

SSA:一种面向 CQF 模型的 TSN 资源调度算法

姜旭艳, 严锦立, 全巍, 孙志刚
(国防科技大学 计算机学院, 湖南 长沙 410073)

摘 要: 时间敏感网络(time sensitive networking, TSN)的目标是在工业控制和 5G 领域为时间敏感流提供低延迟、低抖动的确定性传输服务. 资源调度是保证 TSN 服务质量的关键技术之一. 为此, TSN 标准中提出循环队列转发模型(cyclic queuing and forwarding, CQF), 但是缺乏相应的资源调度算法对多流进行合理的资源分配以避免传输冲突. 由于时间敏感流量具有周期性、特征预知的特点, 主要从时间维度研究如何将 CQF 模型中的队列资源合理分配给多条数据流. 将该问题抽象为多约束条件下的资源规划最大化问题, 提出基于起始时隙分配的轻量级资源调度算法(start-slot assignment based scheduling algorithm, SSA). SSA 通过对端系统上发送时隙进行调节, 在满足约束条件的前提下最大化对队列资源的使用, 避免了端到端逐跳的时隙分配. 实验结果表明, 与不控制时隙的直接调度方式相比, 该算法能够将成功调度的流数目平均提高 41.84%.

关 键 词: 时间敏感网络; 确定性传输; 循环队列转发模型; 网络资源调度; 最大化问题

中图分类号: TP 393.1 **文献标志码:** A **文章编号:** 1005-3026(2020)06-0784-08

SSA: CQF-oriented Scheduling Algorithm in Time-Sensitive Networking

JIANG Xu-yan, YAN Jin-li, QUAN Wei, SUN Zhi-gang
(College of Computer, National University of Defense Technology, Changsha 410073, China. Corresponding author: JIANG Xu-yan, E-mail: jiangxuyan@nudt.edu.cn)

Abstract: Time-sensitive networking (TSN) is designed for providing determined transmission service with low latency and jitter in industrial control network and 5G. Resource scheduling, the core of TSN, is very crucial to guarantee the quality of service (QoS). Recently, TSN puts forward cyclic queuing and forwarding model (CQF). However, the corresponding scheduling algorithm is demanded for allocating resource rationally to avoid transmission collision. Since the TSN flow is periodic and predetermined, how to allocate resource for multiple TSN flows rationally from time dimension was researched. Resource allocation into multi-constrained maximization problem was abstracted. Then, a start-slot assignment based scheduling algorithm (SSA) was proposed. By adjusting the send slot at end system, SSA maximizes the queue resource utilization with meeting related constraints, which avoids complex time slot allocation hop by hop. Compared with the case without adjusting sending slot, experimental results demonstrate that SSA improves the number of flows scheduled successfully by 41.84% on average.

Key words: time-sensitive networking (TSN); deterministic transmission; cyclic queuing and forwarding model(CQF); network resource scheduling; maximization problem

工业控制网络和 5G 网络中包括时间敏感流、带宽约束型流量和 Best-Effort 流量, 其中优先级最高的时间敏感流通常具有周期性的特点, 对端到端延迟和抖动有着严格的要求, 延迟通常在毫秒级别^[1]. 例如在能源开采应用中, 网络时延在 250 μ s ~ 1 ms 之间, 抖动不能超过 1 μ s. 如果不能保证服务质量(quality of service, QoS), 整个生产过程就会失败, 甚至导致系统瘫痪. 因此, 保证

时间敏感流量的服务质量对系统的安全稳定运行至关重要^[2]。

标准以太网是竞争性网络,数据传输由于排队等因素具有不确定性,只能提供尽力而为的服务,无法保证时间敏感流的服务质量^[3]。IEEE 802.1 时间敏感网络任务组提出的实时以太网技术——时间敏感网络 (TSN),利用全局时间同步、确定性分组转发、帧复制及冗余消除等技术为时间敏感流提供低延迟、低抖动的确定性传输服务^[4],目前已经逐渐应用到工业控制领域和 5G 网络中。保证 TSN 性能的关键在于时间同步和资源调度,时间同步是保证服务质量的前提,资源调度是实现确定性传输的关键。目前时间同步已有详细的标准及算法,而资源调度在不同的调度模型和应用中存在很大差异。随着网络规模不断增长、应用需求愈发复杂,研究高效的 TSN 资源调度算法对 TSN 的大规模应用具有重要意义。

资源调度主要包括数据平面的调度模型和控制平面的调度算法,调度算法根据约束条件将调度模型的资源(如队列资源、时隙资源、带宽资源等)合理分配给网络中的时间敏感流和其他类型的流量,同时要最大化网络资源利用率。资源调度问题可以归纳为一个装箱问题,即从时间和空间两个维度上合理地给报文分配链路资源,因此是一个 NP 完全问题^[5]。

