧为传输单位, 每一帧的载荷不超过以太网的最大传输单元(maximum transmission unit, MTU) 1 500 B. 若 1 次传输的消息大小超过 MTU, 则拆分为多个帧传输. $\mathcal{F}_i^{[v_a, v_b]}$ 为流 s_i 在 $hp^{[v_b, v_a]}$ 内于链路 $[v_a, v_b]$ 上传输的帧传输实例的集合, $f_{ij}^{[v_a, v_b]}$ 则为其中的第 j 个帧传输实例, 可由三元组 $\langle \phi_{ij}^{[v_a, v_b]}, T_{ij}^{[v_a, v_b]}, l_{ij}^{[v_a, v_b]} \rangle$ 定义, 分别表示该实例相对于超周期起始的传输时刻、周期长度和传输时长, 可知 $T_{ij} = T_i$. 每个帧在传输路径的每条链路上均产生 1 个帧传输实例.

流量调度需要保证所有帧在网络中无冲突传输并优化传输指标, 可知调度变量为每条流在每个出端口的队列分配 $p_i^{[v_a, v_b]}$ 和所有实例的传输时刻 $\phi_{ij}^{[v_a, v_b]}$.

以图 2 中的网络和表 1 中的流量为例, s_1 的路径为 $[[ES_1, SW_1], [SW_1, SW_2], [SW_2, ES_3]]$, s_2 的路径为 $[[ES_2, SW_1], [SW_1, SW_2], [SW_2, ES_4]]$, 2 条流在链路 $[SW_1, SW_2]$ 上存在争用; s_1 的周期为 $100 \mu s$, 大小为 $1 500 B$, 即 1 个 MTU, s_2 的周期为 $150 \mu s$, 大小为 $4 500 B$, 即 3 个 MTU, 2 条流所能容忍的最大端到端延时均为 $2 ms$, 最大延时抖动均为 $5 \mu s$. 假设链路传播延时与交换机处理延时均为 $1 \mu s$, 所有链路带宽均为 $1 Gbps$, 则每个帧的传输延时为 $12.336 \mu s$. 图 3 采用甘特图展示了一种

Table 1 Flow Instance

表 1 流量实例

| 流 | 源 | 目的 | 周期 / μs | 尺寸 /B | 最大延时 /ms | 最大抖动时间 / μs |
|-------|--------|--------|--------------|-------|----------|------------------|
| s_1 | ES_1 | ES_3 | 100 | 1 500 | 2 | 5 |
| s_2 | ES_2 | ES_4 | 150 | 4 500 | 2 | 5 |

可行调度方案. 横坐标表示时间, 纵坐标表示链路, 矩形表示帧传输实例.

在 $300 \mu s$ 的超周期内, s_1 重复 3 轮, 而 s_2 重复 2 轮. 每个帧传输实例在每个周期内的起始时间相同, 需要经历传输延时、传播延时和交换机处理延时到达下一跳出端口队列. s_1 的优先级高于 s_2 , 因此 s_1 的帧在周期开始时传输, 并且到达下一出端口后立刻传输; s_2 的帧则在 s_1 不传输时尽快传输. s_2 的第 2 周期内, 帧传输实例 $f_{23}^{[ES_2, SW_1]}$ 需要等待 $f_{11}^{[ES_1, SW_1]}$ 传输完成以避免冲突, 因此与前 2 个传输实例产生间隔.

2.2 调度约束

为保证帧在网络中真实、按序、无冲突传输, $p_i^{[v_a, v_b]}$ 和 $\phi_{ij}^{[v_a, v_b]}$ 需要满足 6 个约束条件.

1) 帧约束. 帧约束为 $\phi_{ij}^{[v_a, v_b]}$ 的基本约束, 要求 $\phi_{ij}^{[v_a, v_b]}$ 非负, 且必须保证 $f_{ij}^{[v_a, v_b]}$ 在其周期内完成传输. 对 $\forall s_i \in S, [v_a, v_b] \in R_i, f_{ij}^{[v_a, v_b]} \in \mathcal{F}_i^{[v_a, v_b]}$, 有:

$$\begin{cases} \phi_{ij}^{[v_a, v_b]} \geq 0, \\ \phi_{ij}^{[v_a, v_b]} \leq T_{ij}^{[v_a, v_b]} - l_{ij}^{[v_a, v_b]}. \end{cases}$$

以图 3 为例, s_1 需要满足 $\phi_{11}^{[ES_1, ES_2]} \geq 0$ 和 $\phi_{11}^{[SW_2, ES_2]} \leq 100 \mu s - 12.336 \mu s = 87.664 \mu s$, s_2 需要满足 $\phi_{21}^{[ES_2, SW_2]} \geq 0$ 和 $\phi_{21}^{[ES_2, SW_2]} \leq 150 \mu s - 12.336 \mu s = 137.664 \mu s$.

2) 链路约束. 链路约束为无冲突传输的核心约束, 要求通过同一链路任意 2 帧传输实例在时间上没有重叠. 对 $\forall [v_a, v_b] \in E, f_{ik}^{[v_a, v_b]} \in \mathcal{F}_i^{[v_a, v_b]}, f_{jl}^{[v_a, v_b]} \in \mathcal{F}_j^{[v_a, v_b]}$, 且 $i \neq j$, 有:

$$\begin{aligned} \phi_{ik}^{[v_a, v_b]} + \alpha T_{ik}^{[v_a, v_b]} &\geq \phi_{jl}^{[v_a, v_b]} + \beta T_{jl}^{[v_a, v_b]} + l_{jl}^{[v_a, v_b]} \vee \\ \phi_{jl}^{[v_a, v_b]} + \beta T_{jl}^{[v_a, v_b]} &\geq \phi_{ik}^{[v_a, v_b]} + \alpha T_{jl}^{[v_a, v_b]} + l_{ik}^{[v_a, v_b]}, \end{aligned}$$

α, β 均为整数, $\alpha \in \left[0, \frac{h_{ij}}{T_i} - 1\right]$, $\beta \in \left[0, \frac{h_{ij}}{T_j} - 1\right]$, h_{ij} 为 T_i 和 T_j 的最小公倍数. 该约束表明同一链路的任意 2 个帧传输实例之间, 其中 1 个实例的传输开始时刻必须大于等于另一个实例的完成时刻. 图 3 中 s_1 和 s_2 通过同一链路 $[SW_1, SW_2]$, 因此需要同时满足:

$$\begin{aligned} \phi_{11}^{[SW_1, SW_2]} + \alpha T_1 &\geq \phi_{21}^{[SW_1, SW_2]} + \beta T_2 + 12.336 \mu s \vee \\ \phi_{21}^{[SW_1, SW_2]} + \beta T_2 &\geq \phi_{11}^{[SW_1, SW_2]} + \alpha T_1 + 12.336 \mu s, \\ \phi_{11}^{[SW_1, SW_2]} + \alpha T_1 &\geq \phi_{22}^{[SW_1, SW_2]} + \beta T_2 + 12.336 \mu s \vee \\ \phi_{22}^{[SW_1, SW_2]} + \beta T_2 &\geq \phi_{11}^{[SW_1, SW_2]} + \alpha T_1 + 12.336 \mu s, \\ \phi_{11}^{[SW_1, SW_2]} + \alpha T_1 &\geq \phi_{23}^{[SW_1, SW_2]} + \beta T_2 + 12.336 \mu s \vee \\ \phi_{23}^{[SW_1, SW_2]} + \beta T_2 &\geq \phi_{11}^{[SW_1, SW_2]} + \alpha T_1 + 12.336 \mu s, \end{aligned}$$

其中, $\alpha = 0, 1, \beta = 0, 1, 2$.

3) 流传输约束. 流传输约束规定了帧通过路径上每条链路的时序. 对 $\forall s_i \in S, [v_a, v_x], [v_x, v_b] \in R_i, f_{ij}^{[v_a, v_x]} \in \mathcal{F}_i^{[v_a, v_x]}, f_{ij}^{[v_x, v_b]} \in \mathcal{F}_i^{[v_x, v_b]}$, 都有:

$$\phi_{ij}^{[v_x, v_b]} \geq \phi_{ij}^{[v_a, v_x]} + l_{jl}^{[v_a, v_x]} + d_{ax} + p_a + \delta,$$

即同一帧在后继链路上传输实例的开始时刻必须大于等于前驱链路上传输实例的完成时刻, 即前驱链路上的开始时刻、传输延时、传播延时之和. d_{ax} 为链路 $[v_a, v_x]$ 的传播延时, p_a 为节点 v_a 的处理延时, δ 为全网内时钟偏差的最大值, 用于在时钟同步存在误差的情况下保证链路传输的时序. 以图 3 中的 s_1 为例, 其帧传输实例在 3 条链路上的开始时间需满足:

$$\begin{aligned} \phi_{11}^{[SW_1, SW_2]} &\geq \phi_{11}^{[ES_1, SW_1]} + 12.336 \mu s + 1 \mu s + 1 \mu s + \delta, \\ \phi_{11}^{[SW_2, ES_3]} &\geq \phi_{11}^{[SW_1, SW_2]} + 12.336 \mu s + 1 \mu s + 1 \mu s + \delta. \end{aligned}$$

