

面向车载时间敏感网络的时间感知最大匹配流量调度算法

郑凌^{1,2}, 魏国栋¹, 潘伟涛², 张科遥¹, 褚宏云¹

(1. 西安邮电大学通信与信息工程学院, 陕西 西安 710061;

2. 西安电子科技大学空天地一体化综合业务网全国重点实验室, 陕西 西安 710071)

摘要: 随着车联网技术的逐步应用, 对车内网络数据传输的实时性、确定性及可靠性提出了更高的要求。针对车载网络中数据转发不及时、分组传输时延大、调度算法复杂度高等问题, 提出了一种时间感知的最大匹配流量调度算法。该算法将输入排队的调度问题转化为具有截止期保障的最大流匹配的问题, 并逐时隙实现最大权重匹配调度。在调度集合中含有两种或多种截止期的调度任务中, 充分利用空闲链路和空闲时隙资源, 预推送截止期较大的分组。该算法有效地提高了交换系统的吞吐率, 降低了分组的平均时延。仿真结果表明, 所提算法相较于传统调度算法吞吐率性能提升了5%~15%、分组平均时延性能提升了15%~46%。

关键词: 车载时间敏感网络; 输入排队系统; 截止期保障; 最大匹配; 流量调度

中图分类号: TN915.08

文献标志码: A

doi: 10.11959/j.issn.2096-3750.2025.00431

Time-aware traffic scheduling algorithm based on maximum matching for in-vehicle time-sensitive networks

ZHENG Ling^{1,2}, WEI Guodong¹, PAN Weitao², ZHANG Keyao¹, CHU Hongyun¹

1. School of Communication and Information Engineering, Xi'an University of Posts and Telecommunications, Xi'an 710061, China

2. State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China

Abstract: With the application of the Internet of vehicles, the real-time, deterministic, and reliable data transmission for the in-vehicle networks become crucial. In response to the issues of untimely data forwarding, high packet transmission delay, and high complexity of scheduling algorithms in-vehicle time-sensitive networks, a time-aware traffic scheduling algorithm based on maximum matching was proposed. This algorithm transformed the input queuing scheduling problem into a maximum flow matching problem with deadline guarantees and achieved maximum weight matching scheduling on a per-slot basis. In scheduling sets containing tasks with two or more deadlines, idle link and slot resources were fully utilized to pre-push packets with larger deadlines. This effectively improved the throughput of the switching system and reduced the average packet delay. Simulation results demonstrate that compared with traditional scheduling algorithms, the proposed algorithm achieves 5%~15% throughput performance improvement and 15%~46% average packet delay performance improvement.

Key words: in-vehicle time-sensitive network, input queuing system, deadline guarantee, maximum matching, traffic scheduling

0 引言

物联网的蓬勃发展和人工智能与新工业革命的

深度融合推动了以数据为驱动的工业新生态的形成^[1-2]。先进辅助驾驶系统(ADAS, advanced driver assistance system)和自动驾驶的快速发展给汽车领

收稿日期: 2024-08-22; 修回日期: 2025-04-30

通信作者: 郑凌, lingzheng@xupt.edu.cn

基金项目: 国家自然科学基金资助项目(No. 62102314); 陕西省自然科学基金基础研究计划项目(No. 2021JQ-708, No. 2022JQ-635)

Foundation Items: The National Natural Science Foundation of China (No. 62102314), Natural Science Basic Research Program of Shaanxi Province (No. 2021JQ-708, No. 2022JQ-635)

域带来了巨大的变革^[3-4]。ADAS通过光探测和测距雷达、超声波探测、全球定位系统（GPS, global positioning system）、激光雷达、高清摄像头传感器和实时数据处理，提供了诸如自动紧急制动、车道保持辅助、自适应巡航控制等功能，大幅度提高了驾驶安全性和舒适性^[5]。自动驾驶技术更进一步，使车辆能够在无人驾驶或半自动驾驶模式下行驶，彻底改变了人们对交通出行的认知和体验^[6]。然而，随着技术的进步和应用场景的扩展，车联网领域面临着有限空间内密集传感器接入、通信网络数据传输的实时性与可靠性、数据安全与隐私保护等诸多挑战。

为了标准化车载以太网的发展并克服总线型网络带宽受限以及现有车载网络标准兼容性不佳等问题，IEEE 802.1 在传统车载以太网的基础上，提出了时间敏感网络（TSN, time sensitive network）标准，旨在满足未来智能汽车对实时、可靠、安全通信的需求^[7]。流量调度是车载 TSN 标准中的核心机制^[8]，车载 TSN 根据不同的车载业务需求，将传输的数据分为时间触发型（TT, time-triggered）流量、音视频桥接（AVB, audio/video bridging）流量、尽力而为型（BE, best effort）流量 3 种类型^[9]。其中，基于时间触发的安全关键型数据流量对时延要求极高，如果安全关键型数据流量没有在预定的截止期前成功传输，那么相关汽车控制指令将无法发挥作用^[10]，这严重影响了车辆的行驶安全。因此，深入研究车载网络中有截止期保障的分组调度技术具有重要意义。在系统线路负载较高的情况下，该技术可以确保车载网络能够稳定、可靠地完成数据流量的实时性传输，进而提高整体车辆系统的性能和安全性。

目前，针对车载时间敏感网络的流量调度，已有很多算法被提出，文献[11]利用逻辑约束形式化了一个调度问题，提出了一种基于可满足性模理论（SMT, satisfiability modulo theory）和混合整数规划的优化方法，该算法适用于作为车载网络的 TSN 以太网，但是该算法的求解速度较慢。文献[12]采用冯诺依曼算法来分解业务矩阵。但是，冯诺依曼算法只考虑了最大匹配，没有考虑截止期的约束。为了解决带有截止期约束的分组调度问题，通常采用最早截止期优先（EDF, earliest deadline first）算法或最小松弛度优先（MLF, minimum laxity first）

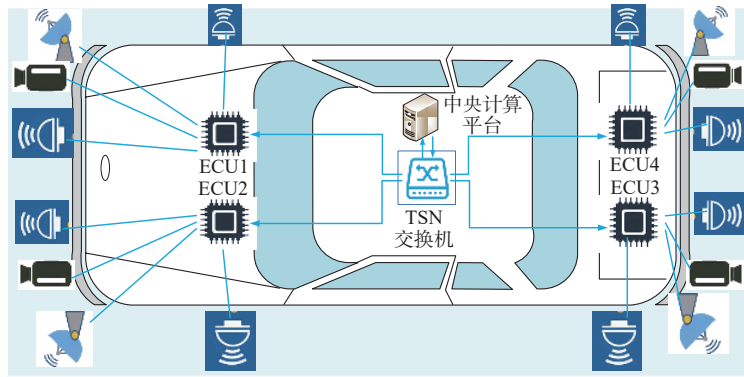
算法及其变体算法^[13-17]。文献[18]在传统 iSLIP 算法的基础上进行了扩展，提出了一种 MC-iSLIP 调度算法，该算法缓解了周期型业务流量对突发型业务流量的阻塞，从而降低了突发型流量的传输时延和抖动，不过，该算法仅对小规模交换进行了仿真实验，且仅适用于单播流量调度场景。文献[19]提出了一种嵌套周期流优化调度算法，该算法通过降维解耦，将多种截止期分组调度问题分解为仅含有两种和一种截止期的分组调度问题，并运用最优化求解器求出调度结果。文献[20-21]将遗传算法应用到输入排队交换系统中，利用遗传算法来求解分组调度中的最大流，以较高的吞吐量完成了仿真实验。文献[22]提出了一种更高效的算法，称为基于流的迭代分组调度（FIPS, flow-based iterative packet scheduling）算法，该算法将截止期较大集合中过载的分组预推送至截止期较小集合的空闲时隙中进行调度，在一定程度上提高了交换系统的交换效率。但是，该算法具有较高的时间复杂度，并且，在多个截止期约束的分组调度中，FIPS 算法要求业务矩阵必须满足可调度条件才能正常工作，因此，该算法对任意流量模型的普适性较低。