现有的资源调度模型主要包括 TTTech 公司提出的时间触发以太网 (time-triggered Ethernet, TTEthernet) 中的交换模型和 TSN 中的时间感知的整形器 (time aware shaper, TAS)。TTEthernet 根据静态的调度表 (schedule table) 对时间敏感流进行逐跳的转发控制。TAS 的原理类似于 TTEthernet 中的调度表,但 TTEthernet 调度的粒度是数据流,而 TAS 调度的粒度则是队列。控制平面的调度算法包括 SMT Solver、OMT Solver、ILP Solver 和禁忌搜索算法等^[6-8],这些算法虽然能够得到可行的调度方案,但是逐跳地分配链路资源的计算复杂度通常较高。为了降低复杂度,通过对 TAS 进行简单配置,提出了循环队列转发模型 CQF。CQF 根据奇偶时隙进行队列切换,每个时隙只有一个队列能够传输报文,另一个队列用于接收报文。目前缺乏基于 CQF 模型的资源调度算法。

1 相关研究

工业控制网络中的实时数据交换从基于总线

的交换向以太网交换演进,标准的以太网技术无法满足实时系统的要求,在标准以太网的基础上发展起来的时间敏感网络 TSN 和时间触发网络 TTEthernet 能够为系统中的时间敏感流提供低延迟、低抖动的传输服务。

在 TSN 和 TTEthernet 中,关于资源调度问题的研究主要从数据平面的调度模型和控制平面的调度算法两个方面展开。调度模型实现确定性转发机制,向控制平面提供可配置的资源。控制平面则根据流量及调度模型的特征构建约束条件,使用调度算法求解可行的调度方案,根据调度方案对数据平面进行配置。文献[5,8-10]采用 SMT 为 TTEthernet 中的时间敏感流寻找满足约束条件的可行解,文献[11]则采用 Tabu 搜索算法寻找最优的调度方案。所有流量特征是预知的,离线地计算满足网络约束(链路资源、交换机资源、端到端延迟等)和应用约束(发送依赖等)的调度方案,为每个数据帧规划沿路由路径的每个交换机上的发送时刻。

TSN 的 TAS 调度模型对应的调度算法是在满足约束条件的前提下配置控制每个队列传输开关的时间门控表。文献[6]发现 TAS 中存在输出端帧交错的问题,需要增加队列隔离、流隔离等方式解决,并采用 SMT 找到满足约束的可行解。除此之外还提出使用 OMT,通过设定最优化目标,求解满足约束条件的最优解。文献[1,7]则使用 ILP 建模求解调度问题。除此之外,为了解决大规模网络下的调度问题,文献[7]提出基于 Tabu 的启发式算法以降低计算开销。上述这些方法都是基于 TAS 模型提出的调度方案,而对于 CQF 模型,目前缺乏相应的资源调度算法。本文将 CQF 模型中的调度问题抽象为多约束条件下的资源规划最大化问题,提出了一种基于起始时隙分配的轻量级的资源调度算法 SSA。SSA 借鉴了贪心算法的思想,与 TAS 中使用启发式方法的调度算法相比,计算开销低。除了贪心算法之外,也可以使用其他方法,例如 Tabu 搜索算法、SMT 等工具求解本文提出的多约束条件下的资源最大化问题。

TSN 和 TTEthernet 中不同调度模型所对应的调度算法的特点及解决方案如表 1 所示。

2 研究动机

IEEE 802.1Qch 提出的循环队列转发模型是对 IEEE 802.1Qci per-stream filtering and policing (PSFP) 和 IEEE 802.1Qbv time aware shaper

表 1 不同调度模型对应的调度算法的特点
Table 1 Characteristics of scheduling algorithms of different scheduling models

项目		TTEthernet	TSN	
			TAS	CQF
调度粒度		单个帧	单个帧	队列中的所有帧
资源配置	位置	端系统及交换机	端系统及交换机	端系统
	粒度	具体时刻	具体时刻	时隙
复杂度		高	高	低
调度方案		SMT solver, Tabu-based solver	SMT solver, OMT solver, Tabu-based solver	SSA

(TAS)的一种简单配置,它通过流量整形为关键数据流提供确定且易于计算的延迟服务^[7]. PSFP 提供过滤、分类和流量监管功能;TAS 提出了门控机制,队列后面的门结构根据门控列表 (gate control list) 指定的时隙来改变状态.

CQF 由一个循环定时器 (cycle timer) 和两个传输队列构成. 在 CQF 模型中,全局时间被划分为等长的时隙,数据传输需要遵循以下两条规则: 1) 上游交换机传输报文的时隙与下游的邻居交换机接收报文的时隙相同; 2) 交换机在某个时隙内接收的报文一定会在下一个时隙传输出去. 基于上述两条规则,奇数时隙,队列 Q7 保存输入端口接收的帧 (接收模式,不发送帧),同时队列 Q6 发送在上一个偶数时隙缓存的数据帧 (发送模式,不接收帧),如图 1a 所示;偶数时隙,操作相反,如图 1b 所示.

