

面向混合多域组网的软件定义时间敏感网络

郭梦杰, 寿国础, 刘雅琼, 胡怡红

(北京邮电大学信息与通信工程学院, 北京 100876)

摘要: 针对混合多域网络中确定性端到端传输问题, 提出一种软件定义时间敏感网络 (SD-TSN) 组网方案。该方案遵循软件定义的范式, 设计了协同控制器和域控制器以实现混合多域之间的连接和互操作, 构建了 SD-TSN 组网实验平台、研发协同配置等关键模块以实现各个域中调度表的精准下发。在多种混合多域场景下进行组网测试, 实验结果表明, 所提方案能够保证高精度的跨域时间同步, 并提供具有有界低时延以及低时延抖动的端到端传输。

关键词: 混合多域组网; 时间敏感网络; 软件定义网络; 确定性传输

中图分类号: TP393

文献标志码: A

DOI: 10.11959/j.issn.1000-436x.2025100

Software-defined time-sensitive networking for hybrid multi-domain networking

GUO Mengjie, SHOU Guochu, LIU Yaqiong, HU Yihong

School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China

Abstract: To address the problem of deterministic end-to-end transmission in hybrid multi-domain networks, a software-defined time-sensitive network organization solution was proposed. Based on the software-defined paradigm, the solution was designed with coordinated and domain controllers to enable connectivity and interoperability between hybrid multi-domains. An software-defined time-sensitive networking (SD-TSN) experimental platform was built, and key modules such as collaborative configuration were developed to achieve accurate delivery of schedules in each domain. Networking experiments were carried out in a variety of hybrid multi-domain scenarios. The experimental results show that the proposed solution can ensure high-precision cross-domain time synchronization and provide end-to-end transmission with bounded low latency and low latency jitter.

Keywords: hybrid multi-domain networking, time-sensitive networking, software-defined networking, deterministic transmission

0 引言

随着工业互联网的快速发展, 越来越多的新兴应用对网络提出确定性传输需求^[1]。一些典型用例包括工业自动化^[2]、触感互联^[3]、远程医疗^[4]等。时间敏感网络 (TSN, time-sensitive net-

working) 被认为是应对这一挑战的关键技术。通过对传统以太网的增强和扩展, TSN 提供了有界低时延、低抖动, 以及高可靠的传输保证^[5]。值得注意的是, TSN 的确定性保证通常仅限于自身网络域内^[6], 即一组使用 TSN 标准传输时间敏感流量的端设备和网桥的集合^[7]。

收稿日期: 2025-02-10; 修回日期: 2025-05-26

通信作者: 寿国础, gcshou@bupt.edu.cn

基金项目: 国家自然科学基金资助项目 (No.92067102)

Foundation Item: The National Natural Science Foundation of China (No.92067102)

然而,在工业互联网背景下,许多应用的端到端传输往往涉及多个网络域^[8]。例如,在工业自动化中,机器、生产单元和生产线可能位于不同的车间中,并且来自不同车间的设备需要互联以支持协同控制等时间关键任务的实施^[9]。文献[10]描述了工业自动化中位于不同 TSN 域中的机器到机器通信用例。文献[11]描述了另一个工业用例,其中来自不同制造即服务(MaaS, manufacturing as a service)提供商的机器部署在各自的 TSN 域中,在执行物理协作任务时,这些机器需要进行跨域通信。与此同时,工业生产系统在硬件方面呈现出生命周期长、技术升级成本高和复杂度大等特征^[6]。因此,针对 TSN 域和非 TSN 域的混合多域场景,构建组网方案以提供确定性传输保证对工业互联网的发展至关重要^[12]。

目前,常见的非 TSN 域主要包括传统以太网、5G、确定性网络(DetNet, deterministic networking)以及灵活以太网(FlexE, flexible Ethernet)等。为此,Wang 等^[13]提出了一种 TSN 集成 5G 架构,其中 5G 系统充当支持 TSN 的逻辑网桥以支持确定性传输。Peng 等^[14]提出了一种确定性的跨域流调度解决方案,以支持跨 DetNet 的确定性传输。文献[15]设计了跨域传输机制,利用确定性 IP(DIP, deterministic IP)连接 TSN 域以实现远距离确定性传输。Huang 等^[16]提出了基于周期的时敏确定性网络(C-TSDN, cycle-based time-sensitive and deterministic network)架构并设计了循环排队转发机制,以实现跨广域的确定性传输。这些研究都有效地解决了跨非 TSN 域的确定性传输问题,但它们大都针对特定的非 TSN 域(如 5G 和 DIP)进行设计,没有提供足够通用的组网方案。

针对上述挑战,软件定义网络(SDN, software-defined networking)集成 TSN 被认为是一个极具潜力的解决方案,吸引了众多学者的关注。目前,SDN 与 TSN 集成研究集中在以下 2 个方面。

1) SDN 融合 TSN 范式

面对新兴工业应用严格的传输需求,在 SDN 范式中融合 TSN 机制,以增强其实时能力是一个热门研究方向。Nayak 等^[17]首次提出了时间敏感软件定义网络(TSSDN, time sensitive software defined network)的设想,消除了流量传输时的相互干扰。文献[18]详细描述了 TSSDN 组网方案,

通过使用带有 SDN 功能的硬件,联合解决时间敏感流量的调度和路由问题。具体来讲,TSSDN 为时间敏感流量计算其位于发送端的调度表。该调度表包含了每个流量的发送时间和传输路径,以确保在每个流量传输期间,传输路径完全为该流所保留。

2) SDN 增强 TSN 机制

SDN 与 TSN 结合的另一个研究方向是应用 SDN 技术增强 TSN 机制,主要集中在时间同步、流量调度以及系统配置等方面。例如,Guo 等^[19]提出了一种基于软件定义时间敏感网络的跨域互联方案,使用软件定义的时间补偿方法,实现了跨非 TSN 域时间同步。汪硕等^[20]提出了基于软件定义的时间敏感网络跨域调度机制,解决域间设备长链路时延带来的时隙对齐问题。Bhattacharjee 等^[21]提出了一种分层架构设计和控制平面框架,以实现多域 TSN 的编排和管理。

上述方法有效地促进了 SDN 与 TSN 的集成,然而,它们大都针对特定目标而优化,没有从整体上讨论如何应对混合多域组网问题。为此,本文提出软件定义时间敏感网络(SD-TSN, software-defined time-sensitive networking)组网方案以应对工业互联网带来的混合多域组网挑战。具体研究内容如下。

1) 提出 SD-TSN 组网方案,引入协同控制器和域控制器,设计软件定义的多域时间同步机制以及多域时间感知调度机制,实现混合多域下确定性传输。

2) 构建 SD-TSN 组网实验平台,研发协同配置等关键模块以支持混合多域调度表的精准下发,实现端到端确定性传输的集中配置。

3) 基于组网平台,设计多种混合多域环境并对所提方案进行实际测试,详细分析不同环境下组网方案的性能。

1 混合多域组网方案

1.1 方案设计

基于软件定义控制与转发分离的范式以及 TSN 的完全集中配置模型,本文设计了分层、逻辑集中的 SD-TSN 组网方案,如图 1 所示。自下而上分为 3 层,分别为物理设施平面、编排配置平面和应用平面。

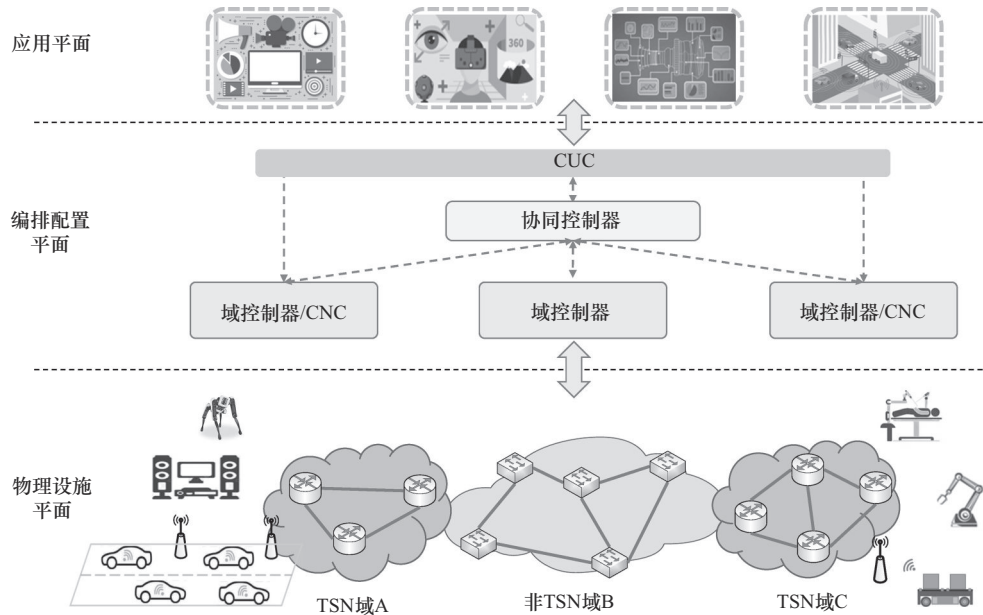


图1 SD-TSN组网方案

1) 物理设施平面

物理设施平面包括 TSN 交换机、非 TSN 交换机以及各种终端站。TSN 交换机支持 IEEE 802.1Q 规定的 TSN 机制（主要包括帧过滤、排队管理、传输选择等），从而能够对时间敏感流量进行精确的整形和调度。在 SD-TSN 中，相互连接的 TSN 交换机组成的网络域被称为 TSN 域，而不支持 TSN 机制的节点组成的网络域则被称为非 TSN 域。各种终端设备生成时间敏感流量并通过混合多域网络进行传输。

2) 编排配置平面

编排配置平面由域控制器、协同控制器和集中用户配置（CUC，centralized user configuration）组成。域控制器负责管理各自网络域，如 TSN 域的集中网络配置（CNC，centralized network configuration）。协同控制器与所有域控制器进行通信，协调它们之间的网络控制和调度策略。协同控制器和域控制器是 SDN 架构中比较常见的形态，针对混合多域场景，本文设计了相关组件以使得端到端传输的编排和组网实施。

域控制器包括拓扑发现、路径规划、流量调度、时间同步和流量统计 5 个关键组件。拓扑发现通过主动探测和被动监听等方式，收集网络设备类型、状态及其互联方式的信息，从而生成拓扑图。路径规划根据网络拓扑、流量需求以及调度策略，为流量计算传输路径。流量调度负责根据传输需求

以及网络状态生成域内流量传输调度表。时间同步负责管理和配置时间同步，如时间同步路径计算和维护、同步参数的计算以及配置等功能。流量统计负责收集和监测网络中的数据流量信息，用于分析流量模式、吞吐量和时延等关键指标，为编排管理提供支持。