针对上述问题，本文提出了一种时间感知最大匹配流量调度算法（TASA, time-aware traffic scheduling algorithm based on maximum matching），当调度集合中仅存在一种截止期时，TASA 算法逐时隙调用最大权重匹配算法^[23]，保证在截止期内的每一个时隙中传输最多数量的分组。当调度集合中含有两种或多种截止期时，TASA 算法通过有上下界约束的最大流算法求出最大合法流，然后将截止期较大的分组预推送至截止期较小的分组的空闲时隙中调度，充分利用时隙资源。仿真结果表明，与传统的截止期保障调度算法相比，在相同条件下，TASA 算法能以更高的吞吐率和更小的平均时延完成车载 TSN 中有截止期保障的分组调度任务。

1 系统模型

1.1 车载时间敏感网络架构

随着汽车智能网联化的发展，未来的电子电气架构将基于区域控制器及中央计算平台，这将对车载网络的带宽、实时性和稳定性提出更高的要求^[24]。车载时间敏感网络架构模型^[25]如图 1 所示，

图1 车载时间敏感网络架构模型^[25]

主要由传感器、电子控制单元以及中央计算平台组成，传感器节点包括光探测和测距雷达、超声波传感器、GPS和高清摄像头等，这些传感器可以帮助汽车在驾驶过程中持续获得对周边道路环境的正确感知。传感器采集的数据信息将通过车载网络传输到汽车的4个电子控制单元（ECU, electronic control unit），ECU将采集和汇总的数据流量通过TSN交换机发送到中央计算平台（CCP, central computing platform）。中央计算平台对数据信息进行整合、分析、决策，并下发相应的控制指令，来完成车身姿态调整以及动力重新分配等功能。通常可将车载传感器采集的关键数据以及中央计算平台发送的控制指令映射为优先级最高的TT流量，将摄像头采集的视频流映射为BE流量。

1.2 车载TSN交换机模型

车载TSN交换系统模型如图2所示。TSN交换机有 N 个输入端口和 N 个输出端口，每个端口维护从Q0~Q7共8个不同优先级的队列。首先，交换系统在IEEE802.1AS标准下进行时钟同步，根据调度算法计算出流量调度表，并将其映射为门控列表。随后，在交换系统中的输入侧，利用IEEE802.1Qci标准进行逐流过滤和监管；利用IEEE802.1Qcr标准进行异步流量整形^[26]，有效地管理和控制不同类型的数据流量。在输出侧，根据不同优先级的流量特性，利用IEEE802.1Qch标准完成循环队列转发^[27]，或是IEEE802.1Qav基于信用整形机制以及IEEE802.1Qbv基于时间感知整形机制实现对实时关键型数据流的可靠性传输^[28]。最后，这些队列依据提前规划好的门控列表对业务流进行调度，门控列表中“1”表示门控开关处于打开状态，可以进行帧的传输；“0”表示门控开关处于关闭状态，不可以进行帧的传输。门控开关的状态按照门控列表

周期性地重复。最后，出队的分组通过优先级选择器以及IEEE802.1CB帧复制与消除协议进行冗余管理之后到达输出端口。

在该模型中，安全控制型流量，即TT流的传输优先级最高，按照预先规划的门窗窗口的传输时间点精确转发。AVB流和BE流的优先级较低，在TT流传输窗口的空隙中进行传输，以满足车辆安全控制型交互信息的传输实时性和确定性。由于控制TT流传输时间点的门控开关直接作用在交换系统的输出端口位置，如果输入端口的分组没能在门控开关打开的时间段内被交换到输出端口，则认为该分组调度失败，调度失败的分组将被丢弃。

车载交换机输入端口和输出端口的匹配可抽象为二分图模型，二分图匹配模型示意图如图3所示，二分图模型如图3(a)所示，该模型将交换机的输入、输出端口划分为两个不相交的子集 U 和 V ，设 $G=(V, E)$ ，其中，输入端口为 U 集合， $U=\{u_1, u_2, \dots, u_N\}$ ，输出端口为 V 集合， $V=\{v_1, v_2, \dots, v_N\}$ ， N 为交换机输入/输出端口数量。每个端口只属于其中一个子集。二分图匹配法为交换机的输入端口匹配唯一的输出端口^[29]，使各个端口可以进行无争用的分组传输，提高了交换系统的吞吐量，一种可行的匹配结果如图3(b)所示。

该二分图匹配问题可以通过构建业务矩阵，并调用最大流算法来解决。输入排队交换模型与业务矩阵的转换关系如图4所示，车载交换机的输入排队交换模型如图4(a)所示，分组到达输入端口后在队列中排队等待调度，每个分组均具有一个截止期 d_p ，只有在截止期到达前将分组调度至对应的输出端口，才能视为调度成功，否则该分组将被丢弃。可以将图4(a)中交换机输入端口的队列建模为图4(b)

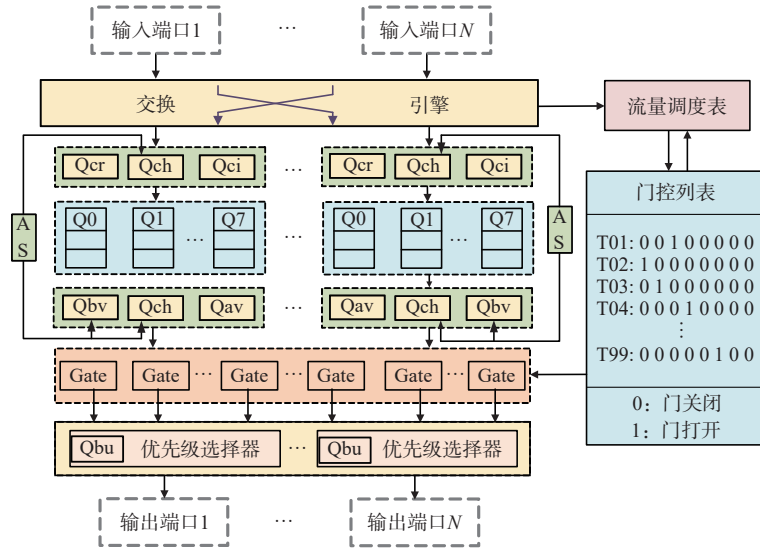


图2 车载TSN交换系统模型

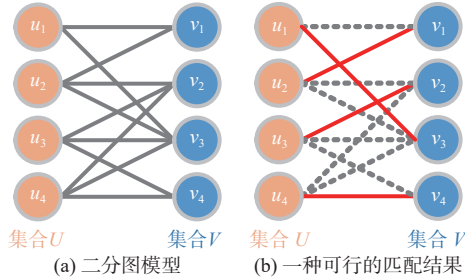


图3 二分图匹配模型示意图

中的业务矩阵 A 。业务矩阵 A 中， u_i ($1 \leq i \leq N$) 表示交换机的输入端口索引， v_j ($1 \leq j \leq N$) 表示交换机的输出端口索引。 $p_{i,j}^{d_p}$ 表示在输入端口 i 处等待被传输至输出端口 j 的分组数量。 $p_i^{d_p} = \sum_{j=1}^N p_{i,j}^{d_p}$ 表示在第 i 个输入端口等待调度的总分组数， $q_j^{d_p} = \sum_{i=1}^N p_{i,j}^{d_p}$ 表示发送往输出端口 j 的总分组数，上标 d_p 表示集合中分组的截止期。

