

文章编号:1007-130X(2023)09-1553-10

时间敏感网络下的一种新型灵活门控机制^{*}

林佳烁^{1,2}, 李伟超², 成 剑², 詹双平², 冯景斌², 王 涛³, 黄倩怡⁴, 唐 博¹, 汪 漪^{2,5}

(1. 南方科技大学计算机科学与工程系, 广东 深圳 518055; 2. 鹏城实验室, 广东 深圳 518055;

3. 广东工业大学自动化学院, 广东 广州 510006; 4. 中山大学计算机学院, 广东 广州 510006;

5. 南方科技大学未来网络研究院, 广东 深圳 518055)

摘 要: 时间敏感网络流调度算法通常需要生成大量的门控事件。这超出了网络设备的能力, 使得调度算法难以在实际网络中部署。针对此问题, 提出一种新型灵活门控机制, 通过放松实时业务流与尽力而为业务流严格隔离的约束, 允许部分节点对实时业务流不启用门控机制。该机制允许灵活地选择在各个网络设备端口对各实时业务流使能门控, 能够将调度实时业务流所需的门控事件最多减少 91.6%, 其甚至允许实时业务传输路径上存在不支持门控调度的网络设备, 实现与普通网络的混合组网。

关键词: 时间敏感网络; 确定性网络传输; 流量调度; 灵活门控; 门控列表

中图分类号: TP393

文献标志码: A

doi: 10.3969/j.issn.1007-130X.2023.09.004

A novel flexible gate control mechanism for time-sensitive networking

LIN Jia-shuo^{1,2}, LI Wei-chao², CHENG Jian², ZHAN Shuang-ping², FENG Jing-bin²,
WANG Tao³, HUANG Qian-yi⁴, TANG Bo¹, WANG Yi^{2,5}

(1. Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055;

2. Peng Cheng Laboratory, Shenzhen 518055;

3. School of Automation, Guangdong University of Technology, Guangzhou 510006;

4. School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006;

5. Institute of Future Network, Southern University of Science and Technology, Shenzhen 518055, China)

Abstract: Time-sensitive networking flow scheduling algorithms often generate a large number of gate control events, exceeding the capabilities of network devices and making it difficult to deploy scheduling algorithms in practical networks. To address this issue, a novel flexible gate control-based flow scheduling strategy is proposed, which relaxes the strict isolation constraints between real-time traffic flows and best-effort traffic flows, allowing some nodes to not enable the gate control mechanism for real-time flows. This strategy can flexibly select to enable the gate control for real-time business at various network device ports, reducing the gate control events required for scheduling real-time traffic flows by up to 91.6%. It even allows the existence of network devices on the transmission path of real-time traffic that do not support gate control scheduling, achieving a mixed deployment with ordinary networks.

Key words: time-sensitive networking; deterministic network transmission; flow scheduling; flexible gate control; gate control list

^{*} 收稿日期: 2023-01-18; 修回日期: 2023-04-14

基金项目: 广东省重点领域研发计划(2020B010164001); 国家自然科学基金(62002150)

通信作者: 李伟超(liwc@pcl.ac.cn)

通信地址: 518055 广东省深圳市南山区西丽街道鹏城实验室

Address: Peng Cheng Laboratory, Xili Street, Nanshan District, Shenzhen 518055, Guangdong, P. R. China

1 引言

随着工业物联网 IIoT (Industrial Internet of Things) 的兴起以及异构网络的融合, 传统的以太网难以满足不同业务流的网络传输要求, 越来越多场景和用户开始关注时间敏感网络 TSN (Time-Sensitive Networking)。TSN 允许周期性数据与非周期性数据、实时数据与非实时数据在同一网络中传输, 使得标准以太网具有确定性传输的优势, 并通过了厂商独立的标准化进程, 已成为广泛聚焦的关键技术^[1,2]。

为了使时间敏感业务实现尽可能低的时延和抖动, TSN 提供了时间同步^[3]、全局时隙调度^[4]和门控机制^[5] 3 种技术来保证关键实时业务的确定性。首先, 时间同步为网络提供统一的时间基准; 其次, 全局的时隙调度统一规划资源, 在时间上(划分时隙)和空间上(规划路由)将实时业务流进行隔离, 以避免互相干扰; 最后, 门控技术可以避免实时业务流在传输过程中受到其他尽力而为 BE (Best Effort) 流的干扰, 具体措施为在实时业务流时隙开始之前提前关闭 BE 流队列的门控开关, 以阻止 BE 流的发送, 保证实时业务流时隙开始时链路是空闲状态, 从而避免了 BE 流对实时业务流的干扰。

门控机制能够很好地将实时业务流与尽力而为业务流隔离开, 从而实现实时业务流的确定性网络服务。典型的调度算法^[6,7] 通过门控机制将实时业务流与 BE 流在空间和时间上严格地进行隔离, 保证实时业务流不受到 BE 流的干扰, 同时也保证实时业务流之间的严格隔离。其引入了 No-wait 约束来保证实时业务流在网络中的队列时延为 0, 使得实时业务流的端到端时延为最小值。然而, 这种严格的隔离机制导致其在交换机中部署的门控制列表 GCL (Gate Control List) 条目数量巨大。例如, 为 1 个实时业务流预留 1 个时隙就需要 2 个门控开关事件。当网络中实时业务流的数量较多时, 由于交换机的设备能力有限, 无法支持大量的 GCL 列表事件, 这使得其难以对所有的实时业务流都进行精确的门控事件控制。为此, 如何在有限的门控列表资源下尽可能多地调度实时业务流成为了一个热点问题。文献[8]放松了网络约束, 保持交换机的门控开关常开, 然后使用基于可满足性模理论 SMT (Satisfiability Modulo Theory) 的方法进行求解。文献[9, 10]放松了实时业务流 0 抖动的约束, 允许预留时隙窗口重叠,