协同控制器由拓扑管理、路径规划、同步管理、调度管理、连接管理和资源管理 6 个关键组件组成。拓扑管理通过收集每个域控制器上传的拓扑及节点信息，构建整个网络的拓扑信息图。路径规划根据全局拓扑信息和传输需求，计算并维护跨端到端传输路径。同步管理依托全局视图，将时间消息传递和误差补偿解耦，从而支持跨非 TSN 域的时间同步。调度管理以业务需求和网络状态为输入，协同编排各域并生成域内及域间调度策略，以实现有界低时延端到端传输。连接管理负责连接建立与管理等功能，以在不同网络域之间建立并维护连接关系，支持域间互操作。资源管理包括资源发现、分配以及管理等功能，具有整个网络的全局视图，为其他功能组件提供支持。

针对混合多域环境的复杂性，域控制器将根据不同非 TSN 域的特征进行设计与实现，以提供足够的适应性。例如，对于由 OpenFlow 交换机组成的非 TSN 域，域控制器应支持基于 OpenFlow 协议的规则定义、流表下发以及流量统计等功能。协同控制器具有全局视图，通过转换接口和域控制器交

换控制信令,从而实现确定性端到端编排。不同域控制器之间相互独立,有效地避免了单点故障导致的确定性服务停止。针对协同控制器可能存在的单点故障风险,可通过使用热备用和配置回退等方式解决^[22]。

CUC 和应用平面通信,负责将服务需求(如截止时间和循环周期)翻译并传递给域控制器和协同控制器,用于生成调度策略。当 CUC 收到服务请求时,它会识别出生成和接收流量的终端设备(即发送者和接收者)。随后,CUC 将发送者、接收者和业务需求信息发送给域控制器和协同控制器,以生成各个域流量传输调度表。最后,CUC 根据这些调度通知相应的终端设备,域控制器负责在各自域中下发调度表。

3) 应用平面

应用平面涵盖智能制造、远程医疗、汽车网络等新兴应用。借助编排配置平面的能力,操作人员可以通过高级语言提出各种需求,而无须关心底层实现细节。这些服务需求通过标准应用程序编程接口(如 RESTful API)传输到 CUC。经过解析编排后被进一步传递到协同控制器和域控制器中以进行进一步的策略规划。

1.2 多域网络模型

如图 1 所示,混合多域网络被建模为 $G^n = (V^n, E^n), n = 0, 1, 2$, 其中, V^n 是第 n 个网络的网络节点集合, E^n 是相应网络中的网络链路的集合。 G^0 表示非 TSN 域, $G^n (n \geq 1)$ 表示 TSN 域。 E^n 中的链路 $[v_a^n, v_b^n]$ 表示一条连接发送节点 v_a^n 和接收节点 v_b^n 的链路。考虑到任何端口最多连接到一个链路,端口与其关联的链路被认为是等效的。因此,链路 $[v_a^n, v_b^n]$ 所对应源节点 v_a^n 上端口的物理队列可表示为 $q_m^{[v_a^n, v_b^n]}$ 。网络中的周期性时间敏感流量 F 表示为

$$f_i = \{ \text{src}_i, \text{dst}_i, \text{dl}_i, T_i, l_i \}, \forall f_i \in F, i = 1, 2, \dots, |F| \quad (1)$$

其中, src_i 是源节点, dst_i 是目标节点, dl_i 是截止时间要求, T_i 是流的周期, l_i 是流长度。 f_i 在其源节点和目的节点之间的传输路径用 path_i 表示,包含从 src_i 到 dst_i 的一系列网络节点。在端到端传输中,它按照网络域被划分为多个路径段。在每个网络域内,出口和入口节点分别表示为 $v_{x_i}^n$ 和 $v_{y_i}^n$ 。此外, $\text{sp}(v_a^n)$ 表示 f_i 在 G^n 中的路径部分,该路径开始于 $v_{y_i}^n$ 、结束于 $v_{x_i}^n$ 。

在时间敏感流量的传输中, f_i 在链路 $[v_a^n, v_b^n]$ 上的传播时延和排队时延分别被表示为 $\text{pd}^{[v_a^n, v_b^n]}$ 和 $\text{qd}_i^{[v_a^n, v_b^n]}$ 。流 f_i 在节点 v_a^n 处的发送时延和处理时延分别表示为 $\text{td}_i^{v_a^n}$ 和 $\text{fd}_i^{v_a^n}$, 将 $[v_a^n, v_b^n]$ 处的带宽表示为 $B_{[v_a^n, v_b^n]}$, 则可以得到 $\text{td}_i^{v_a^n} = \frac{l_i}{B_{[v_a^n, v_b^n]}}$ 。考虑到网络中的

传播、处理和发送时延可以根据网络参数计算得到,本文进一步设置了一个参数表示链路 $[v_a^n, v_b^n]$ 上的这部分时延,具体为

$$\text{cd}_i^{[v_a^n, v_b^n]} = \text{td}_i^{v_a^n} + \text{pd}^{[v_a^n, v_b^n]} + \text{fd}_i^{v_b^n} \quad (2)$$

在 TSN 域中,队列 $q_m^{[v_a^n, v_b^n]}$ 中为流 f_i 设置的门打开时间表示为 $\text{ot}_{i,m}^{[v_a^n, v_b^n]}$ 。门处于打开状态的时间长度定义为门开启窗口,记为 $\text{ow}_{i,m}^{[v_a^n, v_b^n]}$ 。门控制列表(GCL, gate control list)的周期记为 $\text{gc}^{[v_a^n, v_b^n]}$ 。

2 确定性传输技术

2.1 多域时间同步机制

在混合多域环境中,非 TSN 节点普遍缺少对 IEEE 802.1AS 协议的支持,导致难以实现高精度时间同步。为此,本节设计了软件定义多域时间同步机制,将时间消息传递与误差补偿分开,使时间同步过程可编程,实现对同步请求和处理的逻辑集中控制。

1) 时间信息传递

首先,时间同步组件使用最佳主时钟算法从网络中选择一个节点作为基准时钟以提供参考时间,随后根据基准时钟以及节点同步性能构建同步生成树结构。在该同步树结构下,基准时钟周期性地发送同步报文,将参考时间分发给网络中的其他设备。从时钟设备接收到同步报文时会记录接收时间戳,并在自身的主时钟状态端口上启动新的同步报文传输。同时,设备会记录同步报文的发送时间戳。随着同步报文在同步生成树中的传播,每个后续节点执行此过程,记录时间戳并转发同步报文,直到同步生成树的叶节点。在同步报文传递的过程中,TSN 域中任意一对主从节点之间的时间偏移的计算式为

$$\eta = \frac{(T_2 - T_1) - (T_4 - T_3)}{2} \quad (3)$$

根据该值,从时钟节点的时间计算式为

$$T_S = \tau_S + \eta \quad (4)$$

其中, τ_S 为从节点的本地时钟。

2) 误差集中补偿

在混合多域环境中, 域A和域C之间跨域时间同步的时间偏移可以表示为

$$\eta_{AC} = \frac{(T_2^2 - T_1^1) - (T_4^1 - T_3^2)}{2} \quad (5)$$

其中, T_1^1 为时间消息离开域A的时间戳, T_2^2 为时间消息进入域C的时间戳, T_3^2 为时间消息离开域C的时间戳, T_4^1 为时间消息进入域A的时间戳。

如果这些时间戳足够精确, 域A和域C通过时间偏移的计算可以实现高精度的跨域时间同步。然而, 由于缺少对 IEEE 802.1AS 的支持, 同步报文在域B中的传输往往经历不对称的上下行时延, 使得该值难以用于误差补偿。为此, 本节设计了软件定义的集中误差补偿对跨非 TSN 域时间同步偏移值进行补偿。该方法主要包含2个部分: ①域控制器收集时间消息在非 TSN 域中传输时进出节点的时间戳信息; ②协同控制器根据收集的信息计算并补偿非 TSN 域引入的误差。该机制将依赖于时间消息交互的时间偏移补偿抽象到编排配置平面。基于全局视图和集中控制对时间偏移误差进行补偿, 补偿后的偏移值为

$$\hat{\eta}_{AC} = \eta_{AC} + \frac{1}{2} (d_{MS} - d_{SM}) \quad (6)$$

其中, d_{MS} 和 d_{SM} 分别代表时间消息在域B中经历的下行时延和上行时延。具体操作流程如下所述。

在非 TSN 域内, 时间同步组件首先规划同步报文的传输路径 path_{MS} , 令 $\{v_1^0, v_2^0, \dots, v_m^0\}$ 和 $\{e_1^0, e_2^0, \dots, e_n^0\}$ 分别表示同步路径上的 m 个节点和 n 条链路的集合, 其中 $e_j^0 = [v_j^0, v_{j+1}^0]$ 。则同步路径可以进一步表示为

$$\text{path}_{MS} = \{[v_1^0, v_2^0], [v_2^0, v_3^0], \dots, [v_{m-1}^0, v_m^0]\} \quad (7)$$

当同步报文沿着同步路径在非 TSN 域中传输时, 节点将记录步报文到达和离开节点的时间戳 t_{in} 和 t_{out} , 生成时间戳对 (t_{in}, t_{out}) , 并将其发送给时间同步组件。 d_{MS} 和 d_{SM} 分别表示下行时间同步路径和上行时间同步路径上的总时延, 考虑路径上的节点与链路, 有

$$d_{MS} = \sum_{v_i^0 \in \text{path}_{MS}} d(v_i^0)^k + \sum_{e_j^0 \in \text{path}_{MS}} d(e_j^0) \quad (8)$$

$$d_{SM} = \sum_{v_i^0 \in \text{path}_{SM}} d(v_i^0)^k + \sum_{e_j^0 \in \text{path}_{SM}} d(e_j^0) \quad (9)$$

其中, $d(v_i^0)^k$ 和 $d(e_j^0)$ 分别表示第 k 次时间同步过程中第 i 个节点的时间消息驻留时间和第 j 条链路的传播时延, path_{SM} 表示 path_{MS} 路径的相反方向。根据时间同步组件收集的 (t_{in}, t_{out}) , 第 i 个节点中的驻留时间 $d(v_i^0)^k$ 可被表示为

$$d(v_i^0)^k = t_{out}(i)^k - t_{in}(i)^k \quad (10)$$

根据链路时延系数 ω 和链路长度, 每段链路上的传播时延可表示为

$$d(e_j^0) = \omega_j \text{Len}(e_j^0) \quad (11)$$

其中, 时延系数表示单位长度链路上传播 1 bit 所需的时间, 例如, 在光纤网络中该值为 5 ns/m; $\text{Len}(e_j^0)$ 代表链路 e_j^0 的长度。

