

# 多模态网络中时间敏感网络模态的智能调度机制

杨思锦<sup>1</sup>, 庄雷<sup>1</sup>, 宋玉<sup>1</sup>, 王家兴<sup>2</sup>, 阳鑫宇<sup>2</sup>

(1. 郑州大学计算机与人工智能学院, 河南 郑州 450001; 2. 郑州大学网络空间安全学院, 河南 郑州 450002)

**摘 要:** 针对多模态网络中时间敏感网络模态转发调度不确定、求解时间长等问题, 提出了一种基于 CSQF 的时间敏感网络模态的联合路由与调度机制。综合考虑有界时延需求、网络状态和不同的路由机制, 建立联合缓存队列和路由的混合资源调度模型, 旨在优化整个网络的资源使用。基于深度强化学习方法, 利用流量特征与缓存队列利用率来预测下一循环的缓存利用率。此外, 基于多队列 CSQF 转发调度机制和基于缓存利用率的显式路由算法, 提出了一种迭代调度算法, 实现了确定性转发和资源分配。仿真结果表明, 所提机制可以根据网络的资源使用情况有效地调整确定性应用的传输调度, 与其他离线调度机制相比, 具有更好的调度性能。

**关键词:** 时间敏感网络; 多模态网络; 确定性网络; 联合调度; 显式路由

**中图分类号:** TN92

**文献标志码:** A

**DOI:** 10.11959/j.issn.1000-436x.2022078

## Intelligent scheduling mechanism of time-sensitive network modal in polymorphic network

YANG Sijin<sup>1</sup>, ZHUANG Lei<sup>1</sup>, SONG Yu<sup>1</sup>, WANG Jiaying<sup>2</sup>, YANG Xinyu<sup>2</sup>

1. School of Computer and Artificial Intelligence, Zhengzhou University, Zhengzhou 450001, China

2. School of Cyber Science and Engineering, Zhengzhou University, Zhengzhou 450002, China

**Abstract:** For the problems of uncertain forwarding scheduling and long solving time of time-sensitive network modal in polymorphic network, a joint routing and scheduling mechanism of time-sensitive network modal based on CSQF was proposed. Considering the requirement of bounded delay, network state and different routing mechanisms, a hybrid resource scheduling problem of joint cache queue and routing was formulated to optimize the resource usage of the entire network. Then, the traffic characteristics and cache queue utilization was used to predict the cache utilization of the next cycle, which was based on deep reinforcement learning. In addition, by using multi-queue CSQF forwarding scheduling mechanism and explicit routing algorithm based on cache utilization, an iterative scheduling algorithm was proposed to achieve deterministic forwarding and resource allocation. Simulation results show that the mechanism can effectively adjust the transmission scheduling of deterministic applications according to the resource usage of the network, and has better schedulability compared with other off-line scheduling mechanisms.

**Keywords:** time-sensitive network, polymorphic network, deterministic networking, joint scheduling, explicit routing

## 0 引言

随着“互联网+”“工业 4.0”与“网络 5.0”的深度融合发展, 现有互联网基础架构及技术体系存在网络结构僵化、IP 单一承载、未知威胁难以抑制

等基础性问题, 无法通过有限的资源动态化提供智慧、高效、多元的网络差异化服务<sup>[1]</sup>。为打破上述网络发展困境, 加速推进新型网络技术发展, 我国邬江兴院士团队<sup>[2]</sup>提出了一种网络各层功能多模态呈现的网络架构——全维可定义的多模态智慧网

收稿日期: 2022-01-05; 修回日期: 2022-03-23

通信作者: 庄雷, ielzhuang@zzu.edu.cn

基金项目: 国家电网有限公司总部科技基金资助项目 (No.5700-202024176A-0-0-00)

**Foundation Item:** State Grid Corporation of China Science and Technology Project (No.5700-202024176A-0-0-00)

络。其确定了新型网络基线方式，具有底层资源可重构、多模态网络并存、全业务承载、智慧化管理等特点<sup>[3]</sup>。作为未来通信网络的发展范式，多模态网络以开放式的网络架构，通过可编程技术支撑泛在物联网和工业网多业务形态的并存部署，成为学术界和工业界的关注点<sup>[4]</sup>。

目前，在新型多模态智慧网络中，由于不同的网络模态和差异化的业务需求等因素，导致指数增长的数据量对业务控制信息实时性与确定性之间的矛盾。此外，通信平台网络化、共享化和数字化导致各种网络数据流急剧增多，同时也加剧了网络风暴、时延、抖动及网络丢包等问题。例如，智能电网作为多模态网络最具代表性的行业应用场景之一，是典型的多业务共存环境专网，其生产控制区业务和智能变电站业务对控制信令和数据采集有严格的低时延、有界抖动、高可靠等多种需求<sup>[5]</sup>。针对此类确定性的差异化需求，多模态网络需要为其提供包括时延、抖动、丢包率等关键指标在内的确定性业务保障。

当前能够保障确定性传输的技术主要有时间敏感网络 (TSN, time-sensitive network) 和确定性网络 (DetNet, deterministic networking) 等，它们主要研究网络确定性传输与低时延高可靠路径问题。TSN 工作组开发了相关的 IEEE 协议标准，以支持专业的局域网实时应用<sup>[6]</sup>。TSN 通过全网时钟同步、时间感知整形 (TAS, time-aware shaper)、流量调度等机制，实现了局域网范围内的时间敏感流和非实时流的共网传输。为了实现大规模确定性传输需求，IETF DetNet 工作组正在开发确定性体系架构，其扩展了 TSN 协议标准，致力于三层大规模确定性及可靠性传输<sup>[7]</sup>。TSN 和 DetNet 两者相互结合，通过协议扩展，来保证数据传输具有确定的低时延和低抖动，以满足差异化的服务质量 (QoS, quality of service) 要求，现已广泛应用于工业自动化、自动驾驶、智能电网等应用场景<sup>[8]</sup>。但随着新型 IP、工业互联、空天地一体化等大规模时间敏感应用的兴起，如何实现多样化网络模态共生共存环境下的 TSN 模态的流量调度，充分发挥网络特色模态的业务优势，成为当前面临的一个重要挑战。