4) 延时约束. 该约束阐述了实时流量的端到端延时约束. 对 S 中的 $\forall s_i = \langle L_i, J_i, S_i, T_i \rangle$, 其路径 $R_i = [[v_{i1}, v_{i2}], [[v_{i2}, v_{i3}], \dots, [v_{i(n_i-1)}, v_{in_i}]]$, $N_i = \lceil \frac{S_i}{S_{MTU}} \rceil$ 表示 s_i 每次发送的消息所包含的帧数, 有:

$$\begin{aligned} \phi_{iN_i}^{[v_{i(n_i-1)}, v_{in_i}]} + l_{iN_i}^{[v_{i(n_i-1)}, v_{in_i}]} + d_{[v_{i(n_i-1)}, v_{in_i}]} - \\ \phi_{i1}^{[v_{i1}, v_{i2}]} \leq L_i, \end{aligned}$$

消息最后一帧到达接收端的时刻与第 1 帧在发送端传输开始时刻之间的时间间隔即为端到端延时, 必须小于等于流 s_i 能容忍的最大端到端延时 L_i . 满足延时约束的流称为可调度的, 反之则称为不可调度的. 以图 3 中的 s_1 为例, 需要满足:

$$\phi_{11}^{[SW_2, ES_3]} - \phi_{11}^{[ES_1, SW_1]} \leq 2 \text{ ms} - 12.336 \mu s - 1 \mu s.$$

5) 抖动约束. 该约束阐述了实时流量的延时抖

动约束. 对 S 中的 $\forall s_i = \langle L_i, J_i, S_i, T_i \rangle$, 其路径 $R_i = [[v_{i1}, v_{i2}], [[v_{i2}, v_{i3}], \dots, [v_{i(n_i-1)}, v_{in_i}]]$, $N_i = \lceil \frac{S_i}{S_{MTU}} \rceil$ 表示 s_i 每次发送的消息所包含的帧数, D_i 为消息从发送端到最后一跳之前的总传输、传播和处理延时, 有:

$$D_i = \sum_{k=1}^{N_i-1} l_{ik}^{[v_{i1}, v_{i2}]} + \sum_{l=1}^{n_i-2} (l_{iN_i}^{[v_{il}, v_{i(l+1)}]} + \\ d_{v_{il}v_{i(l+1)}} + p_{v_{i(l+1)}}),$$

$\omega_i^{[v_a, v_b]}$ 表示流 s_i 在链路 $[v_a, v_b]$ 上的理想等待时间, 即无抖动状态下最早可传输时间与到达时间的差值, $W_i = \sum_{l=2}^{n_i-1} \omega_i^{[v_{il}, v_{i(l+1)}]}$ 表示从发送端到最后一跳之前总的理想等待时间, 有:

$$\phi_{iN_i}^{[v_{i(n_i-1)}, v_{in_i}]} - \phi_{i1}^{[v_{i1}, v_{i2}]} - D_i - W_i \leq J_i,$$

消息最后一帧从最后一跳传输开始时刻与第 1 帧在发送端传输开始时刻之间的间隔必须小于等于发送端与最后一跳之间的传输延时、理想等待延时和所能容忍的最大延时抖动之和. 以 s_2 为例, $N_2 = 3$, $n_2 = 3$, 在最后一跳之前需通过 2 条链路, 因此 $D_2 = 2 \times 12.336 + 2 \times (12.336 + 1 + 1) = 53.344 \mu s$. 此外 s_2 需要在链路 $[ES_2, SW_1]$ 上分别等待 s_1 的帧传输完成, 等待时间包括 $f_{21}^{[ES_2, SW_1]}$ 和 $f_{23}^{[ES_2, SW_1]}$ 前的间隔 $W_2 = 12.336 + 25.328 = 37.664 \mu s$, 需要满足 $\phi_{23}^{[SW_2, ES_4]} - \phi_{21}^{[ES_2, SW_1]} \leq 5 \mu s + 53.344 \mu s + 37.664 \mu s$.

6) 帧隔离约束. 在同一队列中, 若有多条流的帧同时到达, 则进入队列的顺序是不确定的, 进而出口链路的传输序列也是不确定的, 导致端到端延时的不确定性. 此外, 在实际网络中帧可能丢失, 若来自不同流的帧在队列中交错, 当有丢帧时这些流实际的传输时隙会与预设的调度时隙产生偏差, 同样导致延时的不确定性. 为了避免以上问题, 帧隔离约束规定 1 条队列中同一时刻只能存储来自 1 条流的帧. 对 $\forall [v_a, v_b] \in E, f_{ik}^{[v_a, v_b]} \in \mathcal{F}_i^{[v_a, v_b]}, f_{jl}^{[v_a, v_b]} \in \mathcal{F}_j^{[v_a, v_b]}$ 且 $i \neq j$, $[v_x, v_a]$ 与 $[v_y, v_a]$ 为该链路的任意 2 条前驱链路, 都有:

$$\begin{aligned} \phi_{ik}^{[v_a, v_b]} + \alpha T_i + \delta &\leq \phi_{jl}^{[v_y, v_a]} + \beta T_j + d_{ya} + p_a \vee \\ \phi_{jl}^{[v_a, v_b]} + \beta T_j + \delta &\leq \phi_{ik}^{[v_x, v_a]} + \alpha T_i + d_{xa} + p_a \vee \\ p_i^{[v_a, v_b]} &\neq p_j^{[v_a, v_b]}. \end{aligned}$$

同一队列中的任意 2 条流之间, 只有当 1 条流的帧全部离开队列, 另一条流的帧才可以开始进入队列; 若 2 条流被分配至不同队列中, 则无此要求. 以图 3 中的 s_1, s_2 为例, 在进入交换机 SW_1 时需同时满足:

$$\begin{aligned}
 \phi_{11}^{[SW_1, SW_2]} + \alpha T_1 + \delta &\leq \phi_{21}^{[ES_2, SW_1]} + \beta T_2 + 2 \mu s \vee \\
 \phi_{21}^{[SW_1, SW_2]} + \beta T_2 + \delta &\leq \phi_{11}^{[ES_1, SW_1]} + \alpha T_1 + 2 \mu s \vee \\
 p_1^{[SW_1, SW_2]} &\neq p_2^{[SW_1, SW_2]}, \\
 \phi_{11}^{[SW_1, SW_2]} + \alpha T_1 + \delta &\leq \phi_{22}^{[ES_2, SW_1]} + \beta T_2 + 2 \mu s \vee \\
 \phi_{22}^{[SW_1, SW_2]} + \beta T_2 + \delta &\leq \phi_{11}^{[ES_1, SW_2]} + \alpha T_1 + 2 \mu s \vee \\
 p_1^{[SW_1, SW_2]} &\neq p_2^{[SW_1, SW_2]}, \\
 \phi_{11}^{[SW_1, SW_2]} + \alpha T_1 + \delta &\leq \phi_{23}^{[ES_2, SW_1]} + \beta T_2 + 2 \mu s \vee \\
 \phi_{23}^{[SW_1, SW_2]} + \beta T_2 + \delta &\leq \phi_{11}^{[ES_1, SW_1]} + \alpha T_1 + 2 \mu s \vee \\
 p_1^{[SW_1, SW_2]} &\neq p_2^{[SW_1, SW_2]},
 \end{aligned}$$

其中, $\alpha = 0, 1, \beta = 0, 1, 2$.

2.3 调度目标

流量调度在满足 2.2 节约束条件的同时优化传输性能.由于 TSN 支持的应用种类繁多,对网络性能要求多样,流量调度也有着不同的优化目标.列出当前常用的 TSN 流量调度优化目标:

1) 最小化端到端延时.一种典型的目标为最小化实时流量的端到端延时.尽管实时流量具有明确的延时上界,但部分实时应用期望在此基础上进一步减小该延时以加速应用运行,因此有:

$$\min \sum_{s_i \in S} (\phi_{iN_i}^{[v_i(n_i-1), v_{in_i}]} + l_{iN_i}^{[v_i(n_i-1), v_{in_i}]} - \phi_{i1}^{[v_{i1}, v_{i2}]}),$$

即最小化所有实时流端到端延时之和.

