

时间敏感网络时隙感知循环排队转发流量整形机制

蔡岳平* 任志文

(重庆大学微电子与通信工程学院 重庆 400030)

摘要: 时间敏感网络是智能工厂内网的核心技术之一。智能工厂内存在多种需求各异的业务流。为保证关键业务流的性能,同时提升网络带宽利用率,该文提出一种时隙感知循环排队转发流量整形机制(TSA-CQF)。TSA-CQF通过将低优先级流量插入CQF奇偶队列中剩余可用时隙中传输提高带宽利用率。TSA-CQF机制包括低优先级流量的时隙感知插入和全局流量规划两个部分。低优先级流量的时隙感知插入是在CQF队列出队时,通过感知奇偶队列剩余时隙的大小,将低优先级流量插入到奇偶队列的剩余时隙进行传输。将全局流量规划建模为多条件约束目标优化问题,通过模拟退火算法求解,完成全局流量的调度,提高可调度流数目,进一步提高资源利用率。仿真结果表明,在混合流量条件下TSA-CQF比传统CQF机制平均提高了带宽利用率11.29%。与传统的CQF相比,TSA-CQF在牺牲一定调度策略生成时间的前提下,能明显提高带宽利用率并且降低最坏端到端时延。

关键词: 工业互联网; 时间敏感网络; 流量整形; 带宽利用率

中图分类号: TN915

文献标识码: A

文章编号: 1009-5896(2023)06-1999-08

DOI: 10.11999/JEIT220530

Traffic Shaping Mechanism Based on Time Slot-Aware Cyclic Queuing and Forwarding in Time-Sensitive Networking

CAI Yueping REN Zhiwen

(School of Microelectronics and Communication Engineering, Chongqing University, Chongqing 400030, China)

Abstract: Time sensitive network is one of the core technologies of future smart factories. There are multiple business flows with different requirements in smart factories. To ensure the performance of critical traffic flows and improve network bandwidth utilization, a Time Slot-Aware Cyclic Queuing and Forwarding (TSA-CQF) mechanism is proposed. TSA-CQF improves bandwidth utilization by inserting low-priority traffic into the remaining available time slots of the CQF queues. TSA-CQF mechanism includes slot-aware insertion of low-priority traffic and global traffic planning for low-priority traffic. The first part of TSA-CQF is to insert low-priority traffic into the remaining time slots of the CQF queues. The global traffic planning is modeled as a multi-conditional objective optimization problem, and it is solved by the simulated annealing algorithm to accept as many flows as possible to increase the bandwidth utilization. Simulation results show that TSA-CQF improves the bandwidth utilization by 11.29% on average compared with the traditional CQF mechanism under mixed traffic conditions.

Key words: Industrial internet; Time-Sensitive Network (TSN); Traffic shaping; Bandwidth utilization

1 引言

随着网络通信技术的发展,更多智能设备通过网络连接,人们的生产生活变得更加智能化,极大地推动了人类社会的进步和发展。在工业4.0(第4次工业革命)的背景下,智能工厂内部的智能设备

的数量不断增多,其通信带宽与时延需求也不断提升,现有的网络技术已经无法满足^[1]。为了满足这些新需求,业界开始采用交换式以太网作为一种新的解决方案。但是不确定的排队时延阻碍了交换式以太网提供确定性转发服务^[2,3]。确定性转发服务对包传输过程中的时延、丢包和时延变化(抖动)有严格的要求,但传统的以太网的数据包转发是不确定的,在网络发生拥塞时,传统以太网通过缓存机制来减少数据包的丢失,排队和缓存会带来时延和抖动等问题^[4]。尽管尽力而为服务提供了较高的链路利用率,较大的端到端吞吐量和简单的实现方式,

收稿日期: 2022-04-27; 改回日期: 2022-10-25; 网络出版: 2022-11-18

*通信作者: 蔡岳平 caiyueping@cqu.edu.cn

基金项目: 国家重点研发计划(2020YFB1710900)

Foundation Item: The National Key Research and Development Program of China (2020YFB1710900)

但不能保证端到端的延迟。时间敏感网络(Time-Sensitive Network, TSN)作为一种在以太网上进行扩展的2层网络技术^[5],有着良好的可扩展性,而且能满足确定性转发服务的时延和抖动需求。

TSN流量整形机制是实现流量转发性能保证的关键技术之一。目前TSN的同步整形方法有两种,一种是IEEE 802.1Qbv标准推荐的TAS(Time-Aware Shaper)^[6],另一种是IEEE 802.1Qch标准推荐的CQF(Cyclic Queuing and Forwarding)^[7]。两种同步流量整形方法都是对入口端的流量在全局时间同步的条件下进行入队和出队操作,达到流量整形的目的,完成端到端流的确定性交付。TAS的算法复杂性相对于TTE(Time-Trigger Ethernet)更加简单,但是它需要动态配置全局的门控列表(Gate Control List, GCL),已有相关工作表明,GCL的生成是一个NP难题。在终端节点较多的网络中,有较大概率无法生成合理的GCL,而且TAS的QoS(Quality of Service)分析也较为复杂,所以它的复杂性和可扩展性问题仍待解决^[8]。在基于CQF模型的TSN中,可以通过对包到达时间进行分类,将其有序传输到对应的奇偶队列中,而不考虑包的输入和输出时间。通过对包到达时间和速率的平缓控制,实现流量整形和周期转发,保证关键流的时延抖动性能。

