

引用格式:冯泽坤,龚龙庆,徐丹妮,等. 时间敏感网络中基于 ILP 的动态流量均衡调度算法[J]. 微电子学与计算机,2021,38(6):33-37. [FENG Z K, GONG L Q, XU D N, et al. ILP-based dynamic flow balancing scheduling algorithm in time sensitive network[J]. Microelectronics & Computer, 2021, 38(6): 33-37.]

时间敏感网络中基于 ILP 的动态流量均衡调度算法

冯泽坤, 龚龙庆, 徐丹妮, 刘 扬
(西安微电子技术研究所, 陕西 西安 710054)

摘 要: 针对目前时间敏感网络(Time Sensitive Network, TSN)中多采用离线调度,在处理动态需求时调度开销过大的问题,提出了一种适用于 TSN 的基于整数线性规划(Integer Linear Programming, ILP)的动态流量均衡调度算法.该算法以已生成的离线调度表为基础,通过一种增量计算方式,有效降低拓扑和业务变化带来的开销;同时,针对链路负载可能出现的极端情况,制订了流量均衡策略,避免出现延时瓶颈问题,进而提升通信实时性.实验数据表明,与传统静态调度算法相比,在处理动态流量时的运算时间减少,链路负载更加均衡.

关键词: TSN; ILP; 动态均衡调度; 链路负载; 运算时间

中图分类号: TP393.11

文献标识码: A

文章编号: 1000-7180(2021)06-0033-05

ILP-based dynamic flow balancing scheduling algorithm in time sensitive network

FENG Ze-kun, GONG Long-qing, XU Dan-ni, LIU Yang

(Xi'an Microelectronics Technology Institute, Xi'an 710054, Shaanxi China)

Abstract: In order to solve the problem that off-line scheduling is often used in Time Sensitive Network (TSN), and the scheduling overhead is too high when dealing with dynamic demands, a dynamic flow balancing scheduling algorithm based on integer linear programming (ILP) for TSN is proposed. Based on the generated offline scheduling table, the algorithm can effectively reduce the overhead caused by topology and service changes through an incremental calculation method. At the same time, according to the extreme situation of the link load, the traffic balancing strategy is formulated to avoid the delay bottleneck problem, and then improve the real-time communication performance. Experimental data shows that compared with the traditional static scheduling algorithm, the computing time is greatly reduced and the link load is balanced better.

Key words: TSN; ILP; dynamic balancing scheduling; link load; computing time

1 引言

随着网络通信的不断发展,许多应用场景,例如工业物联网、车载网络、航空电子网络以及移动前传网络等,对网络的实时性和确定性要求日益提升,传统以太网已经无法满足人们的需求^[1].为了满足实时性,以 EtherCAT、Profinet、TTE 等为代表的一些实时通信技术被应用在一些对实时性要求高的场

景,这些技术都基于以太网,但同时又具有一些专用机制,使得相互之间并不能兼容运行,也就限制了实时性以太网的进一步发展.在这种背景下,IEEE802 TSN 任务组提出了时间敏感网络的概念,意图构建一个统一的数据链路层协议,通过标准化使其在各个领域都能够同构运行,提供实时的数据传输.

TSN 目前主要开发了时钟同步、流量调度、网络配置系列标准集,网络配置可以将调度信息配置

到各节点,各节点在时钟同步的基础上用各种调度方法来实现实时性与确定性.其中流量调度方法中最重要的便是 IEEE 802.1 Qbv^[2] 基于门控的时间感知整形器(Time Aware Shaper, TAS),定义了一种基于队列传输选择的门控机制,每个传输队列与一个传输门相关联,传输门的状态一共有两种:open:该队列门打开,若该队列中有待传输的帧,可以进行传输;close:该队列门关闭,该队列无法进行传输.传输门的状态由 GCLs(Gate Control Lists, GCLs)控制,门控列表周期性的调度每个队列的流量,并控制每个队列门打开的时间,门控机制如图 1 所示.此方法的实现依赖于 GCLs 的配置,但是协议中并没有如何得到 GCLs 的方法.