2) 最小化实时流量占用队列数.从帧隔离约束中可看出,若 2 条流被分配至同一队列,则需要在时间上进行隔离;若分配至不同队列,则 2 条流已经在空间上隔离,无需时间上的隔离.时间上的隔离为满足实时流量的端到端延时约束带来压力,减小了实时流量传输的解空间.空间上的隔离能够避免该问题,但同时会增加实时流量占用的队列.然而,TSN 中还存在 AVB 和 BE 流量,这些非实时流量通过使用更多队列能够显著提升时间性能和灵活性.因此在混合流量的 TSN 中,一种优化目标为在满足实时流量延时约束的前提下最小化其占用的队列数,将更多队列留给非实时流量,适用于同时关注非实时流量性能的场景.令 $\kappa^{[v_a, v_b]}$ 表示出端口实时流量所需队列数,有:

$$\min \sum_{[v_a, v_b] \in E} \kappa^{[v_a, v_b]},$$

即最小化全网所有交换机中实时流量占用的总队列数,同时引入约束:对 $\forall [v_a, v_b] \in E, s_i \in S$, 有:

$$\begin{aligned}
 \kappa^{[v_a, v_b]} &\leq n_{ab}, \\
 p_i^{[v_a, v_b]} &\leq \kappa^{[v_a, v_b]}.
 \end{aligned}$$

3) 最小化总传输时长.在实时流量未传输的所有时隙,由 AVB 和 BE 流量按严格优先级传输以填

充带宽.每个出端口的 GCL 中,非实时队列的开放时间可由所有实时队列开放时间的并集取反得到,在此期间非实时队列中缓存的流量按严格优先级传输.在非抢占场景中,每次实时队列开放前均需设置保护带,是 TSN 带宽浪费的主要来源.并且若实时队列频繁开放和关闭,会导致非实时队列的传输时隙碎片化,甚至由于时隙过小无法传输完整的帧,同时也会使 GCL 过长,对交换机内存造成压力.因此,应当在满足实时传输约束的条件下尽可能减小门开放的次数,将实时流量合并至尽可能少的时间窗口内传输.为实现该目的,一种常用的优化目标为最小化实时流量的总传输时长.由于实时流量为固定模式的周期流量,最小化总传输时长即等价于最小化保护带宽、最小化门开放次数以及最小化 GCL 长度,同时提高带宽利用率、非实时队列传输时隙,减小内存压力.总传输时长为所有实时流传输完成时刻的最大值,有:

$$\min \max_{s_i \in S} (\phi_{iN_i}^{[v_i(n_i-1), v_{in_i}]} + l_{iN_i}^{[v_i(n_i-1), v_{in_i}]}).$$

4) 最小化帧缓存时间.TSN 交换机中不仅存储 GCL,还存储已到达但暂未传输的帧.然而由于片上存储价格昂贵,交换机内部的内存容量通常较小,存储过多数据帧时会导致溢出和丢帧,破坏传输约束.因此在设计调度方案时需要考虑内存容量,尽可能减小帧缓存对内存的压力.因此一种优化目标为最小化帧在交换机中缓存的时间.定义 c_{ij} 为流 s_i 的第 j 帧传输完成时刻, $c_{ij} = \phi_{ij}^{[v_i(n_i-1), v_{in_i}]} + l_{iN_i}^{[v_i(n_i-1), v_{in_i}]}$, 则优化目标可表示为

$$\min \sum_{s_i \in S} \sum_{j=1}^{N_i} (c_{ij} - \sum_{k=1}^{n_i-1} (l_{ij}^{[v_{ik}, v_{i(k+1)}]} + d_{(ik)(i(k+1))})),$$

每帧的完成时间 c_{ij} 减去该帧在通过的所有链路上的传输延时和传播延时即为该帧在所有交换机中的缓存时间,该目标最小化网络中所有帧的总缓存时间.

以上优化目标分别反映了 TSN 应用或网络本身所要求的性能指标,既可单目标优化也可结合多目标共同优化.多目标优化可以采用加权、Pareto 前沿、字典序优先级等方法处理不同目标的优先级.

3 TSN 流量调度机制

在定义流量调度问题后,需要对该问题进行求解以得到流量传输方案和 GCL.GCL 的合成已被证明是 NP 完全问题,本节详细介绍 TSN 流量调度机制,所采用的求解算法可以分为 5 类:整数线性规划

(integer linear programming, ILP)、启发式算法 (heuristic)、可满足性模理论 (satisfiability modulo theories, SMT)/优化模理论 (optimization modulo theories, OMT)、禁忌搜索 (tabu search, TS) 和贪心随机自适应搜索 (greedy randomized adaptive search procedure, GRASP)。ILP 为数学优化方法, 通过遍历约束条件定义的整个解空间可获得理论最优解, 但求解时间与变量数成指数关系, 存在扩展性问题, 适用于小规模网络寻找最优解; 启发式算法根据一定规则每次将 1 条流加入 GCL 中, 直到所有流都被调度或某条流不可调度。由于只需探索一种流顺序空间, 启发式算法具有多项式求解时间, 但无法获得最优解也不能保证解的质量, 主要适用于 TT 流量负载较低的大规模网络。SMT/OMT, TS, GRASP 属于元启发式算法, 求解质量和速度介于 ILP 和启发式算法之间, 从初始解开始通过迭代搜索解空间提升解的质量, 无需探索整个解空间, 能够在可接受的求解时间内获得高质量的解, 适用于大规模网络寻找近似最优解。3 种元启发式算法的区别在于初始解生成和搜索的规则不同, OMT 通过 SMT 检验和区域搜索探索整个解空间, 迭代次数多但可获得最优解; TS 和 GRASP 达到终止条件即停止迭代, 保证较低的求解时间但只能获得探索过的最优解。也有调度机制采用以上多种方法的复合方法对调度问题进行求解。表 2 列出了 5 种求解算法的优势、缺点和适用场景, 具体算法将在每一类机制中详细阐述。

Table 2 Scheduling Algorithm Comparison

表 2 调度算法对比

| 算法 | 求解质量 | 求解速度 | 存在缺点 | 适用场景 |
|-----------|------|------|-------|----------|
| ILP | 高 | 慢 | 可扩展性低 | 小规模网络 |
| Heuristic | 低 | 快 | 解空间缩小 | 低负载大规模网络 |
| SMT/OMT | 高 | 较慢 | 迭代次数多 | 大规模网络 |
| TS | 中 | 较快 | 探索空间小 | 大规模网络 |
| GRASP | 中 | 较快 | 探索空间小 | 大规模网络 |

3.1 基于 ILP 的流量调度机制

ILP 是由整数变量、线性目标函数和线性不等式约束组成的数学规划。线性不等式构成定义完整解空间的多面体, 可行解为其中的整数点。在最小化问题中, 最优解为取得最小目标函数值的可行解, 在最大化问题中则为取得最大目标函数值的可行解。ILP 的求解方法为遍历所有可行解以寻找最优解, 尽管可采用分支定界等技术限制解空间的探索次

数, 但最坏情况下依然需要枚举所有可行解。在 TSN 流量调度问题中, 求解时间与变量数成指数关系, 因此 ILP 存在可扩展性问题, 主要适用于小规模问题寻找最优解。

Pop 等人^[16]提出了基于 ILP 的 TSN 流量调度机制, 该文献的调度目标包括最小化 TT 流占用的队列数 $\sum_{[v_a, v_b] \in E} (\kappa_{ab} - 1)$ 及端到端延时与其下界的差距 $\sum_{s_i \in S} (\lambda_i - \underline{\lambda}_i)$, 将目标函数表示为二者的加权和, 将流量调度问题形式化为多目标组合优化问题, 采用 ILP 求解器求解调度方案。Schweissguth 等人^[17]提出了 TSN 中基于 ILP 的流量路由与调度联合机制, 以最小化端到端延时为目标, 以 TT 帧到链路的映射和传输开始时间为调度变量, 形成路由与调度联合决策的 ILP 问题。通过删除无关的流与链路的映射关系以及无争用流之间的资源约束缩小问题规模, 同样采用 ILP 求解器求解路由和调度联合方案。文献^[18]关注任务与流量的整体调度, 扩展应用模型, 将应用抽象为计算与通信的任务链, 增加任务链时序约束, 并扩展优化目标为应用响应时间与端到端延时的综合目标, 将任务与流量整体调度问题形式化为多目标混合整数线性规划 (mixed integer programming, MIP) 问题并采用 MIP 求解器直接求解调度方案。

3.2 基于启发式算法的流量调度机制

启发式流量调度的核心思想为贪心算法: 从空的 GCL 开始, 每次调度 1 条流, 以最优化当前性能指标为原则确定该流的调度方案并写入相关的 GCL 中; 对所有流依次调度, 已调度流的方案作为未调度流的先验条件; 直到所有流均被调度或某条流不可调度时, 算法停止并返回最新的 GCL。启发式算法每次调度只需探索很小的解空间, 并且调度次数为有限值, 求解时间较小; 但由于调度每条流时并未考虑后续流, 可能做出不恰当的决策导致全局次优解, 解空间的缩小也可能令本可调度的流变为不可调度。因此, 启发式算法主要适用于 TT 流量负载较低的大规模问题。

文献^[16]中 TT 流按截止时间排序, 截止时间相同的流按周期从小到大排序, 依次进行调度。以最小化 TT 流占用的队列数为第 1 目标, 在调度每条流时首先将其分配至所有出端口的第 1 队列; 在该队列分配下以最小化延时为第 2 目标, 采用尽快传输策略从发送端链路到接收端链路依次确定最早的满足链路顺序约束且与已调度流无冲突的传输时间;