因此, 在第 k 次时间同步中, 下行链路与上行链路之间的时延差可以表示为

$$\begin{aligned} d^k &= d_{MS} - d_{SM} = \\ &= \sum_{v_i^0 \in \text{path}_{MS}} d(v_i^0)^k + \sum_{e_j^0 \in \text{path}_{MS}} d(e_j^0) - \\ &= \left(\sum_{v_i^0 \in \text{path}_{SM}} d(v_i^0)^k + \sum_{e_j^0 \in \text{path}_{SM}} d(e_j^0) \right) = \\ &= \sum_{v_i^0 \in \text{path}_{MS}} (t_{out}(i)^k - t_{in}(i)^k) + \sum_{e_j^0 \in \text{path}_{MS}} \omega_j \text{Len}(e_j^0) - \\ &= \sum_{v_i^0 \in \text{path}_{SM}} (t_{out}(i)^k - t_{in}(i)^k) - \sum_{e_j^0 \in \text{path}_{SM}} \omega_j \text{Len}(e_j^0) \end{aligned} \quad (12)$$

根据该值进行集中补偿, 域C的时间为

$$T_C = \tau_C + \eta_{AC} + \frac{1}{2} \times d^k \quad (13)$$

总体而言, SD-TSN 通过软件定义的范式将时间消息的传递和误差补偿解耦, 以支持跨非 TSN 域的时间同步。即使时间同步消息在非 TSN 域中经历严重拥塞, 集中式的误差测量仍能通过时间戳对信息补偿时间消息在非 TSN 域中经历的上下行不对称时延, 从而保证时间同步精度。此外, 面对同步路径上的节点故障, TSN 域可通过 IEEE 802.1AS 协议机制实现同步的恢复。在非 TSN 域中, 域控制器通过更新同步路径并收集相应的时间戳对实现同步恢复。

2.2 多域时间感知调度机制

在流量的端到端传输中, 时延由传播时延、发送时延、处理时延和排队时延4个部分组成, 其中前3个时延可以通过全局视图计算并被确定。当多

个流量试图同时通过同一输出端口传输时，其输出端口上就会发生排队。网络中的排队时延通常是不确定，这也是传输不确定的主要原因。

为此，本文设计了多域协同时间感知调度机制，通过协调 TSN 域中的 GCL 时间和非 TSN 域中的传输时隙，以限制端到端传输中的排队时延从而实现有界低时延和低抖动。具体来讲，在 TSN 域内，SD-TSN 采用 TAS 机制对时间敏感流量进行精准调度。在非 TSN 域内，域控制器首先根据流量特征划分传输时隙。针对从 TSN 域到非 TSN 域的传输，联合考虑注入时隙和路径规划，以确保时间敏感流量在非 TSN 域内的传输路径上，其传输时隙被专属保留，即不受其他流量的干扰。考虑到不经 TSN 域直接进入非 TSN 域的非时间敏感流量，SD-TSN 依托全局视图进行时延测量，并在路径规划阶段选择时延最短路径进行传输。TSN 域控制器还将该测量值作为 GCL 计算的输入参数，以消除非 TSN 域中非时间敏感流量导致的时延抖动。此外，非 TSN 域控制器还可使用入口整形和流量控制等方式进一步优化时间敏感流量的传输，以限制非时间敏感流量的干扰。

在 TSN 域中，多域时间感知调度机制规划了周期约束、传输约束以及无冲突约束等域内传输条件以实现 TSN 域内的精准调度（详细约束参考文献[23]）。对于非 TSN 域内的传输，SD-TSN 通过规划流量进入该域的传输时隙，以确保所有时间敏感流量在传输过程中互不干扰。首先，每个时间敏感流进入非 TSN 域的注入时间不应相互重叠，以防止出现流量拥塞，可以描述为

$$\begin{aligned} & \forall G^n, n \geq 1, \forall f_i, f_j \in F, v_{x_i}^n = v_{y_j}^n, i \neq j, \\ & \forall \alpha \in \left[0, \frac{\text{gc}^{[v_{x_i}^n, v_{y_i}^n]}}{T_i} - 1 \right], \forall \beta \in \left[0, \frac{\text{gc}^{[v_{x_j}^n, v_{y_j}^n]}}{T_j} - 1 \right]: \\ & (\text{ot}_{i,m}^{[v_{x_i}^n, v_{y_i}^n]} + \alpha \times T_i - \delta \geq \text{ot}_{j,m'}^{[v_{x_j}^n, v_{y_j}^n]} + \text{ow}_{j,m'}^{[v_{x_j}^n, v_{y_j}^n]} + \beta \times T_j) \vee \\ & (\text{ot}_{j,m'}^{[v_{x_j}^n, v_{y_j}^n]} + \beta \times T_j - \delta \geq \text{ot}_{i,m}^{[v_{x_i}^n, v_{y_i}^n]} + \text{ow}_{i,m}^{[v_{x_i}^n, v_{y_i}^n]} + \alpha \times T_i) \end{aligned} \quad (14)$$

其中， $\alpha \times T_i$ 代表流 f_i 在 GCL 周期内第 α 次出现时对应的整数周期倍， $\beta \times T_j$ 代表流 f_j 在 GCL 周期内第 β 次出现时对应的整数周期倍。

其次，结合协同控制器的全局视图，传播时延、发送时延、处理时延和传输路径是已知的。通

过进一步调整 TSN 域内出口节点流的离开时间，可以达到零排队时延传输的目标，约束描述为

$$\begin{aligned} & \forall [v_a^0, v_b^0] \in E^0, \forall f_i, f_j \in F^{[v_a^0, v_b^0]}, i \neq j, \\ & \forall \alpha \in \left[0, \frac{\text{hp}(i,j)}{T_i} - 1 \right], \forall \beta \in \left[0, \frac{\text{hp}(i,j)}{T_j} - 1 \right]: \\ & (\text{st}_i^0 + \sum_{k \in \text{sp}_i^0} \text{cd}_i^{[k,k+1]} + \text{td}_i^{v_a^0} + \alpha \times T_i \geq \\ & \text{st}_j^0 + \sum_{k \in \text{sp}_j^0} \text{cd}_j^{[k,k+1]} + \beta \times T_j) \vee \\ & (\text{st}_j^0 + \sum_{k \in \text{sp}_j^0} \text{cd}_j^{[k,k+1]} + \text{td}_j^{v_a^0} + \beta \times T_j \geq \\ & \text{st}_i^0 + \sum_{k \in \text{sp}_i^0} \text{cd}_i^{[k,k+1]} + \alpha \times T_i) \end{aligned} \quad (15)$$

其中， $\text{st}_i^0 = \text{ot}_{i,m}^{[v_{x_i}^0, v_{y_i}^0]} + \text{cd}_i^{[v_{x_i}^0, v_{y_i}^0]}$ 表示流 f_i 在非 TSN 域开始传输的时间， $\text{hp}(i,j) = \text{Lcm}(T_i, T_j)$ 是 f_i 和 f_j 的最小公倍数，也称为流 f_i 和 f_j 的超周期。

考虑到非时间敏感流可能随时生成，从而影响时间敏感流量的传输，本文设计了基于集中控制的时延测量补偿优化策略。协同控制器通知发送端周期性地发送用于非 TSN 域时延测量的时敏测量流，记为 f_0 ，该流的周期被称为测量周期。每个网络节点 v_a^n 的入口时间戳 $t_{\text{in}}^{[v_a^n - 1, v_a^n]}$ 和出口时间戳 $t_{\text{out}}^{[v_a^n, v_a^n + 1]}$ 都被记录并构造为时间戳对 $(t_{\text{in}}^{[v_a^n - 1, v_a^n]}, t_{\text{out}}^{[v_a^n, v_a^n + 1]})$ 。时间戳对由域控制器收集并发送到协同控制器以构建时延视图。根据收集到的时间戳，在第 h 个测试周期中非 TSN 域内的路径段的时延为

$$\text{ed}_{i,h}^0 = t_{\text{out},h}^{[v_{x_i}^0, v_{x_i}^0 + 1]} - t_{\text{in},h}^{[v_{y_i}^0 - 1, v_{y_i}^0]} - \gamma_i \quad (16)$$

其中， $\gamma_i = \sum_{k \in \text{sp}(v_{x_i}^0)} (\text{cd}_i^{[k,k+1]} - \text{cd}_0^{[k,k+1]})$ 反映了 f_i 和 f_0 之间时延的差异。则非 TSN 域中流 f_i 的最坏情况时延为

$$\theta_i^0 = \max \{ \text{ed}_{i,h}^0 | h = 1, 2, \dots, H \} \quad (17)$$

基于该测量时延，为了消除尽力而为流带来的抖动波动，从非 TSN 域到 TSN 域的域间时隙对齐约束为

$$\begin{aligned} & \forall f_i \in F, \forall [v_{x_i}^n, v_{y_i}^0], [v_{x_i}^0, v_{y_i}^{n'}], n \geq 1, n' \geq 1: \\ & \text{ot}_{i,m}^{[v_{y_i}^n, v_{y_i}^{n'} + 1]} \geq \text{st}_i^0 + \theta_i^0 + \text{pd}^{[v_{x_i}^0, v_{y_i}^{n'}]} + \text{fd}_i^{v_{y_i}^{n'}} \end{aligned} \quad (18)$$

3 SD-TSN 组网实验平台

SD-TSN 在工业互联网中的推广和应用，既依

赖于理论研究的创新，也离不开在实际环境中的功能验证和性能评估。为此，本节构建了SD-TSN组网平台，研发了时间同步、流量调度和协同配置等关键模块，如图2所示。

3.1 关键模块

1) 时间同步模块

基于时间同步相关组件，使用LinuxPTP套件管理和配置时间同步相关功能，主要包括精确时间协议（PTP，precision time protocol）、用于Linux的PTP（ptp4l，PTP for Linux）、PTP硬件时钟到系统（phc2sys，PTP hardware clock to system）等模组。主要功能包括：①收集节点时间同步状态信息、配置设备节点同步功能、配置更新和时钟源选择等；②根据节点信息与网络拓扑进行同步路径管理，负责生成同步生成树以及相应的管理策略等；③根据多域时间同步机制进行同步集中管理，实时监测网络时间同步性能，收集网络节点和网络链路的时间戳数据并计算时间不对称时延，对跨非TSN域传输中的不对称时延进行集中补偿。此外，网络接口卡（NIC，network interface card）上的接地（GND，ground）和串行数据端口（SDP，serial data port）为设备提供了串行通信数据传输。

2) 流量调度模块

基于流量调度相关组件，该模块以流量传输需求、时间同步精度以及网络状态等参数作为输入，采用多域协同调度机制生成调度表对流量端到端传输进行统一编排调度。主要求解器包括GNU线性编程工具包（GLPK，GNU linear programming kit）、稳定基线3（SB3，stable-baseline3）和优化工具（OR-Tool，optimization tool）。Linux内核需支持英特尔千兆网卡（igb，Intel Gigabit）驱动、时间感知优先级整形（Taprio，time-aware priority shaper）以及最早TxTime优先（ETF，earliest tx-time first）等功能。NIC需支持发送时刻以允许数据在精确的时间点被硬件发送。生成的调度策略包含端设备的发送时刻，应通知流量生成模块进行精准发送，等时线（Isochron）被用于完成该功能。此外，以太网工具（ethtool）、iproute2工具（iproute2）以及数据捕获库（libpcap，packet capture library）等工具为流量调度提供了必要的支持。