目前已经有学者提出自己的调度算法,可以得到相对应的 GCLs,但是绝大部分都只是静态调度^[3-5].若网络出现小幅变动,则需重新全部调度,调度时间过长,并不实用.因此,针对网络运行时需要新加入可调度流和交换机某端口故障的情况,以及网络中链路负载可能出现的极端情况,本文在文献[6]的研究基础上,改进了其算法.

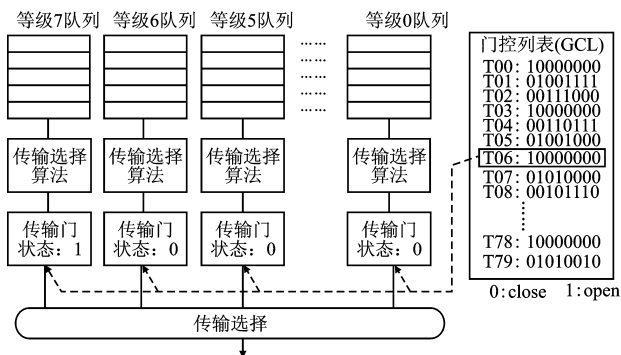


图 1 IEEE802.1 Qbv 门控机制

2 系统模型

实时网络中的应用数据一般分为两种,一种是系统控制的实时数据,即 TSN 数据流(简称 TSN 流),还有一种是用于辅助服务的尽力而为数据,即 BE 数据流. TSN 流具有固定的周期和大小,必须在其截止时间内到达其目的节点并满足抖动约束.相比之下,BE 数据流不需要时间保证.所以,在调度 TSN 流时,可以取它们周期的最小公倍数为一个集群周期,保证在一个集群周期内,满足这些流的实时性与确定性.

首先对时间敏感网络和 TSN 流建模.将时间敏感网络建模为无权有向图 $G=\langle V, E \rangle$,其中 V 是

网络节点的集合,由终端(End Station, ES)和交换机(Switch, SW)组成, $E = \{[v_i, v_j], v_i, v_j \in V, \text{并且 } v_i, v_j \text{ 通过一个链路连接}\}$,是定向网络链路集,比如:一条链路 $[v_i, v_j]$ 传输流时,将流从源 v_i 传输到目的 v_j . $|V|$ 和 $|E|$ 分别是节点数和链路数.

在时间敏感网络中,考虑一组具有时延和抖动约束的 TSN 流 F ,每个 $f_a \in F$ 由元组 $\langle S_a, D_a, T_a, l_a, ddl_a \rangle$ 表示,其中 S_a 表示 f_a 的源节点; D_a 表示 f_a 的目的节点; T_a 表示 f_a 的周期; l_a 表示 f_a 的长度; ddl_a 表示 f_a 的传输延迟上限.为了保证实时性,需要保证 f_a 在 ddl_a 内从 S_a 传输到 D_a .

分别引入两类变量来说明路由和调度问题.二进制决策变量 $lo_{[v_i, v_j], f_a}$ 表示链路 $[v_i, v_j] \in E$ 是否被 $f_a \in F$ 占用:

$$lo_{[v_i, v_j], f_a} = \begin{cases} 1: & \text{流 } f_a \text{ 占用路径 } [v_i, v_j] \\ 0: & \text{流 } f_a \text{ 未占用路径 } [v_i, v_j] \end{cases} \quad (1)$$

为了描述调度问题,在 $[0, T_a]$ 引入实数型决策变量 $t_{[v_i, v_j], f_a}$ 来表示调度结果.如果 f_a 通过 $[v_i, v_j]$ (即, $lo_{[v_i, v_j], f_a} = 1$) 时, $t_{[v_i, v_j], f_a} \in [1, T_a]$ 是流 f_a 在链路 $[v_i, v_j]$ 上的发送时刻,否则, $t_{[v_i, v_j], f_a} = 0$,由上可得,调度 S 由以下元组唯一确定:

$$S = \{ \langle lo_{[v_i, v_j], f_a}, t_{[v_i, v_j], f_a} \rangle \mid \forall [v_i, v_j] \in E, \forall f_a \in F \} \quad (2)$$