若端到端延时不大于该流截止时间,则调度成功并写入可行解,否则将该流重新分配至下一队列并重复尽早传输策略(as soon as possible, ASAP),直到该流成功调度或遍历所有队列均不可调度.所有流可调度时返回整体调度方案,否则仅返回部分可行的调度方案,整个过程具有多项式运行时间.文献[19]关注 TSN 的保护带问题,在已知 GCL 的基础上利用非实时帧填充其中的保护带以提高带宽利用率.针对每段保护带,以最大化填充帧的总优先级或总大小为目标,在保护带容量约束下,构造带有顺序约束的背包问题.首先提出基于动态规划的理想求解算法,为减小求解时间复杂度进而提出基于贪心算法的快速求解算法,依次从所有非实时队列头部选择优先级或尺寸最大的帧进行填充,实现了保护带的充分利用;最后分别提出基于比较二叉树结构和有序列表结构的硬件调度器实现该快速算法.文献[20]改进了 TSN 交换机的队列结构,通过增加出端口队列将非实时流量队列扩展为由多条队列组成的队列集,并设定阈值将同一优先级的非实时帧根据大小分配至不同队列中,修改 GCL 将扩展的队列与原队列采用优先级调度器共同调度.由于对非实时队列进行了细粒度划分,帧尺寸较小的队列其保护带也能够进一步减小,对带宽利用率的提高有很大帮助.文献[21-22]关注计算任务与流量的整体调度,在文献[18]的基础上进一步开放了计算任务的节点放置以及路由选择.文献[21]以计算任务节点和 TT 流的路径组合作为基因组,以所有 TT 流的总传输时长作为适应度函数,以单点交叉算子生成种群,采用遗传算法求解计算节点、路由和 GCL 的联合调度方案.文献[22]将计算任务按通信量从小到大排序,依次为每个任务规划其计算节点、通信流的路径和每一跳的传输时间.以最小化 TT 流总传输时长为目标,采用启发式列表调度算法依次遍历任务可行的计算节点,进而在每个节点规划下遍历通信流的可选路径,在每条路径上计算与已规划流不冲突的最早可传输时间,选择完成时间最小的节点、路径和传输时间作为该任务的联合调度方案.

3.3 基于 SMT/OMT 的流量调度机制

SMT 是求解约束满足问题的一种元启发式理论方法,用于判断由逻辑或数学语言(如线性整数代数、位向量代数)组成的一阶公式能否成立,若可以成立则给出一组使得公式成立的变量取值(称为真值指派).在优化问题求解中 SMT 通常用于检验约束条件能否满足,真值指派即对应可行解.OMT 将

SMT 作为基础组成部分,能够在给出满足约束的变量取值基础上找到最小化目标函数的最优解.

SMT 要求被检验的公式写作所有子句的合取范式,如 $\neg p \vee (p \wedge q)$;每个子句则要求为文字的析取范式,每个文字既可能为布尔变量或取反,也可能为数学公式的简单命题或取反.将每个简单数学命题用布尔变量表示,则整个取值空间只包括所有 n 个布尔变量的 2^n 种赋值.将取值空间组织为二叉树,每个节点表示 1 个布尔变量,每个分支表示 1 种取值,共 2^n 个叶子节点.SMT 通过断定、传播和回溯系统性搜索该空间,在找到布尔变量公式的真值指派后还需检验对应的命题是否真正成立.OMT 通过多次 SMT 迭代缩小目标函数的取值范围.每次迭代将新的目标取值范围作为约束条件与原始约束合并,并采用 SMT 检验新的约束是否成立:若成立,则代入真值指派后采用区域搜索确定约束区域内的最优目标值作为下一轮迭代的目标取值范围并重复 SMT 检验;若不成立,则在另一半取值空间上重复 SMT 检验,直到目标函数取得最小值或判断原始约束不成立.

基于 SMT/OMT 的 TSN 流量调度机制将调度问题的每个约束条件看作一个子句,改写为不等式的析取范式,将所有约束条件用合取符号连接作为 SMT 要求的合取范式.Steiner 等人^[8,23-26]在该方向进行了持续研究,提出了一系列基于 SMT/OMT 的流量调度机制.文献[8,23]将调度变量均定义为整数,采用 SMT 搜索满足约束条件的调度方案.为提高算法的可扩展性,在 SMT 中引入增量回溯算法:每次将一条流的变量和约束加入 SMT 公式,检验最新的约束集是否成立,一旦找到可行解即将其固定为常数,并继续加入新的流;若加入新流后无可行解则进行回溯,在更大的取值空间内搜索可行解;重复以上过程直到所有流均被调度或判断原问题无可行解.该方法减小了 SMT 的平均检验次数,在实时流量负载较低的场景中显著提高了算法性能.文献[24]同样采用 SMT 搜索可行调度方案,同时提出了分解的 SMT 进一步提高对约束条件数量的可扩展性.将实时帧按固定数量分为不同子集,将子集的变量和约束构成独立的 SMT 公式,依次采用 SMT 搜索可行解,不同子集的帧传输时间不重叠;若某个子集无可行解,则可直接判断原问题不可调度.该方法相比增量回溯具有更快的求解速度,但禁止不同子集中帧和时隙的穿插利用,也相应地降低了调度方案的品质.文献[25]进而在静态的实时流量调度

中引入周期性的固定时间片传输非实时流量.一方面在实时流量调度问题中增加约束条件定义恰当的空闲间隔,规定实时帧均不在该间隔内传输,并采用SMT搜索引入空闲间隔后的调度方案;另一方面在GCL合成后对其进行后处理,在不破坏原始约束的条件下增大GCL的循环周期,引入更多空闲间隔,进一步帮助非实时流量降低延时.文献[26]采用的是以时间窗口为调度对象的形式化方法,调度变量为帧到窗口的分配以及窗口开闭时刻;将每条链路上时间窗口的开/闭时刻按序写入数组,将约束条件用一阶数组理论的语法和运算符表示,采用OMT搜索变量取值空间,给出最小化所有实时流延时抖动加权和的最优解.文献[27]革新了现有调度算法以链路超周期作为GCL时长的现状,提出以基本周期循环GCL,即链路所有TT流周期的最大公约数.每个基本周期内GCL的开闭完全一致,保证每条流在其自身周期内到达后均能通过端口.依次在网络中每个出端口局部应用SMT计算满足调度约束的时间窗口分配,遍历所有出端口后形成全局调度方案,相比全局SMT求解大幅减小了计算开销.文献[28]同时考虑TT流和BE流,提出了松弛时间的概念,即在每个TT帧传输之后引入一段空白时隙用来传输BE帧,针对松弛时间引入新的调度约束,规定同一链路上不同流的帧传输时间和松弛时间均不重叠、链路总松弛时间具有给定的上下界等,并以最大化松弛时间为调度TT流,采用OMT求解调度方案,在保证TT流延时约束的条件下降低BE流的延时.文献[29-30]提出任务与流量的整体调度机制,将应用建模为端系统上的计算任务与网络流量传输的序列,将计算任务的截止时间约束、顺序约束、无冲突约束等与流量传输约束合并,构成统一的混合整数线性规划问题,并采用SMT算法求解最小化应用响应时间的整体调度方案.

3.4 基于TS的流量调度机制

TS也为元启发式算法,其核心思想是在解空间中引导式搜索最小化代价函数的可行调度方案.从初始解开始,通过迁移不断生成临近解来探索解空间;若当前解的代价函数小于历史最优解,则将其存储为当前最优解;为防止陷入局部最优解,TS采用禁忌表存储产生历史最优解的迁移,避免重复访问;当在某个局部区域内多步迭代均未探索到更优解时,转移至未探索的区域;当达到终止条件(如搜索时间达到阈值,迭代次数达到阈值)时,停止搜索并返回当前最优解.

文献[31-32]提出基于TS的流量调度机制,定义代价函数为 $w_{TT}\delta_{TT} + w_{RC}\delta_{RC}$, δ_{TT} 和 δ_{RC} 分别为TT帧与RC帧的可调度程度,定义为帧传输延时与截止时间的差值, w_{TT} 和 w_{RC} 则为对应的权重;并定义了前驱帧/当前帧传输提前、当前帧/后继帧传输延后4种迁移;以尽快传输策略生成初始调度方案,对当前方案中不可调度的帧进行提前传输,对可调度程度最高的帧进行延后传输,生成临近方案候选列表,并选择其中代价函数最小的临近方案;若临近方案代价函数小于当前方案,则更新当前方案并将产生该临近方案的迁移写入禁忌表;在搜索过程中持续更新最优方案,当搜索时间达到阈值时返回最新的最优方案.此外,文献[31]将TT流和RC流的路径选择也定义为传输方案的一部分,将重路由也定义为迁移,采用TS算法搜索路由与调度联合方案.文献[33]提出基于TS的流量类型分配与调度联合机制,探索流量类型分配空间.TSN中硬实时流量必须在截止期限内完成传输,一旦超过则效用立刻降为零,而在期限内任何时间完成的效用没有区别;软实时流量在完成时间超过截止期限时效用不会立刻降为零而是随完成时间增大逐渐降低;为混合流量重新分配TT或AVB的类型可以保证硬实时流量在期限内完成的同时为软实时流量提供更多传输的机会,最大化软实时流量的效用.定义代价函数为硬实时流量可调度程度与软实时流量效用的加权和,并定义了流类型切换、修改GCL、流等级切换3种迁移;初始方案为硬实时流量分配TT类型而软实时流量分配AVB类型,采用尽快传输策略生成该类型分配下的调度方案.每次搜索首先生成固定数量的临近方案,并选择其中代价函数最大的;若临近方案代价函数大于当前方案,则更新当前方案并将产生该临近方案的迁移写入禁忌表;在搜索过程中持续更新最优方案,当搜索时间达到阈值时返回最新的最优方案.