流量调度是实现确定性传输的核心机制，其通过特定的调度算法在选定的所有交换机出口确定每个数据帧的传输时隙，保证所有帧在出口链路上依次无冲突地传输，同时满足流量各自的时延和带

宽等要求，使不同类别的业务需求在同一网络上共存。调度与路由联合规划是实现大规模流量调度的主要发展趋势之一，其被排除在 IEEE 802.1 标准之外。联合调度的一个关键难题是如何将动态时隙分配与产生的大量路径相协调，在大规模场景下，该问题计算复杂度高，难以满足时延调度需求。因此，本文引入深度强化学习 (DRL, deep reinforcement learning) 算法预测队列的利用率，生成显式路由方案，并与基于循环队列与转发机制相结合，设计了一种多模态网络场景下面向时间敏感网络模态的联合路由与调度机制，从而使业务流在满足差异化 QoS 目标上实现快速调度。

本文主要研究工作如下。

1) 基于多模态体系架构和循环队列与转发机制，在时间敏感网络模态下开展数据流的全局静态队列规划与局部时隙动态分配联合调度，实现大规模场景下的时间敏感流的自动化控制与调度。

2) 建立机器学习辅助的联合调度优化模型，引入改进的 DRL 算法建立流量状态与网络状态及路由队列状态之间的关系，以预测下一时隙队列利用率，实现在线智能的显式路由决策。

3) 通过基于 NeSTiNg 模型的仿真来验证本文调度机制。仿真结果表明，本文调度机制可以很好地为时间敏感应用提供低时延、低抖动的服务，同时提高了流的可调度性和资源利用率；与经典调度机制对比，本文调度机制的有效性更好。

## 1 背景与相关工作

### 1.1 CSQF 调度机制

DetNet 工作组提出了适应大规模网络标准的循环指定排队和转发 (CSQF, cycle specified queuing and forwarding) 机制。不同于 TSN 中的时间紧迫触发整形机制，如 TAS、循环排队与转发 (CQF, cycle queuing and forwarding)，CSQF 放松了严格的同步约束，通过增加一个接收队列、一个发送队列和多个额外队列来实现灵活的传输调度。其中，额外队列可以增加数据包缓存，并具有一定的处理抖动和流量突发的能力。CSQF 允许使用段路由对数据包进行路由和调度，可同时动态协调每个端口的传输周期。

DetNet 草案<sup>[9]</sup>描述了在大规模确定性网络中，所有网络设备通过 IEEE 1588v2 精确网络时间协议 (PTP, precise time protocol) 实现了全局频率同

步。这使路由器能够根据本地时间将时间域划分为多个等长的周期,以便合理安排转发行为进行周期映射。并且相邻设备之间有稳定的周期映射关系,在通过自适应学习获得映射关系后进行后续业务的转发。在流量整形与转发方面,草案规定时间敏感流的每个数据包都携带一个具有当前周期号的标识符,以便设备确定再次转发的周期。在转发周期内,只有发送队列中缓存的数据包可以发送,而其余队列只能接收或等待转发。

## 1.2 相关工作

以前大多数的联合调度相关研究都是基于严格的时间同步,利用 TSN 机制实现局域网内的确定性业务。文献[10]将时间触发流和速率约束流分离,优化了帧到虚拟链路的分配和虚拟链路路由的静态调度,采用禁忌搜索算法搜索路由与调度联合方案。文献[11]提出了一种基于遗传算法的启发式调度方法,以时间触发流的路由和调度约束组合作为基因,生成了静态全局调度。文献[12]设计了一种解决路由和调度的整数线性规划(ILP, integer linear programming),以最小化端到端时延为目标,通过删减流与链路无关的映射条件来缩小问题的规模,并通过 ILP 求解器获得具有较低通信时延的调度方案。文献[13]提出了一种音视频桥接(AVB, audio video bridging)流和时间触发(TT, time-triggered)流共存的联合调度方法。在考虑 AVB 流的同时解决了 TT 流的联合路由与调度问题,其结合 K-最短路径算法和基于贪婪随机自适应搜索过程的元启发式算法,提出了联合路由与调度策略以决策 TT 流的路径和门控列表(GCL, gate control list)。联合路由与调度策略迭代执行集成的路由算法并搜索最优的 GCL,最后在迭代过程中不断更新最优路由与 GCL 方案。文献[14]提出了一种启发式列表调度器用于 TSN 的全局调度,将调度和路由约束合并为一组约束,并在单个步骤中利用该约束计算特定于端口的 GCL;根据终端系统之间的最短路径,采用启发式列表调度算法依次遍历可行的计算节点,并在每个节点下遍历流的可选路径,在满足最小化流总传输时延的目标基础上,选择完成时间最小的路径和传输时延。文献[15]提出了一种基于配置冲突图的方法,将路由和调度问题归约为冲突图,通过在冲突图中搜索一个独立的顶点集来解决原始的流量规划问题;在导出配置冲突图过程中,将路径和传输开始时间作为节点,若同时采用 2 个节点