但这又引入了实时业务流冲突的问题, 此时需要使用网络演算工具^[11,12] 来考虑实时业务流在最差情况下的排队时延, 使问题复杂化。

对于大规模时间敏感网络下的流量调度场景, 交换机的 GCL 条目数量成为了限制调度算法部署的关键资源, 如何减少调度算法所需的门控事件数量, 提高可行性是 TSN 网络流量调度的关键问题。本文首先对实时业务流在非 TSN 交换机上的转发过程进行了深入的时延分析。基于该分析, 本文提出了一种灵活门控机制。这种灵活门控机制允许网络管理员灵活选择网络交换机对部分实时业务流使能门控, 可适用于基于 No-wait 约束的调度算法等, 以减少网络中的门控事件数量(即交换机的 GCL 条目数量)。并且, 由于该机制允许在某些交换机不使能门控, 从而可以在一些非 TSN 网络上部署 TSN 业务流。在实际网络部署中, 可以根据网络中的流量特征, 对某些实时业务流不部署门控事件, 保持门控开关常开, 这减少了网络调度所需的门控事件数量。对于未部署门控事件所引入的网络抖动, 可以在相应节点设置门控来进行消除。对一个典型工业网络拓扑的仿真分析表明, 该门控机制能最多减少门控条目数量 91.6%, 大大降低了 TSN 网络中流量调度所需的 GCL 资源, 利于 TSN 网络在实际中的部署。

2 相关背景

本节首先对 IEEE 802.1Qbv 网络中交换机的转发模型进行简单介绍, 然后介绍 TSN 中的调度模型。

2.1 IEEE 802.1Qbv 转发模型

一个典型的支持 TSN 网络 IEEE 802.1Qbv 协议的交换机转发模型如图 1 所示。其中, 交换机在将网络报文转发到对应的出端口中时, 根据报文中的优先级字段 PCP (Priority Code Point) 将其置入对应的队列中。

在 TSN 交换机的每个端口中, 有 8 个队列用于传输报文, 其中队列的开关由 GCL 列表来控制。只有当队列的门控开关被打开时, 即此时 GCL 列表中该队列的值为 1, 队列中的报文才会被转发, 否则不会被转发。如图 1 所示, GCL 列表由多个条目组成, 每一个条目指定了各个队列的开关状态, 及该条目的持续时间。交换机根据该 GCL 列表周期性地设置各个队列的门控开关。通过这种机制, 控制器中的调度引擎可以为网络中各个交换

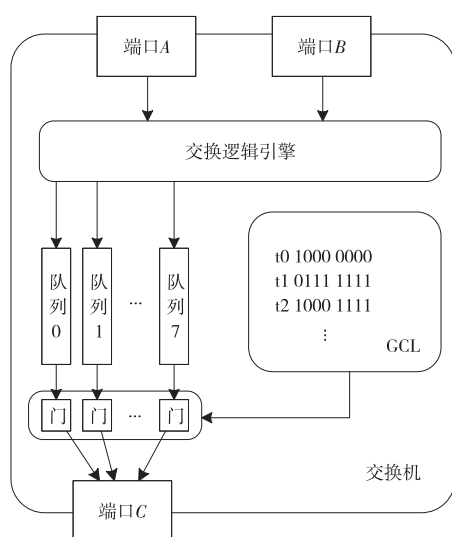


Figure 1 Forwarding model of typical TSN switch

图1 典型的 TSN 交换机转发模型

机生成对应的 GCL 列表来对网络业务流进行调度,以确保实时业务流的确定性转发服务。与文献[6]相似,本文使用单个队列(队列7)作为时间敏感报文的转发队列,而将其他队列(队列0~6)作为尽力而为业务流的转发队列。

2.2 TSN 调度模型

典型的混合 TSN 网络架构如图2所示,包含 TSN 交换机(深灰色节点)与非 TSN 交换机(白色节点)。在该网络中,控制器拥有对下层网络设备的全局信息。控制器中部署了调度引擎对网络中业务流进行调度并下发配置到交换机中。

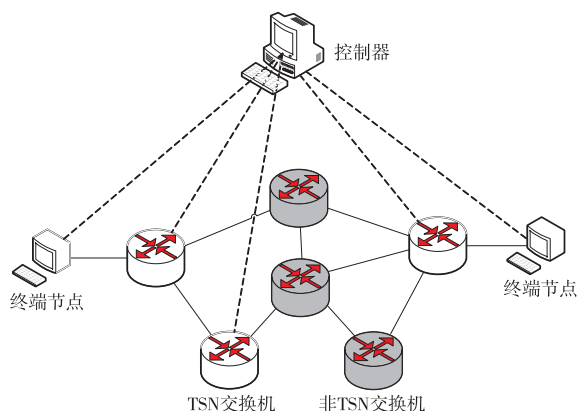


Figure 2 Architecture of hybrid TSN

图2 混合 TSN 网络架构

非 TSN 交换机不支持 TSN 协议标准如 IEEE 802.1Qbv 等,因而无法部署门控从而对业务流进行精确的流量开关控制。尽管这种交换机无需支持 IEEE 802.1Qbv 协议,但其必须支持时间同步协议如 IEEE 802.1AS 等来进行时间同步信息的传递,即必须保证整个网络中的设备都进行

了时间同步,从而使得各个 TSN 交换机的 GCL 门控队列都具有相同的基准时间。

在 TSN 中,一个典型的流调度过程如下:终端节点根据其应用特征发起业务流调度请求,调度器根据其调度请求为该业务流预留资源,并更新网络中 TSN 交换机的门控列表 GCL。最后,终端节点根据调度结果在对应的时间节点发送业务流报文。

3 研究动机

3.1 研究问题

本文旨在研究基于 IEEE 802.1Qbv 协议的 TSN 网络中的实时业务流调度问题。在给定网络状态和实时业务流需求下,综合考虑网络拓扑及业务需求等因素进行调度计算,得到网络中的 GCL 门控列表。然而,传统的调度算法没有考虑实际生成的门控数量,其容易超过实际设备的能力,从而难以在实际网络中部署。本文对传统的基于 No-wait 约束的调度算法进行改进,旨在满足 No-wait 约束的同时,减少生成的 GCL 门控数量。

在调度计算过程中,实时业务流的调度结果需要满足各种约束条件^[7,13],其中与时间敏感网络相关的约束如下:

约束1 避免实时业务流之间的互相干扰,这一点由调度算法来保证,所有实时业务流分配的时隙均不能重叠。

约束2 避免实时业务流受到 BE 流的干扰,这一点通过门控机制来实现,即,在实时业务流发送时隙开始时,提前将 BE 流的门关闭(引入保护带)以避免干扰。

3.2 研究挑战

传统的时隙调度方法需要对实时业务流进行精确的流调度控制,在网络交换机中生成了大量的门控制事件,超过了设备能力的需求。主要的挑战如下:

(1) 门控事件数量要求较高。门控开关的动作是由门控列表 GCL 来控制的,一个时隙需要打开和关闭 2 个动作,其反映在 GCL 列表中就是 2 个表项。显然,对于某一个节点,其 GCL 列表表项与经过该节点的实时业务流的数量成正比。随着网络中实时业务流的数量增加,时隙的数量也会增加,因此门控数量也会相应增加。

另外,门控事件的数量除了与实时业务流数量相关外,与业务流应用周期的关系也非常大。当网

络中业务流的应用周期相差较大时,门控事件的条目将急剧增加。例如,如图3所示,2条业务流的应用周期分别是2 ms和3 ms,此时,GCL列表的周期为这2条业务流应用周期的最小公倍数6 ms,需要在这整个宏周期内预留时隙。*Flow1*在整个宏周期内占用了2个时隙,而*Flow2*在宏周期内占用了3个时隙,因此一共需要5个时隙共10个GCL表项。考虑在另一场景下,存在3条业务流,其应用周期分别是1 ms,7 ms和20 ms,则此时宏周期长度为 $LCM(1,7,20)=140$ ms,此时3条业务流共需要 $140+20+7=167$ 个时隙共334个GCL表项,这已超过了大部分交换机所能支持的门控列表数量^[8]。并且,随着网络中实时业务流数量的增加及业务流应用周期差异的增大,所需的门控表项数量会进一步急剧增加,远超设备能力。

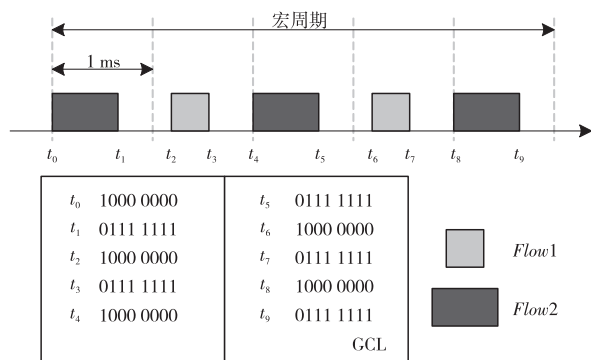


Figure 3 GCL expansion due to the hypercycle

图3 宏周期引起的GCL表项膨胀

(2) 保护带对网络资源消耗较严重。由于每一个实时业务流时隙开始之前,都需要为其设置保护带,以确保时隙开始时链路是空闲状态,保证实时业务流可以在确定时刻转发。随着时隙数量的增加,大量的门开关动作就需要大量的保护带,而保护带又不允许任何新的BE流的传输,因此大量的保护带将会对网络资源造成严重的浪费。

(3) GCL配置时间较长。GCL列表过长也增加了网络配置所需要的时间。尽管目前一个典型的TSN交换机设备仅能支持256个GCL表项,但退一步说,即使设备支持上万个GCL表项来满足时隙调度的需求,但此时新增或者删除一条流往往需要完全重写整个GCL表,特别是在宏周期发生了变化的情况下,这会导致新增或者删除业务流的配置时间很长,影响业务部署效率。

4 灵活门控机制及部署

本文提出一种新型的灵活门控机制,可以针对

不同的流在不同的节点设置是否开启门控。本节首先介绍基础的时延分析,接着介绍几种典型的灵活门控部署策略:无门控机制,即在所有节点均不开启门控;尾节点门控机制,只在网络最后一跳开启门控,其他节点均不开启门控;灵活门控机制,在部分节点开启门控。网络管理员可以结合网络中的设备能力、业务流的特征以及性能需求等因素灵活选择不同的部署策略。

4.1 时延分析

时间敏感报文在TSN网络中的端到端时延一般包括:链路传播时延LD(Link Delay)、交换机处理时延PD(Processing Delay)、传输时延TD(Transmission Delay)及队列时延QD(Queueing Delay)等。其中,LD指的是信号在单条网络链路上传播所需要的时延,其与传输介质及信号在传输介质中的传播速度有关;PD指的是交换机收到报文后进行处理直到报文在对应端口开始传输的时延,其与交换机的处理能力相关;TD指的是报文在交换机出端口处的传输时延,其与报文的长度及该网络链路的链路速率相关;QD指的是报文在交换机端口处队列中的排队时延,其与交换机实时的负载相关。本文采用基于No-wait约束的网络模型^[6],该约束保证了实时业务流的QD为0。

图4展示了时间敏感报文从源节点(*src*)经中间节点(B_1)到目标节点(*dest*)的传输过程。其中,源节点在 t_0 时刻将报文发出,经过 LD_1 时间后,报文的第1个字节到达 B_1 节点;再经过 TD_1 时间后报文的最后1个字节到达 B_1 节点。此时,报文已经被 B_1 节点完整接收,开始进入 B_1 节点的转发流程,经过交换机处理时延 PD_1 后,报文到达 B_1 的出端口队列7中,此时可以开始发送报文。在 t_4 时刻, B_1 出端口的门控开关被打开,此时报文将从 B_1 节点的出端口转发出去,开门的持续时间等于报文在 B_1 出端口的传输时延,即 TD_2 。报文发送完成后,在 t_5 时刻将门关闭,允许其他业务转发。最后,报文在确定的 t_7 时刻到达目标节点。

需要注意的是,为了确保在 t_4 时刻开门时,时间敏感报文可以准确地在该时刻开始传输,需要提前将其他队列的门关闭以确保此时链路上没有其他报文在传输。这个提前量也被称为保护带GB(Guard Band)。在保护带内,不允许开始任何新的报文传输。保护带的长度与节点的转发行为有关,如果节点不支持帧抢占机制^[13],则该保护带需要足够长,以确保能够传输完一个最大传输单元MTU(Maximum Transmission Unit)的BE数据