3.5 基于GRASP的流量调度机制

GRASP类似于TS算法,也是在解空间中引导式搜索最优解的元启发式算法,与TS的区别在于初始解生成和邻域搜索算法不同.GRASP适用于求解组合优化问题,初始解通过贪心算法生成,每次迭代包含构造和搜索2个阶段.构造阶段每次通过贪心随机算法生成一个新的初始可行解;搜索阶段采用爬山算法对该初始解进行邻域搜索并持续优化,得到局部最优解,若新的局部最优解比当前最优解代价函数更小,则更新当前最优解;多次迭代,构造

与搜索交替执行,直到满足终止条件时返回当前最优解。

Pop 等人也设计了基于 GRASP 的调度机制^[16,34]。文献[16]以提升 TT 流和 AVB 流的可调度程度为目标,以 TT 流可调度的方案为可行解,以 AVB 流的可调度程度为代价函数来判断可行解的质量。文献[34]将目标函数写为三元组($|S| - |x|$, K_x , Λ_x),以 TT 流的可调度程度、占用队列数和端到端延时为评价指标,按字典序进行比较。文献[16, 34]针对尽快传输策略和最迟传输策略分别设计了 5 种变体,共 12 种启发式策略作为候选算法。在构造阶段采用贪心随机算法,为每条流预测 12 种算法对目标函数的增量,选择增量最小的 γ 种组成候选列表,再从中随机选择一个算法调度该流,形成初始解。在搜索阶段,每次从当前解中移除最多 π 条流,采用贪心随机算法重新调度移除的流形成临近解;一旦临近解的代价函数小于当前解即结束本轮搜索,以临近解为当前解开始下一轮搜索,达到搜索的终止条件后进入下一个构造阶段。在达到 GRASP 的终止条件后,返回历史最优调度方案。

3.6 基于复合方法的流量调度机制

部分调度机制采用以上多种方法的复合方法求解调度问题中不同的子问题,或在一种方法迭代的每一步采用其他方法计算当前调度方案和指标。

文献[35]以最小化 TT 流占用队列数为调度目标来为 AVB 和 BE 流留出尽可能多的队列,将 TT 流调度问题形式化为 ILP 问题,并采用 ILP 求解器求解调度方案,并在此基础上以 AVB 流可调度、最小化 AVB 流端到端延时及最大化网络带宽利用率为目,构建代价函数为 3 个指标的加权和,采用 GRASP 为 AVB 流选择路径。文献[36]关注 AVB 流的可调度性,采用网络演算理论^[37]对 TT 流对 AVB 流的阻塞反映到 AVB 流的最坏情况端到端延时上界中,并以所有 AVB 流的端到端延时上界与截止时间的差值作为目标函数,提出了联合路由与调度策略 JRS 决策 TT 流的路径和 GCL。JRS 迭代执行 K-最短路径算法选择路径以及 GRASP 算法搜索该路由方案下的最优 GCL,计算目标函数值并在迭代过程中不断更新最优路由与 GCL 联合方案。文献[38-40]均关注无等待调度问题,要求帧一旦从发送端发出,在中间节点上无等待,因此调度变量只包括帧在发送端的传输开始时间,而在后续端口的传输时间可通过累加链路传输延时、传播延时和交换机处理延时直接得到。文献[38]以最小化所

有 TT 流的总传输时长为目标,将问题分解为流排序和时间表制定,采用 TS 与启发式算法联合求解。TS 搜索 TT 流的顺序空间:根据每条 TT 流在通过所有节点上的总处理时间和最大处理时间分别升序和降序排列,结合随机排序算法共生成 5 个初始序列;分别从每个初始序列开始,将最后完成的流作为关键流,将其插入前驱流之前或与前驱流交换位置生成临近序列,并将关键流写入禁忌表;每次生成新序列后采用贪心算法按序为每条流分配与前流无冲突的最早传输开始时间,形成该序列下的调度方案,计算整体传输时长作为该序列的评价指标,选择传输时长最小且关键流不在禁忌表中的临近序列进行迁移;当多次迭代未能更新最优序列时停止搜索,返回当前最优序列的调度方案。文献[39]将 TT 流的路由和调度问题归约为冲突图,将每条流的每种配置,即路径和传输开始时间作为节点,节点间的冲突边表示同时采用 2 个节点的配置会破坏约束,即存在冲突。冲突图中包含所有流的无关点集即为有效的路由和调度方案,采用 ILP 与启发式算法联合求解。初始冲突图包括每条流的初始配置,通过迭代遍历每条流的路由和开始时间,扩充冲突图,并依次调用启发式算法和 ILP 在扩充图中寻找最大无关点集,直到该点集包含所有流,返回对应的路由和开始时间方案。文献[40]将 TT 流用图表示,以流为节点,流之间的相关程度为边的权重,提出启发式图划分算法将所有流划分为相关度最少的多个分组;采用 GRASP 算法探索多径路由空间,初始路由随机生成,在当前路由方案下对流进行划分,将对相关度贡献最大的流替换路由方案进行邻域搜索,多次迭代得到冲突最小的路由和分组方案。每组 TT 流独立采用 ILP 求解各自的约束,前驱组的可行解作为后续组的先验约束,直到所有分组的可行解共同构成调度方案。

3.7 小 结

表 3 总结了 TSN 流量调度机制的调度目标、调度对象、求解方法等特点。从 3.1~3.6 节分析中可以看出,现有的 TSN 流量调度机制能够从机理上保证流量在网络中按序、无冲突传输,且满足实时流量确定性延时与有界延时抖动的性能要求,在此基础上针对端到端延时、带宽利用率、内存占用时间等性能指标分别优化且均实现了不同程度的性能提升。然而也具有 3 个共性问题:

1) 现有流量调度机制大多基于时间感知整形器 TAS 设计,通过端口的 GCL 控制各队列出口的

开闭,为不同队列分配不重叠的传输时隙对不同优先级的流量进行隔离.在 TAS 整形框架中,调度机制需要严格规定每个 TT 帧到达和离开每个端口的精确时间且必须满足严苛的约束条件,导致变量与约束数量众多且解空间受限,为调度方案质量与求解时间的有效权衡造成了困难.

2) 现有流量调度机制均采用离线方法计算静态的流量调度方案,导致缺少对动态环境下帧到达和离开端口的时间偏差的补偿能力,难以抵抗在线运行时网络与流量固有且难以避免的不确定性,如帧丢失、时间同步误差、紧急事件触发的突发流量等.一旦某一帧未能按调度方案给定的时刻到达指定端口,则可能导致该帧及后续帧无法在分配的时

隙完成传输,扰乱既定的流发送时序,诱发连锁反应导致不可预期的传输结果,具有较弱的灵活性和鲁棒性.

3) 现有流量调度机制主要关注实时流量,而非实时流量大多只是简单填充实时流量的剩余带宽和时隙.然而非实时流量也承载着业务语义,同样具有业务定义的优先顺序,以及延时、带宽保证等差异化的性能要求;缺少对非实时流量的个性化关照会导致非实时流量难以准确充分地利用网络资源和实现相关业务的效用最大化.

针对上述问题,部分研究工作提出了新的调度思路,扩展了 TSN 流量调度机制的设计空间.第 4 节对该部分研究工作进行了阐述.