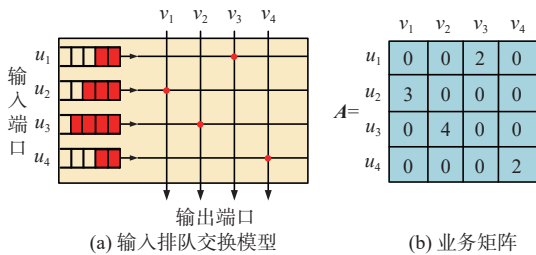


图4 输入排队交换模型与业务矩阵的转换关系

定义1 设业务矩阵输入端口 i 的过载度为 a_i ， $a_i = p_i^{d_p} - (d_p + 1)$ 。输出端口 j 的过载度为 b_j ， $b_j = q_j^{d_p} - (d_p + 1)$ 。当 $a_i > 0$ 或 $b_j > 0$ 时，表示业务矩阵

输入端口 i 或输出端口 j 有过载现象。当 $a_i < 0$ 或 $b_j < 0$ 时，则表示业务矩阵各端口仍有空位。

定义2 设业务矩阵输入端口 i 的空余度为 c_i ， $c_i = (d_p + 1) - p_i^{d_p}$ 。输出端口 j 的空余度为 e_j ， $e_j = (d_p + 1) - q_j^{d_p}$ 。当 $c_i > 0$ 或 $e_j > 0$ 时，表示业务矩阵输入端口 i 或输出端口 j 有空位，反之则表示业务矩阵各输入、输出端口过载。

1.3 问题建模

基于以上车载TSN输入排队系统交换模型，研究有截止期保障的业务流调度算法的目标是根据输入业务流量的业务矩阵，通过流量调度算法，在有截止期约束的条件下，使尽可能多的分组完成调度，最大化车载交换机吞吐率。为了完成这个调度任务，本文提出以下假设，并将它们作为设计和实施算法的基础。

假设1 时间轴被划分为等间隔的时隙，并且每个时隙的长度等于一个分组发送或接收所需的时间。时隙的序号从0开始。

假设2 设在每个时隙中，交换机一对一建立从输入端口到输出端口的连接，每个输入端口或输出端口在一个时隙中最多可以发送或接收一个分组，并且分组只能在时隙开始时发送或接收。假设所有分组都有固定的大小。

假设3 设每个分组可用三元组 $[i, j, d_p]$ 表示，其中， d_p 表示分组的截止期。设截止期集合为 $D = \{d_1, d_2, d_3, \dots, d_p\}$ ，且 $d_1 < d_2 < d_3 < \dots < d_p$ 。有截止期保障意味着分组必须在截止期到来前从输入端

口 i 传输到输出端口 j , 错过截止期的分组将被丢弃。

假设 4 设截止期越小的分组具有越高的优先级, 反之, 优先级越低。交换系统会优先调度较小截止期的分组集合, 然后调度较大截止期的集合, 最终完成所有分组集合的调度。

定理 1 对于分组集合 S , 对应的截止期集合为 $D = \{d_1, d_2, d_3, \dots, d_p\}$, 且 $d_1 < d_2 < d_3 < \dots < d_p$ 。当满足以下条件时, 集合可称为非过载的

$$\begin{cases} \sum_{i=1}^m p_i^{d_i} \leq d_m + 1, \\ \sum_{j=1}^m q_j^{d_j} \leq d_m + 1 \end{cases} \quad (1)$$

$\forall i, j \in \{1, 2, \dots, N\}, \forall m \in \{1, 2, \dots, p\}$ 。

定理 1 很容易被证明, 因为 $\sum_{i=1}^m p_i^{d_i}$ 表示截止至时隙 d_m 到达第 i 个输入端口的总分组数, $\sum_{j=1}^m q_j^{d_j}$ 表示截止至时隙 d_m 发送至第 j 个输出端口的总分组数, 而 d_m+1 表示可以使用的总时隙数量。当到达任意一个输入端口或发送至任意输出端口的总分组数小于或等于总的可用时隙数时, 交换系统的负载不超过 1, 称系统是非过载的, 否则, 交换系统处于过载状态。当系统为非过载状态时, 存在一个可行解使所有分组满足截止期约束。而当系统处于过载状态时, 则一定存在丢包。

根据上述对研究目标、假设以及定理的分析, 车载 TSN 输入排队交换系统中有截止期保障的业务流调度问题可以描述为式(2)

$$\max \sum_{i=0}^{d_p} \sum_{j=1}^N \sum_{i=1}^N p_{i,j}(t) \quad (2)$$

在任何时隙中输入端口和输出端口之间最多可以传输一个分组, 因此, 对于任何时隙, 需满足以下约束条件, 如式(3)~式(5)所示

$$p_{i,j}(t) = \{0, 1\}, \forall i, j \in \{1, 2, \dots, N\} \quad (3)$$

$$\sum_{j=1}^N p_{i,j}(t) \leq 1, \forall i \in \{1, 2, \dots, N\} \quad (4)$$

$$\sum_{i=1}^N p_{i,j}(t) \leq 1, \forall j \in \{1, 2, \dots, N\} \quad (5)$$

其中, $p_{i,j}(t)$ 表示在时隙 t 中调度的总分组数, i 和 j 分别表示交换机输入端口、输出端口的序号索引。

通过对上述的调度模型进行分析, 得出求解该问题的主要思路如下。首先, 根据业务流量的输入端口、输出端口、截止期、分组数量等参数建立输入业务矩阵。然后, 依据调度集中分组的截止期

数量分类讨论。若调度集中所有分组的截止期都相同, 则将其定义为单类分组调度问题, 可以利用最大权重匹配算法分解业务矩阵来进行求解; 若调度集中分组具有两种或两种以上的截止期, 则将其定义为两类或多类分组调度问题, 此时可充分利用业务矩阵的空余度, 提前调度截止期较大的分组, 从而最大化交换系统的吞吐率, 降低转发时延。

2 算法设计

2.1 单类分组调度

单类分组调度指在调度集中, 所有分组有相同的截止期 d_1 。通过在 $[0, d_1]$ 中逐时隙调用最大权重匹配算法来求出最大流, 对业务矩阵进行分解, 保证在每个时隙中, 使尽可能多的分组在截止期到来以前完成调度, 从而得到最大吞吐量的调度结果。单类分组调度算法如算法 1 所示, 单类分组调度算法的时间复杂度取决于步骤 2) 的二分图最大权重匹配, 该算法的复杂度为 $O(N^3)$ 。