3 基于 ILP 的动态流量均衡调度算法

ILP 是一种广泛接受的,成熟的方法,可以理论上用于解决所有可以用整数决策变量建模的问题.因此,本文基于 ILP 设计了该调度算法.完成对系统模型的建立后,在考虑多种链路约束和延迟要求下,将路由与调度问题规划成一个 ILP 模型,从而为 TSN 流选择出最佳通信路由,并合理分配流量,从而满足 TSN 流的数据传输延迟要求.

3.1 时间敏感约束

在数据传输过程中,由于网络通信存在传输冲突和转发延迟,若没有延迟约束,则调度无法提供实时性保证.为了确保其实时性,调度应该满足某些约束,这些约束称为时间敏感约束.

时间敏感约束包括路由约束、路径相关约束、无争用约束、延迟约束和链路负载约束,流 f_a 的约束如下:

3.1.1 路由约束

路由约束描述了网络流的传输连续性,保证流 f_a 从源 S_a 通过交换机到达目的节点:

$$\sum_{\forall v_i | [v_i, s_a] \in E} lo_{[v_i, s_a], f_a} = 0, \quad \sum_{\forall v_j | [s_a, v_j] \in E} lo_{[s_a, v_j], f_a} = 1 \quad (3)$$

式(3)保证了单播流 f_a 从源节点 S_a 发出,并且 S_a 不接收 f_a .

$$\forall v_i \in v_{ES} \setminus S_a: \sum_{\forall v_j | [v_i, v_j] \in E} lo_{[v_i, v_j], f_a} = 0 \quad (4)$$

式(4)保证在调度 f_a 时,不是流 f_a 的源节点的终端的发送路径不被占用.

$$\forall v_k \in SW: \sum_{\forall v_i | [v_i, v_k] \in E} lo_{[v_i, v_k], f_a} \leq 1, \quad \sum_{\forall v_i | [v_i, v_k] \in E} lo_{[v_i, v_k], f_a} = \sum_{\forall v_j | [v_k, v_j] \in E} lo_{[v_k, v_j], f_a} \quad (5)$$

式(5)主要定义了中间链路的约束,第一个不等式表示每个交换机最多只从一个端口接收流 f_a , 由于采用单播模型,第二个等式表示每个交换机发送和接收流 f_a 的次数相等.

$$\sum_{\forall v_i | [v_i, D_a] \in E} lo_{[v_i, D_a], f_a} = 1, \quad \sum_{\forall v_j | [D_a, v_j] \in E} lo_{[D_a, v_j], f_a} = 0 \quad (6)$$

式(6)保证了流 f_a 的目的节点 D_a 接收流 f_a , 并且 D_a 不发送流 f_a .

$$\forall [v_i, v_j] \in E: t_{[v_i, v_j], f_a} \leq M \times lo_{[v_i, v_j], f_a} \quad (7)$$

式(7)确保流 f_a 未用到的链路上 $t_{[v_i, v_j], f_a} = 0$, 即在该链路上不传输流 f_a . 其中 M 是一个足够大的整数,在整数线性规划(ILP)中,常用来表示条件约束,引入 M 是确保了该约束只在链路 $[v_i, v_j]$ 被流 f_a 占用时才起作用,后面引入 M 是起到同样的作用.