3) 协同配置模块

协同配置模块负责根据网络设备节点，将流量调度模块生成的端到端调度策略转换为特定于设备的底层配置命令并进行下发。针对网络中存在的不同的网络域，该节点具备接口抽象功能，以屏蔽底

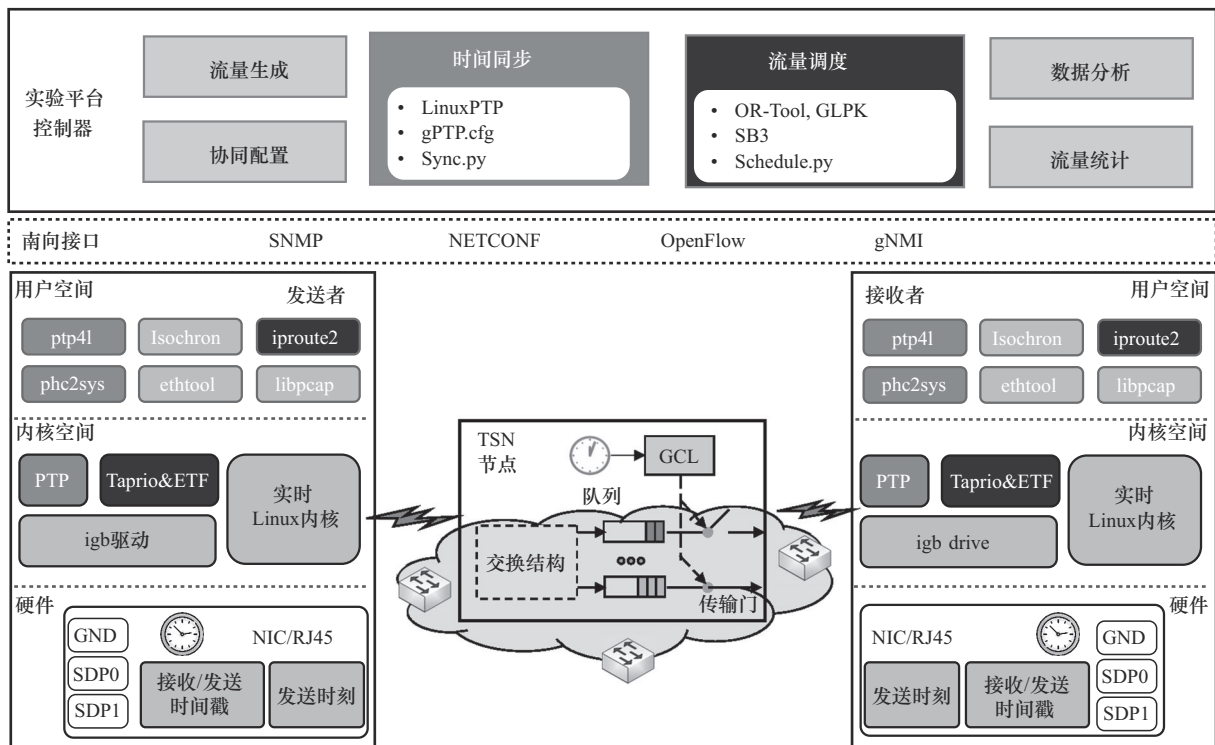


图2 SD-TSN组网平台关键模块

层设备和协议的差异。该模块支持多种主流网络管理接口，包括简单网络协议 (SNMP, simple network management protocol)、网络配置协议 (NETCONF, network configuration protocol)、基于 gRPC 的网络管理接口 (gNMI, gRPC network management interface) 以及 OpenFlow 等。

在实际应用中，一个关键问题是如何保证流量的发送与配置的 GCL 是匹配的，即保证时间敏感流量到达节点时相应 GCL 开始运行，从而使传输门按照调度策略精确地管理流量的传输。在实际测试中，即使控制器已计算出流发送时间以及网络中 GCL 运行时间，但由于时间同步精度、发送端抖动以及门控抖动等不可避免的原因，流量与门控行为之间往往难以对齐。为此，本节设计了周期增量偏移方法以确保测试过程中的门控匹配。

传统配置方式设置流量发送时间以及各个节点上的 GCL 生效时间，从而保证流量在规定时刻内按照门控调度策略实现精准传输。如图 3 中描述的传统配置方式，流 f_1 在 t_0 时刻被发出，链路 $[v_1, v_2]$ 上的 GCL 在 t_1 时刻开始执行，则该时间窗口允许流 f_1 进行传输，并且阻碍了其他流量的传输以防止干扰。然而，流发送时间与 GCL 不对齐问题带来的累积偏差将导致流量的错误传输行为。例如，在链路 $[v_n, v_{dst}]$ 上， f_1 早于 GCL 运行时间到达节点会

导致其遭受额外排队时延； f_1 晚于 GCL 运行时间到达节点，剩余的可用时间窗口资源可能无法支持流量的传输。

周期增量偏移方法结合流量周期和 Unix 时间戳，将流量发送时间与 GCL 的运行时间解耦，如图 3 所示，主要包括 2 个步骤。

① 发送端周期偏移

在发送端，流量发送时间 t_0 的计算式为

$$t_0 = (\text{UnixTS} \bmod f_1.\text{period} + Z) f_1.\text{period}, Z \in \mathbb{Z}^+ \quad (19)$$

其中，UnixTS 代表了当前时间戳， Z 为一个正整数。通过该方式，流 f_1 的发送时刻被安排在从 1970 年 1 月 1 日 00:00:00 UTC 起计算的、即将到来的周期整数倍时刻。

② GCL 运行时间偏移

各个交换机上 GCL 的运行时间的计算式为

$$t_n = (\text{UnixTS} \bmod f_1.\text{period} + Z) f_1.\text{period} + \text{path}_{f_1}^{v_n}.\text{delay}, Z \in \mathbb{Z}^+ \quad (20)$$

其中， $\text{path}_{f_1}^{v_n}.\text{delay}$ 表示流 f_1 从源节点 v_{src} 开始到节点 v_n 所经历的时延。式(20)等号右边第一项通过流量周期和 Unix 时间戳，确保 GCL 初始运行时间和流量发出时间保持一致；第二项通过引入流量在网络中的传输时延，对初始运行时间进行偏移，从而保证 GCL 和流量传输严格匹配。

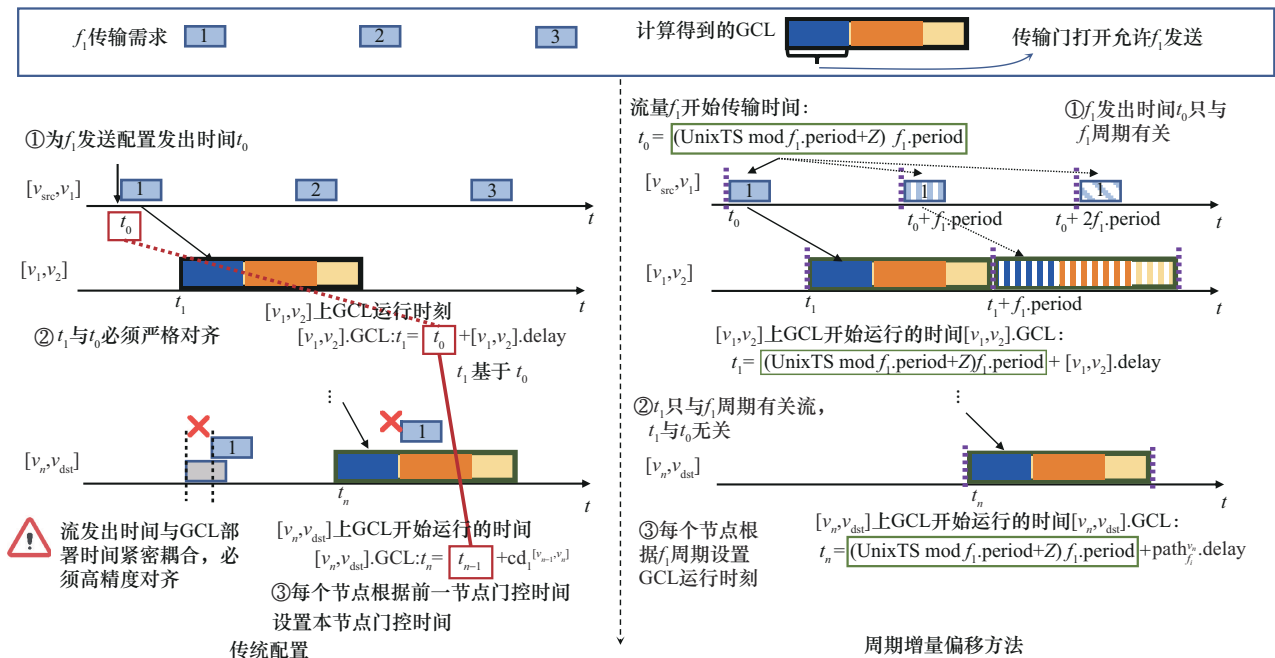


图 3 周期增量偏移方法

3.2 测试流程

基于 SD-TSN 组网平台设计, 一个组网测试流程如图 4 所示, 具体包括以下步骤。

- 1) 测试需求输入与解析: 获取并解析测试项的特征和测试目的, 将相关信息写入资源数据库。
- 2) 时间同步: 使用软件定义多域时间同步机制进行集中误差补偿, 实现跨域时间同步, 并定期校准同步精度, 以确保发送者和接收者的时间保持一致。
- 3) 流量调度: 采用多域时间感知调度机制生成各个网络域的调度表。
- 4) 流量生成: 根据测试项要求和调度表, 生成符合特定场景需求的测试流量, 并通过门控机制实现测试流量精准发送。
- 5) 协同配置: 将调度表转换为特定于设备的底层配置命令, 并通过门控周期偏移方式精准下发到各个域。
- 6) 流量统计: 测试项运行后, 实时收集网络中流量传输的特征和时间戳信息, 将相关数据写入资源数据库中。为时间同步和流量调度等功能的优化提供支撑, 同时为后续分析提供数据基础。
- 7) 数据分析: 对收集的数据进行整理和分析,

得出时延、抖动和可调度性等关键指标的性能, 结合输入的测试项, 评估实验结果。

4 平台搭建及组网实验

4.1 平台搭建

SD-TSN 组网实验平台如图 5 所示, 由控制器、端设备(发送者/接收者)以及多域网络(域 A、域 B 和域 C)组成。其中, 端设备由配备运行频率为 3.4 GHz 的 Intel Xeon E2224 处理器和 16 GB RAM 的服务器充当。控制器是一台配备运行频率为 2.4 GHz 的 Intel Xeon E4210 处理器和 32 GB RAM 的服务器。所有的端设备都装配 Intel I210 网卡以支持硬件时间戳以及高精度的流量操作。混合多域网络包含 3 个不同的网络域, 由 6 个交换机逐个连接组成, 链路为全双工以太网链路, 链路带宽为 1 Gbit/s。具体来讲, 混合多域网络包含 4 种场景。