的配置,则会破坏约束,即 2 个节点存在冲突,冲突图中的所有无关点组合被认为是有效的路由和调度方案,但其同样采用 ILP 联合启发式算法进行求解。文献[16]考虑大规模 TSN 的无等待调度和多路径路由问题,采用离线调度方法,以已知的 TT 流为节点,以流之间的相关程度为边,通过基于迭代 TLP 的可伸缩调度(IIS, iterated integer-linear-programming based scheduling)技术将流分组,并以图划分算法将 IIS 迭代之间的冲突程度降至最低;采用贪心随机自适应搜索算法探索多路径路由技术,经过多次迭代,最终找出冲突最小的分组和多路径方案。其同样采用 ILP 求解每组的约束,最终所有分组的可行解组合为可行的调度方案。上述流量调度机制大多基于 TAS 调度 TT 流,基于 CBS(credit-based shaper)调度 AVB 流,且需要严格的全网时钟同步及复杂的门控调度列表,因此,以上机制在大规模确定性网络中效果不佳。

在最新的大规模确定性网络研究中。Krolikowski 等<sup>[17]</sup>采用 CSQF 与分段路由提出了离线和在线 2 种集中控制平面算法,以最大化流量接受率。其主要关注的是确定性网络的有界时延和抖动,将联合路由与调度问题描述为一个多商品流问题的扩展,通过列生成和动态规划的方法进行问题的求解。但在线快速调度只适用于小部分流。文献[18]在文献[17]的基础上,提出了一种负载均衡的解决方案,在产生候选路径的过程中,考虑链路的负载,在解决单个调度路径路由问题的情况下,生成可行的负载均衡策略。但其未讨论多流竞争资源的问题。文献[19]提出了一种基于软件定义的时间敏感网络跨域调度机制,基于 3 种广域时钟同步技术保障了纳秒级的域间同步;利用时隙无冲突约束条件限制了边缘流量的到达时间,解决了域间链路时延的对齐问题;最后在线计算得到全局的门控调度列表。但其未考虑路由带来的路径瓶颈问题。

综上所述,现有时间敏感网络的流量调度大多基于给定的路由方案,路由算法通常采用最短路径或生成树协议来确定,未考虑多流聚合时的资源争用问题,且主要适用于离线计算的场景。然而,在实际网络中,流量可能实时动态变化,此外,在大规模多模态网络调度场景下,在线智能化的路由与调度联合机制是必要的。为此,本文针对多模态网络中大规模时间敏感网络模态的流量调度问题进行了研究,提出了基于 CSQF 整形机制的时间敏感网络模态智能

调度机制和算法，联合调度路径和队列资源，保证了多业务场景下流量的有界时延及可调度性。

## 2 系统模型

本节针对时间敏感网络模态的流量调度问题，在遵循CSQF的基本转发准则下，设计了基于CSQF的网络模型和数学模型。

### 2.1 基于CSQF的网络模型

针对基于CSQF的确定性多模态基础网络，本文将时间敏感网络模态的联合路由与调度问题定义为在满足调度模型的约束条件下，针对时间触发的网络基础设施和流量模式 $F$ ，寻找一个可行的网络调度，以最大化可调度数据流数量。为不失一般性，本文将网络拓扑建模为有向图 $G=(V,E)$ ，其中， $V=\{v_1, v_2, \dots, v_p\}$ 表示底层多模态路由由节点集合， $E=\{e_1, e_2, \dots, e_E\}$ 表示相互连接的有向边集合。每一个链路 $e=(u,v) \in E$ 包含一个从节点 $u$ 到节点 $v$ 的长距离链路时延和处理时延 $l_e$ 。每条链路 $e$ 的带宽容量为 $C_e$ ，在一个周期 $T$ 内链路传输的最大流量定义为周期流量，用 $C_e T$ 表示。时间敏感流定义为元组 $f=(src, dst, t, period, pnum, D_f) \in F$ ，每个流 $f_i$ 分别由源节点、目的节点、流传输时延、发送周期、数据包数量和最大端到端时延组成。系统参数如表1所示。

表1 系统参数

| 参数                         | 含义                |
|----------------------------|-------------------|
| $V$                        | 节点集合              |
| $E$                        | 有向边集合             |
| $F=(f_1, f_2, \dots, f_n)$ | 流量模式              |
| $D(e_j)$                   | 链路时延              |
| src                        | 源节点               |
| dst                        | 目的节点              |
| $t$                        | 流传输时延             |
| period                     | 发送周期              |
| pnum                       | 数据包数量             |
| $D_f$                      | 最大端到端时延           |
| offset                     | 周期偏移              |
| sst                        | 发送时间偏移            |
| $Q_{size}$                 | 队列大小              |
| $h$                        | 路由跳数              |
| qu( $i$ )                  | 第 $i$ 个转发节点的队列利用率 |
| RANK( $k$ )                | 转发节点 $k$ 的度量值     |
| $N(k)$                     | 转发节点 $k$ 的邻居节点集合  |

为了统一时间敏感流的不同传输周期，设置超周期 $P$ 等于所有时间敏感流的最小公倍数，如式(1)所示。

$$P = \text{LCM}(F.\text{period}) \quad (1)$$

### 2.2 基于CSQF的数学模型

本文调度机制的目的是最大化可调度时间敏感流的数量，其目标函数为

$$\max \sum_{i=1}^n \text{Scheduled}(f_i) \quad (2)$$

其中， $\text{Scheduled}(f_i)$ 表示时间敏感流 $f_i$ 可成功调度。对于每个流 $f \in F$ ，本文将可行路径集表示为PATH，对应于流 $f$ 的源和目的地之间的路径，其端到端时延满足路径 $p \in \text{PATH}$ 上的E2E(end to end)时延约束。在每个超周期内，若分配的时隙在每个边不重叠，并且满足每个流的调度约束，则网络调度是可行的。本文提出的基于CSQF的调度机制必须满足以下核心约束条件。