帧并加上帧间隙 IFG (Interframe Gap); 如果节点支持帧抢占机制, 则保护带长度就是打断 BE 流传输所需的时间加上 IFG。在 IEEE 802.1Qbu 中, 当使能了帧抢占机制, 该值为该节点传输 123 B 所需的时间。

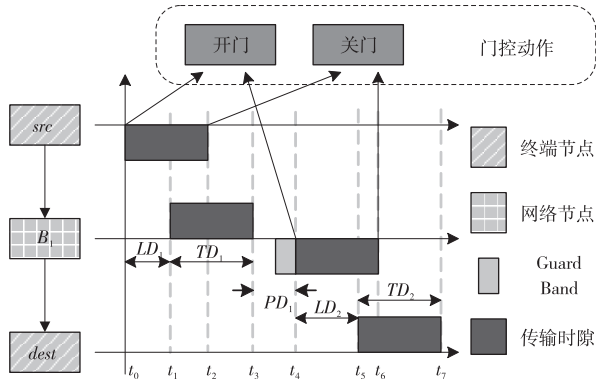


Figure 4 Packet delay analysis in TSN

图4 TSN中的时延分析

4.2 无门控流调度策略

无门控流调度策略允许实时业务流在调度中不使用门控, 放宽了约束 2, 此时实时业务流将会受到 BE 流的干扰。即对于某一实时业务流, 其在某一节点出端口的传输时隙开始时, 该出端口链路可能正在被其他 BE 业务流占用, 此时需要等待其发送完成才能开始发送, 对于这个由于 BE 流干扰引入的时延, 使用 ID (Interference Delay) 来表示。干扰产生的时延与具体的转发机制有关, 如果使能了 IEEE 802.3Qbu 帧抢占机制^[14], 干扰时延 ID 的长度为打断 BE 流传输所花费的时间; 如果未使能帧抢占功能, ID 的长度就是等待其他流传输完成所花费的时间, 与 BE 流的 MTU 和链路速率有关。

由于交换机是根据优先级队列来对业务流进行转发, 所以无论使能帧抢占机制与否, 实时业务流受到的干扰时延最多为传输一个低优先级报文所需要的时间, 而与低优先级业务流的流量特征 (如业务流带宽、持续时间等) 无关。 ID 表示的是可能受到的最大干扰时延, 而实际上的干扰时延会在一定的范围内抖动, 其上界为 ID 。例如, 当一个实时数据帧到达某个交换机时, 若当前无低优先级数据帧正在传输, 则其实际干扰时延为 0。

对于一个实时业务流, 其端到端的抖动等于该流的转发路径上每一跳所可能会引入的抖动之和, 其上界则等于每一跳的最大干扰时延 ID 之和, 如式(1)所示:

$$Jitter_{\max} = \sum_e ID_e \quad (1)$$

其中, e 表示该实时业务流转发路径上的网络链路。

如图 5 所示, 在无门控流调度策略中的 B_1 节点, 由于没有使能门控, 所以实时业务流可能受到其他 BE 流的干扰, 此时, 其可能在 t_0 时刻就能发送, 也有可能受到干扰, 直到 t_1 时刻才能发送, 因此其占用的时隙就需要考虑干扰带来的影响, 即在 $t_0 \sim t_4$ 时间段内, 不允许其他实时业务流调度 (见约束 1), 从而避免了实时业务流之间的影响。此时, 为该实时业务流传输所预留的时隙, 等于其实际传输所耗费的时间 TD_1 , 加上其可能受到的干扰时延 ID_1 , 即 $TS_1 = TD_1 + ID_1$ 。类似地, 当实时业务流到达 B_2 节点后, 抖动将会累积, 占用的时隙会进一步增加为 $TS_2 = TD_3 + ID_1 + ID_2$ 。当到达目标节点时, 报文会产生一定的抖动, 也就是各节点累积的抖动之和, 即 $ID_1 + ID_2$ 。

无门控流调度策略只需要源节点处支持门控, 而对于中间的网络节点, 甚至可以完全不支持 IEEE 802.1Qbv 门控机制, 就可以达到避免实时业务流之间互相干扰的目的。

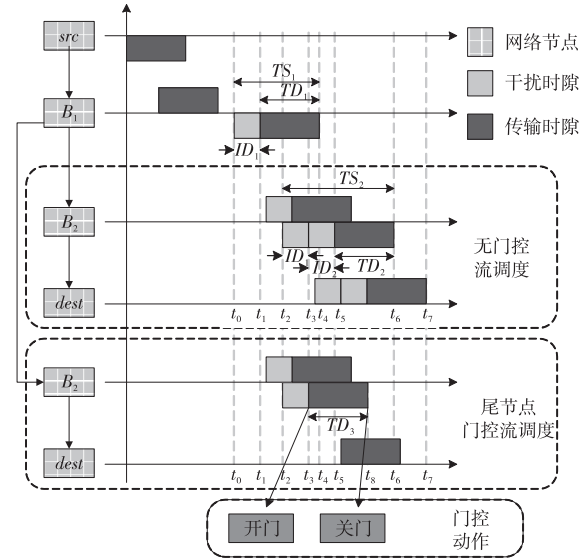


Figure 5 Delay analysis of no gate control and end gate control mechanisms

图5 无门控和尾节点门控机制时延分析

4.3 部分门控流调度策略

4.2 节提出的无门控流调度策略, 尽管其可以使得交换机完全无需支持门控功能, 极大地简化了网络配置, 但也带来了 2 个问题:

问题 1 业务流会产生一定的抖动, 这对于一些 0 抖动要求的实时业务流并不适用。

问题 2 放松约束 2 带来的抖动会随着跳数逐跳累积, 对于某一业务流, 其占用的时隙将随着

跳数增加,这对网络资源造成了浪费,降低了可调度流的数量。

为了解决上述问题,可以在网络中的部分节点启用门控,以消除累积的抖动。

4.3.1 尾节点门控流调度策略

为了满足某些业务流 0 抖动的要求,可以在该业务流转发路径上的最后一跳部署门控,以消除网络抖动。

如图 5 中的尾节点门控流调度策略所示,其中 B_1 节点均未使能门控,因此产生的干扰时延 ID_1 会累积到占用的时隙中,报文到达 B_2 节点时,占用的时隙为 $t_2 \sim t_8$ 。尽管报文到达 B_2 节点的最早可能时间是 t_2 时刻,但是并未在时隙开始时就将门打开,而是延迟一定的时间,以容忍最坏情况下报文的到达时刻,也就是在 t_3 时刻开门,持续 TD_4 时间后将门关闭。此时无论前面累积了多少抖动,都通过门控的方式清零,最终报文在确定的时刻到达目标节点,达到业务流的端到端 0 抖动网络转发。