1) 场景 1: TSN+开发式虚拟交换机(OvS, open vSwitch), 其中域 A 和域 C 中的交换机为 TSN 交换机, 域 B 中为基于 OvS 的交换机。该场景通过 OvS 模拟常见商用交换机, 控制器不对域 B 进行优化与操作, 只从全局层面进行跨域协同操作。

2) 场景 2: TSN+OvS+数据平面开发工具包

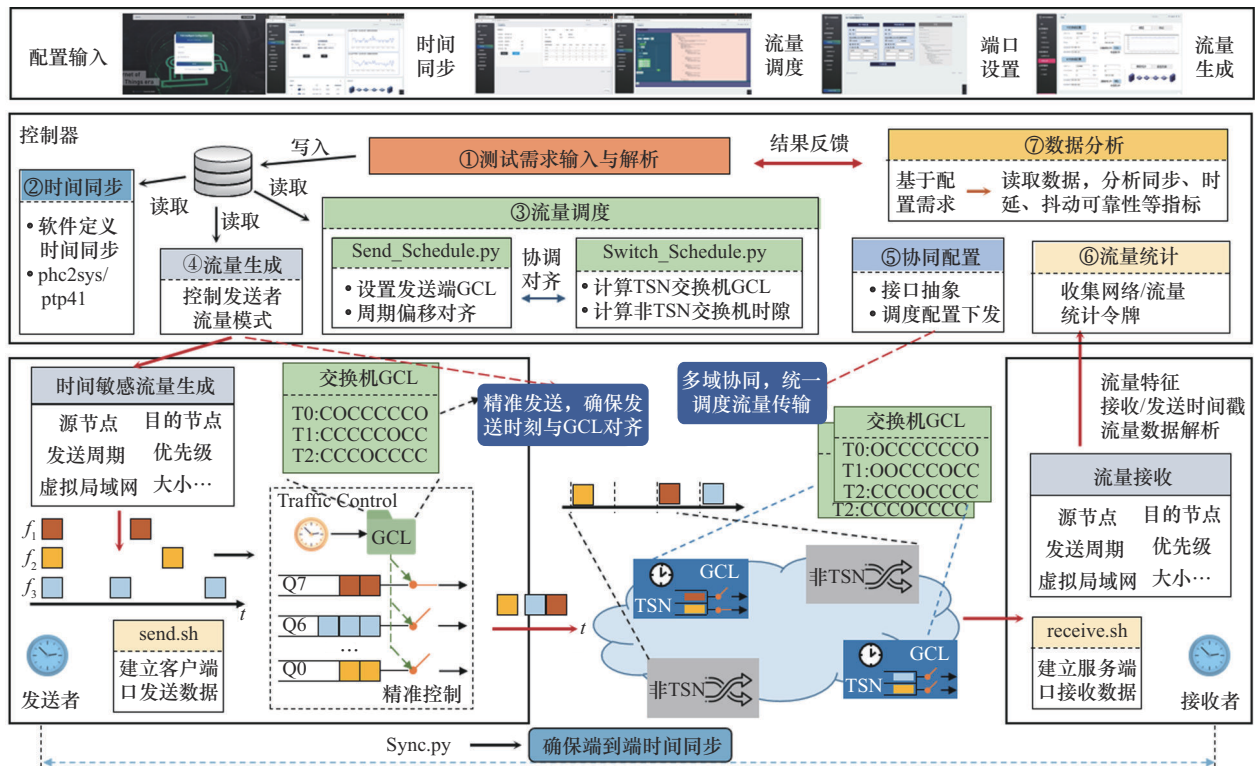


图 4 SD-TSN 组网测试流程

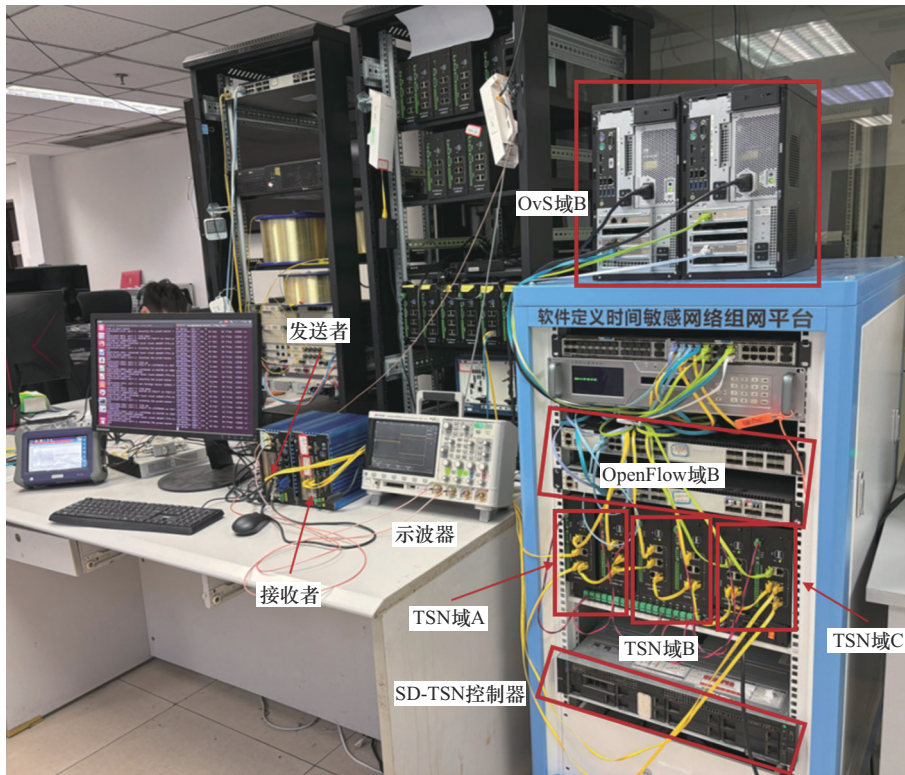


图5 SD-TSN组网实验平台

(DPDK, data plane development kit) + 流量控制 (TC, traffic control), 该场景中域 A 和域 C 中的交换机为 TSN 交换机, 域 B 中的交换机为 OvS。域控制器使用 DPDK 和 TC 对流量转发进行优化, 以增强端到端传输性能。

3) 场景 3: TSN+OpenFlow, 该场景中域 A 和域 C 中的交换机为 TSN 交换机, 域 B 中为 OpenFlow 交换机, 使用速率限制以及队列管理对域 B 进行优化, 以提升端到端传输性能。

4) 场景 4: 对照组。TSN 域 A、域 B 和域 C 中的交换机全部为 TSN 交换机, 该场景为 TSN 多域, 用于对混合多域场景结果进行对比。

4.2 实验设置

发送者将根据测试需求生成若干时间敏感流量, 接收者负责接收。根据 IEC/IEEE 60802 中对工业用例的描述, 测试的时间敏感流量周期从周期集合 {0.5,1,2,4,8,16,32,64,128} ms 中随机选择, 所有时间敏感流量的长度为 64 B, 端到端时延界限被设置为自身的周期。

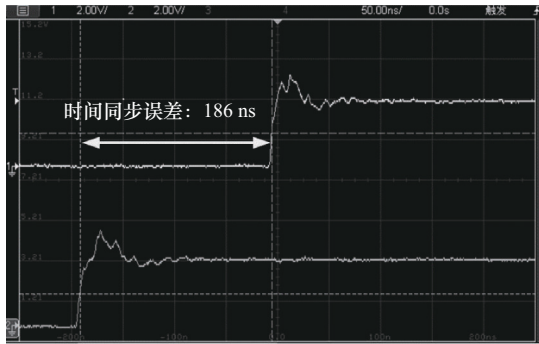
4.3 实验结果

4.3.1 时间同步性能

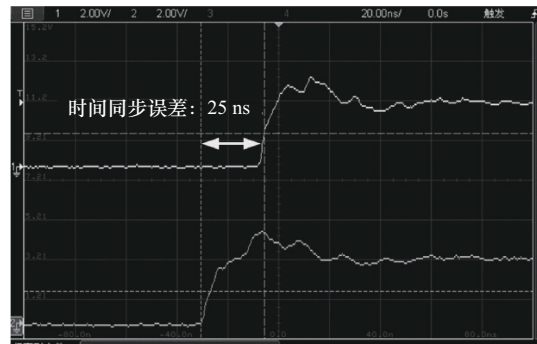
本节测试了 SD-TSN 在各个混合多域场景下的

时间同步性能, 如图 6 所示。在没有背景流量的情况下, 多域场景 1 中的最大时间同步误差达到 186 ns, 在多域场景 2 中为 25 ns, 只有多域场景 1 中的 13.44%。一方面, TC 操作优化了时间同步消息的传递, 避免了过大的上下行不对称时延。另一方面, DPDK 提供了高性能数据包处理能力, 允许设备绕过网络堆栈转发数据包, 进一步增强了 TC 操作的精度。

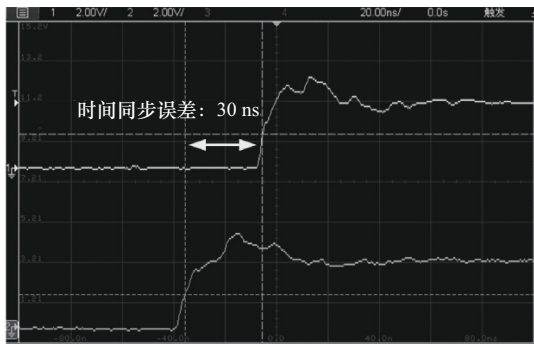
此外, 本节通过流量生成模块向非 TSN 域注入尽力而为流量作为背景流, 以研究 SD-TSN 在非时间敏感流量干扰情况下的同步性能。图 6(e)展示了未进行误差补偿时的最大时间同步误差。由图 6(e)可知, 在 TSN 多域组网中, 门控机制隔离了时间消息和其他流量, 因此背景流量负载没有影响时间同步性能, 其最大时间同步误差保持在 50 ns 以内。在其余混合多域环境中, 最大时间同步误差则在背景流量负载大于 30% 后呈现上升趋势。尤其在 TSN+OvS 混合多域场景中, 当背景流量负载达到 80% 时, 最大时间同步误差为 18 512 ns, 难以支持网络时间同步保持。这主要是因为背景流量的增加, 非 TSN 域内时间消息传递经历了拥塞, 进而导致较大的上下行不对称时延。图 6(f)展示了经过误差集中