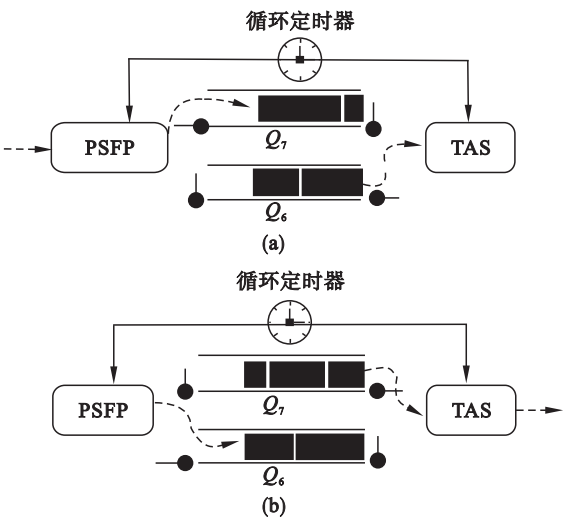


图 1 CQF 的实现方式
Fig. 1 Implementation of CQF
(a)—奇数时隙; (b)—偶数时隙.

基于上述传输方式,CQF 可以确定报文的延迟范围. 假定时隙长度为 d ,链路速率足够大,交换机 SW_i 要将报文 msg 传输给下游的邻居交换

机 SW_j . 在最坏情况下,若 SW_i 在时隙 $slot_k$ 的开始时刻将报文 msg 传输到 SW_j , SW_j 在时隙 $slot_{k+1}$ 的结束时刻将 msg 传输出去,那么 msg 在这两个交换机之间的延迟就是 $2d$;在最好情况下,若 SW_i 在时隙 $slot_k$ 的结束时刻将 msg 传输到 SW_j , SW_j 在时隙 $slot_{k+1}$ 的开始时刻 (即时隙 $slot_k$ 的结束时刻) 将 msg 传输出去,那么 msg 在这两个交换机之间的延迟就是 0 . 将上述情况一般化,假定 h 为跳数 (路由路径上的交换机的数目),那么一个帧的最大延迟 $maxDelay$ 和最小延迟 $minDelay$ 即

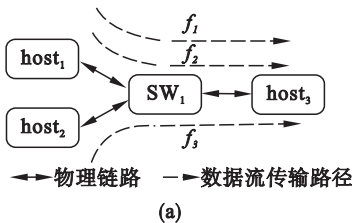
$$maxDelay = (h + 1) d, \tag{1}$$

$$minDelay = (h - 1) d. \tag{2}$$

换言之,一个帧的端到端延迟范围由时隙长度和跳数确定.

图 2a 所示 TSN 网络中包含三个终端 $host1$ 、 $host2$ 和 $host3$ 以及一台交换机 SW_1 . 假定时隙长度为 $150 \mu s$,交换机的输出端口中每个 CQF 队列的长度均为 3.2 kB . 网络中有 f_1, f_2, f_3 三条时间敏感流,流的信息如图 2b 所示,假定三条数据流均从 0 时刻开始传输.

若不采用任何调度策略,将三条流的传输用甘特图表示,如图 3a 所示. 根据 CQF 的传输规则 (1) 和传输规则 (2),上游交换机在时隙 $slot_i$ 内发送的帧会在相同的时隙内,即时隙 $slot_i$ 内被相邻的下游交换机接收,下游的邻居交换机在时隙 $slot_{i+1}$ 将该帧传输出去. 数据流 f_3 第 0 个周期的报文 $f_{3,0}$ 在 $host_2$ 的第 0 个时隙 $slot_0$ 送往 SW_1 , SW_1 本应该在时隙 $slot_0$ 将 $f_{3,0}$ 缓存到 CQF 队列,并在第 1 个时隙将 $f_{3,0}$ 发往 $host_3$. 但在 SW_1 的时隙 $slot_0$ 内,报文 $f_{1,0}$ 和 $f_{2,0}$ 已经占用了 $1000 \text{ Bytes} + 1200 \text{ Bytes} = 2200 \text{ Bytes}$ 的队列资源, SW_1 在时隙 $slot_0$ 上可用的队列资源即 $3200 - 2200 \text{ Bytes} = 1000 \text{ Bytes}$,小于 $f_{3,0}$ 的帧长 1500 Bytes , SW_1 只能丢弃 $f_{3,0}$,所以 $f_{3,0}$ 无法在时隙 1 传输到 $host_3$.



f	周期/ μs	最后期限/ μs	大小/Bytes
f_1	300	600	1 000
f_2	450	450	1 200
f_3	300	750	1 500

图 2 简单网络拓扑及流实例

Fig.2 Simple example of network topology and data flow

(a) —网络拓扑; (b) —流的信息.