3.1.2 路径相关约束

路径相关约束描述流的传输定时约束. 由于流只能在到达当前交换机后转发到下一个交换机,因此流 f_a 的路径相关约束如下所示:

$$\forall v_k \in SW, \forall v_i | [v_i, v_k] \in E, \forall v_j | [v_k, v_j] \in E: \\ t_{[v_k, v_j], f_a} - t_{[v_i, v_k], f_a} \geq t_{tran_a} + \Delta t_p + M \times (lo_{[v_k, v_j], f_a} + lo_{[v_i, v_k], f_a} - 2) \quad (8)$$

3.1.3 无争用约束

无争用约束描述所有链接的独占链接约束. 由于一个链路只能同时传输一个数据流,因此资源限制如下:

$$\forall [v_i, v_j] \in E, \forall f_a, f_b \in F, a \neq b:$$

$$r = (t_{[v_i, v_j], f_a} - t_{[v_i, v_j], f_b}) \bmod (\gcd(T_a, T_b)); \\ t_{tran_b} - M \times (2 - lo_{[v_i, v_j], f_a} - lo_{[v_i, v_j], f_b}) \leq r \leq \gcd(T_a, T_b) - t_{tran_a} + M \times (2 - lo_{[v_i, v_j], f_a} - lo_{[v_i, v_j], f_b}) \quad (9)$$

式(9)中 $\gcd(T_a, T_b)$ 是流周期 $p = T_a, T_b$ 的最大公约数,每两个流的约束意味着两个流的传输在其周期的最大公约数中不冲突. 为了简化表述并保持约束,引入变量 r . r 是流 f_a 与流 f_b 在同一条链路上传输时间差除以它们的周期最大公约数的余数,为了确保两个流的传输时间不冲突,最小传输差 r 必须大于每个流的传输时间.

3.1.4 延迟约束

延迟约束描述了流从发送到接收的时间应该小于延迟上限,因此延迟约束如下:

$$\sum_{\forall v_i | [v_i, D_a] \in E} t_{[v_i, D_a], f_a} - \sum_{\forall v_j | [S_a, v_j] \in E} t_{[S_a, v_j], f_a} \leq ddl_a - t_{tran_a} \quad (10)$$

式(10)中 ddl_a 是流 f_a 的延迟上限,不等式左边表示流 f_a 从源节点到与目的节点相连的交换机的传输时间. 之前的式(3)和式(6)保证了流 f_a 只从源节点发出一次并且目的节点只接收一次. 所以,该约束使得流 f_a 满足延迟上限.

3.1.5 链路负载约束

链路负载约束描述了 TSN 流在交换机间各链路的负载应该小于某阈值,因此链路负载约束如下:

$$\forall v_i, v_j \in SW, [v_i, v_j] \in E: \sum_{f_a \in F} B_{[v_i, v_j], f_a} \leq B_{max} \quad (11)$$

式(11)中 $B_{[v_i, v_j], f_a}$ 表示流 f_a 在链路 $[v_i, v_j]$ 上的带宽利用率; B_{max} 为预设的链路的最大带宽利用率,本文设为 20%. 该约束保证了 TSN 流在交换机间链路上的最大负载不超过 20%,避免在极端情况下,某些链路上的负载过高,而其余链路空闲的情况.

3.2 最优化目标

时间敏感约束仅仅保证了调度是可行的,但不一定是最优的,因此本文算法针对带宽利用率设计了如下优化目标:

$$\min(\sum_{[v_i, v_j] \in E} \sum_{f_a \in F} lo_{[v_i, v_j], f_a}) \quad (12)$$

优化目标为最短路径,因为上面约束(11)已经避免了出现某些链路负载过大的情况,最优化目标

设为最短路径可以更好的满足 TSN 流的延迟要求.

3.3 路由和调度程序

时间敏感约束和优化目标如上所述,在本小节中设计调度过程. 具体的说,给定一个 TSN 网络 G , 现有流的集合 F_{ex} 和新加流的集合 F_{add} , 重新调度好的流集合 F_s , 现有流 F_{ex} 的调度结果 S_{ex} , 故障链路 L_F , 输出为最终的流调度结果 S_v . 程序在不干扰现有流的集合 F_{ex} 中未用到故障链路 L_F 的调度结果的情况下, 为 F_{ex} 中剩余的流和新加的流 F_{add} 生成有效的调度. 调度程序如下所示:

Procedure:

Input: Network G , existing flow set F_{ex} , add flow set F_{add} , schedule flow set F_s , existing schedule S_{ex} , faulty link set L_F .

