

面向业务可达性的广域工业互联网调度算法研究

胡致远*^① 胡文前^① 李香^① 马志^① 王文莉^②
王旭东^① 李春阳^① 黄天聪^①

^①(重庆大学微电子与通信工程学院 重庆 400044)

^②(中冶赛迪重庆信息技术有限公司 重庆 401120)

摘要: 工业互联网业务呈现出小规模、确定性的特征, 通常运行在大规模、异构的网络环境中, 业务的调度与功能链的编排难以与异构承载网资源匹配。基于此该文提出非工作保持型的多节点联合调度模型, 首先采用全路径时间协调算法, 将功能链从空间维度的拓扑编排扩展至时空维度; 其次, 针对网络节点中的同步调度问题, 提出了基于紧急度的流调度算法来平滑时延抖动, 进一步, 将时间触发调度延拓到大规模、异构且非同步的承载网中, 提出了虚拟到达队列编排算法, 利用业务同步机制替代时间同步, 保障了业务确定的可达性需求。仿真实验表明该文所提算法可提升业务的可达性, 保障其满足及时性、准时性、协同性需求。

关键词: 工业互联网; 确定性网络; 可达性; 业务编排; 非工作保持

中图分类号: TN915

文献标识码: A

文章编号: 1009-5896(2021)09-2608-09

DOI: 10.11999/JEIT200583

Research on Wide Area Industrial Internet Scheduling Algorithm Based on Service Reachability

HU Zhiyuan^① HU Wenqian^① LI Xiang^① MA Zhi^① WANG Wenli^②
WANG Xudong^① LI Chunyang^① HUANG Tiancong^①

^①(College of Communication Engineering, Chongqing University, Chongqing 400044, China)

^②(CISDI Chongqing Information Technology Co., Ltd. Chongqing 401120, China)

Abstract: In the large-scale and heterogeneous network environment, the industrial Internet service has the characteristics of small scale and certainty, so it is difficult to match the resources of the heterogeneous bearer network with the scheduling of the service and the orchestration of the function chain. Based on this, a multi-node joint scheduling model based on Non-workconserving is proposed. First, the full-path time coordination algorithm is adopted to extend the function chain from the spatial dimension topological orchestration to the space-time dimension; Then, aiming at the problem of synchronous scheduling in network nodes, a flow scheduling algorithm based on urgency is proposed to smooth delay jittering, furthermore, time-triggered scheduling is extended to large-scale, heterogeneous and non-synchronous bearer networks. A virtual arrival queue scheduling algorithm is proposed, which uses service synchronization mechanism instead of time synchronization to ensure the reachability requirements of service determination. Simulation results show that the algorithm proposed in this paper can improve the accessibility of the service and ensure service can reach in time, on time or cooperatively.

Key words: Industrial internet; Deterministic network; Reachability; Business orchestration; Non-workconserving

收稿日期: 2020-07-16; 改回日期: 2021-07-05; 网络出版: 2021-07-16

*通信作者: 胡致远 hzy@cqu.edu.cn

基金项目: 国家科技重大专项(2017ZX01030204), 重庆市基础研究与前沿探索(cstc2015jcyjA40021)

Foundation Items: The National Science and Technology Major Projects (2017ZX01030204), Chongqing Basic Research and Frontier Exploration (cstc2015jcyjA40021)

1 引言

随着工业互联网的发展和智能终端的泛在接入，小规模且确定性的工业应用对通信网络提出了新的要求，不同业务应满足相应的可达性需求。业务的可达性可定义为业务在端到端传输时对于服务质量(Quality of Service, QoS)的需求程度^[1]，可归纳为及时可达、准时可达和协同可达。及时可达业务需要数据分组尽快到达，准时可达业务需保证到达时延的抖动性，协同可达业务需保证业务间到达的时序逻辑。当前基于TCP/IP协议栈的互联网，其提供尽力而为的网络传送服务和有限的QoS保证导致大规模的底层网络资源分配不合理，难以满足准时与协同可达性需求。时间敏感网络(Time Sensitive Network, TSN)和确定性网络^[2,3](Deterministic Networking, DetNet)凭借在2层桥接和3层路由段上的确定性传输技术等优势，与具有低时延、高可靠特征的5G网络联合应用，成为了未来工业互联网网络架构的研究热点，如图1所示。