虽然 SW_1 在 $slot_0$ 内的队列资源有限,但在 $slot_1$ 内队列资源充足,因此,可以将 $f_{3,0}$ 的发送时间往后偏移一个时隙,从而提高资源利用率. 如果 $f_{3,0}$ 能够在 $slot_0$ 被 SW_1 缓存,并在时隙 $slot_1$ 被 SW_1 送往 $host_3$,根据 CQF 的最大延迟计算公式 (1),其最大的端到端延迟为 $(1+1) \times 150 \mu\text{s} = 300 \mu\text{s}$. 那么将 $f_{3,0}$ 往后偏移一个时隙后,其最大的端到端延迟即 $300 \mu\text{s} + 150 \mu\text{s} = 450 \mu\text{s}$,小于其允许的最大端到端延迟 $750 \mu\text{s}$,因此将 $f_{3,0}$ 的发送时间往后偏移一个时隙是可行的. 将三条流的传输用甘特图表示,如图 3b 所示. 值得注意的是, $f_{3,0}$ 偏移发送时间后的最大端到端延迟变成了 $450 \mu\text{s}$,但 f_3 其余报文的最大端到端延迟为 $300 \mu\text{s}$,会在 $host_3$ 上造成抖动,所以将 f_3 所有报文的发送时间都往后偏移一个时隙.

综上所述,合理地在源端偏移时间敏感流的发送时隙,可以提高网络资源的利用率. 本文根据以上发现,设计基于起始时隙分配的资源调度算法,从而最大化能够成功调度的流的数量.

3 系统模型

本文将 TSN 网络的拓扑抽象为无向图,用 G 表示网络拓扑, $G = \{V, E\}$. V 表示网络节点的集合, $V = SW \cup H$, 其中 SW 表示交换机的集合, H 表示终端的集合. E 表示连接两个网络节点的边的集合. 交换机的每个输出端口包含 8 个队列,其中优先级最高的两个队列用作 CQF 队列,用于缓存时间敏感流,其余队列缓存低优先级的流量. 本文只讨论时间敏感流的调度,所以只对 CQF 队列资源的调度进行讨论. 图 4 所示的 Q_7 和 Q_6 即交

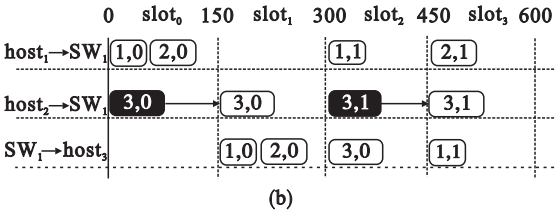
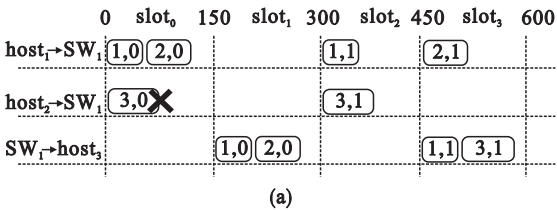


图 3 TSN 调度传输示意图

Fig.3 TSN scheduling flows

(a) —调度前; (b) —调度后.

换机 SW_0 的端口 SW_0^2 的两个 CQF 队列. SW_i^j 表示交换机 SW_i 的第 j 个端口,在端口 SW_i^j 的第 k 个时隙内,处于发送模式的 CQF 队列上可用的队列资源长度用 $SW_i^j[k]$. freeQueue 表示. 在端口 SW_0^2 的输出队列中,PSFP 和 TAS 根据用户配置的时隙控制两个 CQF 队列的切换.

终端是数据流的来源及目的地,它对每条流报文的发送时隙进行配置,调度器 Scheduler 根据配置信息和全局时钟调度缓存队列 Buffer 中的报文,如图 4 中的终端 H_0 所示. 终端也能够对数据流进行准入控制,数据流被标记为无法调度时,调度器不会调度属于该数据流的帧. 如果属于同一条流的数据帧的偏移量不同,那么它们的端到端延迟也不同,从而产生抖动,因此本文假定数据同一数据流的所有数据帧在源端的发送偏移量相同. 交换机和终端均使用 IEEE 1588 Precision Time Protocol (PTP) 进行全局的时间同步,时隙的长度是相同的,并且是固定的. 某个时隙内,终端 H_0 和交换机 SW_0 内的数据传输情况如图 4 所示.

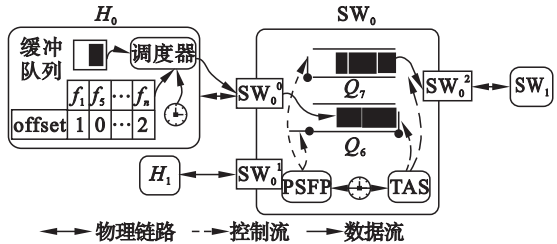


图 4 系统模型

Fig.4 System model

时间敏感流是周期性的^[6,12],属于同一条流的两个数据帧之间的最小发送间隔即为该流的周期,每个周期发送的数据量是固定的.假定流的信息是已知的,流在每个周期只发送一个报文,并且在 0 时刻就开始产生数据.用包含源端、目的端、发送周期、每个周期的数据量大小、传输路径和允许的最大端到端延迟的六元组来描述数据流:

$$f_i = \{f_i. \text{src}, f_i. \text{dest}, f_i. \text{period}, f_i. \text{size}, f_i. \text{path}, f_i. \text{deadline}\}. \quad (3)$$