算法 1 单类分组调度算法 Single-Class-Scheduling(A)

输入 业务矩阵 A

输出 时隙 i 的匹配矩阵 M_i

for $i = 0:d_1$ **do**

调用最大权重匹配算法 $M_i = \text{Maximum-Weight-Matching}(A)$;

输出时隙 i 的匹配矩阵 M_i ;

更新业务矩阵 $A = A - M_i$;

end for

2.2 两类分组调度

如果总分组调度集合 S 中有两种不同截止期 $D = \{d_1, d_2\}$ 并且满足 $0 < d_1 < d_2$, 将截止期为 d_1 的分组构建为业务矩阵 A , 截止期为 d_2 的分组构建为业务矩阵 B , 则称这种情况为两类分组调度。两类调度算法的思路是优先调度截止期较小的分组集合, 如果截止期较小的分组在调度过程中有空闲时隙, 则可以将截止期较大的部分分组预推送至这些空闲时隙中进行调度。两类分组调度的核心是预推送算法, 预推送算法如算法 2 所示。

算法 2 预推送算法 Promote-algorithm(A, B)

输入 调度矩阵 A, B

输出 预推送矩阵 B_x

$c_i = (d_1 + 1) - p_i^{d_1}$, $e_j = (d_1 + 1) - q_j^{d_1}$; //计算

矩阵 A 的空余度

$$a_i = p_i^{d_2} - (d_2 - d_1), \quad b_j = q_j^{d_2} - (d_2 - d_1); \quad // \text{计算}$$

矩阵 B 的过载度

构建四级流网络 $G(s, t, U, V, E)$;

连接源节点 s 和输入端口 u_i ($1 \leq i \leq N$) 的边;

if $0 < a_i < c_i$

$$\text{下界 } l_{si} = \max \{0, a_i\};$$

else

$$\text{下界 } l_{si} = 0;$$

end if

上界 $u_{si} = c_i$;

连接 u_i 和 v_j ($1 \leq i, j \leq N$) 的边;

下界 $l_{ij} = 0$, 上界 $u_{ij} = p_{i,j}^{d_2}$;

连接 v_j ($1 \leq j \leq N$) 和 t 的边;

if $0 < b_j < e_j$

$$\text{下界 } l_{jt} = \max(0, b_j);$$

else

$$\text{下界 } l_{jt} = 0;$$

end if

上界 $u_{jt} = e_j$;

调用上下界最大流算法求出最大可行流

\max_flow ;

根据最大可行流 \max_flow 得出预推送矩阵 B_x

(从矩阵 B 预推送至矩阵 A 的分组集合)

算法2步骤3)中构建的四级流网络 $G(s, t, U, V, E)$

包含5个参数。其中, 源点 s 为流网络的第一级。

第二级流网络存在一个包含 N 个节点的集合 $U, U =$

$\{u_1, u_2, \dots, u_N\}$, 其中, u_i 表示第 i 个输入端口。

第三级流网络存在一个包含 N 个节点的集合 $V, V =$

$\{v_1, v_2, \dots, v_N\}$, 其中, v_j 表示第 j 个输出端口。汇点

t 为流网络的第四级。集合 E 表示两节点之间的边,

给集合 E 中的每一条边都分别设置下边界和上边界。

步骤5)~9)中设置上下边界的目的是保证从矩阵 B

输入端口 i 预推送到矩阵 A 输入端口调度的分组数

量不超过矩阵 A 空余度 c_i 的前提下, 尽可能确保矩

阵 B 各输入端口 i 不出现过载的情况。步骤10)和步

骤11)意味着可以从 B 类分组 $p_{ij}^{d_2}$ 集合中选择部分分

组预推送到矩阵 A 的空余时隙中调度。步骤13)~

17)的含义是在矩阵 B 的 j 端口, 最多有 e_j 个分组会

被提升到矩阵 A 与 A 类分组一起传输调度。

业务矩阵 A, B 如图5所示, 为了更清晰地解释

算法2的工作过程, 以业务矩阵 A, B 为例。假设 A 和 B 为两个 4×4 的业务矩阵, 其中, A 类分组的截止期 $d_1 = 9$, B 类分组的截止期 $d_2 = 14$ 。可以看出矩阵 A 的出端口和入端口均存在空余度, 而矩阵 B 的出端口和入端口存在过载度。可调用预推送算法将 B 类矩阵中的流量推进至 A 类矩阵进行调度。由业务矩阵 A, B 构建的流网络如图6所示, 最大可行流如图7所示, 为调用最大流算法求出的最大合法流, 这意味着有7个 B 类分组将会从 $[d_1 + 1, d_2]$ 时隙中预推送到 A 类矩阵中并与 A 类分组一起在 $[0, d_1]$ 时隙中调度。更新后的业务矩阵 A, B 如图8所示。

	v_1	v_2	v_3	v_4	$P_i^{d_1}$	c_i
u_1	0	2	0	3	5	5
u_2	0	2	1	0	3	7
u_3	0	2	1	2	5	5
u_4	0	0	3	1	4	6
$q_j^{d_2}$	0	6	5	6		
e_j	10	4	5	4		

	v_1	v_2	v_3	v_4	$P_i^{d_2}$	a_i
u_1	2	0	1	1	4	-1
u_2	2	3	0	3	8	3
u_3	3	2	3	0	8	3
u_4	1	3	0	2	6	1
$q_j^{d_2}$	8	8	4	6		
b_j	3	3	-1	1		

图5 业务矩阵 A, B

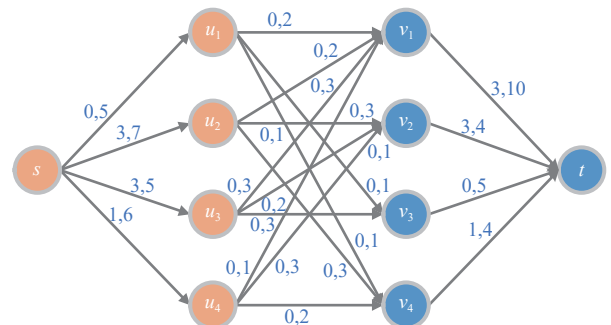


图6 由业务矩阵 A, B 构建的流网络

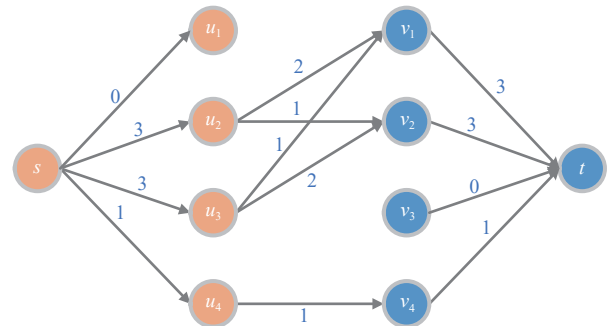


图7 最大可行流

预推送算法的时间复杂度取决于具有上下界的最大流算法的复杂度, 常见的有源汇有上下界最大流算法的复杂度为 $O(N^2 \times E)$, 其中, N 是节点数量, E 是边数量。