在即将到来的工业4.0背景下,大量机器与应用程序相互协作,因此工厂内网络既需要提供机器间水平通信服务,同时也需要提供用户到边缘云和数据中心(Data Center, DC)的垂直通信服务^[9]。文献^[10]说明了智能工厂中用于控制、协调、操作、服务、并行处理5种流量类型,其中用于服务的流量是尽力而为(Best-Effort, BE)流量,其他4种为时间敏感流量(时敏流量)。文献^[11]为了处理以错误周期到达的不同步数据包(由于处理时延带来的变化),通过3-Queue CQF额外添加一个队列来缓冲这种类型的数据包。此方法可以减少错误时隙数据包的数量,提高整体流量整形的容错,但是会造成额外的资源开销并且提高端到端时延。文献^[12]考虑在无法进行全局同步的情况下,提出用Paternoster机制创建4个队列,每个交换机之间保持频率同步,即周期持续时间在节点之间是相同的。与CQF相比,Paternoster机制不需要时钟同步并且能降低平均时延,但是无法保证流量的确定性交付。文献^[13]提出了一种注入时间规划(Injection Time Planning, ITP)机制,此机制在CQF的基础上通过启发式算法进行全局流量规划,在中心协调器的计算下将指令下发到每个传输开始端口,改变

数据源的传输开始时间,减少数据在链路的冲突,提高了整体流量的可调度数量和带宽利用率。但是其分析的流量中没有高优先级流量和低优先级流量混合的情况,此方案可能导致高优先级流的最坏端到端时延增大。文献^[14]提出了一种新的时间感知循环排队(Time-Aware Cyclic-Queuing, TACQ)机制,并设计了一种新颖的无等待整形器(No-Wait Shaper, NWS)解决同步流低抖动和循环流低时延的需求。这种方案用尽可能少的时隙处理同步流,对循环流则通过扩展CQF进行流量整形。与CQF机制相比,评估结果表明TACQ能够严格地将同步流的抖动限制为零,同时仍然保持循环流的有界时延传输。但是其设计存在速率受限问题,且在处理流量调度问题时,和文献^[15]一样是将所有的流量看作时敏流量,没能考虑到不同优先级流量混合传输的问题。

本文面向时敏流量与非时敏流量共网传输的智能工厂网络,提出了一种时隙感知循环排队转发流量整形机制(Time Slot-Aware Cyclic Queuing and Forwarding, TSA-CQF)。TSA-CQF机制是通过将非时敏流量插入CQF的奇偶队列的剩余时隙中进行传输提高带宽利用率,包括低优先级流量的时隙感知插入和全局流量规划两个部分。低优先级流量的时隙感知插入是在CQF队列出队时,通过感知奇偶队列剩余时隙的大小,将低优先级流量插入到奇偶队列的剩余时隙进行传输。全局流量规划是将调度策略问题抽象成一个多条件约束目标优化问题,在求解多条件约束目标优化问题时,考虑到局部最优解问题将直接影响期望的时延要求能否达到,采用具有概率的全局优化性能的模拟退火算法,该算法在计算时具有渐进收敛性和并行性,且能赋予搜索过程时变且趋于零的概率突跳性,从而能有效避免陷入局部最优解。得到调度策略后,完成全局流量的整形和调度,进一步提高资源利用率,最后通过仿真验证了所提出机制的正确性和优越性。

2 系统模型

2.1 CQF问题分析

流量为网络中传输的数据,具有相同5元组(源MAC地址、目的MAC地址、源IP地址、目的IP地址、端口号)的数据包属于同一条流量。在系统模型中,将背景流、尽力而为转发流、音视频流等归类为非时敏流,将网络控制、同步信号等高优先级流量归类为时敏流。IEEE 802.1Qch标准定义的CQF机制通过底层奇偶队列之间的循环接收与发送操作实现数据的传输,但是存在如下问题:

(1)周期循环流量在某一端队列大量聚集。如图1所示。许多周期循环流量(时敏流量的其中一种)由于其周期转发的特殊性，很容易在一个队列汇合，导致另一队列空闲，缓存利用率不高。

(2)固定时隙导致的带宽浪费。如图2所示。由于CQF调度的特殊性，需要将整个端口传输带宽划分为固定的时隙，此固定时隙为了满足时敏流量的传输安全性，会预留一部分带宽作保护带宽。通常选择大于最大报文的时隙长度，但是当链路上传输较小报文时，时隙利用率低。

2.2 交换机模型

本方案在CQF机制的基础上增加了一个出队整形模块，如图3所示，流量通过流分类在入队门控的控制下进入对应的队列，将多优先级队列划分为时敏流量和非时敏流量两种队列，在有时敏流量进行传输时，其他队列流量通过优先级调度排队出队。在有时敏流量进行传输时，时敏流量进入奇偶队列进行传输。奇数周期内，时敏流量进入奇数队列，而偶数队列进行数据传输。偶数周期内，时敏流量进入偶数队列，而奇数队列进行数据传输。非时敏流量通过入队整形，插入时敏流量的空余时隙中进行传输。