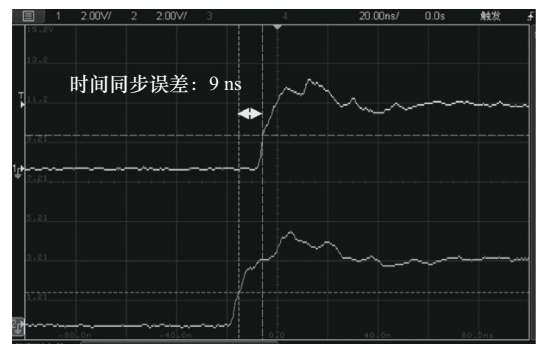
(a) TSN+OvS混合多域时间同步误差



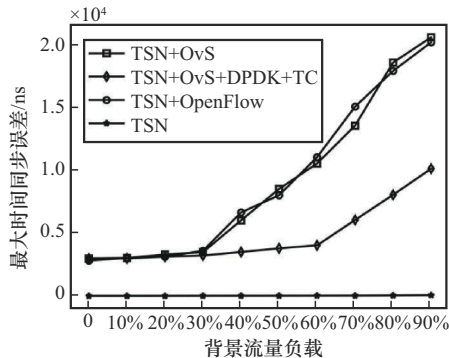
(b) TSN+OvS+DPDK+TC混合多域时间同步误差



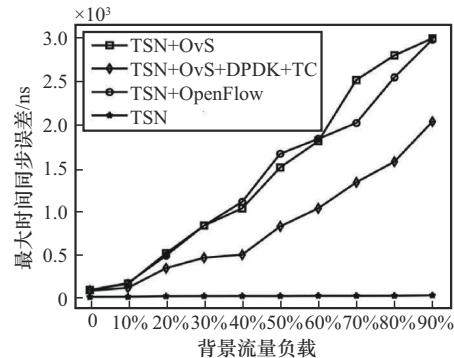
(c) TSN+OpenFlow混合多域时间同步误差



(d) TSN多域时间同步误差



(e) 未进行误差补偿时的最大时间同步误差



(f) 误差集中补偿后的最大时间同步误差

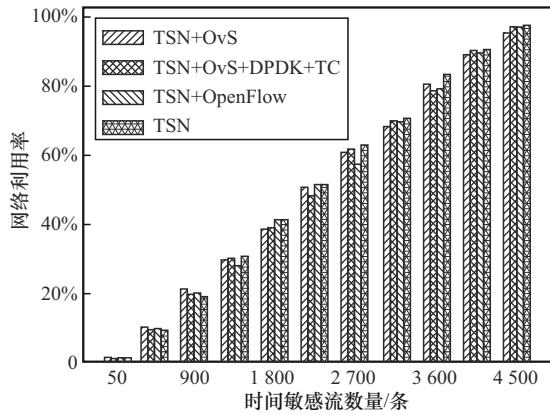
图6 时间同步性能

补偿后，SD-TSN在混合多域网络中的时间同步性能。在TSN+OvS多域环境中，集中误差补偿有效地将最大时间同步误差限制在3 000 ns以内，减少了大约85%。在TSN+OvS+DPDK+TC多域环境中，TC操作为时间同步消息的传递预留了时间窗口，减轻了背景流量的干扰。尤其是，在背景负载小于60%时，最大时间同步误差被控制在1 000 ns以内，能够有效支持高精度流量调度。

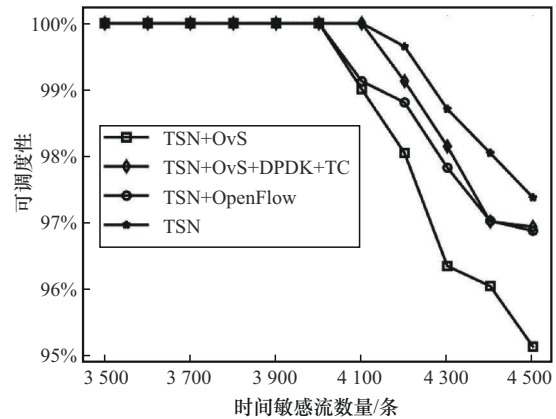
4.3.2 流数量与端到端传输性能

本节探究了在混合多域环境中，SD-TSN在不同时间敏感流数量下的性能，如图7所示。图7(a)

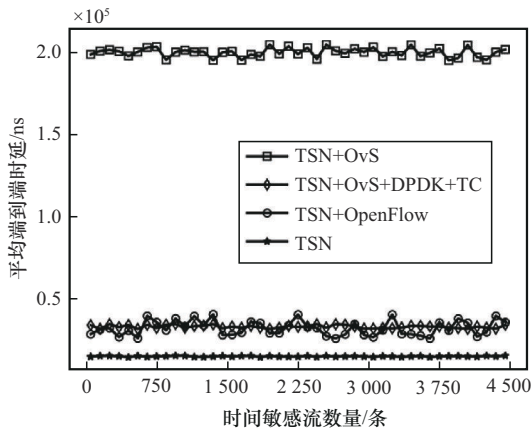
展示了不同流数量下网络利用率。由图7(a)可知，随着流数量的增加网络利用率呈上升趋势，当流数量为4 500时，SD-TSN在各个混合多域中的网络利用率平均为97%，表明所提方案能够充分利用网络资源。图7(b)展示了可调度性与时间敏感流数量的关系，其中可调度性代表成功调度的流量占需要调度流量的比例。在流数量小于4 000时，所有流量都能够被成功调度。之后，网络容量开始减少，导致无法被成功调度的时间敏感流数量开始增加，因此可调度性开始下降。端到端传输时延结果如图7(c)和图7(d)所示。由图7(c)和图7(d)可知，



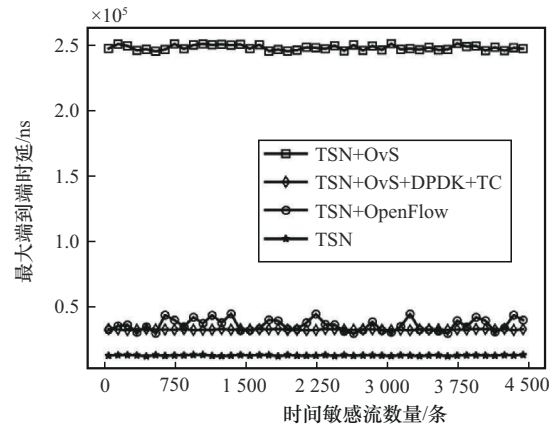
(a) 不同时间敏感流数量下的网络利用率



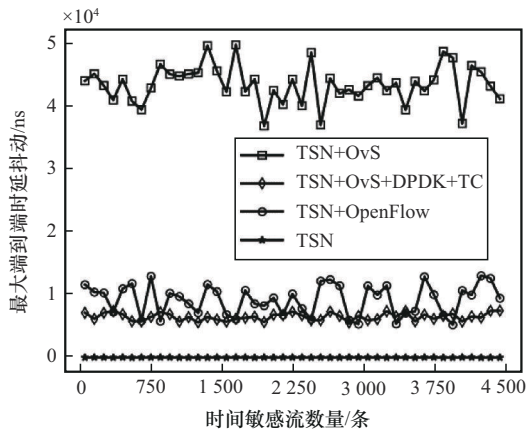
(b) 不同时间敏感流数量下的可调度性



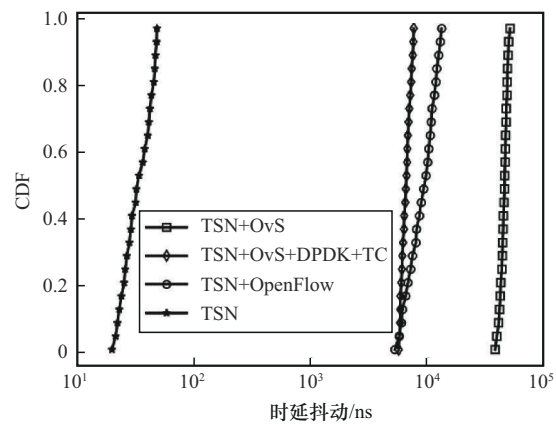
(c) 不同时间敏感流数量下平均端到端时延



(d) 不同时间敏感流数量下最大端到端时延



(e) 不同时间敏感流数量下的最大端到端时延抖动



(f) 抖动的累积分布函数

图 7 不同流数量下端到端传输性能

SD-TSN 在所有混合多域环境中都具有有界的最大时延。其中，在 TSN+OvS 混合多域场景中，最大端到端时延为 251 585 ns。在混合多域场景 2 中，SD-TSN 使用 DPDK 对 OvS 转发进行优化，并在相应端口设置了 TC 优化。因此，最大端到端时延上界为 34 986 ns，缩小了约 86%。图 7(e) 展示了不同流数量下最大端到端时延抖动。在未对非 TSN 域

进行优化的 TSN+OvS 混合多域环境中，SD-TSN 可将端到端时延抖动控制在 49 648 ns 以内。在进行 TC 优化后，TSN+OvS+DPDK+TC 多域环境中的最大端到端时延抖动减少了 80%，被限制在 7 489 ns 以内。图 7(f) 是这些抖动的累积分布函数 (CDF)。由图 7(f) 可知，TSN+OvS 多域组网环境中的抖动范围最大且抖动值最高。

4.3.3 负载与端到端传输性能

本节进一步探究非时间敏感流量干扰下的端到端传输性能，设置网络中传输的时间敏感流数量遵循正态分布，均值为1 500，标准差为33，即时间敏感流数量在[1 400,1 600]范围波动。非时间敏感流数量逐步增加。具体来讲，非时间敏感流数量通过iperf3生成，并直接注入到非TSN域内。负载 ρ 表示非时间敏感流与时间敏感流所占带宽之比， ρ 越大代表背景流干扰越强。

图8(a)展示了不同负载 ρ 下的网络利用率。当 ρ 为2.0时，SD-TSN在各个混合多域中的网络利用率平均为90.86%，此时时间敏感流占用带宽平均为33.96%，非时间敏感流占用带宽平均为56.9%。这表明所提方案在传输时间敏感流数量的同时也能为非时间敏感流数量的传输提供足够的支持。图8(b)展示了不同负载 ρ 下的可调度性。在TSN+OvS混合多域中，SD-TSN的可调度性在负载 ρ 大

于1.2之后开始下降。通过对非TSN域优化，SD-TSN在其他2种混合多域环境中则表现出更强的可调度性，在负载 ρ 大于1.4之后开始下降，此时网络利用率为75%左右。非时间敏感流量干扰下的端到端传输时延和时延抖动如图8(c)和图8(d)所示。其中，TSN+OvS场景中的端到端性能对 ρ 值最为敏感，即随着 ρ 值的增加上升趋势最为明显。这是因为时间敏感流量在非TSN域中会被过多的背景流量干扰，导致了额外的排队时延。然而，在控制优化后，即在TSN+OvS+DPDK+TC多域场景中，组网方案的传输性能得到了显著增强，最大时延和时延抖动分别只有165 251 ns和25 145 ns。

4.3.4 综合对比

本节选择C-TSDN^[16]、确定性跨域流调度^[14](DCDFS, deterministic cross-domain flow scheduling)以及TSSDN^[18]3种典型方案作为对比对象，对所提SD-TSN方案进行了综合对比验证。