$f_i \in F$, 用 $f_{i,j}$ 表示 f_i 的第 j 个数据帧. 由于属于一个数据流的所有数据帧在源端的偏移是相同的, 因此用 $f_i. \text{offset}$ 表示数据帧从源端发送时对应的时隙. 本文的调度粒度是交换机输出端口上的 CQF 队列, 因此在描述传输路径时需要携带传输端口. 用 f_i 传输经过的交换机端口来表示路径, 那么路径的表达式如下:

$$f_i. \text{path} = \{H_i, \text{SW}_k^m, \text{SW}_{k+1}^n, \dots, H_i\}. \quad (4)$$

经过端口 SW_k^m 的流 f_i 用 $f_i^{\text{SW}_k^m}$ 表示.

根据 CQF 的传输规则, 对时隙长度 lenOfSlot 进行分析. 由于以时隙为基本单位对数据进行偏移, 假定 lenOfSlot 能够整除所有数据流的周期. 将所有数据流的周期记作 $P, P = \{f_0. \text{period}, f_1. \text{period}, \dots, f_n. \text{period}\}$, 那么最大的时隙长度为

$$\text{MAX}(\text{lenOfSlot}) = \text{GCD}(P). \quad (5)$$

即 lenOfSlot 的最大值为数据流周期的最大公约数. CQF 规定, 报文在相邻两个节点的发送时隙与接收时隙相同. 那么, 时隙的长度至少需要保证队列中的最后一个报文在相邻节点之间的发送时隙与接收时隙相同, 因此最小的时隙长度为

$$\min(\text{lenOfSlot}) = \frac{\text{lenOfCQF}}{B} + d^{\text{proc}} + d^{\text{prop}} + \text{synPrec}. \quad (6)$$

其中: lenOfCQF 是 CQF 队列的长度; B 是链路发送速率; $\frac{\text{lenOfCQF}}{B}$ 即队列资源全部被占用时, 最后 1 比特发送到链路上所花费的时间; d^{proc} 和 d^{prop} 是处理延迟和传播延迟; synPrec 为同步精度.

4 问题描述

时间敏感流是周期性流量, 正常情况下会循环往复地运行, 直接计算出数据流在无限个时隙内的传输计划是不现实的. 考虑到数据流的传输也是周期性的, 因此在描述约束条件前, 需要确定调度周期 T . 只要在 T 内数据流的传输符合约束条件, 那么在整个过程中流的传输都符合约束条

件. 一般情况下, 数据流周期的值是任意的, 因此, 将调度周期 T 设置为所有流的周期的最小公倍数, 即 $T = \text{LCM}(P)$. 这样, 调度计划每隔 T 循环执行一次即可. 在上述假设下, 数据流的传输主要包括以下 4 个约束条件.

约束 1: 帧发送偏移

每个数据帧在源端的偏移量以时隙为单位. 偏移量应不超过每个周期内时隙的数量, 即

$$f_i. \text{offset} < f_i. \text{period} / \text{lenOfSlot}. \quad (7)$$

在源端偏移时隙, 可以提高资源利用率, 但是, 当偏移量过大时, 下一个数据帧已产生, 上一个数据帧却还未发送, 终端上需要缓存大量的数据帧, 耗费的存储开销大.

约束 2: 端到端延迟约束

TSN 必须要保证服务质量. 因此, 在发送端偏移的时隙加上 CQF 模型最大的延迟不能超过该数据流允许的最大端到端延迟, 即

$$f_i. \text{offset} \times \text{lenOfSlot} + (f_i. \text{hop} + 1) \times \text{lenOfSlot} \leq f_i. \text{deadline}. \quad (8)$$

其中, $f_i. \text{hop}$ 为数据流从源端至发送端的总跳数.

约束 3: 接收窗口约束

根据描述的 CQF 传输规则(1), 上游交换机发送报文的时隙应该和相邻下游交换机接收报文的时隙相同. 满足 CQF 传输规则的最小的时隙可以保证该约束.