Output: Schedule S_v

```
1: while  $F_{ex} \cup F_{add} \neq \emptyset$  do
2:  $F_{ex} \leftarrow F_{ex} \setminus f_a$  OR  $F_{add} \leftarrow F_{add} \setminus f_a$ ,  $C \leftarrow \emptyset$ 
3: if  $f_a$  don't need schedule again then
4:  $S_v \leftarrow S_v \cup S_{ex, f_a}$ ,  $F_s \leftarrow F_s \cup f_a$ 
5: else
6:  $C \leftarrow \text{timesensitiveconstraints}(3) - (11)$ 
7:  $S_{f_a} \leftarrow \text{ILP}(C)$ 
8: if  $S_{f_a} \neq \emptyset$  then
9:  $S_v \leftarrow S_v \cup S_{f_a}$ ,  $F_s \leftarrow F_s \cup f_a$ 
10: else
11: return false
12: endif
13: endif
14: endwhile
15: return  $S_v$ 
```

4 实验验证与分析

4.1 实验环境

4.1.1 计算环境

计算环境为 Windows 7 的 PC 机, 处理器为 Intel i7-3770 四核 3.40 GHz 和 8 GB 的内存. 使用 CPLEX 12.8 对算法的性能进行验证评估, CPLEX 是 IBM 的一个 ILP 解算器.

4.1.2 网络结构

为了方便观察测试结果, 本文假设网络结构如图 2 所示, 网络结构使用 3 个交换机, 8 个终端和 11 条全双工的链路组成, 链路传输速度假设为 1 Gbps, 交换机处理延迟均设为 $20 \mu s$, 后面的测试结果均使用该网络结构.

4.2 链路负载比较

链路负载的使用情况决定着调度结果的好坏. 本文算法保证了在可调度的情况下, 若某条链路上的负载超过 20% 时, 则会绕过该链路, 选择其他链

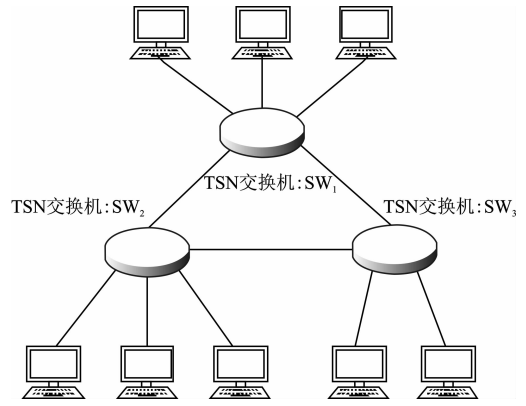


图 2 网络结构

路. 即保证了所有链路上的负载均不超过 20%, 图 3, 图 4 分别是使用基于 ILP 的传统静态调度算法和使用本文算法生成调度结果的链路最大负载和平均负载比较:

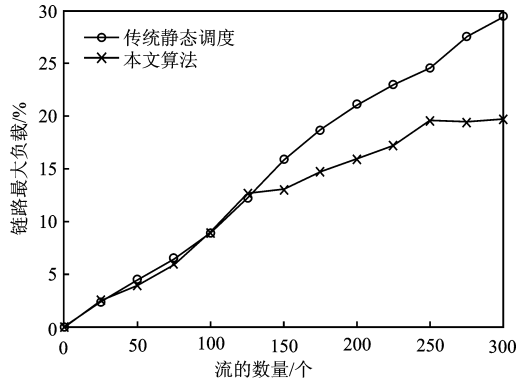


图 3 链路最大负载比较

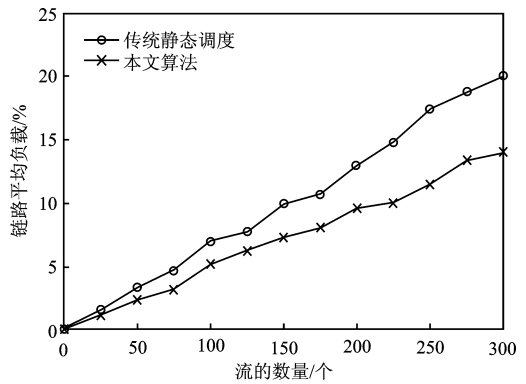


图 4 链路平均负载比较

4.3 动态需求运算时间比较

随机生成 10、20、30 和 40 条流量, 大小为 64~1512 字节, 周期分别为 1 ms、2 ms、4 ms, 延迟上限均设为 0.2 ms. 选取添加流和交换机故障两种典型测试场景, 先利用程序生成四种流量模型的路由和调度结果, 再在其基础上使用基于 ILP 的静态调度

算法和使用本文算法进行测试,两种方法的运算时间对比如表 1,表 2 所示:

表 1 添加流的运算时间比较

流数	运算时间(s)	
	静态调度	本文算法
10+1	0.195	0.016 3
20+1	0.477	0.020 0
30+1	1.11	0.021 0
40+1	4.03	0.025 1

表 2 交换机故障的运算时间比较

流数	运算时间(s)	
	静态调度	本文算法
10	0.024 8	0.013 0
20	0.317 3	0.013 1
30	0.730 3	0.171 0
40	1.510 0	0.333 0

从表 1,表 2 可以看出,在上述两种情况下,使用本文算法的运算时间比使用静态调度的运算时间开销大幅减少.

5 结束语

本文在研究 IEEE802.1 Qbv 协议的基础上,提出了一种基于 ILP 的动态流量均衡调度算法,并基于该方法实现了动态调度,生成了满足实时性的调度表,并且保证了链路的负载均衡. 运算结果表明使用本文算法比静态调度算法可调度性提高,处理动态流量时调度开销大幅减少,更好的保证了其实时性.

参考文献:

[1] 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): 31-42. DOI: 10.11959/j.issn.1000-0801.2019227.
CONG P Z, TIAN Y, GONG X Y, et al. A survey of key protocol and application scenario of time-sensitive

network[J]. Telecommunications Science, 2019, 35(10): 31-42. DOI: 10.11959/j.issn.1000-0801.2019227.
[2] IEEE. 802.1Qbv-2015 IEEE Standard for Local and metropolitan area networks-Bridges and Bridged Networks-Amendment 25: Enhancements for Scheduled Traffic[S]. Piscataway, NJ: IEEE, 2016. DOI: 10.1109/IEEESTD.2016.7440741.
[3] 彭轻羽. 面向工业物联网的 TSN 调度算法研究[D]. 杭州: 浙江理工大学, 2019.
PENG Q Y. Research on TSN scheduling algorithm for industrial internet of things[D]. Hangzhou: Zhejiang Sci-Tech University, 2019.
[4] D? RR 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, NY: ACM, 2016: 203-212. DOI: 10.1145/2997465.2997494.
[5] CRACIUNAS S S, OLIVER R S, CHMEL? K M, et al. Scheduling real-time communication in IEEE 802.1Qbv time sensitive networks[C]//Proceedings of the 24thInternational Conference on Real-Time Networks and Systems. New York:ACM, 2016: 183-192. DOI: 10.1145/2997465.2997470.
[6] YU Q H, GU M. Adaptive group routing and scheduling in multicast time-sensitive networks[J]. IEEE Access, 2020, 8: 37855-37865. DOI: 10.1109/ACCESS.2020.2974580.

作者简介:

冯泽坤 男,(1996-),硕士研究生.研究方向为嵌入式计算机应用技术. E-mail:fzk0316@163.com.
龚龙庆 男,(1962-),博士生导师,研究员.研究方向为空间计算机技术.
徐丹妮 女,(1989-),硕士,工程师.研究方向为实时以太网通信技术.
刘 扬 男,(1991-),博士研究生.研究方向为嵌入式计算机系统结构.