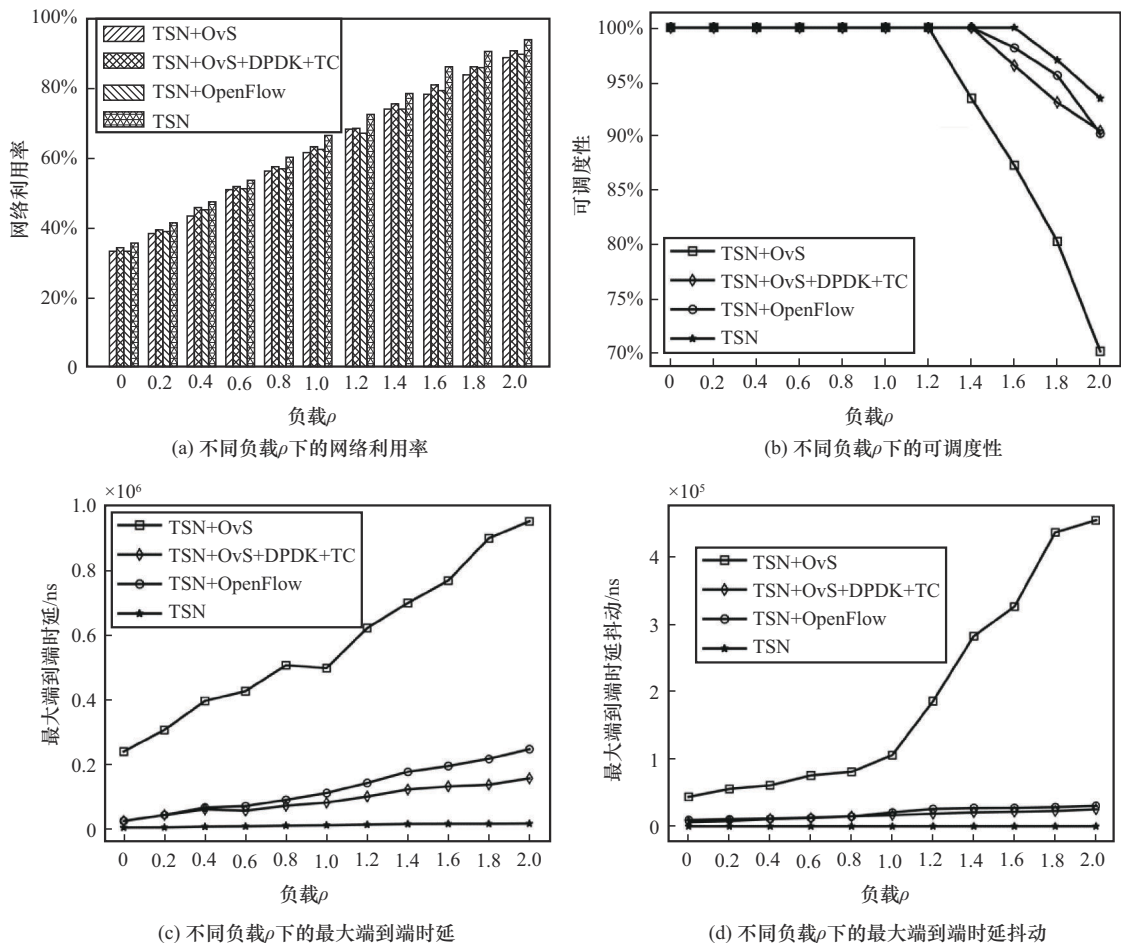


图8 不同负载与端到端传输性能

图9展示了不同混合多域下各个方案的可调度性结果。首先，得益于各个网络域的集中编排管理，SD-TSN的可调度性优于现有方案。相比之下，C-TSDN和DCDFS中的循环切换乒乓队列机制导致部分流量无法满足其传输需求，从而降低了可调度性。TSSDN在各种混合多域环境中都取得了相似的结果。这是因为该方案只在发送端处理时间敏感流量，对网络节点没有特别要求。然而，由于节点流量处理能力的缺失，TSSDN的可调度性下降最为显著。

图10比较了不同混合多域环境中各个方案的最大端到端传输时延。C-TSDN和DCDFS在各个域规划接收时隙和发送时隙，通过循环切换乒乓队列实现流量整形，从而保证有界时延。这种方式易用性高但是也导致了最低传输时延 $(hop_{num}-1)T_{slot}$ 和最高传输时延 $(hop_{num}+1)T_{slot}$ 的时延范围，其中 hop_{num} 表示流量传输路径上交换机的数量， T_{slot} 表示时隙的长度。因此，在各个混合多域中，C-TSDN和DCDFS的端到端时延都处于一定范围内，最小值

为560 110 ns，最大值为715 364 ns。TSSDN采用无等待传输方式在发送端调度时间敏感流量，即端到端传输中的排队时延为0。因此，该方式取得了最短的传输时延，平均仅有7 300 ns。此外，SD-TSN在各个混合多域中都能够实现有界低时延传输。尤其在TSN多域环境中，SD-TSN取得了14 833 ns的平均时延，与无等待传输十分接近。而在TSN+OvS混合多域环境中，由于OvS交换机的时间同步稳定性和转发精度等因素，SD-TSN的平均时延上升到248 641 ns，但相对于DCDFS方案仍减少了64.51%。

图11展示了非时间敏感流干扰下不同混合多域中各个方案的性能。具体来讲，时间敏感流的数量遵循正态分布，均值为1 000，标准差为33，负载 ρ 设置为1.0。图11(a)比较了不同混合多域下各个方案的可调度性。得益于集中控制和全局优化，SD-TSN在各个混合多域中的调度性都达到了100%。由于队列的循环切换，C-TSDN和DCDFS中部分流量无法在其截止时间内完成传输，因此平