基于预推送算法的两类分组调度算法如算法3

	v_1	v_2	v_3	v_4	$p_i^{d_i}$	c_i
u_1	0	2	0	3	5	5
u_2	2	3	1	0	6	4
u_3	1	4	1	2	8	2
u_4	0	0	3	2	5	5
$q_j^{d_i}$	3	9	5	7		
e_j	7	1	5	3		

	v_1	v_2	v_3	v_4	$p_i^{d_i}$	a_i
u_1	2	0	1	1	4	-1
u_2	0	2	0	3	5	0
u_3	2	0	3	0	5	0
u_4	1	3	0	1	5	0
$q_j^{d_i}$	5	5	4	5		
b_j	0	0	-1	0		

图8 更新后的业务矩阵 A, B

所示，主要思路是循环调用预推送算法，直到找不到合法的可行流。

算法3 基于预推送算法的两类分组调度算法 Two-Class-Scheduling(A, B)

输入 业务矩阵 A, B

输出 时隙 i 的匹配矩阵 M_i

while any($c_i > 0$) || any($e_j > 0$)

$B_x = \text{Promote-algorithm}(A, B)$; //调用预推送

算法

更新 A, B 业务矩阵: $A = A + B_x, B = B - B_x$;

if 找不到合法的可行流

break;

end if

end while

在时隙 $[0, d_1]$ 中调用 Single-Class-Scheduling(A),

在时隙 $[d_1 + 1, d_2]$ 中调用 Single-Class-Scheduling(B);

由于两类分组调度算法需要多次调用预推送算法，迭代次数与网络流的节点数成正比，因此时间复杂度为 $O(N^3 \times E)$ 。

2.3 多类分组调度

多类分组调度，即在总的分组调度集合 S 中，存在 3 个或 3 个以上不同截止期的分组调度问题。多类分组调度问题是两类分组调度问题的延伸，可以将多类调度问题分解为多个具有两个截止期的调度问题，通过循环调用两类分组调度算法并更新业务矩阵的方式来解决。以三类分组调度问题为例，首先按照截止期从小到大将集合中的分组分为高优先级 S_1 ，次高优先级 S_2 ，以及低优先级 S_3 这 3 个类别的业务矩阵，先对优先级较高的矩阵 S_1 和 S_2 调用预推送算法，当 S_2 中再也找不到可以预推送的分组，同时集合 S_1 输入、输出端口仍然存在空余度时，再对 S_1 和 S_3 调用预推送算法，直到 S_1 输入、输出端口空余度 $c_i, e_j = 0$ ，或预推送算法找不到最大合法流，此时在 $[0, d_1]$ 时隙中调用最大权重匹配算法分解业务矩阵 S_1 。接着将 S_2 作为高优先级矩阵，

S_3 作为次高优先级矩阵，重复以上过程，在 $[d_1 + 1, d_2]$ 时隙中调度 S_2 。最后，在 $[d_2 + 1, d_3]$ 时隙中调度 S_3 。多类分组调度算法如算法 4 所示。

算法4 多类分组调度算法 Multi-Class-Scheduling($S_1, S_2, S_3, \dots, S_p$)

输入 业务矩阵 $S_1, S_2, S_3, \dots, S_p$ 以及 $D = \{d_1, d_2, d_3, \dots, d_p\}$

输出 时隙 i 的匹配矩阵 M_i

令 $d_0 = -1$;

计算矩阵 S_1 的空余度;

计算矩阵 S_2 的过载度;

for $l = 1:p - 1$ **do**

while 矩阵 S_l 空余度大于 0

for $t = l + 1:p$ **do**

$B_x = \text{Promote-algorithm}(S_t, S_l)$; //调用预

推送算法

if 找不到可行流

continue;

else

$S_l = S_l + B_x$; //更新矩阵 S_l

$S_t = S_t - B_x$; //更新矩阵 S_t

更新矩阵 S_l 的空余度，矩阵 S_t 的过

载度;

end if

end for

if 找不到可行流且所有矩阵都已迭代

break;

end if

end while

在时隙 $[d_{l-1} + 1, d_l]$ 中调用 Single-Class-Scheduling(S_l);

end for

在时隙 $[d_{p-1} + 1, d_p]$ 调用 Single-Class-Scheduling(S_p);

多类分组调度算法的总体时间复杂度可近似为 $O(p^2 \times N^3 \times E)$ ，因为多类分组调度算法可以视为循环调用两类分组调度算法，最多执行 p^2 次。

3 仿真及分析

为了验证所提算法的性能，本文构建了包含一个车载 TSN 核心交换节点及多个数据终端节点的网络仿真场景。终端节点以固定周期产生定长的

分组用于模拟车载传感器的数据采集，并将车载TSN交换机转发处理每个定长分组的时间记为一个时隙的长度。仿真实验在MATLAB环境中进行。为了测试所提算法在大规模网络场景中的性能，取核心交换节点的端口密度范围为 $N=[4,128]$ ，并在此基础上随机生成具有不同截止期类别的业务矩阵进行实验，每种类别分别取等间隔分布与不等间隔分布共3种不同情况，以验证所提算法对不同流量场景的适应性。每次实验均进行10次独立仿真，取平均值以确保结果的稳定性和准确性。仿真参数的设置见表1。

表1 仿真参数设置

参数名称	参数取值	
交换机节点数	1个	
ECU终端节点数	8~256个	
交换机端口数量 N	4~128个	
线路负载load	0.5~1.0	
调度集合截止期类别 $ D $	$ D =1, 2, 3$	
截止期取值	$ D =1$	情况1: $d_1=36$
		情况2: $d_1=57$
		情况3: $d_1=109$
	$ D =2$	情况1: $d_1=25, d_2=50$
		情况2: $d_1=82, d_2=124$
		情况3: $d_1=162, d_2=244$
	$ D =3$	情况1: $d_1=50, d_2=70, d_3=90$
		情况2: $d_1=72, d_2=146, d_3=182$
		情况3: $d_1=144, d_2=290, d_3=361$

本文模拟了在不同交换规模、不同截止期数量以及不同线路负载的条件下，不同调度算法的性能表现^[30]，对比方法包括输入排队交换的经典算法iSLIP，在实时调度领域的典型算法EDF-RR、MLF-RR、EDF-SDR、MLF-SDR^[13]，以及FIPS算法^[22]。具体性能指标包括调度成功率、丢包率、分组平均时延以及吞吐率。

3.1 基于交换规模的仿真实验

首先，本文研究了交换规模，即交换机端口数量对算法性能的影响，在本次实验中，交换规模 N 的取值范围设定为4~128，以模拟评估不同网络流量规模下的算法性能表现。单类调度中成功率与交换规模的关系如图9所示，其中，调度成功率定义

为调度算法能够在截止期的约束内成功调度所有分组的实验次数与总实验次数的比值。由图9可见，随着交换规模的增大，分组调度成功率呈现出明显的下降趋势。这是由于交换规模 N 的增加导致业务矩阵中待调度分组的数量以平方级增长，进而显著地提升了调度决策的复杂性，使调度算法难以在规定的截止期内成功调度所有分组，导致调度成功率逐步下降。本文所提的TASA算法能够尽可能充分利用空闲时隙，因此在不同交换规模下均表现出较为优异的调度性能。在 $N=16$ 的情况下，调度成功率能够达到100%，即使在 $N=128$ 的大规模交换条件下，仍能维持约40%的调度成功率，高于对比算法。