4.3.2 灵活门控流调度策略

尾节点门控虽然消除了业务流的抖动,但沿途的抖动依然存在,在其路径上的交换机仍然占用了较大的时隙,特别是越接近目标节点的中间节点,其占用时隙会越大,这浪费了网络资源,降低了可调度的流数量。并且,当网络中的某一个终端节点成为热点时(例如网络中存在应用周期差异较大的业务流并且这些业务流的目标节点相同),与该终端节点相连的交换机将会承受较大的流量压力,此时尾节点门控流调度策略已无法有效地减少 GCL 表项数量。

在实时业务流转发路径上的中间节点处部署门控,一方面可以消除前面累积的抖动,从而减小实时业务流的时隙占用;另一方面可以有效地将 GCL 分散到网络中的各个节点中,以减轻某些热点节点的压力。

需要注意的是,当使用灵活门控机制时,其生成的 GCL 表项的逻辑与传统的完全门控机制是相反的。这是因为对于某一网络节点,其可能仅对网络中的部分实时业务流使能门控,而对其他实时业务流保持正常转发。如图 6 所示,对于某一网络节点,其对实时业务流 $Flow2$ 使能门控,而对流 $Flow1$ 不使能门控。由于 $Flow1$ 与 $Flow2$ 的优先级都是一致的,即这 2 个业务流的数据帧中 PCP 字段的值是相同的,所以这 2 个业务流的数

据帧将会映射到同一实时队列中。尽管该节点只对 $Flow2$ 使能门控,但其仍然需要保持实时队列的门控开关为打开状态,以保证业务流 $Flow1$ 的正常传输。所以,该网络节点的门控开关状态如图 6 所示,当实时业务流 $Flow2$ 的预留时隙到达时,需要首先关闭门控开关,然后在 $Flow2$ 的可能最晚到达时间才开启门控,以保证无论 $Flow2$ 在 t_0 时刻还是 t_1 时刻到达,都只能在 t_1 时刻才能被转发,达到消除抖动的目的。

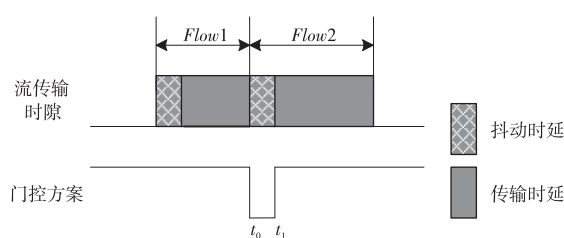


Figure 6 Gate should keep opened in flexible gate control mechanism

图 6 灵活门控机制必须保持门控开关常开

4.4 示例分析

4.4.1 示例分析 1:消除累计抖动

选择实时业务流转发路径上的中间节点来使能门控,可以有效地消除累积抖动。如图 7 所示,实时业务流在中间节点 B_k 使能门控,而在其他节点(如 B_{k-1} 和 B_{k+1})未使能门控。在 B_{k-1} 节点中,由于其未使能门控,并且该节点处已经累积了大量的干扰时隙,所以在该节点处,实时业务流的最早发送时刻为 t_0 ,最晚发送时刻为 t_3 ;在 B_k 节点中,由于使能了门控,其提前在 t_1 时刻把门控开关关闭,并在 t_4 时刻才把门控开关打开,允许实时业务流进行传输,所以在该节点处,实时业务流在之前累积的抖动将会被清零,并且以完全确定的时刻 t_4 向 B_3 发送报文。

4.4.2 示例分析 2:减少门控数量

使用灵活门控机制,可以针对网络业务流的应用周期特点,选择特定交换机来对指定实时业务流使能门控,从而减少门控数量。如图 8 所示,网络中有 3 个实时业务流,应用周期分别为 1 ms, 2 ms 和 10 ms。若使用完全门控,由于宏周期为 10ms,在每一个节点,对于上述 3 个实时业务流分别各需要 10,5,1 个时隙,共 16 个时隙 32 个 GCL 条目。若使用尾节点门控机制,尽管在 B_1 节点和 B_2 节点处无需部署门控,但在 B_3 节点处也同样需要 32 个 GCL 条目。若仅在 B_3 节点处对流 $Flow1$ 和 $Flow2$ 进行门控,而在 B_2 节点处对流 $Flow3$ 进

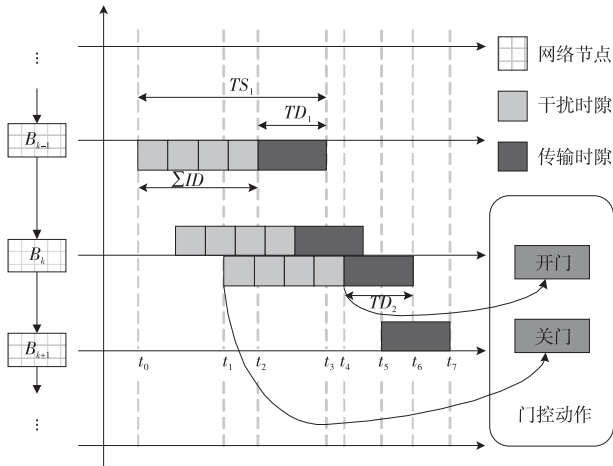


Figure 7 Delay analysis of gate control at the intermediate node

图7 中间节点门控时延分析

行门控,则此时在 B_3 节点所需部署的 GCL 周期为 2 ms,表项数量为 $(2+1) \times 2=6$ 个,而 B_2 节点的 GCL 周期为 10 ms,表项数量为 $1 \times 2=2$ 个。尽管此时 Flow3 的端到端时延会略有抖动,但此时整个网络所需的 GCL 表项数量为 $6+2=8$ 个,相比传统的完全门控机制 96 (32×3) 和尾节点门控机制的 32 个 GCL 表项,分别减少了 91.6% 和 75%,极大地缓解了网络中交换机设备的 GCL 门控压力。