约束 4: 队列资源约束

交换机输出端口的缓存队列的资源是有限的. 换言之, 某段链路在一个时隙能够传输的时间敏感型数据帧的数量是有限的, 在这个时隙内传输的数据的总长度不应该超过缓存队列的长度. 若流 f_i 的第 j 个报文在时隙 t 经过 SW_k 的端口 1, 则 $O(i, j, k, l, t) = 1$, 否则为 0. 队列资源约束的形式化描述如下:

$$\begin{aligned} & \forall f_i \in F, \forall f_{i,j} \in f_i, \forall j \in [0, \frac{T}{f_i. \text{period}} - 1], \\ & \forall \text{SW}_k \in \text{SW}, \forall \text{SW}_k^l \in \text{SW}_k, \forall t \in [0, \frac{T}{\text{lenOfSlot}} - 1], \\ & \sum_{i=0}^{n-1} O(i, j, k, l, t) \times f_i. \text{size} < \text{lenOfCQF}. \end{aligned} \quad (9)$$

t 与 i, j, k, l 之间存在如下关系:

$$t = \left(f_i. \text{offset} + \frac{j \times f_i. \text{period}}{\text{lenOfSlot}} + f_i^{\text{SW}_k}. \text{hop} \right) \bmod \frac{T}{\text{lenOfSlot}}.$$

其中: $\frac{T}{f_i. \text{period}}$ 为数据流 f_i 在调度周期 T 内需要传输的数据帧的个数; $\frac{T}{\text{lenOfSlot}}$ 为调度周期 T 内时隙的个数; $f_i^{\text{SW}_k}. \text{hop}$ 为数据流经过 SW_k^l 时的

跳数。

在满足上述约束条件的情况下,算法要能够使调度的流的数量最大化。因此,链路时隙分配问题实际上是一个优化问题。

5 算法描述

根据资源调度模型和约束条件,提出了基于起始时隙分配的资源调度算法,如算法 1 所示。

算法 1 SSA 调度算法

```
输入:  $F, G$ 
输出:  $F$ . OFFSET

BEGIN
1   $T = \text{compute\_LCM}(P)$ ;
2   $F = \text{flow\_sort}(F)$ ;
3  FOREACH  $f_i$  IN  $F$  DO:
4    FOREACH offset IN  $[f_i.\text{period} - 1, 0]$  DO:
5      latency = count_e2e_latency( $f_i$ , offset, slot_cycle);
6      IF latency <  $f_i$ . deadline THEN:
7         $f_i$ . flag = sw_queue_constraint( $f_i, G, T$ );
8        IF  $f_i$ . sched_flag == SUCCESS THEN:
9          update_global_resource( $f_i, G, T$ );
10         BREAK;
11        ELSE
12          CONTINUE;
13        END IF
14      ELSE
15         $f_i$ . sched_flag = FAIL;
16        CONTINUE;
17      END IF
18    END FOR
19  END FOR
END
```

首先,计算调度周期(第 1 行),调度周期 T 为所有时间敏感流的周期的最小公倍数,调度方案在这个周期内循环执行。接着,按照设定的参数(路径长度、报文大小、允许的最大端到端延迟、周期大小)进行排序(第 2 行),位置靠前的数据流优先分配资源。然后,按照既定的顺序为数据流分配资源(第 3~19 行)。4 个约束条件中,帧发送偏移约束对应第 4 行,端到端延迟约束对应第 5 行和第 6 行,队列资源约束则对应第 7 行。为了让 CQF 模型能够正常传输报文,用户设定的时隙一定大于允许的最小时隙长度,接收窗口约束得到满足。

算法借鉴了贪心算法的思想。对流进行排序时(第 2 行),优先映射占用资源较少的数据流,根据这一思想提出了 4 种贪心策略,即按照路径长度升序、报文大小升序、允许的最大端到端延迟升序、周期大小降序的方式调整流的顺序。以报文大小升序为例,排序后,每次映射时选择的总是当前尺寸最小的报文。在源端对时隙进行偏移时,采用偏移量递减的贪心策略,从最大的偏移量对数据流进行映射(第 4 行),为后映射的流保留位置相对靠前的时隙。

6 实验评估

6.1 实验场景

目前应用 TSN 的工业控制网络中,大多是环形拓扑,例如环形以太网列车网络(Ethernet train bone, ETB)。因此,本文模拟搭建了一个环形网络,如图 5 所示。

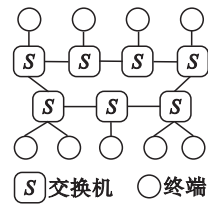


图 5 环形网络拓扑
Fig. 5 Experiment ring topology

环形网络中包括 7 个支持 CQF 转发模型的 TSN 交换机,每个交换机连接的终端数量从集合 $\{1, 2, 3\}$ 中随机选择,时隙长度设置为 $125\ \mu\text{s}$ 。网络中时间敏感流数量从集合 $\{100, 150, 200, 250\}$ 中选择。由于目前没有开源的支持 TSN 测试的数据集,文中的数据流为随机产生。每条数据流中每个周期产生的报文数量为 1。数据流的周期长度为时隙的整数倍,周期范围为 2~9 个时隙,报文大小在 64~1 500 Bytes 之间,允许的最大端到端延迟至少要大于 CQF 的最大端到端延迟。在上述场景中运行 SSA,为时间敏感流计算出调度计划,并根据调度成功率对算法的质量进行评估。

6.2 实验对比与分析

不在发送端控制帧的发送时隙,先映射的流先分配队列资源,队列资源不足时直接将流标记为不可调度,称这种调度方式为直接调度(direct scheduling, DS)。接下来从不同的维度展示 SSA 的调度效果,并对实验结果进行分析。

将交换机 CQF 队列的长度均设置为 15 kB,在上述实验场景中分别生成 100, 150, 200, 250 条数据流,按照不同的参数对流进行排序(周期降

序、数据帧大小升序、路径长度升序、允许的最大端到端延迟升序),使用 SSA 计算调度方案,统计调度成功率.同时,计算 DS 成功率,与 SSA 进行对比,如图 6 所示.不论使用何种排序方式,SSA 的调度成功率均高于 DS.计算得到,SSA 能够使 DS 的调度成功率平均提高 41.84%.