#### 1) 周期约束

为了限制周期 $T$ 的上限和下限，本文在CSQF中设置了等长的时隙，如式(3)所示。

$$T \in \left[ \frac{Q_{size} \text{MTU}}{C_e}, \text{LCM}(F.\text{period}) \right] \quad (3)$$

其中，MTU为最大数据包长度。

#### 2) 截止时间约束

对于每个确定性数据流，该约束是实现有界时延的确定性转发关键。对于每一条输入流，其发送时延、处理时延和链路时延组成了端到端时延，三者之和应该小于或等于端到端最大时延。该约束定义为

$$T \left( \text{sst} + \sum_p \text{offset} \right) + \sum_{e \in p} l_e \leq D_f \quad (4)$$

#### 3) 分配配置约束

为了确保多个流不会被同时调度到同一条链路，在基于CSQF实现的调度中，一次只能为一个时间敏感流 $f_i$ 保留一个时隙资源。该约束定义为

$$\sum_{f \in F} \sum_{p \in \text{PATH}} R_e^t(f_i) \leq C_e \frac{P}{T} \quad (5)$$

$$\sum_{i=1}^n \text{Scheduled}(f_i) f_i.pnum \leq Q_{size} \text{qnum} \quad (6)$$

其中， $R_e^t$ 为链路 $e$ 在第 $t$ 个时隙内被占用的资源， $R_e^t(f_i)$ 为流 $f_i$ 与 $R_e^t$ 的资源匹配关系，qnum为CSQF

机制使用的队列数。由于一次只能为一个时间敏感流保留一个时隙资源，如果链路重叠的 2 条路径被分配到同一时隙内，可能会引起冲突，因此链路上任一时隙内最多只允许一个时间敏感流进行传输，则有约束条件式(5)成立。约束条件式(6)表示对于可调度的时间敏感流，其数据包总数不能超过该节点的队列资源容量。

#### 4) 时隙无重叠约束

在基于 CSQF 的调度中，要确保每个流在每个输出端口上的时隙必须比输入链路上的时隙调度得晚，即在任何传输周期内，流  $f_i$  在下一跳的传输时间应大于后续流  $f_j$  的出端口时间。式(7)确保了流  $f_i$  在出端口的传输时间和其他任何流  $f_j$  的出端口时间都不重叠。该约束定义为

$$\forall f_i, f_j \in F (i \neq j), e \in E$$

$$f_i.sst + \frac{l_e}{f_i.period} \geq f_j.sst + \frac{l_e}{f_j.period} + t_j \quad (7)$$

当网络基础设施的规模和复杂性增加，将多个业务的时间敏感流映射到某个网络路径上时，可能稍后才会发现该路径上的单个链路过载，或者因时隙重叠阻塞了流的传输，从而使数据流无法满足其时延约束。此外，如果选择的下一跳不需要排队就能转发，那么就网络节点而言，较长的可行路径甚至会产生更低的时延。为了解决路由和调度之间的相互依赖，一种显式路由策略是必要的。下面将介绍基于预测队列缓存利用率的显式路由策略，联合动态资源调度，以确定最佳路由和调度选项。

### 3 时间敏感网络模式的联合调度机制

针对时间敏感网络模式的联合调度问题，本文设计了时间敏感网络模式的智能调度机制和算法。首先，在基于 CSQF 的多队列转发平面中，来自不同上游节点的数据包可以映射到相同的发送周期并共享相同的队列，因此会产生聚合调度问题。此时，队列缓存是产生数据包丢失的主要因素。为了解决该问题，本文设计了一个基于预测队列利用率的显式路由解决方案，以获得端到端时延与网络负载之间的均衡，同时解决静态路由造成的资源争用问题。该方案以 DRL 为基础，依据当前队列缓存状态和网络指标，预测下一跳队列利用率，并根据

当前和预测的队列利用率实现负载均衡的显式路由。其次，时间敏感流与资源分配关系是 NP 难问题，因此，基于显式路由方案，本文设计了一种时间敏感网络模式的在线多约束迭代调度算法，按流的固有到达顺序逐流进行调度，从而减少了计算时间，获得满足约束的局部最优解，提高了可调度数据流的数量。

#### 3.1 基于 DRL 的显式路由

本文提出的显式路由方案考虑当前和预测的队列缓存利用率，决定缓冲区中数据包的下一跳。当转发节点接收到数据包时，它会根据多个指标（如队列利用率和最大化时延）选择下一跳候选集，以避免数据包丢失、降低排队时延或达到无等待传输，减少抖动。

在基于 DRL 预测整形缓存利用率的路由问题中，状态、动作、奖励的具体含义如下。

1) 状态。在时间敏感网络模式中，状态指数据分析代理解析出的网络中的流请求信息和网络信息，状态用  $s_t = [F_t, bw_t, qu_t, pri]$  表示，其中， $F_t$  为  $t$  时刻流请求集合， $bw_t$  为  $t$  时刻带宽容量， $qu_t$  为当前节点队列整形缓存利用率， $pri$  为流的优先级。

2) 动作。动作指将描述的状态信息  $s_t$  作为输入，根据优化策略函数指定每个流的下一跳，并生成一个流量分配配置操作。强化学习智能体在获得当前环境状态  $s_t$  后，根据策略函数  $\pi$  生成动作向量  $a_t$ ，并将动作的奖励  $r_t$  反馈给智能体。动作向量  $a_t$  可表示为

$$a_t = \pi(s_t) \quad (8)$$

其中， $a_t = [a_1, a_2, \dots, a_n]$ ，每个分量  $a_i (0 \leq i \leq n)$  表示邻居节点之间可用队列缓存的比重。