服务功能链是业务功能的有序集合，可通过功能链编排技术^[4]，在大规模异构的DetNet中定义局部应用场景的拓扑结构。现有研究大多在空间域上实现了对路径或参考点的编排^[5,6]，但未考虑业务在时间维度上要求的及时、准时与协同可达性要求。此外，针对在定义的拓扑中如何保障QoS指标，近年来国内外关于TSN中流量调度算法的研究已有初步成果。文献^[7-9]分析了工业实时通信中音视频桥接(Audio Video Bridging, AVB)网络的时延性能和使用网络微积分计算了802.1Qav策略中基于信誉值的整形机制(Credit Based Shaper, CBS)所带来的时延及抖动，表明CBS可有效降低控制流的时延抖动。文献^[10]在不同传播时延和不同流量强度的条件下，研究了802.1Qch中轮询转发队列(Cyclic Queuing and Forwarding, CQF)机制的时延上下边界值。文献^[11]采用802.1Qbv方案中时间

感知整形(Time Awareness Shaper, TAS)先入先出顺序，保障控制类业务流时延与抖动。针对802.1Qbv中门控制列表生成的复杂性，文献^[12]提出了软件定义网络(Software-Defined Networking, SDN)和802.1Qbv融合方案，利用SDN中控制平面的逻辑集中化来动态计算时间触发流的路由和调度，可快速在线更新调度时间表。文献^[13,14]中提出了一种简单的时分复用策略实现门控列表效果，在传输周期内为每条流分配固定时隙，各流只在对应时隙内传输，但该算法未能很好解决时隙分配冲突问题，也没有明确给出时隙大小划分方式。文献^[15]提出了梳状拟合(Comb-FITting, C-FIT)时隙分配算法，通过寻找最近可用时隙方式解决时隙分配冲突问题，但本质仍属于TAS机制，固定的报文优先级分类使得发生流突发与阻塞时，对于抖动影响较大。综上，当前对功能链编排的研究未考虑业务时间性需求；在交换设备同步的TSN中，未考虑流量特征变化对时延与抖动的影响；在大规模的DetNet中，时钟源多种且异构，难以实现全网时钟一致，如公专融合网络(Public Network Integrated Non-Public Network, PNI-NPN)，不可直接沿用基于时间同步要求的TSN调度算法，若改造设备用以传输同步，成本较高，目前国内外对此研究相对较少。

针对上述问题，本文提出确定性协调调度模型，为小规模且确定的业务规划转发路径和每一节点中的排队时延，以保证端到端业务的及时、准时与协同可达，包括在全网时钟同步的TSN中，编排承载面的业务流，引入动态紧急度优化TAS机制，从而调整业务调度时隙，既能保障上述规划的每一节点中的排队时延，也能够针对流的突发或阻塞进行时延补偿从而有效平滑时延抖动。在无统一时标的大规模DetNet中，编排用户面的业务入网顺序，在交换线卡处将同步的业务时钟作为调度驱动时钟，以实现业务的协同到达。