Table 3 Scheduling Mechanisms Summary

表 3 调度机制总结

| 调度机制 | 调度目标 | 调度流量类型 | 求解方法 | 路由 | 任务 |
|--------------|---------------------------|--------|-------------------------------|----|----|
| 文献[16]调度机制 | 最小化 TT 队列数与延时加权和 | TT | ILP | 否 | 否 |
| | 最小化 TT 队列数、延时 | TT | Heuristic(贪心算法) | 否 | 否 |
| | 最大化 AVB 可调度性 | TT | GRASP | 否 | 否 |
| 文献[17]调度机制 | 最小化延时 | TT | ILP | 是 | 否 |
| 文献[18]调度机制 | 最小化应用响应时间、延时 | TT | ILP | 否 | 是 |
| 文献[19]调度机制 | 最大化带宽利用率 | AVB,BE | Heuristic(贪心算法) | 否 | 否 |
| 文献[20]调度机制 | 最大化带宽利用率 | AVB,BE | Heuristic(严格优先级) | 否 | 否 |
| 文献[21]调度机制 | 最小化总传输时长 | TT | Heuristic(遗传算法) | 是 | 是 |
| 文献[22]调度机制 | 最小化总传输时长 | TT | Heuristic(列表算法) | 是 | 是 |
| 文献[8,23]调度机制 | 满足约束条件 | TT | SMT | 否 | 否 |
| 文献[24]调度机制 | 满足约束条件 | TT | SMT | 否 | 否 |
| 文献[25]调度机制 | 最大化空闲间隔 | TT | SMT | 否 | 否 |
| 文献[26]调度机制 | 最小化延时与抖动加权和 | TT | OMT | 否 | 否 |
| 文献[27]调度机制 | 减小 GCL 长度 | TT | SMT | 否 | 否 |
| 文献[28]调度机制 | 最大化松弛时间 | TT | OMT | 否 | 否 |
| 文献[29]调度机制 | 最小化应用响应时间之和 | TT | SMT | 否 | 是 |
| 文献[30]调度机制 | 满足约束条件 | TT | SMT | 否 | 是 |
| 文献[31]调度机制 | 最大化 TT 和 RC 的可调度性 | TT,RC | TS | 是 | 否 |
| 文献[32]调度机制 | 最大化 TT 和 RC 的可调度性 | TT,RC | TS | 否 | 否 |
| 文献[33]调度机制 | 最大化软实时流量效用函数 | TT,AVB | TS | 否 | 否 |
| 文献[34]调度机制 | 最大化 TT 可调度性、最小化 TT 队列数、延时 | TT | GRASP | 否 | 否 |
| 文献[35]调度机制 | 最小化 TT 队列数 | TT | ILP 与 GRASP 复合方法 | 是 | 否 |
| 文献[36]调度机制 | 最大化 AVB 可调度性 | TT | Heuristic(K-最短路径)与 GRASP 复合方法 | 是 | 否 |
| 文献[38]调度机制 | 最小化总传输时长 | TT | TS 与 Heuristic(贪心算法)复合方法 | 否 | 否 |
| 文献[39]调度机制 | 满足无等待约束条件 | TT | ILP 与 Heuristic(极大独立点集)复合方法 | 是 | 否 |
| 文献[40]调度机制 | 满足无等待约束条件 | TT | GRASP 与 ILP 复合方法 | 是 | 否 |

4 TSN 流量调度发展趋势

在现有 TSN 流量调度机制的基础上,流量调度的发展趋势主要包括动态分配传输时隙、混合流量整体调度、流量调度与路由联合规划、流量与任务联合调度、其他整形器框架下的调度、可靠性感知的流量调度、融合网络中的流量调度等。

1) 以时间窗口为调度单位.为了缓解离线计算帧传输时隙的静态调度方案求解复杂、解空间受限的问题,已有少量研究提出了以时间窗口为单位的调度问题形式化定义^[41].采用以时间窗口为调度对象的形式化方法,离线计算帧到时间窗口的分配及窗口的开闭时刻,而每一帧在时间窗口内的具体传输时隙则在传输过程中动态分配.网络模型、流量模型均与 2.1 节中所描述的相同,而调度变量为帧在出端口的时间窗口分配 $I_{ijk}^{[v_a, v_b]}$ 和时间窗口的开闭时刻 $l_k^{[v_a, v_b]}$ 和 $r_k^{[v_a, v_b]}$;规定每个帧在通过的每条链路上被分配至唯一窗口,且窗口大小需大于等于被分配至该窗口的所有帧的传输时间,并将帧约束、链路约束、流传输约束、延时约束、抖动约束和帧隔离约束均改写为由新变量表示的不等式;离线计算满足所有约束的时间窗口开闭时刻,而帧的传输时刻 $\phi_{ij}^{[v_a, v_b]}$ 则根据帧到达时刻、端口状态、其他帧的状态根据局部调度规则实时分配.时间窗口大小的计算留有裕量,可一定程度抵抗网络不确定因素.即使帧未能按时到达端口,其他已到达的帧可以先传输,且一定范围内的延误不影响帧在分配的窗口内完成传输,避免了连锁反应和延时坍塌,增强调度方案的灵活性和鲁棒性.

2) 混合流量的整体调度.为进一步优化非实时流量的传输性能,实现非实时流量的效用最大化,部分研究工作在 TT 流调度的基础上关注 AVB, RC 流所要求的延时性能和 BE 流所要求的带宽利用率.一方面在 TT 流的调度中考虑其他流量的性能指标,在保证自身可调度的条件下为其他流量留出最大队列数或空闲时隙;另一方面直接将非实时流量的性能指标写入优化目标,将 AVB, RC 流的传输方案与 TT 流共同决策.可见已有研究开始关注 RC 流和 AVB 流的性能,但相关调度算法主要关注延时指标,在传输效用的个性化关照方面还有进一步完善的空间.此外,当前对于 BE 流主要以提高网络整体的带宽利用率为主,同样缺少对 BE 流自身性能要求的细化分析和针对性优化.

3) 流量调度与路由联合规划.现有流量调度机制大多基于给定的路由方案,通常采用生成树协议或最短路径路由算法确定;在此基础上流量调度执行端口队列和帧传输时隙分配.由于路由机制只考虑路由相关指标,如路径长度、负载均衡等,并未考虑每条流的分类和性能要求,因此可能产生不利于甚至阻碍调度机制提高流传输性能的路由方案.若路由机制将多条延时紧迫的 TT 流分配至同一链路,则由于严重的时隙争用难以满足所有流的截止时间,更难以为其他流量留出优化的空间.因此,已有部分研究开始考虑将流量调度与路由联合规划,增大传输解空间,使更多流量可调度并且取得更优的性能.可以看出,路由与调度联合机制的主要思路是将路径选择设为变量,与调度方案针对同一目标共同决策,主要适用于离线计算的场景.在实际中网络可能发生实时动态变化,可以进一步探索在线路由与调度的联合机制.

4) 任务与流量整体调度.实时应用不仅要求网络传输的确定性低延时,应用本身也需要确定性的低响应时间.为将延时的确定性从网络扩展到应用层,就需要运行在端系统上的计算任务与网络中的流量整体调度,否则任务执行与流量传输不匹配会导致应用响应时间的增大和不确定性.因此,少量研究将端系统计算任务采用与流量传输采用统一的方法建模,将应用抽象为计算与传输的任务序列,将计算任务的执行时隙也作为调度变量,根据计算任务特性与流量传输的依赖关系定义任务调度约束,以整体应用性能为目标合成任务与流量的联合调度方案.实时系统应用的计算与通信模式多种多样且较为复杂,在现有工作的基础上可以进一步讨论不同应用下的计算任务与流量传输模型以获得更为精准的调度结果.

5) 基于其他整形器框架的调度.现有流量调度机制大多基于 TAS 调度 TT 流和基于 CBS 调度 AVB 流,而基于其他整形器的调度机制较少.然而 CQF 和 ATS 也是 TSN 中的通用整形器,分别适用于周期性流量以及周期与非周期性混合流量整形,因此有研究开始关注基于其他整形器的流量调度机制.少数研究提出了基于 ATS 的流量调度机制,包括流的队列分配和队列的优先级分配 2 部分决策.根据 ATS 的队列分配规则设计端口的队列分配约束,分析 ATS 作用在流的端到端延时上界设计延时约束,并分别采用 SMT 求解器和拓扑排序求解器

(topology rank solver, TRS)求解调度方案^[42].在此基础上,基于CQF和ATS的流量调度机制还可继续丰富和完善.

6) 可靠性感知的调度.为增强TSN流量传输的可靠性,采用逐流控制实现零拥塞丢弃是一种基本技术手段.分布式流预留协议(Stream Reservation Protocol, SRP)^[43]中的连接准入控制可为特定流按需预留网络资源,避免因不同流争用网络资源为确定性传输带来负面影响.此外,TSN工作组还开发了帧副本和消除(FRER)策略^[44],通过在多个不相交的路径发送帧副本,为无法容忍帧丢失的应用程序提供冗余高可靠传输服务,接收的第1个消息将被正常处理,其他副本将被丢弃.基于FRER策略,文献[45-46]提出可靠性感知的路由方案,考虑链路的失效概率,计算传输路径的可靠性指标,分别采用ILP和蚁群算法为每条流选择可靠性高、相互冲突小、延时小的路径,为流量调度提供更大的可行解空间.然而,FRER支持的消息副本多路径冗余传输虽可显著增强确定性可靠传输,但引入的过量带宽开销是一个不可回避的问题,需要进一步研究解决方案和替代策略来加以克服.此外,也可将调度与可靠性感知的路由结合,求解具有可靠性保证的联合方案.