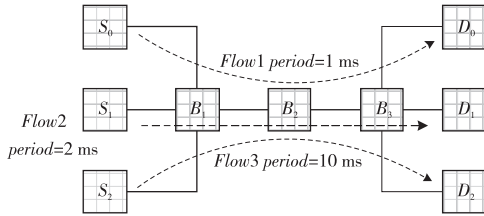


Figure 8 Flexible gate control mechanism can effectively reduce the length of GCL

图8 灵活门控机制可以有效减少 GCL 表项数量

通过该示例可以看出,灵活门控调度策略在减少 GCL 门控条目数量方面有较大的优势,特别是当网络中的业务流应用周期差异较大时效果显著。但是,该策略同时会导致业务流遭受一定的抖动,并且减少了网络能够调度的实时业务流数量。在实际应用中,需要根据业务流特征和网络拓扑灵活选择门控策略,后续将研究如何平衡网络调度性和 GCL 条目数量之间的关系。

4.5 门控数量分析

考虑使用随机的门控机制的流调度算法,即对于每条实时业务流在其转发路径上的每个交换机,假设使能门控机制概率为 p 。此时,对于一个转发

路径跳数为 n 的业务流,对其使能门控机制的交换机个数的期望值为 $p \times n$ 。而使能门控机制需要开门和关门 2 个 GCL 表项,即对于该实时业务流,其在每个应用周期所需的全局门控开关数量的期望值为 $2 \times p \times n$ 。

对于文献[6]采用的策略,由于其使用的是传统的完全门控流调度策略,可以认为其 p 值为 1。而对于本文所提出的尾节点门控流调度策略,由于该策略仅在实时业务流转发路径上的末尾节点进行门控,则对于每一条流,其在每个应用周期所需的全局门控事件数量仅为 2。

5 实验与仿真

5.1 实验设置

实验设置的网络拓扑如图 9 所示,由 16 台交换机和 16 台终端组成。其中,交换机 $B_0 \sim B_3$ 之间的网络链路为 10 GE,其余链路为 1 GE。这是一种典型的工业网络拓扑,其中 $B_0 \sim B_3$ 可认为是工业网络中的核心转发交换机,而剩下的 12 台交换机各自组成了 2 个小环状网络。网络链路传播时延为 $1 \mu\text{s}$ 。在实际部署中,交换机由于硬件能力的限制,在处理报文时会产生一定的抖动。交换机处理时延抖动设为 $0.85 \mu\text{s}$,其中最小处理时延和最大处理时延分别为 $0.45 \mu\text{s}$ 和 $1.3 \mu\text{s}$ 。这种参数的选择参照了典型工业 TSN 交换机的硬件能力。

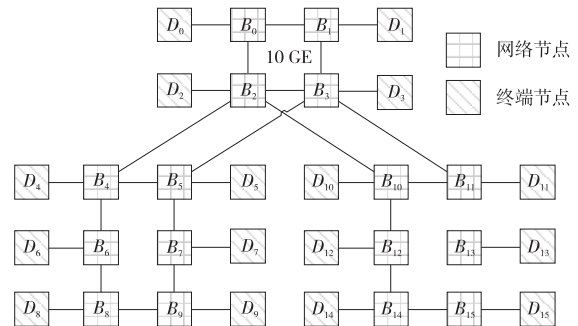


Figure 9 Experimental topology

图9 实验网络拓扑结构

实验所设置的参数如表 1 所示。在实验 1 和实验 2 中,网络中随机生成的实时业务流数量从 100 条增加到 500 条,步长为 50,共 9 组实验。在实验 3 中的 6 组实验中,业务流的数据大小分别为 100,200,500,1 000,1 500 和随机,其中第 6 组实验的流数据帧大小在 100,200,500,1 000,1 500 中随机选取。在实验 2 和实验 3 中,实时业务流的应用周期分别从 0.1 ms, 1 ms, 10 ms 和 0.1 ms,

0.5 ms, 1 ms 中随机选取。其中,随机生成实时业务流的终端节点从 $D_0 \sim D_{15}$ 中随机选取。为保证实验结果的准确性和可靠性,所有实验均进行 20 次并取平均值。

Table 1 Experimental parameters setting

表 1 仿真实验参数设置

实验	流数量	流周期/ms	流数据包大小/B
实验 1	100~500	10	100
实验 2	100~500	0.1,1,10	100
实验 3	-	0.1,0.5,1	100,200,500, 1 000,1 500,random

本文所有实验均经过仿真工具 Omnet 和 TSN 扩展框架 Nesting 验证。在仿真验证中,随机添加了 BE 业务流对实时业务流进行干扰,仿真结果显示实时业务流的时延符合预期,验证了调度结果的正确性。

在对比实验中,本文主要选择文献[6]中的算法进行对比,该文献中提出的 No-wait 约束调度算法是一种典型的 TSN 调度算法。以下实验是在文献[6]的基础上应用灵活门控机制。需要注意的是,灵活门控机制并不只局限于 No-wait 约束算法,由于可以灵活选择节点进行门控,因此灵活门控机制可以广泛地应用在各种算法中,以降低其门控数量,提高其部署能力。

5.2 实时业务流数量对门控数量影响分析

实时业务流数量对门控数量影响的实验结果如图 10 所示,其中随机门控中的 p 指的是对于某一条实时业务流,在某一条交换机上使能门控机制的概率为 p 。

随着网络中实时业务流数量的增加,对于 4 种流调度策略,网络中的全局门控事件数量都呈线性增加。其中文献[6]中的策略由于采用的是全局门控机制来精确调度实时业务流,所以其门控数量在 4 种策略中上升较快。相比之下,本文所提出的灵活门控机制(随机门控和尾节点门控)的门控事件数量显著减少,其中,尾节点门控由于仅在实时业务流转发路径上最末端的交换机使能门控,所以无论实时业务流的转发路径有多长,其全局门控数量均为实时业务流数量的 2 倍,最终呈现在图上时,其曲线几乎是一条直线,而随机门控机制由于其随机性,其曲线呈现了一定的波动。