TSA-CQF机制将一个包的大小映射成固定的时长，完成空间到时间的映射。在CQF机制中需要设定一个固定的周期时隙，此时隙也是计算端到

端最坏时延的一个重要参数。中心协调器(用于管理TSN的高性能全局处理模块)可以通过固定时隙和传输时敏数据的大小来感知固定时隙中所剩余的时隙大小，从而利用剩余的时隙来传输非时敏流量，如图4所示。

2.3 TSA-CQF流量整形机制

TSA-CQF流量整形机制步骤如图5所示。

(1)系统设置初始的固定时隙长度，初始化网络拓扑、流量模型、链路最大带宽、奇偶队列时隙大小。流分类将有确定性交付保证的流量放入奇偶队列(根据到达时隙的具体时间戳放入奇队列或偶队列)，其他流量放入非时敏流量队列中。

(2)系统对流量进行约束判断，遍历所有约束条件，对每个流量进行初始发送时间戳的调整，直至所有流量能顺利发送。

(3)系统对非时敏流量进行固定长度的切片(切片大小尽量小以保证其能顺利插入)，非时敏流量在符合奇偶队列剩余带宽时插入，并随着奇偶队列中时敏流量的出队而出队。

(4)完成流量的映射后系统通过启发式算法进行求解，最后更新整个系统的可配置资源，进而将消息从主控制器发送到每个交换机的协调器。

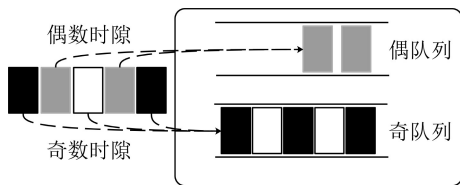


图1 周期流量在某一端队列大量聚集

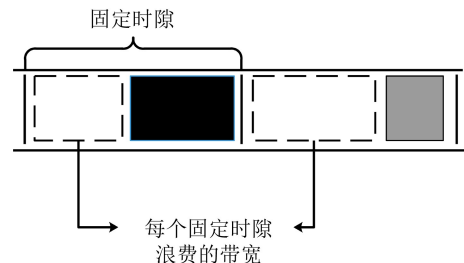


图2 固定时隙导致的带宽浪费

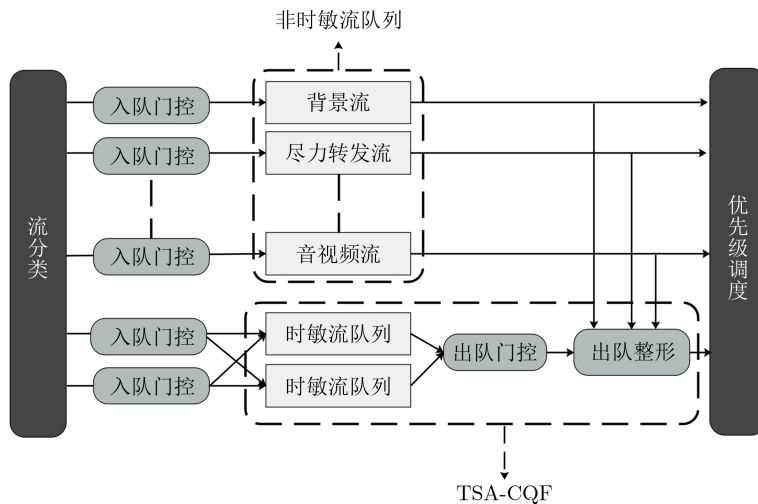


图3 TSA-CQF机制交换机模型

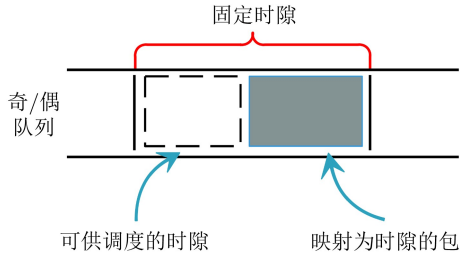


图4 固定时隙中可供调度的时隙

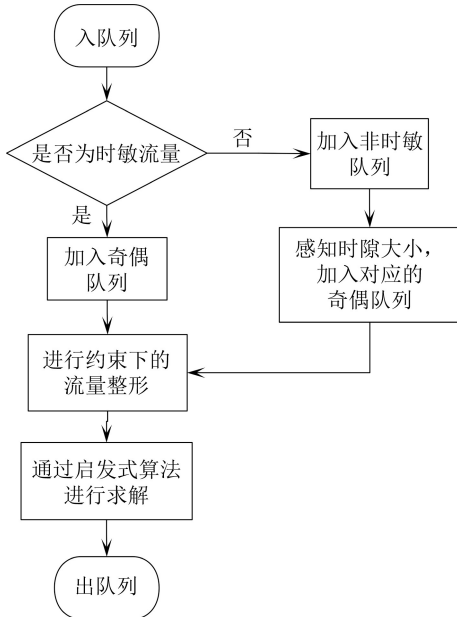


图5 TSA-CQF方案步骤

3 TSA-CQF流量整形建模

此方案可以抽象为一个条件约束问题，优化的目标为所有进行调度的流量的最小端到端最坏时延之和。TSA-CQF机制通过对调度的条件进行约束，然后对约束方程进行求解得到最终的调度结果。此方案将该网络建模为一个有向图 $G(V, E)$ ，其中节点(交换机和终端)是图顶点 (V) 的集合，节点之间的连接通过图边表示。表1对目标函数以及约束条件中的参数进行了说明。