3) 奖励。奖励是对选择动作所获收益的评价。本文模型的目标是让数据包的时延降低到所需时间间隔内，同时最大限度地提高整形缓存队列的利用率，因此，在节点  $i$  和时间  $t$  内，将属于流  $f_j$  缓存中的数据包的转发到下一节点，其奖励为

$$r_t = \alpha_1 \frac{bw_{i,t}}{Bw_i} + \alpha_2 \frac{q_{i,t}}{Q_i} + \alpha_3 \frac{p_{max}^j}{P_j} \quad (9)$$

其中， $bw_{i,t}$  表示剩余链路容量， $q_{i,t}$  表示剩余缓存资源， $Bw_i$  和  $Q_i$  分别表示节点的总容量与总缓存资源， $P_j$  表示流  $f_j$  的长度。

本文采用三层 DQN 神经网络来解决多状态和

动作问题，将网络状态作为输入层，通过与特征对应的计算，网络模型得到下一时隙缓存队列的预测值。为了取得较好的训练效果，采用策略梯度的方法，即用神经网络将策略参数化，寻找最优的参数使所有数据流的端到端时延最小。损失函数则是对参数化后的目标函数求梯度，使其以最快的速度向目标函数收敛。此时搜索空间相比于初始阶段较小，故损失函数为

$$\nabla \text{loss} = \sum_t \nabla_{\theta} P(t, \theta) r_t \quad (10)$$

其中， $P(t, \theta)$  为在时刻  $t$  策略选择执行动作  $a_t$  的概率， $r_t$  为在时刻  $t$  执行动作  $a_t$  后得到的奖励值。

针对下一跳选择机制，使用队列利用率  $\text{qu}(k)$  和带宽利用率来度量路由器，记为 RANK，并基于该 RANK 值从候选集中选择下一跳。路由器  $k$  的 RANK 为

$$\text{RANK}(k) = \alpha \text{qu}_i(k) + \beta \text{qu}_{i+1}(k) + \gamma \frac{\text{bw}_k}{\text{BW}_k} \quad (11)$$

其中，参数  $\alpha + \beta + \gamma = 1$ 。每个路由器根据式(12)定义的邻居节点  $N(k)$  确定下一跳候选集  $H_k$ ，即

$$H_k = \{h_n(d) \leq h_k(d), \text{qu}_i(n), \text{qu}_{i+1}(n+1) < 1\} \quad (12)$$

其中， $h_n(d) \leq h_k(d)$  表示路由器  $n$  距离目的节点比路由器  $k$  近， $\text{qu}_i(n), \text{qu}_{i+1}(n+1) < 1$  表示路由器  $n$  有可用的缓存资源。

基于队列缓存利用率的路径选择算法允许每个路由器选择具有较低队列利用率的下一跳。虽然所选路径不一定是路由跳数最少的，但考虑缓存队列的利用率及链路利用率，可以实现更低的排队时延，有效利用底层网络资源。队列缓存的显式路由算法如算法 1 所示。在离散时间段内，对于每个路由器，采用链路容量、业务模式、当前队列缓存利用和流优先级作为输入，经过运行阶段和中央控制器的信息分发，每个确定性路由节点可以根据神经网络的输出做出路由决策，实现负载均衡的路径方案。

**算法 1** 基于队列缓存的显式路由算法

输入  $G, s_i = [F_i, \text{bw}_i, \text{qu}_i, \text{pri}]$

输出 可行路径集 PATH

- 1) 初始化路径集 PATH
- 2) 对任意流的中间节点，基于 DRL 算法预测下一时隙队列利用率  $\text{pqu}$
- 3) 每个路由器根据式(11)计算各自的 RANK 值

- 4) 基于式(12)和路由器的 RANK 值，选择下一跳候选节点  $\text{nexthop}$

- 5) while  $\text{nexthop} \neq \text{null}$  do

- 6)  $\text{nexthop} = \text{Next}(\text{nexthop}, \text{qu}, \text{pqu})$

- 7)  $\text{path.add}(\text{nexthop})$

- 8)  $\text{next.buffer.push}(s_i)$

- 9) end while

- 10) if  $\text{nexthop} \neq \text{null}$  then

- 11) return PATH

- 12) else

- 13) return null

- 14) end if

### 3.2 动态调度算法设计

时间敏感网络模态的联合路由与调度机制的整体流程如算法 2 所示。

**算法 2** 时间敏感网络模态的联合路由与调度机制

输入  $G=(V, E), F$

输出  $F_{\text{sched}}$

- 1) 初始化  $F_{\text{sched}} = \{\phi\}$

- 2) 对任意  $f_i \in F$ ，通过算法 1，得出可行路径集 PATH

- 3) for  $i \in \text{len}(\text{PATH})$  do

- 4) 计算每条路径的权重，排序得到  $R_{\text{explicit}}$

- 5) end for

- 6) while  $\text{iteration} < \text{iteration\_max}$  do

- 7) 根据给定的  $R_{\text{explicit}}$  和 CSQF 多队列产生约束条件

- 8) 对式(2)的目标函数进行求解

- 9) if 流  $f$  成功被调度 then

- 10) 更新  $F_{\text{sched}}$

- 11) else

- 12) 将  $F_{\text{fail}}$  中的流重新调度

- 13) end if

- 14) 更新  $F_{\text{sched}}^* = F_{\text{sched}}^* + F_{\text{sched}}$

- 15) end while

- 16) return  $F_{\text{sched}}$

算法 2 的输入为顺序到达的流量模型  $F$ 、网络拓扑  $G$ ，输出为最终可调度的数据流集合  $F_{\text{sched}}$ 。首先，算法初始化可调度流量集合（第 1)行）；随后，通过算法 1 的显式路由算法得到可行的调度路径集