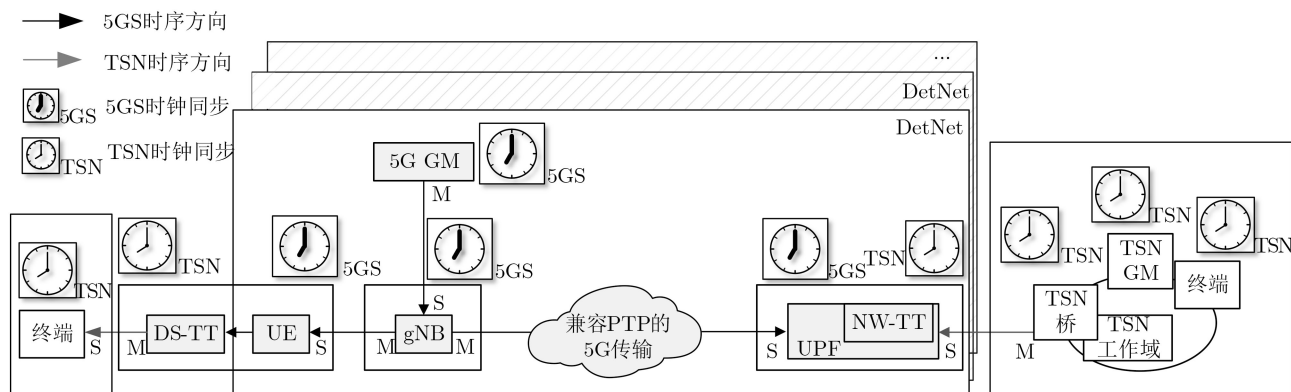


图1 未来工业互联网网络架构

2 确定性协调调度模型

传统网络在QoS保障上采用Workconserving模式与整形机制^[16]相结合的调度模型。表1对比了Workconserving与Non-workconserving模式的性能，后者可通过均衡调度来控制节点中流的抖动与速率，但受限于单节点或局部区域。工业互联网已实现小规模网络中的路径编排，并可获得全局性的时间调度信息，从而为本文将Non-workconserving模式与全路径、多节点的业务编排相结合提供了实现环境，对比模型如图2所示。

相较于采用Workconserving和整形相结合的调度模型，Non-workconserving与考虑时间性的全路径业务编排相结合，能够将承载面具有概率性、局部性的流交换与整形问题转变为确定性、全局性的业务面编排与承载面调度问题。在时钟异步的DetNet中，将小规模 x 、同步 y 且具有全局信息 z 的业务调度问题 A 对偶为大规模 X 、异构 Y 且只有局部信息 Z 的承载网调度问题 B ，利用业务同步机制代替时间同步，实现交换节点(Switch Fabric, SF)由时间驱动转换为事件驱动，从而构建准同步环境，保障业务的协同性。对偶问题如式(1)所示

$$A(x, y, z) \leftrightarrow A^* = B(X, Y, Z) \quad (1)$$

具体实现上，本文采用事先配置、事中控制与事后调整策略。在事先配置中提出全路径时间协调算法，在DetNet域中确定路径中每一节点的等待时延值，在时间基准不一致时采用虚拟到达队列编排算法，调整业务入网时序并校正线卡驱动时钟；在事中控制中提出基于紧急度的IntServ类节点业务流调度算法，计算数据流最优确定性排列调度

表1 工作模式对比

性能	Workconserving	Non-workconserving
触发方式	事件触发	时间触发
链路利用率	高	低
平均时延	低	高
时延抖动	高	低
速率控制	无	有

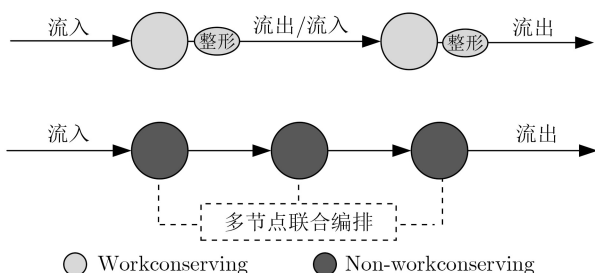


图2 调度模型对比

方案，后将采用虚拟输出队列(Virtual Output Queueing, VOQ)^[17]和迭代滑动(iterative SLIP, iSLIP)^[18]算法进行数据流输入与输出端口的交换；在事后调整中，可综合当前业务体验质量反馈与新业务需求，重新进行全路径时间协调。