(1) 优化目标。优化的目标函数为所有进行调度的流量的最小端到端最坏时延之和，如式(1)所示

$$\min f(X) = T \left(\sum_{n=1}^m x_n \right)_{De}, x_n \in X \quad (1)$$

s.t. $X \in \Omega$

其中， x_n 表示进行调度的流， n 表示当前流的序号， m 代表所调度的流的数目， X 表示当前约束条件， De 表示最坏端到端时延， Ω 代表所有约束条件。

(2) 端到端的最坏时延约束。端到端最坏时延限制指定流的到达时间和发送时间之间的差必须小于或等于指定的最大值。约束如式(2)所示： l 代表

表1 约束参数说明

参数	描述
S	流的集合
$S_{[a,b]}$	流从发出节点 a 到接收节点 b
T_{s_i}	最坏端到端时延
L_{s_i}	数据大小(以字节为单位)
C_{s_i}	流的周期
$f_{[a,b]}$	流中的帧从节点 a 到节点 b
$f_{[a,b]} \cdot \phi$	流从节点 a 到 b 的帧映射到时钟上的偏移量
$f_{[a,b]} \cdot T$	流从节点 a 到 b 的帧映射到时钟上的周期
$f_{[a,b]} \cdot L$	流从节点 a 到 b 的帧映射到时钟上传输大小
$F_{[a,b]}$	流 $S_{[a,b]}$ 和帧 $f_{[a,b]}$ 的集合

最后一帧，本文用 $sr(s_i)$ 表示流的发送链路，用 $d(s_i)$ 表示接收节点之前的最后一个链路。 $F_{d(s_i)} \cdot \phi$ 表示流 S_i 和帧 f_i 从起始节点到终点节点的偏移量， $F_{d(s_i)} \cdot L$ 表示流 S_i 和帧 f_i 从起始节点到终点节点映射在时钟上的传输大小的集合。

$$\begin{aligned} \forall s_i \in S: & sr(s_i) \cdot mt \cdot f_{sr(s_i)} \cdot \phi + T_{s_i} \\ & \geq d(s_i) \cdot mt \cdot (l(F_{d(s_i)}) \cdot \phi \\ & + l(F_{d(s_i)}) \cdot L) \end{aligned} \quad (2)$$

(3) 帧的起始时间约束。网络中调度的任何帧的帧偏移必须大于或等于0，而且整个传输窗口(偏移量加上帧传输时间)必须符合帧周期。约束如式(3)所示

$$\begin{aligned} \forall f_{[a,b]} \in F_{[a,b]}: & (f_{[a,b]} \cdot \phi \geq 0) \cap \\ & (f_{[a,b]} \cdot T \geq f_{[a,b]} \cdot \phi + f_{[a,b]} \cdot L) \end{aligned} \quad (3)$$

其中， S_i 表示流 i ， $f_{[a,b]}$ 表示流中的帧从节点1到节点2， $F_{[a,b]}$ 表示流 $S_{[a,b]}$ 和帧 $f_{[a,b]}$ 的集合， $f_{[a,b]} \cdot \phi$ 表示流从节点 a 到 b 的帧映射到时钟上的偏移量， $f_{[a,b]} \cdot T$ 表示流从节点 a 到 b 的帧映射到时钟上的周期， $f_{[a,b]} \cdot L$ 表示流从节点 a 到 b 的帧映射到时钟上传输大小。

(4) 队列长度约束。队列缓存区中的所有包的长度之和不能超过队列的最大缓存，否则会将后到达的包直接丢弃。约束如式(4)所示

$$\begin{aligned} \forall s_i \in S, \forall [s_a, s_b] \in s_i, \forall f_{[a,b]} \in F_{[a,b]}: \\ \text{Count}(W \cap s_{[a,b]}) \cdot f_{[a,b]} \cdot L \leq W_{QS} \end{aligned} \quad (4)$$

其中， W 代表流路径上的某个交换机， $S_{[a,b]}$ 表示流从发出节点 a 到接收节点 b ， W_{QS} 表示交换机 W 的队列缓存大小。

(5) 链路时隙约束。相邻交换机之间的时隙必须相等，并且网络中通过同一物理链路路由的两个帧不可能在时域上重叠。约束如式(5)所示

$$\begin{aligned}
 &\forall f_{[a,b]} \in F_{[a,b]}, \forall \alpha \in [0, hp_i^j / s_i \cdot T - 1], \\
 &\forall \beta \in [0, hp_j^i / s_j \cdot T - 1]: \\
 &(f_{[a,b]} \cdot \phi + \alpha \cdot f_{[a,b]} \cdot T \\
 &\geq f_{[a,b]} \cdot \phi + \beta \cdot f_{[a,b]} \cdot T) \\
 &\cup (f_{[a,b]} \cdot \phi + \beta \cdot f_{[a,b]} \cdot T \\
 &\geq f_{[a,b]} \cdot \phi + \alpha \cdot f_{[a,b]} \cdot T + f_{[a,b]} \cdot L) \quad (5)
 \end{aligned}$$