(第 2)行), 并基于每个路由器的 RANK 值进行链路的排序, 得到一条显式路由 (第 3)~(5)行)。在进行迭代调度过程中, 先根据预定义的路由和 CSQF 多队列机制, 产生 2.2 节定义的链路无重叠的约束条件式(3)~式(7) (第 7)行)。在满足约束条件下, 求解目标函数得到一个可行的调度方案, 将该调度结果更新到  $F_{\text{sched}}$  中, 并迭代计算未成功调度的具有较高优先级的流 (第 8)~(14)行)。最后得到可调度的全部数据流。

## 4 仿真实验

### 4.1 实验环境

本文在 NeSTiNg<sup>[20]</sup>上实现了调度模型实例。NeSTiNg 是基于 OMNeT++ 离散事件模拟器和 INET 框架的模拟器, 可为网络层的 DetNet 服务和链路层的 TSN 提供构建块, 能够轻松地进行实验。本文使用 Python 和 Tensorflow 搭建 DRL 网络并进行模型的训练。

考虑多样化的综合网络环境, 本文设置了具有 9 个核心节点和 100 个边缘节点的大规模多模态网络场景。接入和核心链路的容量分别为 10 Mbit/s 和 100 Mbit/s, 链路时延为 1~10 ms, 设置每个多模态路由节点的带宽为 100 Mbit/s。底层多模态网元均采用 3 个队列的 CSQF, 用于确定性流量传输, 队列大小为 1 MB, 周期持续时间  $T=10 \mu\text{s}$ , 超周期  $P=10 \text{ms}$ 。依据实际场景业务需求和 TSN 实验台定义的流量特征, 生成具有随机源和目的地的时间敏感流<sup>[21]</sup>, 所有的流连续不断地产生, 均服从均值为  $\lambda$  的泊松分布。本文在网络中部署了 2 000 条具有 4 种不同周期的流, 周期分别为 1 ms、2 ms、5 ms、10 ms。所有应用程序的带宽要求遵循泊松分布, 持续时间服从指数分布, 模拟时间设置为 1 800 s。

### 4.2 结果和分析

为了证明时间敏感网络模态的联合调度机制的有效性, 本文将其与基于 CSQF 的最短路径 (SP-CSQF, short path CSQF-based) 机制和随机列生成 (CG-RR-CSQF, column generation and randomized rounding-CSQF) 机制<sup>[17]</sup>进行了结果对比。调度性能的评价指标是运行时间与调度成功率。

#### 4.2.1 流量规模对调度性能的影响

调度成功率随 TS 流数量的变化如图 1 所示。随着流数量的增加, 3 种策略的调度成功率均有下降趋势, 但本文机制一直保持最高的调度成功率。

当网络中时间敏感流数量为 200 时, 本文机制与 CG-RR-CSQF 机制均能调度 98% 的流量, 而 SP-CSQF 机制只能成功调度约 80% 的流量。当流数量大于 200 时, 会产生资源争用问题, 但本文机制采用的智能显式路由策略会根据不同的流量特征选择下一跳, 相比于最短路径, 路由节点和链路不会成为流量瓶颈, 而且采用的整形机制错开了 TS 流之间的传输时隙, 因此, 本文机制调度成功率相比于 SP-CSQF 和 CG-RR-CSQF 分别提高了 31% 和 7%。

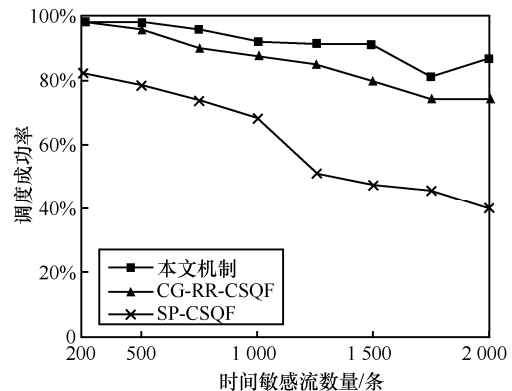


图 1 调度成功率随 TS 流数量的变化

随着流数量的增多, 3 种调度机制的运行时间变化如图 2 所示。从图 2 中可以看出, 本文机制可以在较短的时间内提供解决方案, 在降低求解时间的同时保证了稳定的调度成功率。这对于在线调度来说是合理的。CG-RR-CSQF 和 SP-CSQF 均采用离线调度方式, 而且需要预知所有流的信息, 计算时间长; 而本文机制在线动态调度逐流进行计算, 减少了计算时间。综合来看, 本文基于预测队列的联合调度机制是有效的。

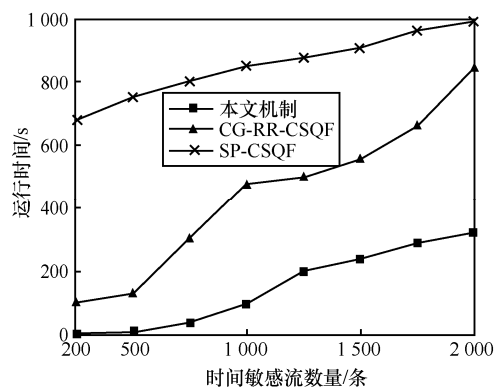


图 2 运行时间随 TS 流数量的变化

#### 4.2.2 不同网络规模对调度性能的影响

为了探究网络规模对调度性能的影响, 本文随机生成 3 个不同规模的网络, 分别包含 50 个、

100 个和 150 个路由节点，待处理流的数量为 1 000 个。

网络规模对丢包率的影响如图 3 所示。随着网络规模的增大，更多的下一跳候选节点可以带来更好的性能，所以 3 种调度机制的丢包率都略有下降，但本文机制具有较低的丢包率。3.1 节的理论描述可以很好地解释这一结果，本文使用的神经网络具有较强的学习能力，具有与非线性关系的关联能力，与 CG-RR-CSQF 和 SP-CSQF 相比，避免了缓冲区资源较少的路由作为下一跳，因此降低了丢包率。