7) 融合网络中的流量调度.由于TSN互联互通的优势,目前TSN与新型网络的融合已成为研究热点.2017年IEC与IEEE联合成立了IEC/IEEE60802工作组,负责制定TSN在工业自动化领域应用的规范.OPC UA基金会则成立了现场级通信(FLC)工作组,将TSN相关标准与OPC UA规范融合,提供适用于智能制造和工业互联网领域的工业通信架构.专注通信产品认证的产业联盟AVnu也开始重点关注TSN相关技术和产品专业音视频传输、汽车和工业自动化等领域的市场应用.融合网络中流量的类型与特征各异,需要根据性能要求将融合网络流量与TSN标准机制进行映射,按TSN流量类型调度.文献[47]将工业自动化和控制系统流量总结为8类:同步通信流量、循环流量、事件触发流量、网络控制流量、配置与诊断流量、尽力而为流量、视频流量、音频流量,并为每种流量分别映射了TSN标准中的严格优先级、转发和排队增强、时间同步、时间感知整形、帧抢占、流预留、逐流过滤和监管、帧复制和消除以及直通交换等机制.文献[48]遵循文献[47]中的流量分类,将每个网络周期划分为多个阶段,每个阶段单独分配给一类流量;将工厂自动化网

络分为机器层、生产单元层和主干层,并将流量分为层内流量和跨层流量,同一类型的层内和跨层流量分别占用其阶段的前半部分和后半部分传输,层内流量往往比跨层流量具有更低的延时要求,因此采用每流调度,而跨层流量采用整体调度以降低计算复杂度,实现了工厂自动化网络的混合流量在融合网络中的隔离和高性能传输.文献[33]将信息物理系统中的应用分为3类:硬实时应用、软实时应用和非时间关键应用,根据截止时间和效用函数基于TS算法为混合流量分配TSN中的TT或AVB类型,并采用尽快传输策略确定流量传输时间.

针对当前TSN流量调度存在的问题,可以考虑改进TSN流量调度问题的结构加以完善.TSN中的流量规划与调度需要做的决策主要包括每条流到队列的分配以及每个帧实例在每个端口的传输时隙分配.可以对流量调度任务进行合理分解,开展全局静态规划与局部动态调度联合调度方法的研究,基本思路是:将无需全局信息的传输时隙分配从全局规划中移除,由端口的局部调度完成.全局规划只决策每流的队列分配,探索对目标函数最有利的队列分配方案,大幅降低变量与约束条件的数量,并将端到端延时与抖动约束分解到每个端口上,为局部调度提供指导;局部调度根据端口和帧的实时状态动态分配传输时隙,利用以太网固有流量控制机制在相邻节点间传递资源与流量信息,利用出端口的优先级转发和帧抢占机制为不同队列的帧实时分配可用的时隙或带宽资源,并根据本地的延时与抖动约束进行适当延时补偿以保证传输确定性.

5 总 结

为实现确定性低延时的网络传输需求,IEEE 802 TSN工作组提出了一系列增强机制与策略,将以太网扩展为TSN.流量调度是TSN的核心机制,通过规定帧传输的顺序和时隙保证网络内无冲突传输,满足延时与带宽要求,并优化传输性能.本文系统地阐述了TSN流量调度的标准框架,形式化描述了TSN流量调度问题,包括典型的调度约束和调度目标.在此基础上对现有的TSN流量调度机制按采用的算法分类,分别进行了分析与总结,最后讨论了TSN流量调度的设计空间和发展趋势,为后续的研究奠定了基础.随着计算系统算力的提高,TSN流量调度机制的可扩展性将大幅提升,并且对调度问题的建模与求解更加完善.

作者贡献声明: 张彤负责提出综述思路、设计综述内容、论文撰写和最后版本修订; 冯佳琦负责收集和分析时间敏感网络流量控制标准相关的文献资料; 马延滢负责收集和分析时间敏感网络流量调度形式化问题相关的文献资料; 渠思源负责收集和分析时间敏感网络流量调度算法相关的文献资料; 任丰原负责对文章学术性和技术性内容进行审阅和关键性修订。

参 考 文 献

- [1] Wollschlaeger M, Sauter T, Jasperneite J. The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0 [J]. IEEE Industrial Electronics Magazine, 2017, 11(1): 17-27
- [2] Ge Xiaohu. Ultra-reliable low-latency communications in autonomous vehicular networks [J]. IEEE Transactions on Vehicular Technology, 2019, 68(5): 5005-5016
- [3] Finzi A, Mifdaoui A, Frances F, et al. Network calculus-based timing analysis of AFDX networks with strict priority and TSN/BLS shapers [C/OL] //Proc of the 13th IEEE Int Symp on Industrial Embedded Systems (SIES). Piscataway, NJ: IEEE 2018 [2021-07-08]. <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8442080>
- [4] Teener M D J. IEEE 802 Time-sensitive networking: Extending beyond AVB [OL]. [2021-03-10]. http://www.hit.bme.hu/~jakab/edu/litr/TimeSensNet/IEEE802_08_Teener_TSN.pdf
- [5] Alliance of Industrial Internet. Time sensitive network (TSN) industry white paper v1.0 [OL]. [2021-03-10]. http://www.aii-alliance.org/static/upload/202009/0901_165_010_961.pdf (in Chinese)
(工业互联网产业联盟. 时间敏感网络(TSN)产业白皮书 V1.0 版 [OL]. [2021-03-10]. http://www.aii-alliance.org/static/upload/202009/0901_165_010_961.pdf)
- [6] Cong Peizhuang, Tian Ye, Gong Xiangyang, et al. A survey of key protocol and application scenario of time-sensitive network [J]. Telecommunications Science, 2019, 35(10): 1-42 (in Chinese)
(丛培壮, 田野, 龚向阳, 等. 时间敏感网络的关键协议及应用场景综述 [J]. 电信科学, 2019, 35(10): 1-42)
- [7] Ashjaei M, Bello L L, Daneshbalab M, et al. Time-sensitive networking in automotive embedded systems-state of the art and research opportunities [J/OL]. Journal of Systems Architecture, 2021[2021-07-08]. <https://www.sciencedirect.com/science/article/pii/S1383762121001028>
- [8] Craciunas S S, Oliver R S. An overview of scheduling mechanisms for time-sensitive networks [OL]. [2021-07-08]. http://www.cs.uni-salzburg.at/~scraciunas/pdf/techreports/craciunas_ETR17.pdf
- [9] Time-Sensitive Networking Task Group. IEEE Std 802.3br-2016 (Amendment to IEEE Std 802.3-2015) IEEE Standard for Ethernet Amendment 5: Specification and Management Parameters for Interspersing Express Traffic [S]. Piscataway, NJ: IEEE, 2016
- [10] Time-Sensitive Networking Task Group. IEEE Std 802.1Qbu-2016 (Amendment to IEEE Std 802.1Q-2014) IEEE Standard for Local and Metropolitan Area Networks-Bridges and Bridged Networks-Amendment 26: Frame Preemption [S]. Piscataway, NJ: IEEE, 2016
- [11] Time-Sensitive Networking Task Group. IEEE Std 802.1Qbv-2015 (Amendment to IEEE Std 802.1Q) IEEE Standard for Local and Metropolitan Area Networks-Bridges and Bridged Networks-Amendment 25: Enhancements for Scheduled Traffic [S]. Piscataway, NJ: IEEE, 2016
- [12] Time-Sensitive Networking Task Group. IEEE Std 802.1Qav-2009 (Amendment to IEEE Std 802.1Q-2005) IEEE Standard for Local and Metropolitan Area Networks-Virtual Bridged Local Area Networks-Amendment 12: Forwarding and Queuing Enhancements for Time-Sensitive Streams [S]. Piscataway, NJ: IEEE, 2009
- [13] Time-Sensitive Networking Task Group. IEEE 802.1Qch-2017 (Amendment to IEEE Std 802.1Q-2014) IEEE Standard for Local and Metropolitan Area Networks-Bridges and Bridged Networks-Amendment 29: Cyclic Queuing and Forwarding [S]. Piscataway, NJ: IEEE, 2017
- [14] 802.1 WG-Higher Layer LAN Protocols Working Group. IEEE 802.1Qcr-2020 IEEE Standard for Local and Metropolitan Area Networks-Bridges and Bridged Networks-Amendment 34: Asynchronous Traffic Shaping [S]. Los Alamitos, CA: IEEE Computer Society, 2020
- [15] Specht J, Samii S. Urgency-based scheduler for time-sensitive switched Ethernet networks [C] //Proc of the 28th Euromicro Conf on Real-Time Systems (ECRTS). Los Alamitos, CA: IEEE Computer Society, 2016: 75-85
- [16] Raagaard M L, Pop P. Optimization algorithms for the scheduling of IEEE 802.1 time-sensitive networking (TSN) [OL]. [2021-07-08]. <http://www2.compute.dtu.dk/~paupo/publications/Raagaard2017aa-Optimization%20algorithms%20for%20th.pdf>
- [17] Schweissguth E, Danielis P, Timmermann D, et al. ILP-based joint routing and scheduling for time-triggered networks [C] //Proc of the 25th Int Conf on Real-Time Networks and Systems. New York: ACM, 2017: 8-17
- [18] Zhang Licong, Goswami D, Schneider R, et al. Task-and Network-level schedule co-synthesis of Ethernet-based time-triggered systems [C] //Proc of the 19th Asia and South Pacific Design Automation Conf (ASP-DAC). Piscataway, NJ: IEEE, 2014: 119-124