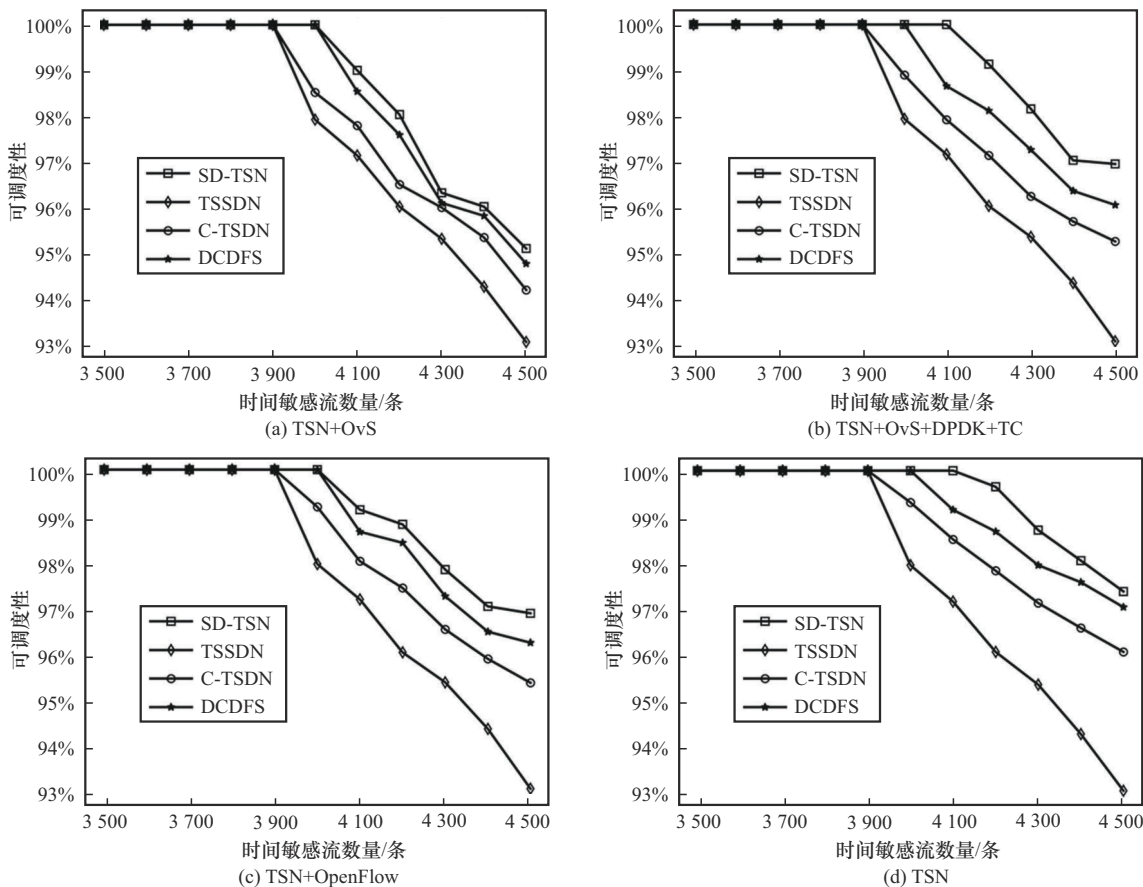


图9 不同混合多域可调度性对比

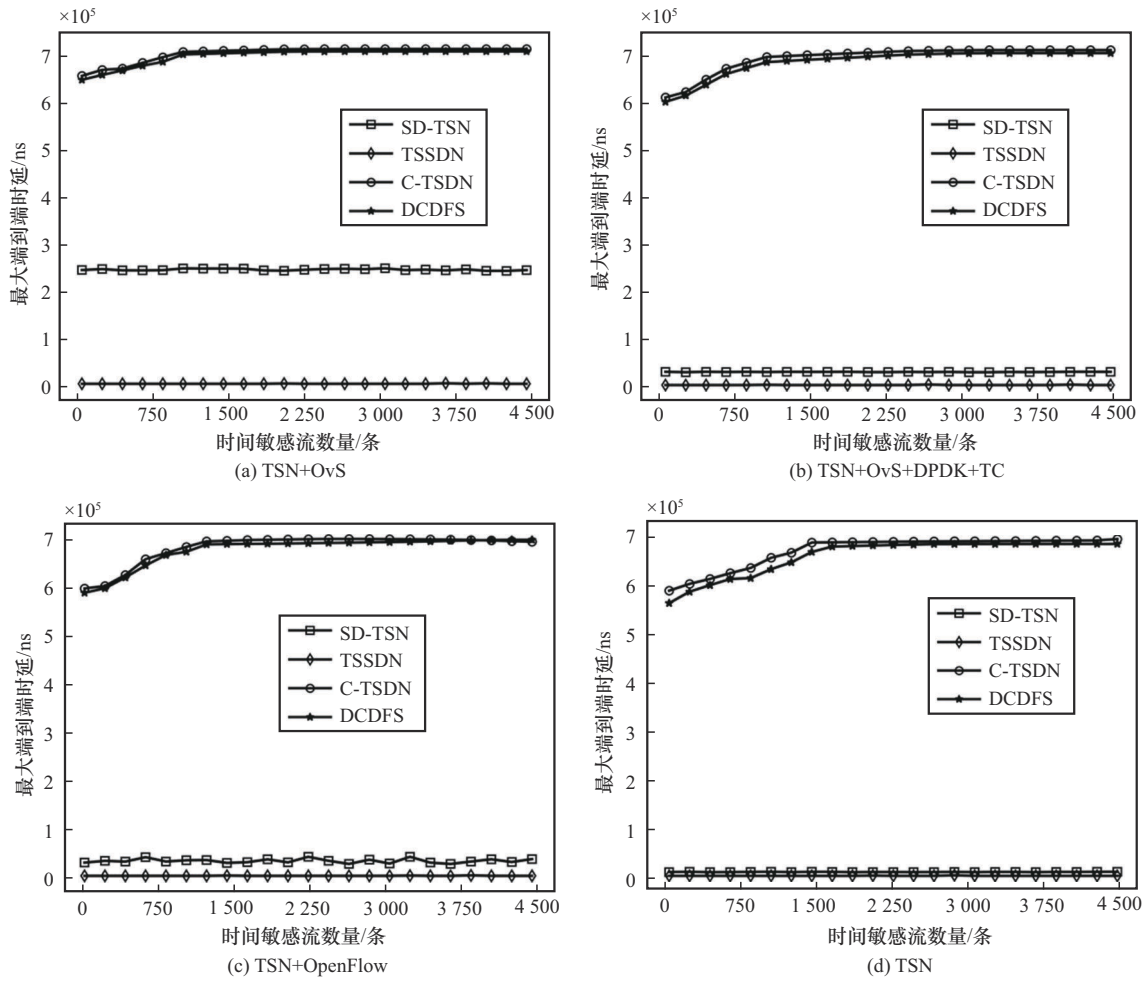


图 10 不同混合多域端到端时延对比

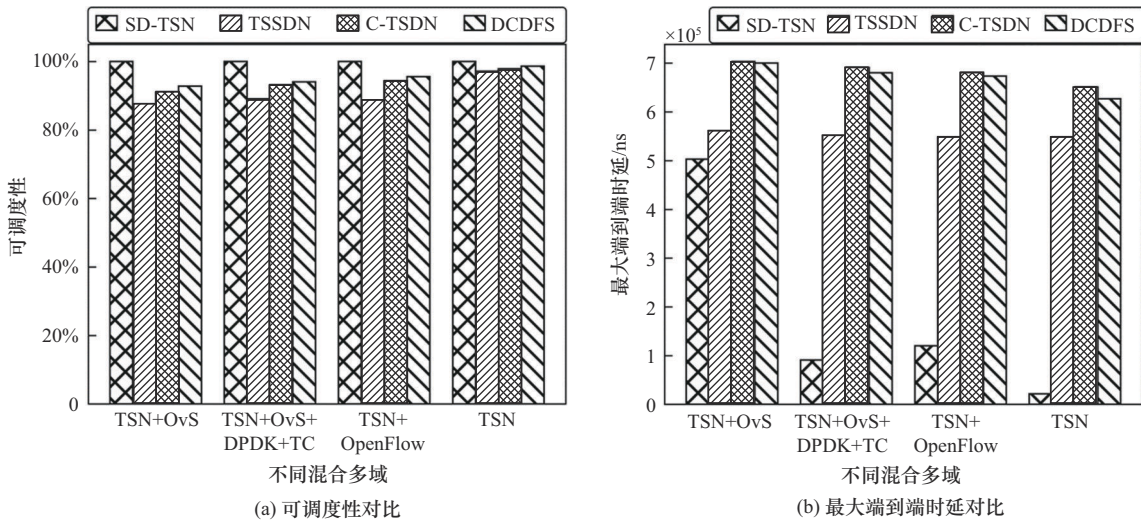


图 11 不同混合多域下载载 $\rho=1.0$ 时性能对比

均可调度分别为 94.11% 和 95.28%。由于无法对非 TSN 域进行进一步优化, TSSDN 的平均可调度性只有 90.56%。最大端到端时延结果如图 11(b) 所示,

SD-TSN 在各个混合多域中的平均最大端到端时延为 183 983 ns, 相较于 TSSDN 减少了 66.65%。这是因为 TSSDN 无法控制非 TSN 域中非时间敏感流

量对时间敏感流量的干扰, 导致了更大的排队时延。此外, 尽管 C-TSDN 和 DCDFS 能够对非 TSN 域进行有效控制, 但由于乒乓队列的循环切换机制, 虽然传输时延可以被限定在一定范围内, 其数值仍高于 SD-TSN 和 TSSDN。

5 结束语

本文提出了面向混合多域网络的 SD-TSN 组网方案, 设计了域控制器和协同控制支持混合多域环境下的确定性端到端传输。为验证所提方案, 构建了 SD-TSN 组网实验平台, 研发了协同配置等关键模块实现各个域内调度表的精准下发。基于此, 搭建了 SD-TSN 组网平台, 并设计了多种混合多域环境对组网方案进行了验证。实验结果表明, 所提方案能够在各种混合多域场景下保证有界低时延和低时延抖动的端到端传输。

参考文献:

- [1] 蔡岳平, 李栋, 许驰, 等. 面向工业互联网的 5G-U 与时间敏感网络融合架构与技术[J]. 通信学报, 2021, 42(10): 43-54.
CAI Y P, LI D, XU C, et al. Integrating 5G-U with time-sensitive networking for industrial Internet: architectures and technologies[J]. Journal on Communications, 2021, 42(10): 43-54.
- [2] WOLLSCHLAEGER M, SAUTER T, JASPERNEITE J. The future of industrial communication: automation networks in the era of the Internet of Things and industry 4.0[J]. IEEE Industrial Electronics Magazine, 2017, 11(1): 17-27.
- [3] FETTWEIS G P. The tactile Internet: applications and challenges[J]. IEEE Vehicular Technology Magazine, 2014, 9(1): 64-70.
- [4] GUO M, GU C J, HE S B, et al. MSS: exploiting mapping score for CQF start time planning in time-sensitive networking[J]. IEEE Transactions on Industrial Informatics, 2023, 19(2): 2140-2150.
- [5] YANG D, CHENG Z R, ZHANG W T, et al. Burst-aware time-triggered flow scheduling with enhanced multi-CQF in time-sensitive networks[J]. IEEE/ACM Transactions on Networking, 2023, 31(6): 2809-2824.
- [6] XUE J L, SHOU G C, LI H X, et al. Enabling deterministic communications for end-to-end connectivity with software-defined time-sensitive networking[J]. IEEE Network, 2022, 36(2): 34-40.
- [7] BÖHM M, OHMS J, WERMESER D. Multi-domain time-sensitive networks - an east-westbound protocol for dynamic TSN-stream configuration across domains[C]//Proceedings of the 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA). Piscataway: IEEE Press, 2019: 1363-1366.
- [8] SCHRIEGEL S, KOBZAN T, JASPERNEITE J. Investigation on a distributed SDN control plane architecture for heterogeneous time sensitive networks[C]//Proceedings of the 2018 14th IEEE International Workshop on Factory Communication Systems (WFCS). Piscataway: IEEE Press, 2018: 1-10.
- [9] BHATTACHARJEE S, ALEXANDRIS K, BAUSCHERT T. Multi-domain TSN orchestration and management for large-scale industrial networks[J]. IEEE Transactions on Network and Service Management, 2024, 21(6): 6480-6492.
- [10] LO BELLO L, STEINER W. A perspective on IEEE time-sensitive networking for industrial communication and automation systems[J]. Proceedings of the IEEE, 2019, 107(6): 1094-1120.
- [11] BÖHM M, WERMESER D. Multi-domain time-sensitive networks: control plane mechanisms for dynamic inter-domain stream configuration[J]. Electronics, 2021, 10(20): 2477.
- [12] 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.
- [13] WANG X L, YAO H P, MAI T L, et al. Reinforcement learning-based particle swarm optimization for end-to-end traffic scheduling in TSN-5G networks[J]. IEEE/ACM Transactions on Networking, 2023, 31(6): 3254-3268.
- [14] PENG G Y, WANG S, LI Z Q, et al. Deterministic cognition: cross-domain flow scheduling for time-sensitive networks[J]. IEEE Transactions on Cognitive Communications and Networking, 2024, 10(4): 1481-1495.
- [15] TAN W Q, WU B W, WANG S, et al. Large-scale deterministic transmission among IEEE 802.1Qbv time-sensitive networks[C]//Proceedings of the ICC 2022 - IEEE International Conference on Communications. Piscataway: IEEE Press, 2022: 2315-2320.
- [16] HUANG Y D, WANG S, HUANG T, et al. Cycle-based time-sensitive and deterministic networks: architecture, challenges, and open issues[J]. IEEE Communications Magazine, 2022, 60(6): 81-87.
- [17] NAYAK N G, DÜRR F, ROTHERMEL K. Software-defined environment for reconfigurable manufacturing systems[C]//Proceedings of the 2015 5th International Conference on the Internet of Things (IOT). Piscataway: IEEE Press, 2015: 122-129.
- [18] NAYAK N G, DÜRR F, ROTHERMEL K. Incremental flow scheduling and routing in time-sensitive software-defined networks[J]. IEEE Transactions on Industrial Informatics, 2018, 14(5): 2066-2075.
- [19] GUO M J, SHOU G C, XUE J L, et al. Cross-domain interconnection with time synchronization in software-defined time-sensitive networks[C]//Proceedings of the 2020 Asia Communications and Photonics Conference (ACP) and International Conference on Information Photonics and Optical Communications (IPOC). Piscataway: IEEE Press, 2020: 1-3.
- [20] 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]. Jour-

nal on Communications, 2021, 42(10): 1-9.

- [21] BHATTACHARJEE S, ALEXANDRIS K, BAUSCHERT T. Hierarchical control plane framework for multi-domain TSN orchestration[C]// Proceedings of the 2023 IEEE 9th International Conference on Network Softwarization (NetSoft). Piscataway: IEEE Press, 2023: 26-34.
- [22] HACKEL T, MEYER P, KORF F, et al. Software-defined networks supporting time-sensitive in-vehicular communication[C]// Proceedings of the 2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring). Piscataway: IEEE Press, 2019: 1-5.
- [23] GUO M J, SHOU G C, LIU Y Q, et al. Software-defined time-sensitive networking for cross-domain deterministic transmission[J]. Electronics, 2024, 13(7): 1246.

[作者简介]



郭梦杰 (1994-), 男, 河南辉县人, 北京邮电大学博士生, 主要研究方向为确定性网络、跨域时间同步、动态时间感知调度、系统配置等。



寿国础 (1965-), 男, 浙江诸暨人, 博士, 北京邮电大学教授, 主要研究方向为接入网络与边缘计算、光纤与无线网络虚拟化、网络构建与路由、移动互联网与应用等。



刘雅琼 (1988-), 女, 河南驻马店人, 博士, 北京邮电大学副教授, 主要研究方向为车联网、空间通信和边缘智能等。



胡怡红 (1961-), 女, 湖北武汉人, 北京邮电大学副教授, 主要研究方向为光纤与无线网络技术、网络监控与管理、软件定义网络架构与技术等。