其中, α 表示流 S_i 的周期, β 表示流 S_j 的周期, T 表示最坏端到端时延, hp 表示流 S_i 和流 S_j 周期的最小公倍数。

(6)可填充帧大小约束。填充进奇偶队列的非时敏帧切片后的长度必须小于剩余可供调度的时隙大小。约束如式(6)所示: 此约束只针对非时敏流量。

$$\begin{aligned}
 &\forall s_i \in S, \forall [s_a, s_b] \in s_i: \\
 &(BE_{sl} \cap s_{[a,b]}) \leq B_{re} \quad (6)
 \end{aligned}$$

综上, 此问题最终转变为满足上述约束条件的最优化问题, 接下来的算法分析会针对此问题进行进一步的求解。

4 基于模拟退火的启发式算法

普通的贪思想能完成优化目标的求解, 但是此方法容易陷入局部最优解, 难以得到全局的最优解。为了避免陷入局部最优解, 本方案通过模拟退火算法的思想来求解约束条件下的最优解。模拟退火算法(Simulated Annealing, SA)最早是由Metropolis等人^[16]于1953年提出的。模拟退火算法是通过赋予搜索过程一种时变且最终趋于零的概率突跳性, 从而有效避免陷入局部极小并最终趋于全局最优的串行结构的优化算法^[17]。SA-TSA算法的流程如图6所示。

基于模拟退火的思想解决多约束优化问题, 调度算法如算法1所示。

此算法先通过流和约束的双重循环, 计算次数为 $5n$, 再进行模拟退火思想的 m 次迭代, 所以最终计算次数为 $m \times 5n$, 由于迭代次数 m 为常数, 所以最终的算法时间复杂度为 $O(n)$ 。

5 仿真与结果分析

5.1 仿真设置

5.1.1 网络拓扑设置

工业控制网络主要基于线型、环型、星型3种拓扑结构, 如图7所示, 本方案的仿真采取线型和环型拓扑。线型拓扑结构中所有设备连接到一条连接介质上, 每个节点上的网络接口板硬件均具有收、发功能。环型拓扑结构是各个节点形成一个闭合环, 环形网中各节点通过环路接口连在一条首尾相连的闭合环形通信线路中, 环上任何节点均可请求发送信息。在仿真实验中两种拓扑结构的最大跳数都为6。

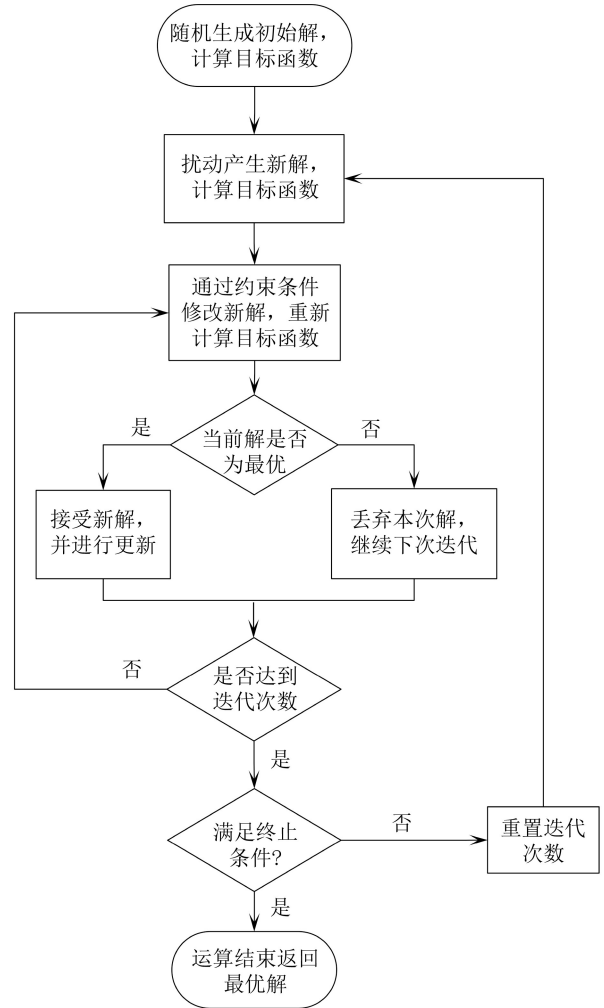


图6 SA-TSA算法流程

5.1.2 流设置

时延敏感流量设置为周期为定长的循环流量模型, 语音流量或视频流量通常用马尔可夫调制泊松过程来描述^[18], 针对非时敏流量本方案采用马尔可夫调制泊松分布模型。

5.1.3 资源设置

实验仿真对资源的分析主要分为以下几个方面: 链路的最大带宽, 交换机中奇偶队列的长度, 非时延敏感队列长度, 奇偶队列的时隙大小。为了满足绝大部分工业场景的流量传输需求, 本方案将链路的最大带宽设置为1000 Mbit/s。奇偶队列由于其传输时延敏感流的特殊性, 不能将缓存设置得太大, 仿真设置为1 MByte。奇偶队列的时隙大小至少要大于时敏流量的周期的最小公倍数, 仿真时设置为250 μ s。仿真的主要参数设置如表2所示。