- [19] Zhang Chuwen, Wang Yi, Yao Ruyi, et al. Packet-size aware scheduling algorithms in guard band for time sensitive networking [J]. *CCF Transactions on Networking*, 2020, 3 (1): 4-20
- [20] Heilmann F, Fohler G. Size-based queuing: An approach to improve bandwidth utilization in TSN networks [J]. *ACM SIGBED Review*, 2019, 16(1): 9-14
- [21] Pahlevan M, Obermaisser R. Genetic algorithm for scheduling time-triggered traffic in time-sensitive networks [C] //Proc of the 23rd IEEE Int Conf on Emerging Technologies and Factory Automation (ETFA). Piscataway, NJ: IEEE, 2018: 337-344
- [22] Pahlevan M, Tabassam N, Obermaisser R. Heuristic list scheduler for time triggered traffic in time sensitive networks [J]. *ACM SIGBED Review*, 2019, 16(1): 15-20
- [23] Craciunas S S, Oliver R S, Chmelik M, et al. Scheduling real-time communication in IEEE 802.1 Qbv time sensitive networks [C] //Proc of the 24th Int Conf on Real-Time Networks and Systems. New York: ACM, 2016: 183-192
- [24] Pozo F, Steiner W, Rodriguez-Navas G, et al. A decomposition approach for SMT-based schedule synthesis for time-triggered networks [C/OL] //Proc of the 20th IEEE Conf on Emerging Technologies & Factory Automation (ETFA). Piscataway, NJ: IEEE, 2015 [2021-07-08]. <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7301436&tag=1>
- [25] Steiner W. Synthesis of static communication schedules for mixed-criticality systems [C] //Proc of the 14th IEEE Int Symp on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops. Los Alamitos, CA: IEEE Computer Society, 2011: 11-18
- [26] Oliver R S, Craciunas S S, Steiner W. IEEE 802.1 Qbv gate control list synthesis using array theory encoding [C] //Proc of the 24th IEEE Real-Time and Embedded Technology and Applications Symp (RTAS). Los Alamitos, CA: IEEE Computer Society, 2018: 13-24
- [27] Li Qing, Li Dong, Jin Xi, et al. A simple and efficient time-sensitive networking traffic scheduling method for industrial scenarios [J]. *Electronics*, 2020, 9(12): 2131-2149
- [28] Houtan B, Ashjaei M, Daneshtalab M, et al. Synthesising schedules to improve QoS of best-effort traffic in TSN networks [C/OL] //Proc of the 29th Int Conf on Real-Time Networks and Systems (RTNS). New York: ACM, 2021 [2021-07-08]. http://www.es.mdh.se/pdf_publications/6159.pdf
- [29] Craciunas S S, Oliver R S. Combined task-and network-level scheduling for distributed time-triggered systems [J]. *Real-Time Systems*, 2016, 52(2): 161-200
- [30] Craciunas S S, Oliver R S. SMT-based task-and network-level static schedule generation for time-triggered networked systems [C] //Proc of the 22nd Int Conf on Real-Time Networks and Systems (RTNS). New York: ACM, 2014: 45-54
- [31] Tamas-Selicean D, Pop P, Steiner W. Design optimization of TTEthernet-based distributed real-time systems [J]. *Real-Time Systems*, 2015, 51(1): 1-35
- [32] Tamas-Selicean D, Pop P, Steiner W. Synthesis of communication schedules for TTEthernet-based mixed-criticality systems [C] //Proc of the 8th IEEE/ACM/IFIP Int Conf on Hardware/software Codesign and System Synthesis. New York: ACM, 2012: 473-482
- [33] 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
- [34] Gavrilut V, Pop P. Scheduling in time sensitive networks (TSN) for mixed-criticality industrial applications [C/OL] //Proc of the 14th IEEE Int Workshop on Factory Communication Systems (WFCS). Piscataway, NJ: IEEE, 2018 [2021-07-08]. <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8402374>
- [35] Pop P, Raagaard M L, Craciunas S S, et al. Design optimisation of cyber-physical distributed systems using IEEE time-sensitive networks [J]. *IET Cyber-Physical Systems: Theory & Applications*, 2016, 1(1): 86-94
- [36] Gavrilut V, Zhao Luxi, Raagaard M L, et al. AVB-aware routing and scheduling of time-triggered traffic for TSN [J]. *IEEE Access*, 2018, 6: 75229-75243
- [37] Zhao Luxi, Pop P, Zheng Zhong, et al. Timing analysis of AVB traffic in TSN networks using network calculus [C] //Proc of the 24th IEEE Real-Time and Embedded Technology and Applications Symp (RTAS). Los Alamitos, CA: IEEE Computer Society, 2018: 25-36
- [38] Dürr F, Nayak N G. No-wait packet scheduling for IEEE time-sensitive networks (TSN) [C] //Proc of the 24th Int Conf on Real-Time Networks and Systems (RTNS). New York: ACM, 2016: 203-212
- [39] Falk J, Dürr F, Rothermel K. Time-triggered traffic planning for data networks with conflict graphs [C] //Proc of the 26th IEEE Real-Time and Embedded Technology and Applications Symp (RTAS). Piscataway, NJ: IEEE, 2020: 124-136
- [40] 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
- [41] Craciunas S S, Oliver R S, Steiner W. Formal scheduling constraints for time-sensitive networks [J]. arXiv preprint, arXiv:1712.02246, 2017

- [42] Specht J, Samii S. Synthesis of queue and priority assignment for asynchronous traffic shaping in switched Ethernet [C] // Proc of the 38th IEEE Real-Time Systems Symp (RTSS). Los Alamitos, CA: IEEE Computer Society, 2017: 178–187
- [43] Time-Sensitive Networking Task Group. IEEE Std 802.1Qat-2010 (Revision of IEEE Std 802.1Q-2005) IEEE Standard for Local and Metropolitan Area Networks-Virtual Bridged Local Area Networks Amendment 14: Stream Reservation Protocol (SRP)[S]. Piscataway, NJ: IEEE, 2010
- [44] Time-Sensitive Networking Task Group. IEEE Std 802.1CB-2017 IEEE Standard for Local and Metropolitan Area Networks-Frame Replication and Elimination for Reliability [S]. Piscataway, NJ: IEEE, 2017
- [45] Atallah A A, Hamad G B, Mohamed O A. Reliability-aware routing of AVB streams in TSN networks [C] // Proc of the 31st Int Conf on Industrial, Engineering and Other Applications of Applied Intelligent Systems. Berlin: Springer, 2018: 697–708
- [46] Huang Kai, Wan Xinming, Wang Ke. Reliability-aware multipath routing of time-triggered traffic in time-sensitive networks [J]. Electronics, 2021, 10(25): 125–142
- [47] Alliance of Industrial Internet. Time sensitive networks for flexible manufacturing testbed characterization and mapping of converged traffic types [OL]. [2021-05-02]. https://www.iiiconsortium.org/pdf/IIC_TSN_Testbed_Char_Mapping_of_Converged_Traffic_Types_Whitepaper_20180328.pdf (in Chinese)
(工业互联网产业联盟. 面向柔 性制造试验台的时间敏感网络融合流量类型表征与映射 [OL]. [2021-05-02]. http://www.ii-alliance.org/static/upload/202009/0901_165_010_961.pdf)
- [48] Hellmanns D, Glavackij A, Falk J, et al. Scaling TSN scheduling for factory automation networks [C] // Proc of the 16th IEEE Int Conf on Factory Communication Systems (WFCS). Piscataway, NJ: IEEE, 2020 [2021-07-08]. <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9114415>



Zhang Tong, born in 1992. PhD, associate professor. Member of CCF. Her main research interests include time-sensitive networking and data center network.

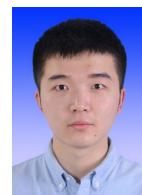
张彤,1992 年生.博士,副研究员,CCF 会员.主要研究方向为时间敏感网络和数据中心网络.



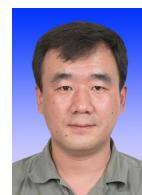
Feng Jiaqi, born in 1999. Master candidate. His main research interests include time-sensitive networking and industrial Internet.
冯佳琦,1999 年生.硕士研究生.主要研究方向为时间敏感网络和工业互联网.



Ma Yanying, born in 1997. Master candidate. Her main research interests include time-sensitive networking and industrial Internet.
马延萍,1997 年生.硕士研究生.主要研究方向为时间敏感网络和工业互联网.



Qu Siyuan, born in 1997. Master candidate. His main research interests include time-sensitive networking and industrial Internet.
渠思源,1997 年生.硕士研究生.主要研究方向为时间敏感网络和工业互联网.



Ren Fengyuan, born in 1970. PhD, professor, PhD supervisor. Senior member of CCF. His main research interests include network traffic management and control, control in computer networks, wireless networks and wireless sensor networks.

任丰原,1970 年生.博士,教授,博士生导师,CCF 高级会员.主要研究方向为网络流量管理与控制、计算机网络控制、无线网络、无线传感器网络.