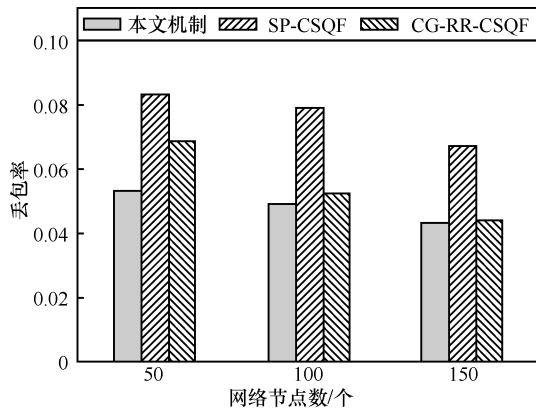


图 3 网络规模对丢包率的影响

网络规模对调度成功率的影响如图 4 所示。随着网络规模的增大，可选的下一跳及可用的链路资源也越来越多，因此 3 种调度机制的调度成功率均有上升。通过与 SP-CSQF 和 CG-RR-CSQF 对比可以看出，本文机制采用的动态路由方案在处理大规模网络时具有较高的调度成功率。

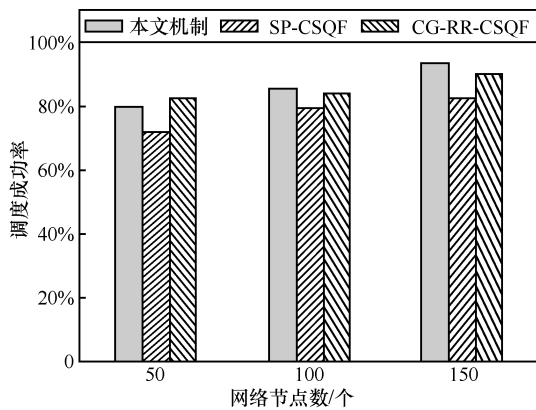


图 4 网络规模对调度成功率的影响

### 4.2.3 调度机制对时延的影响

时延是确定性网络需要解决的首要因素，因

为确定性时延应用所需的时延是有界的，这为网络调整数据传输提供了一定程度的灵活性。为了评估调度机制对时延的影响，本节以数据包在网络中经历的平均跳数为基准，对比了不同调度机制的性能，结果如图 5 所示。本文机制的平均跳数大于 CG-RR-CSQF 和 SP-CSQF。这说明根据网络状态调整传输时延的机制是有效的，即本文考虑队列利用率的方案可以避免选择队列缓冲资源较少的路由器，并基于预测的队列利用率选择满足有界时延的下一跳，虽然增加了跳数，但减少了排队时延，降低了端到端时延。而 CG-RR-CSQF 和 SP-CSQF 虽然经历的平均跳数相对较少，但其不能根据网络状态调整传输时延，因此会增大排队时延，造成较高的丢包率。此外，这也从侧面证明了本文机制适合长距离传输，且随着时间的推移趋于收敛且具有稳定性。

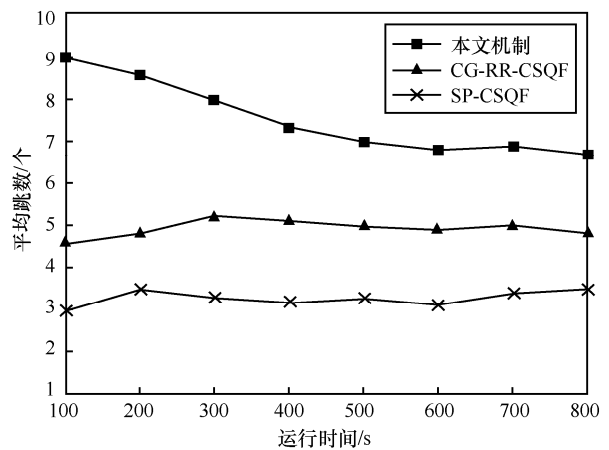


图 5 不同调度机制的性能

### 4.2.4 不同层数的 DQN 对调度性能的影响

本节评估不同层数的 DQN 对调度性能的影响。DQN-1、DQN-2、DQN-3 分别表示层数为 1、2、3 的 DQN，共有 100 个训练周期，每个周期迭代 1 000 次。不同 DQN 层数调度成功率随训练周期的变化如图 6 所示。通过对比可知，从宏观来看，随着 DQN 层数的增加，调度机制的整体性能逐渐提高；且随着训练次数的增加，调度成功率也显著增加。尤其当运行 60 个训练周期后，DQN-3 的性能明显优于 DQN-1 和 DQN-2，这说明在运行时间随训练周期增加而增加的条件条件下，本文使用的 DQN-3 具有明显的实效，且收敛性良好。

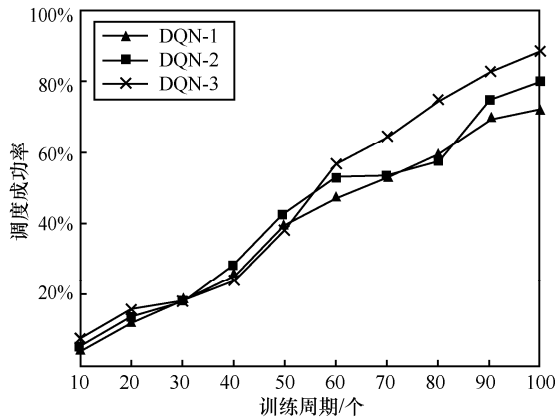


图 6 不同 DQN 层数调度成功率随训练周期的变化

## 5 结束语

本文依据 TSN 相关理论基础和标准,对多模态网络中的时延敏感流调度机制进行了扩展和改进,提出了一种基于 CSQF 的大规模时间敏感网络模态联合路由与调度机制,以适应多样化网络模态共生共存的环境。该机制利用 DRL 预测转发节点的队列缓存利用率,将联合路由与调度机制分解为全局静态和局部动态的结合。在动态调度过程中,基于预测的队列利用率生成不同的显式路由,解决了由于流量聚合而造成的数据丢包率高的问题。同时,智能选择下一跳机制可消除排队时延,保证时延的有界性及确定性。另外,基于 CSQF 的迭代调度机制解决了离线调度求解时间长的问题。实验结果表明,在多网络模态融合的多模态场景下,本文联合路由与调度机制可以在较短的时间内得出调度结果且具有较高的资源利用率和调度成功率。