本文进一步探讨了分组平均时延与交换规模之间的关系，其中，时延为分组在交换机内部等待转发的排队时延，以时隙为单位。这里分别针对业务矩阵具有不同的截止期数量进行了仿真验证。单类调度中分组平均时延与交换规模的关系如图10所示，在单类调度任务中，业务矩阵具有相同的截止期。随着交换规模的增大，调度分组集合的总时延

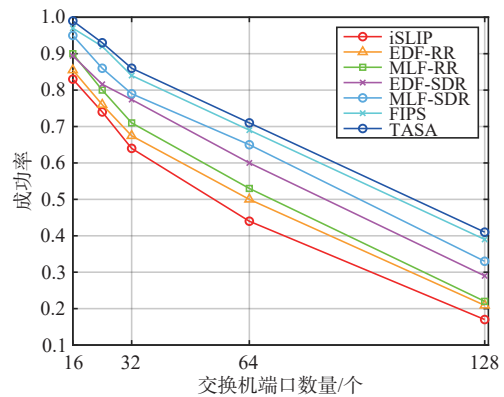


图9 单类调度中成功率与交换规模的关系

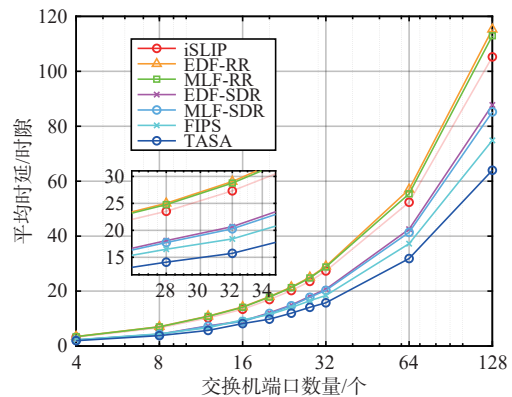


图10 单类调度中分组平均时延与交换规模的关系

呈显著增长趋势，因而分组平均时延与交换规模呈正相关。当交换规模较小时，iSLIP、EDF-RR、MLF-RR、EDF-SDR、MLF-SDR、FIPS以及TASA的分组平均时延相差不大，但是TASA算法始终保持最低的平均时延，当 $N=128$ 时，分组平均时延大约为64时隙。

两类调度中分组平均时延与交换规模的关系如图11所示，业务矩阵具有两个不同截止期。随着交换规模增大，分组平均时延呈上升趋势， $N \geq 32$ 时，几种对比算法之间的分组平均时延差距逐渐增大，当 $N=128$ 时，TASA算法的平均时延不超过200时隙每包，平均时延性能比FIPS算法提升了约18%，比传统对比算法提升了约24%。

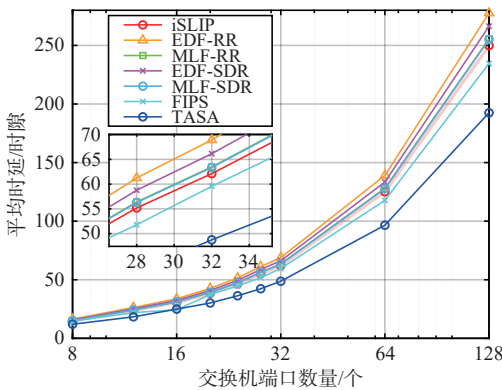


图11 两类调度中分组平均时延与交换规模的关系

三类调度中分组平均时延与交换规模的关系如图12所示，此时业务矩阵具有3种不同截止期。在交换规模较小时，几种算法的平均时延性能接近，TASA算法的性能稍微优于其他对比算法。但是，当 $N \geq 64$ 时，TASA算法的平均时延性能相对于其他几种对比算法具有较大优势。此时，TASA算法的平均时延性能比其他传统算法提升了大约15%，交换规模越大，TASA算法的分组平均时延性能优势越明显。由于三类调度的业务集合中具有多种不同的截止期约束，决策复杂度更高，所以在相同交换规模条件下，三类调度任务的分组平均时延大于两类调度任务的分组平均时延。

此外，研究了分组丢包率随交换规模的变化关系。三类调度中丢包率与交换规模的关系如图13所示，三类调度的分组丢包率随交换规模的增大呈现下降的趋势。从交换机理的角度分析其原因，分组在交换机内部排队的原因是交换调度出现端口冲突的现象，而一旦排队时延大于流量截止期则会导

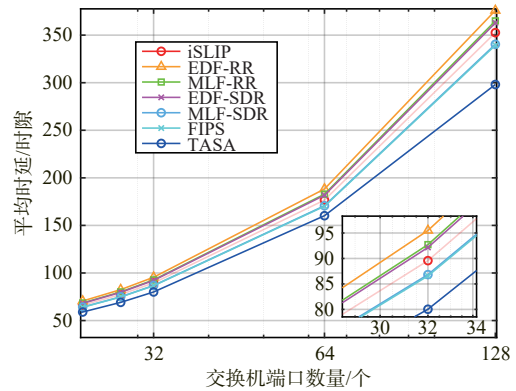


图12 三类调度中分组平均时延与交换规模的关系

致分组无法满足调度截止期约束而产生丢弃。而随着交换规模的增加，交换机在进行输入端口到输出端口的匹配时，产生冲突的概率降低，丢包率也会随之降低。从算法的角度来分析，在FIPS算法和TASA算法中，随着交换规模的增大，较大截止期的分组有更多的机会预推送到较小截止期分组的空闲时隙中进行提前调度，所以这两种算法的分组丢包率具有较大改善，当 $N=128$ 时，TASA算法的丢包率性能比iSLIP、EDF-RR等传统算法提升了5%~9%。

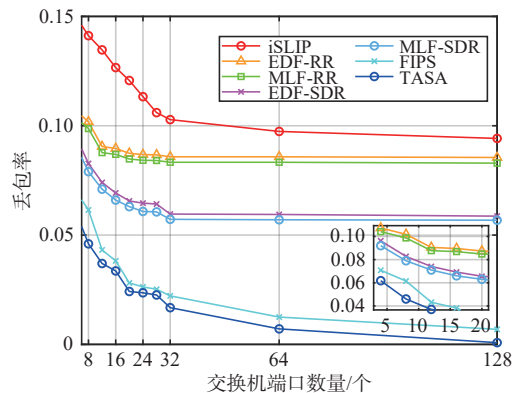


图13 三类调度中丢包率与交换规模的关系

三类调度中吞吐率与交换规模的关系如图14所示。结果显示，随着交换规模的增加，系统的吞吐率呈上升趋势，当 $N \geq 32$ 时，TASA算法的吞吐率最高，FIPS算法的性能次之，传统EDF-RR、MLF-RR、EDF-SDR、MLF-SDR算法的吞吐率较低。iSLIP算法的吞吐率最低，这是由于iSLIP算法采用迭代的方式计算最大匹配，过程中并不考虑分组截止期约束，导致更多的分组由于无法在截止期前完成调度，吞吐率性能下降。