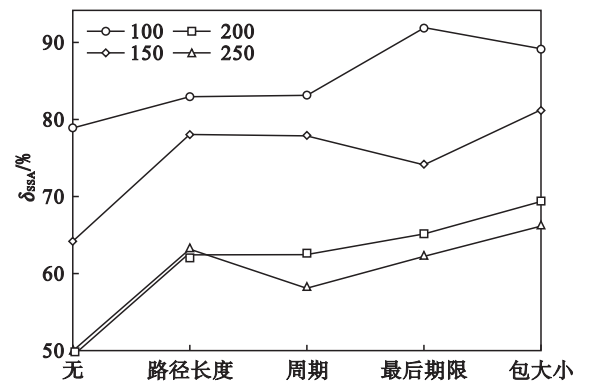


图 6 SSA 调度与 DS 调度的对比
Fig.6 SSA versus DS scheduling

在偏移源端的发送时隙时,采用了偏移量递减的贪心策略.为了验证这种方式的有效性,本文设计了偏移量递增的分配方式.在上述实验场景中生成了 200 条数据流,CQF 队列长度同样为 15 kB,分别运行两种算法,结果如图 7 所示.偏移量递减比偏移量递增的调度成功率平均高 3.89%,偏移量递减的调度方式比不调整偏移量的调度方式的调度成功率平均高 18.32%.与按照偏移量递增的方式相比,偏移量递减的方式能够为周期较小的报文留下调度周期中相对靠前的时隙,因此调度成功率比偏移量递增的方式高.在周期大小降序的流映射方式中,与偏移量递增相比,偏移量递减的方式能够将调度成功率提高 9.65%.按周期大小排序,位置靠后的报文的周期小,可选择的偏移量的范围也小,周期大的报文占用了靠后的时隙,留下了相对靠前的时隙资源,因

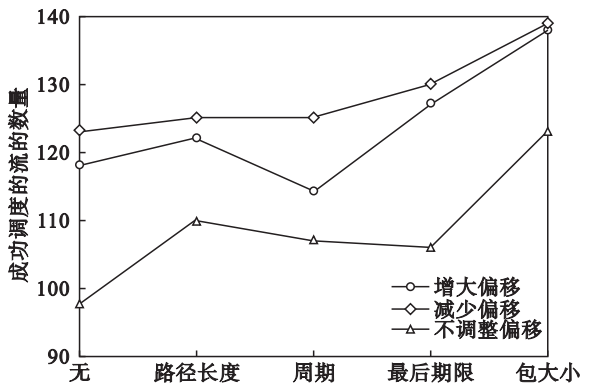


图 7 offset 升序与降序的对比

Fig.7 Increasing offset versus decreasing offset

此调度成功率得到明显提高.

为了研究队列资源对调度成功率的影响,本文调整队列长度,将 CQF 队列长度分别设置为 8,10,12,14 kB,生成 200 条数据流,统计调度成功的流的数量,如图 8 所示.实验表明,随着队列资源的增多,调度成功率也随之提高.CQF 队列长度相同时,4 种排序方式之间进行对比,按照报文大小排序的调度成功率远大于其他 4 种方式,即报文大小对调度成功率的影响最大.

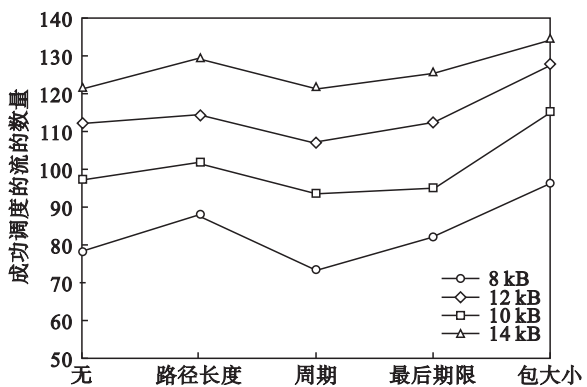


图 8 队列长度对成功调度的流的数量影响

Fig.8 Number of scheduled flows of different length of queue

将队列长度分别设置为 2,4,6,8,10,12,14,16,18,20 kB,采用包大小升序的排序方式、偏移量降序的调度方式,计算 200 条时间敏感流的成功调度的数量,实验结果如图 9 所示.本文设定时隙长度为 125 μs,在千兆以太网中,一个时隙可以传输的最大的时间敏感流大小为 15.625 kB.在时隙长度不变的情况下,当队列长度小于 15.625 kB 时,增长队列长度,相当于增加分配给时间敏感流的带宽,如图 9 中的理论带宽所示,那么能够成功调度的流的数量也随之增加.当 CQF 队列的长度超过 15.625 kB 时,由于链路速率的限制,即使再增加队列长度,调度成功率也不会继续增长.