## 参考文献:

[1] 郭江兴, 兰巨龙, 程东年. 新型网络体系结构[M]. 北京: 人民邮电出版社, 2014.  
WU J X, LAN J L, CHENG D N. Novel network architecture [M]. Beijing: Posts and Telecommunications Press, 2014.

[2] WU J X. Thoughts on the development of novel network technology[J]. Science China Information Sciences, 2018, 61(10): 1-11.

[3] 胡宇翔, 伊鹏, 孙鹏浩, 等. 全维可定义的多模态智慧网络体系研究[J]. 通信学报, 2019, 40(8): 1-12.  
HU Y X, YI P, SUN P H, et al. Research on the full-dimensional defined polymorphic smart network[J]. Journal on Communications, 2019, 40(8): 1-12.

[4] 兰巨龙, 程东年, 胡宇翔. 可重构信息通信基础网络体系研究[J]. 通信学报, 2014, 35(1): 128-139.  
LAN J L, CHENG D N, HU Y X. Research on reconfigurable information communication basal network architecture[J]. Journal on

Communications, 2014, 35(1): 128-139.

[5] 任罡, 李世倩, 季宁, 等. 智能电网层次化广域保护系统的关键技术研究[J]. 电测与仪表, 2020, 57(17): 93-98.  
REN G, LI S Q, JI N, et al. Key technologies of hierarchical wide-area protection system for smart grid[J]. Electrical Measurement & Instrumentation, 2020, 57(17): 93-98.

[6] 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.

[7] GROSSMAN E. Deterministic networking use cases[R]. RFC Editor, 2019.

[8] 黄韬, 汪硕, 黄玉栋, 等. 确定性网络研究综述[J]. 通信学报, 2019, 40(6): 160-176.  
HUANG T, WANG S, HUANG Y D, et al. Survey of the deterministic network[J]. Journal on Communications, 2019, 40(6): 160-176.

[9] FINN N, THUBERT P, VARGA B, et al. Deterministic networking architecture[R]. RFC Editor, 2019.

[10] 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.

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

[12] SCHWEISSGUTH E, DANIELIS P, TIMMERMANN D, et al. ILP-based joint routing and scheduling for time-triggered networks[C]//Proceedings of the 25th International Conference on Real-Time Networks and Systems. New York: ACM Press, 2017: 8-17.

[13] GAVRILUȚ V, ZHAO L X, RAAGAARD M L, et al. AVB-aware routing and scheduling of time-triggered traffic for TSN[J]. IEEE Access, 2018, 6: 75229-75243.

[14] 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.

[15] FALK J, DÜRR F, ROTHERMEL K. Time-triggered traffic planning for data networks with conflict graphs[C]//Proceedings of 2020 IEEE Real-Time and Embedded Technology and Applications Symposium. Piscataway: IEEE Press, 2020: 124-136.

[16] 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.

[17] KROLIKOWSKI J, MARTIN S, MEDAGLIANI P, et al. Joint routing and scheduling for large-scale deterministic IP networks[J]. Computer Communications, 2021, 165: 33-42.

[18] CHEN S, LEGUAY J, MARTIN S, et al. Load balancing for deterministic networks[C]//Proceedings of 2020 IFIP Networking Conference (Networking). Piscataway: IEEE Press, 2020: 785-790.

[19] 汪硕, 黄玉栋, 黄韬, 等. 基于软件定义的时间敏感网络跨域调度机制[J]. 通信学报, 2021, 42(10): 1-9.  
WANG S, HUANG Y D, HUANG T, et al. Software-defined cross-domain scheduling mechanism for time-sensitive networking[J]. Journal on Communications, 2021, 42(10): 1-9.

- [20] FALK J, HELLMANN S D, CARABELLI B, et al. NeSTiNg: simulating IEEE time-sensitive networking (TSN) in OMNeT++[C]// Proceedings of 2019 International Conference on Networked Systems (NetSys). Piscataway: IEEE Press, 2019: 1-8.
- [21] ADEMAJ A, PUFFER D, BRUCKNER D. Time sensitive networks for flexible manufacture in testbed-description of converged traffic types[S]. IEEE, 2021.



宋玉(1969- )，男，河南南阳人，博士，郑州大学副教授、硕士生导师，主要研究方向为下一代互联网、人工智能、机器学习等。

#### [作者简介]



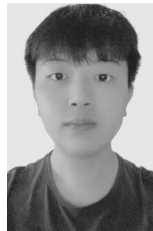
杨思锦(1990- )，女，河南商丘人，郑州大学博士生，主要研究方向为下一代互联网、时间敏感网络、确定性网络。



王家兴(1998- )，男，河南周口人，郑州大学硕士生，主要研究方向为时间敏感网络、下一代互联网。



庄雷(1963- )，女，河南郑州人，博士，郑州大学教授、博士生导师，主要研究方向为未来网络架构、时间敏感网络、网络虚拟化等。



阳鑫宇(1998- )，男，湖南邵阳人，郑州大学硕士生，主要研究方向为确定性网络、下一代互联网。