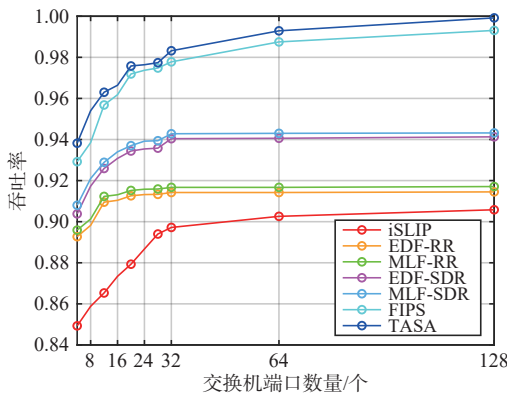


图14 三类调度中吞吐量与交换规模的关系

3.2 基于系统线路负载的仿真实验

本文在 16 × 16 规模的交换机上研究了线路负载对算法丢包率、平均时延、吞吐量等方面性能的影响。对线路负载从 0.5 增加到 1.0 的不同情况下进行仿真。线路负载 load 即线路繁忙程度，定义为本次传输调度总的分组数与可用的总时隙数之间的比值，如式(6)所示

$$load = \frac{to_packets}{to_slots} \quad (6)$$

其中，to_packets 表示本次调度中传输的总分组数，to_slots 表示本次调度中在截止期前可以利用的总时隙数。

首先，分析单类调度场景中系统线路负载与丢包率的关系，单类调度中丢包率与线路负载的关系如图 15 所示。随着交换系统的线路负载升高，施加于交换机的流量压力增大，从而丢包率提高。当系统的线路负载比较低时，各种算法的丢包率都处于较低的水平，当 load=1 时，TASA 算法的丢包率最低，保持在 6%。

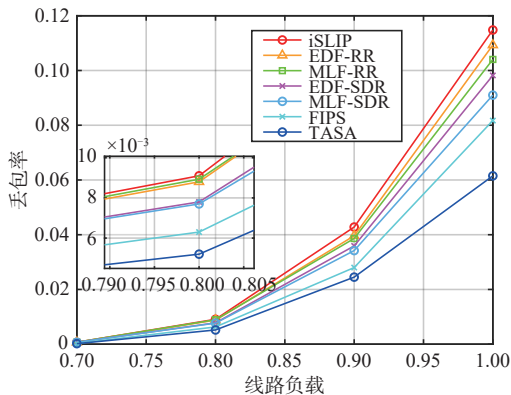


图15 单类调度中丢包率与线路负载的关系

其次，两类调度中丢包率与线路负载的关系如图 16 所示。可以看出，在两类分组调度中，线路负载越大，交换系统的丢包率越高，其中，TASA 算法相对于对比算法的丢包率性能表现更好，并且线路负载越高，TASA 算法的丢包率性能优势就越明显。当 load=1 时，TASA 算法的丢包率性能比 iSLIP 算法提升了约 10%。

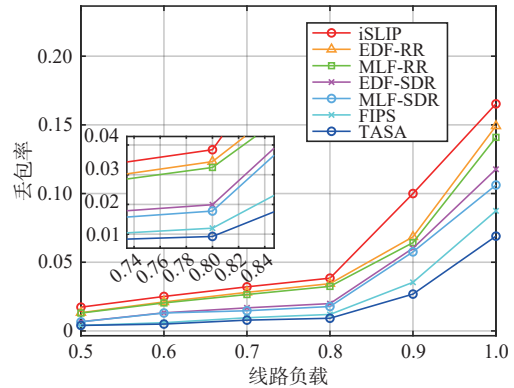


图16 两类调度中丢包率与线路负载的关系

三类调度中丢包率与线路负载的关系如图 17 所示，与两类调度的仿真结果类似，三类调度中系统的丢包率与线路负载呈正比例相关，并且三类调度的丢包率高于两类调度，当 load=1 时，TASA 算法的丢包率性能相对于对比算法提升了约 10%。

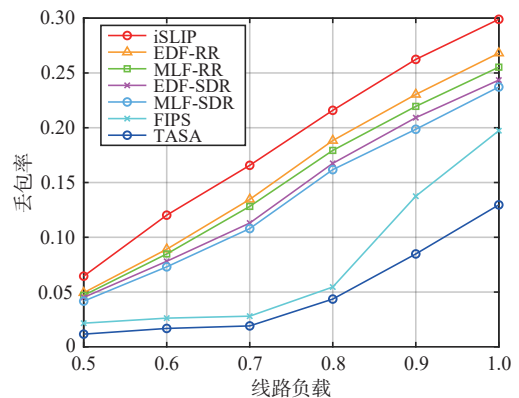


图17 三类调度中丢包率与线路负载的关系

最后，三类调度中吞吐量与线路负载的关系如图 18 所示，总体上，TASA 算法的吞吐量最高，可以保持在 85% 以上，FIPS 算法次之，EDF-RR、MLF-RR 等几种传统调度算法的吞吐量相对较低，iSLIP 算法的吞吐量最低。在相同负载条件下，TASA 算法可以使更多的分组在有截止期保障的条件下完成调度任务。

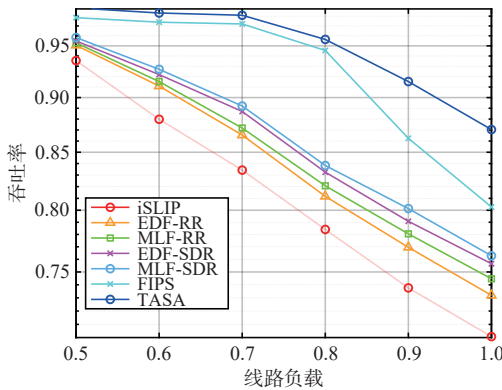


图 18 三类调度中吞吐量与线路负载的关系

4 结束语

针对传统车载网络吞吐率低、分组平均时延大、调度算法复杂度高等问题，本文提出了一种时间感知最大匹配流量调度算法，该算法将输入排队分组调度问题转化为业务矩阵分解问题，并将截止期较大的分组预推送至截止期较小的业务分组的空闲时隙中发送，在保障车载安全关键型业务最大化流量调度的同时降低转发时延。仿真结果表明，在多种流量场景中，TASA 算法能够以更低的丢包率、更小的平均时延保证分组在截止期到来之前完成分组调度任务。本文所提算法在车载时间敏感网络中的流量调度中具有较好的应用潜力。本文主要以输入排队作为底层交换模型，在未来的工作中，将进一步考虑如交叉节点排队等更多的交换模型，以及具有多跳转发的大规模网络场景。

参考文献:

[1] 李宗辉, 杨思琪, 喻敬海, 等. 时间敏感网络中确定性传输技术综述[J]. 软件学报, 2022, 33(11): 4334-4355.
 LI Z H, YANG S Q, YU J H, et al. State-of-the-art survey on deterministic transmission technologies in time-sensitive networking[J]. Journal of Software, 2022, 33(11): 4334-4355.

[2] HUANG C J, CHENG H-W, LIEN Y H, et al. A survey on video streaming for next-generation vehicular networks[J]. Electronics, 2024, 13(3): 649.

[3] PRAKASH J, MURALI L, MANIKANDAN N, et al. A vehicular network based intelligent transport system for smart cities using machine learning algorithms[J]. Scientific Reports, 2024, 14(1): 468.

[4] KIM H J, CHOI M H, KIM M H, et al. Development of an Ethernet-based heuristic time-sensitive networking scheduling algorithm for real-time in-vehicle data transmission[J]. Electronics,

2021, 10(2): 157.