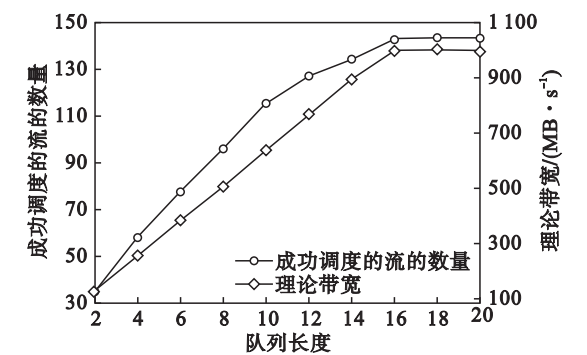


图 9 带宽资源对成功调度的流的数量影响

Fig.9 Number of scheduled flows of different bandwidth

7 结 语

本文研究发现,在基于 CQF 模型的资源调度中,需要从时间维度上对资源进行分配.如果在端系统上直接发送数据流,某些时隙内的流量会超过队列长度,而在另一些时隙中存在大量的空闲队列资源.通过延迟端系统的发送时间,可以利用这些空闲资源,成功调度更多的流量.为此,本文将基于 CQF 模型的资源调度问题抽象为多约束条件下的资源规划最大化问题,将约束条件归纳为:1)帧发送偏移约束;2)接收窗口约束;3)端到端延迟约束;4)队列资源约束.

本文借鉴贪心算法思想,提出了基于起始时隙分配的资源调度算法(SSA),通过对端系统的时隙进行调度,提高队列资源利用率,以最大化成功调度的流的数量.本文分别从周期、报文大小、路径长度、允许的最大端到端延迟4个角度出发,提出了4种贪心策略,使占用资源少的报文能够被优先调度.

本文构建了工业控制网络中典型的环形网络拓扑对算法的有效性进行评估,通过与不控制发送时隙的分配策略相比,本文提出的 SSA 算法平均可以提高 41.84%.通过对4种策略的对比结果发现,报文大小对资源调度的影响最大.

参考文献:

- [1] Nasrallah A, Thyagaturu A, Alharbi Z, et al. Ultra-low latency (ULL) networks: The IEEE TSN and IETF Det Netstandards and related 5G ULL research [J]. *IEEE Communications Surveys & Tutorials*, 2018, 21(1): 88 - 145.
- [2] Nayak N G, Dürr F, Rothermel K. Incremental flow scheduling and routing in time-sensitive software-defined networks[J]. *IEEE Transactions on Industrial Informatics*, 2017, 14(5): 2066 - 2075.
- [3] Chowdhury S R. Packet scheduling algorithms for a software-defined manufacturing environment[D]. Stuttgart: University of Stuttgart, 2015.
- [4] Wilfried S, Silviu C, Ramon S O. Traffic planning for time-sensitive communication [J]. *IEEE Communications Standards Magazine*, 2018, 2(2): 42 - 47.
- [5] Steiner W. Synthesis of static communication schedules for mixed-criticality systems [C]//14th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops. Newport Beach, 2011: 11 - 18.
- [6] Craciunas S S, Oliver R S, Chmelfik M, et al. Scheduling real-time communication in IEEE 802.1 Qbv time sensitive networks [C]//Proceedings of the 24th International Conference on Real-Time Networks and Systems. Brest: ACM, 2016: 183 - 192.
- [7] 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. Brest: ACM, 2016: 203 - 212.
- [8] Steiner W. An evaluation of SMT-based schedule synthesis for time-triggered multi-hop networks [C]//2010 31st IEEE Real-Time Systems Symposium. San Diego, 2010: 375 - 384.
- [9] Pozo F, Rodriguez-Navas G, Hansson H, et al. SMT-based synthesis of TTEthernet schedules: A performance study [C]//10th IEEE International Symposium on Industrial Embedded Systems (SIES). Siegen, 2015: 1 - 4.
- [10] Pozo F, Steiner W, Rodriguez-Navas G, et al. A decomposition approach for SMT-based schedule synthesis for time-triggered networks [C]//IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA). Luxembourg, 2015: 1 - 8.
- [11] Tamas-Selicean D, Pop P, Steiner W. Synthesis of communication schedules for TTEthernet-based mixed-criticality systems [C]//Proceedings of the 8th IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis. Tampere: ACM, 2012: 473 - 482.
- [12] Oliver R S, Craciunas S S, Steiner W. IEEE 802.1 Qbv gate control list synthesis using array theory encoding [C]//IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS). Porto, 2018: 13 - 24.