由于本实验的各组实验都采用同一网络拓扑,则随机生成的实时业务流的转发路径的跳数期望值 $E(n)$ 相等,同时其网络周期也相等,根据 4.5 节的分析,对于随机门控机制,网络中的全局门控

事件数量应与概率 p 成正比。

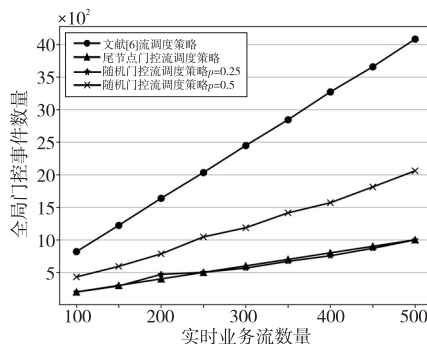


Figure 10 Experiment 1: Effect of the number of flows on the length of GCL

图 10 实验 1——流数量对 GCL 列表门控数量的影响

如图 10 所示,随机门控流调度策略 $p = 0.5$ 的曲线约为文献[6]流调度策略曲线的一半,而随机门控流调度策略 $p = 0.25$ 的曲线又约为随机门控流调度策略 $p = 0.5$ 曲线的一半,符合预期。另外,随机门控流调度策略 $p = 0.25$ 的曲线与尾节点门控流调度策略的曲线重叠度较高,可以根据此计算出网络中的实时业务流转发路径的跳数,即此时 $2 * p * n = 2$,代入 $p = 0.25$,得到 $n = 4$ 跳,即此时网络中实时业务流的平均跳数约为 4 跳,符合本实验采用的网络拓扑结构。

本文所提出的灵活门控流调度策略允许对业务流灵活地进行门控,从而减少调度策略所需的 GCL 表项资源。在上述实验中,采取最简单的随机分配门控策略,使用随机因子 0.25,可以将全局门控事件数量有效减少约 75%。在实际的网络部署中,需要综合考虑设备能力和网络业务流 QoS 约束,对门控进行更加精细的控制。

5.3 业务流应用周期对门控数量影响分析

本节实验主要研究业务流应用周期对门控数量的影响,实验结果如图 11 所示。在该实验中,本文所提出的灵活门控流调度策略针对实时业务流的应用周期特点,对网络中的交换机分别使能门控。具体地,将所有交换机平均分为 3 组,分别对应应用周期为 0.1 ms,1 ms 和 10 ms 的实时业务流。每一组交换机仅对对应的实时业务流使能门控,这使得尽管网络中的流的宏周期为 10 ms,但网络中各个交换机端口的 GCL 列表宏周期仅与使能门控的实时业务流相关。

文献[6]所采取的传统的全门控流调度策略导致了网络中所有端口的 GCL 宏周期都为 10 ms。此时,对于应用周期为 0.1 ms 的实时业务流,尽管其在每一个应用周期所需的 GCL 表项仅

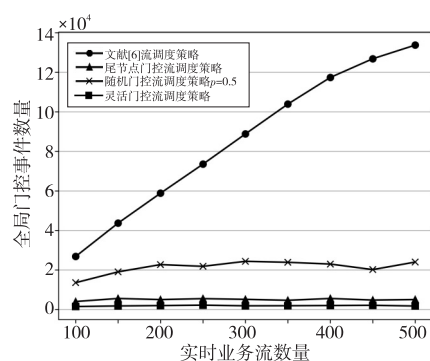


Figure 11 Experiment 2: Effect of the number of flow cycles on the length of GCL

图 11 实验 2——实时业务流应用周期对门控数量的影响为 2,但在宏周期中,其所需的 GCL 表项膨胀到了 200 个,这导致了全局门控事件急剧增加,远远超过了交换机的设备能力。而本文所使用的灵活门控流调度策略,通过将实时业务流根据应用周期分组在对应的端口进行门控,解决了门控事件膨胀的问题。在该实验中,基于灵活门控机制设计的分组门控流调度,有效避免了因业务流周期不同导致的 GCL 表项膨胀问题。这种根据业务流周期特点进行分组的流调度设计,相比随机门控流调度,节省了更多的 GCL 表项资源。

5.4 最大调度流数量分析

本节实验主要研究本文所提出的灵活门控流调度策略对网络可调度流的数量影响。如前文所述,当交换机不对实时业务流使能门控时,其需要为其预留出可能受到干扰的时隙。当使能了帧抢占功能时,需要预留出的额外时隙为 123 B、帧间隙以及前导码的传输所需要的时间。

在本节实验设置中,实时业务流的应用周期从 0.1 ms, 1 ms, 10 ms 中随机选取。其中,网络增量式地生成实时业务流给调度器进行调度计算,当无法调度时停止运行,此时得到网络的最大可调度流数量,实验结果如图 12 所示。

使用随机门控流调度,将使能门控的概率 p 从 1 到 0 逐渐下降。当 $p=1$ 时,随机门控流调度策略等价于文献[6]中的完全门控流调度策略;而当 $p=0$ 时,网络中所有交换机都不使能门控,即此时交换机保持所有队列为开启状态,等价于一个传统的以太网交换机。在这种情况下,尽管网络中无门控,但调度器在进行调度计算时,仍然需要将所预留的时隙考虑进去,以保证实时业务流之间的严格隔离。

可以看到,随着概率值的下降,网络中的可调

度流数量呈下降趋势。其中,当业务流数据大小为 100 B 时,无门控情况下可调度流数量比完全门控情况下减少了约 82.44%,对网络的可调度性造成了严重的影响。另一方面,当业务流数据大小为 1 500 B 时,是否使能门控对可调度流的数量几乎没有影响。这是由于当不使能门控时,需要为业务流预留出至少 123 B 的额外传输时间,并且这个额外的时隙是逐跳累计的。即,对于未使能门控的每一跳,都需要为其预留出额外的干扰时隙。对于业务流数据量较小的场景,该额外预留时隙占比较大,所以不使能门控对网络可调度流数量会造成较大影响,而当实时业务流的数据量较大时,该影响就微乎其微了。在实际的网络部署中,管理员需要权衡网络的实时业务流数量和设备能力,以选择适合实际网络需求的方案。