5.2 评价指标

根据本方案解决的主要问题, 评价指标分为如下两个方面:

- (1)时敏流量端到端最坏时延

算法1 SA-TSA 算法

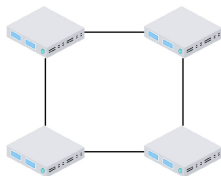
```

输入:  $G$  //网络拓扑
       $F$  //流资源设置
       $Q$  //队列资源设置
       $R_i$  //流约束设置,  $i \in [1,5]$ 
输出:  $F_{\text{offset}}$  //流偏移量
 $F_{\text{cur}} \leftarrow \{G, F, R\}_{\text{init}}$ ; //通过输入初始化当前解
for ( $i=0; i < N; i++$ ) //循环每一个流
  for ( $j=1; j \leq 5; j++$ ) //循环每一个约束条件
    if  $F_i$  contain  $R_j$  //判断是否满足约束条件
       $Q_{\text{cur}} \leftarrow Q_{\text{fresh}}$ ; //满足则更新可用资源数
      Continue;
    else
      while  $F_i$  contain  $R_j$  //不满足则进行偏移循环
         $F_{\text{cur}} \leftarrow F$  do offset;
      end while;
    end if
  end for
end for
while  $\text{num}_{\text{cur}} < \text{num}_{\text{itera}}$  AND  $F_i$  contain  $R_i$  //迭代
   $F_{\text{new}} \leftarrow SA \text{ random}()$  compute; //扰动产生新解
  if  $F_{\text{new}}$  better than  $F_{\text{cur}}$  //对最优解进行更新
     $F_{\text{cur}} = F_{\text{new}}$ ;
  end if
   $\text{num}_{\text{cur}}++$ ; //当前迭代次数加1
end while
return  $F_{\text{cur}}$ ; //完成迭代, 输出最优解

```



(a) 线型拓扑



(b) 环型拓扑

图7 仿真拓扑图

$$T_{\text{su}} = T_{\text{se}} + T_{\text{sp}} + T_{\text{d}} + T_{\text{q}} \quad (7)$$

其中, T_{su} 表示时敏流量端到端最坏总时延, T_{se} 表示发送时延, T_{sp} 表示传播时延, T_{d} 表示处理时延, T_{q} 表示排队时延。

计算所有调度流量的端到端时延的最坏情况,即时延的最大值。可以通过时敏流量最坏端到端时延大小反映TSN调度机制的优劣。

(2)带宽利用率

表2 仿真参数

参数	分布	取值
链路带宽(Mbit/s)	-	1 000
网络负载	-	[0.1,1]
帧数量	-	5×10^4
帧大小(Byte)	均匀分布	[64,500]
帧到达过程	泊松分布	500
奇偶时隙大小(μs)	-	250
队列长度(MByte)	-	1

$$\eta = \frac{U_{\text{real}}}{U_{\text{max}}} \quad (8)$$

带宽利用率 η 是衡量算法能否保障网络链路负载均衡的重要指标^[17],是指单位时间内实际通过的数据量与链路带宽之比。

5.3 对比分析

对比分析分为两个部分,一部分是在TSA-CQF机制上的SA算法与其他算法在最坏时延上的比较,另一部分是TSA-CQF机制与CQF机制在带宽利用率上的比较。网络负载反映了当前网络的负载大小,数值上为吞吐量与网络带宽的比值,该比值越接近1,则表示当前网络的负载大小越接近网络提供带宽的极限,达到网络饱和状态。过大的网络负载反映了当前网络承受着巨大的带宽分配压力,同时也给调度机制的运行带来挑战。分析调度机制在不同网络负载下的端到端最坏时延,能够得到调度机制处理不同大小数据传输量下的能力。为了证明SA-TSA的有效性,将本文提出的SA-TSA算法与其他两个算法进行了比较:Native-TSA算法、Greedy-TSA算法。这两种算法也是基于TSA-CQF机制实现的。Native-TSA算法是在报文生成后立即发送报文,没有进行过任何调度策略上的优化。而Greedy-TSA算法是基于贪心思想进行调度策略优化的算法,没有进行优化的迭代,容易陷入局部的最优解。

5.3.1 端到端最坏时延分析

图8显示了Native-TSA, Greedy-TSA, SA-TSA 3种算法端到端最坏时延与网络负载的关系。可以看出,随着网络负载的增大,端到端的最坏时延都逐步上升。其中Native-TSA由于没有进行调度的优化,端到端的最坏时延最大,Greedy-TSA由于采用了贪心的思想进行优化,得到相较于Native-TSA更低的端到端最坏时延,而SA-TSA通过其升温退火机制进行迭代,能得到相较于Greedy-TSA更优的解,其端到端的最坏时延也是3种算法中最低的。