[5] FANG Z W, LIANG J H, TAN C P, et al. Enhancing robust driver assistance control in distributed drive electric vehicles through integrated AFS and DYC technology[J]. IEEE Transactions on Intelligent Vehicles, 2024, PP(99): 1-14.

[6] LIANG M F, SU J C, SCHULTER S, et al. AIDE: an automatic data engine for object detection in autonomous driving[C]//Proceedings of the 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2024: 14695-14706.

[7] FENG Z W, CAI M Y, DENG Q X. An efficient pro-active fault-tolerance scheduling of IEEE 802.1Qbv time-sensitive network[J]. IEEE Internet of Things Journal, 2022, 9(16): 14501-14510.

[8] SUN W J, ZOU Y, ZHANG X D, et al. Optimization design framework for in-vehicle time-sensitive networking architecture[J]. IEEE Internet of Things Journal, 2024, 11(16): 27840-27853.

[9] DENG L B, XIAO X R, LIU H, et al. A low-delay AVB flow scheduling method occupying the guard band in time-sensitive networking[J]. Journal of Systems Architecture, 2022, 129: 102586.

[10] PATTI G, BELLO L L, LEONARDI L. Deadline-aware online scheduling of TSN flows for automotive applications[J]. IEEE Transactions on Industrial Informatics, 2023, 19(4): 5774-5784.

[11] 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.

[12] CHANG C S, CHEN W J, HUANG H Y. Birkhoff-von Neumann input buffered crossbar switches[C]//Proceedings of the Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Piscataway: IEEE Press, 2000: 1614-1623.

[13] ZHANG B X, WAN X L, LUO J Z, et al. A nearly optimal packet scheduling algorithm for input queued switches with deadline guarantees[J]. IEEE Transactions on Computers, 2015, 64(6): 1548-1563.

[14] ATAR R, SHADMI Y. Fluid limits for earliest-deadline-first networks[J]. Stochastic Processes and Their Applications, 2023, 157: 279-307.

[15] ZHANG Y W, MA J P, ZHENG H, et al. Criticality-aware EDF scheduling for constrained-deadline imprecise mixed-criticality systems[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2024, 43(2): 480-491.

[16] CHETTO M, EL OSTA R. Earliest deadline first scheduling for real-time computing in sustainable sensors[J]. Sustainability, 2023, 15(5): 3972.

[17] CHEN N J, KURNIAWAN C, NAKAHIRA Y, et al. Smoothed least-laxity-first algorithm for electric vehicle charging: online decision and performance analysis with resource augmentation[J]. IEEE Transactions on Smart Grid, 2022, 13(3): 2209-2217.

[18] 林晚晴, 李峭, 赵露茜, 等. 基于 Crossbar 的混合关键性流量调

度算法[J]. 电光与控制, 2016, 23(7): 6.

LIN W Q, LI Q, ZHAO L X, et al. Mixed-criticality flows scheduling algorithm for crossbar switches[J]. Electronics Optics & Control, 2016, 23(7): 6.

- [19] 吴俊, 李斌. 输入队列交换机中嵌套周期流优化调度问题的复杂性分析[J]. 计算机学报, 2010, 33(1): 55-62.

WU J, LI B. On the complexity of optimal scheduling multi-rate nested periodic traffic in an input-queued switch[J]. Chinese Journal of Computers, 2010, 33(1): 55-62.

- [20] MAJIDI S, OBERMAISSER R. Genetic algorithm for scheduling communication networks in time-triggered systems-of-systems[C]// Proceedings of the 2021 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW). Piscataway: IEEE Press, 2021: 121-128.

- [21] PAHLEVAN M, OBERMAISSER R. Genetic algorithm for scheduling time-triggered traffic in time-sensitive networks[C]// Proceedings of the 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA). Piscataway: IEEE Press, 2018: 337-344.

- [22] LEE Y, LOU J Y, LUO J Z, et al. An efficient packet scheduling algorithm with deadline guarantees for input-queued switches[J]. IEEE/ACM Transactions on Networking, 2007, 15(1): 212-225.

- [23] ASSADI S. A simple $(1: \epsilon)$ -approximation semi-streaming algorithm for maximum (weighted) matching[M]// 2024 Symposium on Simplicity in Algorithms (SOSA). Philadelphia, PA: Society for Industrial and Applied Mathematics, 2024: 337-354.

- [24] WEN Y, ZHANG X D, ZOU Y, et al. Modeling and traffic scheduling problems for in-vehicle time-sensitive network communication architectures[C]// Proceedings of the 2023 8th International Conference on Intelligent Computing and Signal Processing (ICSP). Piscataway: IEEE Press, 2023: 1905-1908.

- [25] GUO Y, LUO F, WANG Z T, et al. Research on automated testing of in-vehicle time-sensitive network time synchronization mechanism[J]. SAE International Journal of Connected and Automated Vehicles, 2024, 7(4): 12-7-04-0027.

- [26] YOSHIMURA A, ITO Y. QoS evaluation of ATS in IEEE 802.1TSN on in-vehicle Ethernet by comparing with CBS and TAS[J]. IEICE Communications Express, 2024, 13(9): 355-358.

- [27] GONZALEZ MARIÑO A, FONS F, MORENO AROSTEGUI J M. Vehicular network processor design for scalability & automation: Elastic Gateway SoC concept & builder[J]. Journal of Systems Architecture, 2024, 146: 103038.

- [28] GALILEA TORRES-MACÍAS A, SEGARRA FLOR J, LUIS BRIZ VELASCO J, et al. Fast IEEE802.1Qbv gate scheduling through integer linear programming[J]. IEEE Access, 2024, 12: 111239-111250.

- [29] WU G, GAN X. MatchBG: a boundary subgraph-based maximal

matching algorithm for bipartite graphs[C]// Proceedings of the International Conference on Database Systems for Advanced Applications. Singapore: Springer Nature Singapore, 2024: 463-473.

- [30] WEI Q, LIU M, LIANG Z, et al. A performance simulation and verification method of packet scheduling algorithms for data stream based on QoS[J]. Evolutionary Intelligence, 2020: 1-6.

[作者简介]



郑凌(1989-), 男, 博士, 西安邮电大学通信与信息工程学院讲师, 西安电子科技大学空天地一体化综合业务网全国重点实验室博士后, 主要研究方向为确定性网络、高性能路由交换、光计算赋能的智能信息处理、基于FPGA的人工智能算法硬件加速。



魏国栋(2000-), 男, 西安邮电大学通信与信息工程学院硕士生, 主要研究方向为时间敏感网络、高性能交换。



潘伟涛(1981-), 男, 博士, 西安电子科技大学空天地一体化综合业务网全国重点实验室副教授, 主要研究方向为与高速可编程网络相关的FPGA及芯片实现、网络协议卸载加速、RDMA网卡、星载路由器、星载数字透明处理器、时间触发以太网/TSN FPGA及芯片实现。



张科遥(2000-), 男, 西安邮电大学通信与信息工程学院硕士生, 主要研究方向为时间敏感网络、高性能交换。



褚宏云(1984-), 女, 博士, 西安邮电大学通信与信息工程学院讲师, 主要研究方向为无线通信系统、电磁信号处理、图像信号处理。