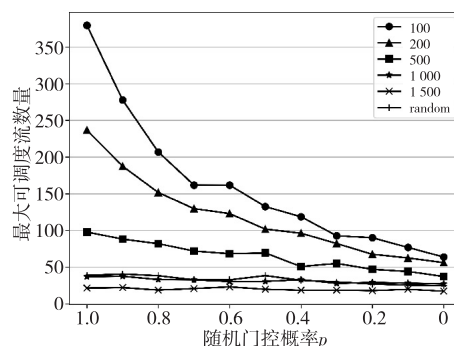


Figure 12 Experiment 3: Effect of flexible gate control on the number of scheduled flows

图 12 实验 3——灵活门控对网络中可调度流数量的影响

6 结束语

针对大规模时间敏感网络中 GCL 表项资源紧张的问题,本文提出了一种新型的灵活门控机制,该机制在保证满足实时业务流确定性网络传输的前提下,允许在特定的节点不支持门控机制转发或特定端口对部分实时业务流不使能门控,有效地减少了调度机制所需的 GCL 表项资源。

基于该灵活门控机制,可以针对网络中的流量周期特性,在各个交换机分别对不同的实时业务流使能门控,以减少设备 GCL 表项数量。从本文实验结果可知,尽管不使能门控可以有效减少网络交换机所需的 GCL 条目数量,但其增加了业务流的抖动,同时也减少了网络可调度业务流的数量。

在实际应用中,网络管理员必须根据实际场景的业务流流量特征、网络设备能力等信息综合考虑门控部署方案。TSN 中的流量调度问题本就是一

个 NP 难问题,本文所提出的灵活门控机制又引入了一个新的维度空间(即网络业务流的门控策略选择)。这种灵活门控机制在减少调度策略所需 GCL 表项资源的同时,也带来了新的调度计算问题。未来将进一步研究针对业务流特征和网络特点的调度优化算法。

参考文献:

- [1] Messenger J L. Time-sensitive networking: An introduction [J]. IEEE Communications Standards Magazine, 2018, 2(2): 29-33.
- [2] Finn N. Introduction to time-sensitive networking[J]. IEEE Communications Standards Magazine, 2018, 2(6): 22-28.
- [3] IEEE 802.1 Working Group. Timing and synchronization for time-sensitive applications in bridged local area networks: IEEE Std 802.1ASTM-2011[S]. New York: IEEE Standards Association, 2011.
- [4] Steiner W, Craciunas S S, Oliver R S. Traffic planning for time-sensitive communication [J]. IEEE Communications Standards Magazine, 2018, 2(2): 42-47.
- [5] IEEE 802.1 Working Group. Bridges and bridged networks—Amendment 25: Enhancements for scheduled traffic: IEEE Std 802.1QbvTM-2015 [S]. New York: IEEE Standards Association, 2015.
- [6] Dürr F, Nayak N G. No-wait packet scheduling for IEEE time-sensitive networks (TSN)[C]//Proc of the 24th International Conference on Real-Time Networks and Systems, 2016: 203-212.
- [7] Craciunas S S, Serna R, Martin O, et al. Scheduling real-time communication in IEEE 802.1Qbv time sensitive networks [C]//Proc of the 24th International Conference on Real-Time Networks and Systems, 2016: 183-192.
- [8] Jin X, Xia C, Guan N, et al. Real-time scheduling of massive data in time sensitive networks with a limited number of schedule entries[J]. IEEE Access, 2020, 8: 6751-6767.
- [9] Reusch N, Zhao L, Craciunas S S, et al. Window-based schedule synthesis for industrial IEEE 802.1Qbv TSN networks [C]//Proc of IEEE International Workshop on Factory Communication Systems, 2020: 1-4.
- [10] Serna Oliver R, Craciunas S S, Steiner W. IEEE 802.1Qbv gate control list synthesis using array theory encoding[C]//Proc of the IEEE Real-Time and Embedded Technology and Applications Symposium, 2018: 13-24.
- [11] Zhao L, Pop P, Craciunas S S. Worst-case latency analysis for IEEE 802.1Qbv time sensitive networks using network calculus[J]. IEEE Access, 2018, 6: 41803-41815.
- [12] Maile L, Hielscher K S, German R. Network calculus results for TSN: An introduction[C]//Proc of 2020 Information Communication Technologies Conference, 2020: 131-140.
- [13] Nayak N G, Dürr F, Rothermel K. Time-sensitive software-defined network (TSSDN) for real-time applications [C]//

Proc of the 24th International Conference on Real-Time Networks and Systems, 2016: 193-202.

- [14] IEEE 802.1 Working Group. Bridges and bridged networks—Amendment 26: Frame preemption: IEEE Std 802.1QbuTM-2016[S]. New York: IEEE Standards Association, 2016.

作者简介:



林佳烁(1995 -),男,广东潮州人,博士生,研究方向为计算机网络、确定性网络和时间敏感网络。**E-mail:** linjs2021@mail.sustech.edu.cn

LIN Jia-shuo, born in 1995, PhD candidate, his research interests include computer network, deterministic network, and time-sensitive networking.



李伟超(1979 -),男,广东梅州人,博士,副研究员,研究方向为网络测量、移动计算和时间敏感网络。**E-mail:** liwc@pcl.ac.cn

LI Wei-chao, born in 1979, PhD, associate research fellow, his research interests include network measurement, mobile computing, and time-sensitive networking.



成剑(1980 -),男,湖南湘乡人,硕士,高级工程师,研究方向为时间敏感网络和移动承载网。**E-mail:** chengj@pcl.ac.cn

CHENG Jian, born in 1980, MS, senior engineer, his research interests include time-sensitive networking and mobile carrier network.



詹双平(1980 -),男,湖南怀化人,工程师,研究方向为时间敏感网络。**E-mail:** zhanshp@pcl.ac.cn

ZHAN Shuang-ping, born in 1980, engineer, his research interest includes time-sensitive networking.



冯景斌(1977 -),男,广东茂名人,硕士,高级工程师,研究方向为时间敏感网络、确定性网络、网络体系架构和网络芯片/FPGA设计。**E-mail:** fengjb@pcl.ac.cn

FENG Jing-bin, born in 1977, MS, senior engineer, his research interests include time-sensitive networking, deterministic network, network architecture, and network IC/FPGA design.