图9显示了时敏流和非时敏流不同占比对时敏

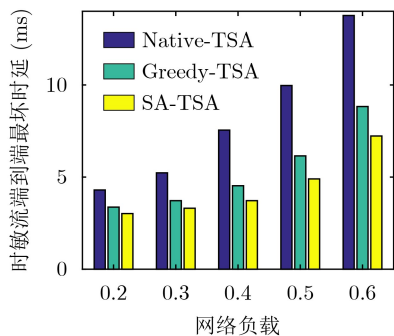


图8 不同网络负载下时敏流量端到端最坏时延

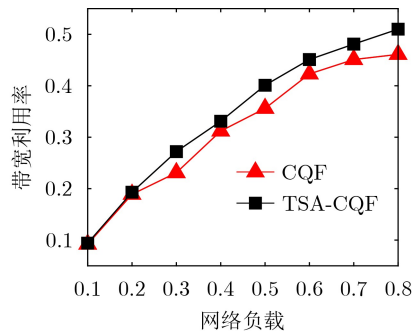


图10 不同网络负载下带宽利用率

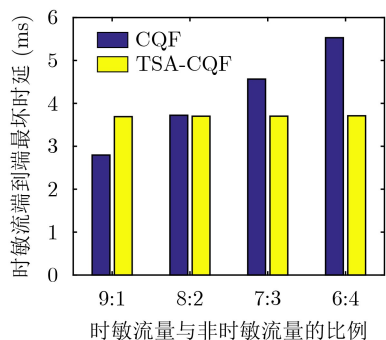


图9 不同流量占比对时敏流量最坏时延的影响

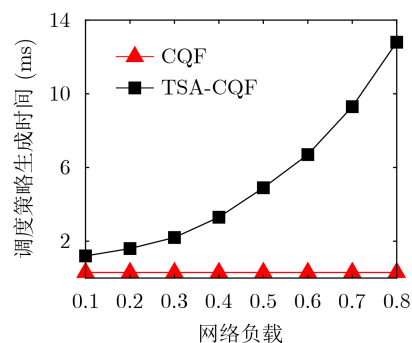


图11 网络负载与流调度结果生成时间的关系

流量的端到端最坏时延的影响。从图中可以看出,在时敏流量占比较大时, TSA-CQF机制时敏流端到端最坏时延比普通CQF机制要高,原因是TSA-CQF机制需要通过剩余时隙传输非时敏流量,固定时隙设置的比普通CQF机制的时隙大,导致最坏端到端时延增大。

但是随着非时敏流的比重增大,普通CQF机制的时敏流最坏时延随着非时敏流占比的增大而增大,而TSA-CQF机制的时敏流最坏时延没有明显的变化,趋于一个稳定的值。

5.3.2 带宽利用率分析

图10显示了普通的CQF和采用了时隙感知的TSA-CQF两种模型网络负载和带宽利用率的关系。从图中可以看到,总体的趋势是随着网络负载的增大,带宽利用率逐步增加。但是由于TSA-CQF采用了时隙感知的插入方法,对于固定的CQF的时隙来说,每个时隙的带宽利用率提高,整体的带宽利用率也随之提高,相较于CQF带宽利用率平均提高了11.29%,所以TSA-CQF的带宽利用率相较于CQF有明显提升。

5.3.3 调度策略生成时间分析

图11显示了普通的CQF和采用了时隙感知的TSA-CQF两种模型网络负载和调度策略生成耗时的关系。从图片可以清晰地看出,普通的CQF机制模型由于没有采用任何调度策略生成的方法,在流量产生就开始发送,所以其在耗时上一不变且

非常低。但是TSA-CQF由于其要进行复杂的调度策略计算,随着网络负载的增大,计算时间会越来越大。本文采用离线方式计算与生成调度策略,以减少实时调度产生的时延。

6 结论

本文提出一种时隙感知的循环排队与转发机制,简称TSA-CQF。TSA-CQF机制通过在奇偶队列的基础上额外增加一个出队整形,在入队口以是否时敏流量为标准进行分类,然后将非时敏流量在出端口插入奇偶队列的空闲时隙部分,最后通过条件约束以模拟退火的思想进行优化求解得到对应的调度策略,完成混合时敏流量和非时敏流量的调度。结果表明,在时敏流量和非时敏流量混合的智能工厂环境中,与传统的CQF相比, TSA-CQF在牺牲一定的调度策略生成时间的前提下,能明显提高带宽利用率并且降低最坏端到端时延。

参考文献

- [1] NIKISHIN K and KONNOV N. Schedule time-triggered ethernet[C]. 2020 International Conference on Engineering Management of Communication and Technology, Vienna, Austria, 2020: 1-5. doi: [10.1109/EMCTECH49634.2020.9261540](https://doi.org/10.1109/EMCTECH49634.2020.9261540).
- [2] LI Ziyang, ZHANG Yiming, ZHAO Yunxiang, et al. Efficient semantic-aware coflow scheduling for data-parallel

- jobs[C]. 2016 IEEE International Conference on Cluster Computing, Taipei, China, 2016: 154–155. doi: [10.1109/CLUSTER.2016.82](https://doi.org/10.1109/CLUSTER.2016.82).
- [3] LI Ziyang, ZHANG Yiming, ZHAO Yunxiang, *et al.* Best effort task scheduling for data parallel jobs[C]. 2016 ACM SIGCOMM Conference, Florianopolis, Brazil, 2016: 555–556. doi: [10.1145/2934872.2959047](https://doi.org/10.1145/2934872.2959047).
- [4] FINN N. Introduction to time-sensitive networking[J]. *IEEE Communications Standards Magazine*, 2018, 2(2): 22–28. doi: [10.1109/MCOMSTD.2018.1700076](https://doi.org/10.1109/MCOMSTD.2018.1700076).
- [5] SAHOO S, BAO Ninghai, BIGO S, *et al.* Deterministic dynamic network-based just-in-time delivery for distributed edge computing[C]. 2020 European Conference on Optical Communications, Brussels, Belgium, 2020: 1–4. doi: [10.1109/ECOC48923.2020.9333219](https://doi.org/10.1109/ECOC48923.2020.9333219).
- [6] IEEE. IEEE std 802.1QchTM-2017 IEEE standard for local and metropolitan area networks - bridges and bridged networks - amendment 29: Cyclic queuing and forwarding[S]. New York: IEEE, 2017.
- [7] IEEE. IEEE Std 802.1QbvTM-2015. IEEE standard for local and metropolitan area networks - bridges and bridged networks - amendment 25: Enhancements for scheduled traffic[S]. New York: IEEE, 2016.
- [8] GARDINER E. The Avnu alliance theory of operation for TSN-enabled industrial systems[J]. *IEEE Communications Standards Magazine*, 2018, 2(1): 5. doi: [10.1109/MCOMSTD.2018.8334911](https://doi.org/10.1109/MCOMSTD.2018.8334911).
- [9] NASRALLAH A, THYAGATURU A S, ALHARBI Z, *et al.* Ultra-Low Latency (ULL) networks: The IEEE TSN and IETF DetNet standards and related 5G ULL research[J]. *IEEE Communications Surveys & Tutorials*, 2019, 21(1): 88–145.
- [10] MESSENGER J L. Time-sensitive networking: An introduction[J]. *IEEE Communications Standards Magazine*, 2018, 2(2): 29–33. doi: [10.1109/MCOMSTD.2018.1700047](https://doi.org/10.1109/MCOMSTD.2018.1700047).
- [11] BIGO S, BENZAOUI N, CHRISTODOULOPOULOS K, *et al.* Dynamic deterministic digital infrastructure for time-sensitive applications in factory floors[J]. *IEEE Journal of Selected Topics in Quantum Electronics*, 2021, 27(6): 6000314. doi: [10.1109/JSTQE.2021.3093281](https://doi.org/10.1109/JSTQE.2021.3093281).
- [12] FINN N, LE BOUDEC J Y, MOHAMMADPOUR E, *et al.* DetNet bounded latency[EB/OL]. <https://datatracker.ietf.org/doc/draft-ietf-detnet-bounded-latency/>, 2022.
- [13] SEAMAN M. Paternoster policing and scheduling[EB/OL]. <https://grouper.ieee.org/groups/802/1/files/public/docs2017/cr-seaman-paternoster-policing-scheduling-0317-v03.pdf>, 2017.
- [14] YAN Jinli, WEI Quan, JIANG Xuyan, *et al.* Injection time planning: Making CQF practical in time-sensitive networking[C]. 2020 IEEE Conference on Computer Communications, Toronto, Canada, 2020: 616–625. doi: [10.1109/INFOCOM41043.2020.9155434](https://doi.org/10.1109/INFOCOM41043.2020.9155434).
- [15] HUANG Yudong, WANG Shuo, WU Binwei, *et al.* TACQ: Enabling zero-jitter for cyclic-queuing and forwarding in time-sensitive networks[C]. 2021 IEEE International Conference on Communications, Montreal, Canada, 2021: 1–6. doi: [10.1109/ICC42927.2021.9500914](https://doi.org/10.1109/ICC42927.2021.9500914).
- [16] WANG Guangjun, XU Caifeng, and LIU Gang. The transient electromagnetic inversion based on the simplex-simulated annealing algorithm[C]. The 37th Chinese Control Conference, Wuhan, China, 2018: 4321–4324. doi: [10.23919/ChiCC.2018.8484067](https://doi.org/10.23919/ChiCC.2018.8484067).
- [17] 谢维, 关嘉欣, 周游, 等. 基于改进模拟退火算法的登机口分配问题[J]. *计算机系统应用*, 2021, 30(5): 157–163. doi: [10.15888/j.cnki.csa.007903](https://doi.org/10.15888/j.cnki.csa.007903).
- XIE Wei, GUAN Jiabin, ZHOU You, *et al.* Gate distribution problem based on improved simulated annealing algorithm[J]. *Computer Systems & Applications*, 2021, 30(5): 157–163. doi: [10.15888/j.cnki.csa.007903](https://doi.org/10.15888/j.cnki.csa.007903).
- [18] YERA Y G, LILLO R E, NIELSEN B F, *et al.* A bivariate two-state Markov modulated Poisson process for failure modeling[J]. *Reliability Engineering & System Safety*, 2021, 208: 107318. doi: [10.1016/j.ress.2020.107318](https://doi.org/10.1016/j.ress.2020.107318).
- 蔡岳平: 男, 副教授, 研究方向为泛在确定性网络。
任志文: 男, 硕士生, 研究方向为时间敏感网络。
- 责任编辑: 马秀强