3 协调算法研究

3.1 全路径时间协调算法

为保障不同业务流的及时性、准时性、协同性等需求，本文提出了全路径时间协调算法，并在业务时间基准不一致时，在源端提出虚拟队列编排算法，即在编排业务报文时不依据其自身携带的时间信息，而采用统一的业务到达时间体系，并依据入口等待时延建立等待队列，以业务时间校正调度时标，如图3所示。

假设研究的承载网中存在 n 个节点，定义图 $G = (V, E)$, V 为所有节点的集合， E 为图 G 所有边的集合， $E = \{(x, y) | x, y \in V\}$ 。

定义 $A = \{a_{x,y} | x, y \in V\}$ 为图 G 的连通矩阵， $a_{x,y}$ 表示终端 x 到终端 y 的连通性

$$A = \begin{pmatrix} 0 & a_{1,2} & \cdots & a_{1,s} \\ a_{2,1} & \ddots & & \\ \vdots & & 0 & \\ a_{s,1} & & a_{s,s-1} & 0 \end{pmatrix} \quad (2)$$

当点 x, y 不连通时， $a_{x,y} = \infty$ 。定义业务流在任意两点间的时延矩阵 $D = \{d_{x,y} | x, y \in V\}$ ，其中 $d_{x,y} = \sum_{i=0}^h (t_i + q_i + p_i)$ ， t_i 为相邻节点路径传播时延， q_i 为节点内排队等待时延， p_i 为节点内处理时延， $h \in [1, s]$ 为节点 (x, y) 间经过的节点数。

定义二进制决策变量 $X_{x,y}^j$ ， $X_{x,y}^j = 1$ 时，说明业务流 j 经过边 $\langle x, y \rangle$ ，反之不过边 $\langle x, y \rangle$ 。为提高网络资源利用率，降低业务流时延和，需优化路径传播时延与排队时延，其目标函数可表示为

$$\min F = \sum_{j=1}^m \sum_{x=0}^s \sum_{y=0}^s d_{x,y} X_{x,y}^j \quad (3)$$

其中， $j \in \{1, 2, \dots, m\}$ 为不同业务流集合。为保障不同业务流确定性QoS需求，需满足约束为

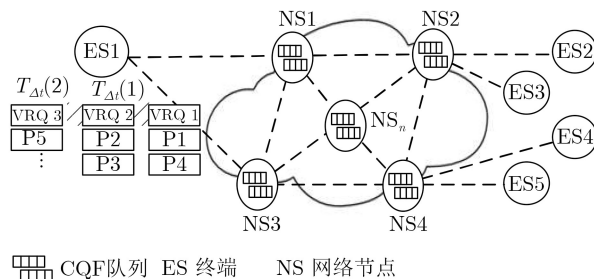


图3 全路径时间协调示意图

$$\left. \begin{aligned} \text{s.t. } & \sum_{x=0}^s \sum_{y=0}^s d_{x,y} X_{x,y}^{j_{\text{int}}} \leq D_{\text{down}}^{j_{\text{int}}}, j_{\text{int}} \in J \\ & D_{\text{up}}^{j_{\text{ont}}} \leq \sum_{x=0}^s \sum_{y=0}^s d_{x,y} X_{x,y}^{j_{\text{ont}}} \leq D_{\text{down}}^{j_{\text{ont}}}, j_{\text{ont}} \in J \\ & T_{\text{in}}^{j_{\text{co1}}} + \sum_{x=0}^s \sum_{y=0}^s d_{x,y} X_{x,y}^{j_{\text{co1}}} < \dots < T_{\text{in}}^{j_{\text{co2}}} \\ & \cdot \sum_{x=0}^s \sum_{y=0}^s d_{x,y} X_{x,y}^{j_{\text{co}}} , j_{\text{co}} \in J \end{aligned} \right\} (4)$$

其中, j_{int} 为及时确定性需求业务, 需保障时延下界, $D_{\text{down}}^{j_{\text{int}}}$ 为 j_{int} 的时延下界; j_{ont} 为准时确定性需求业务, 需保障时延上界, $D_{\text{up}}^{j_{\text{ont}}}$ 与 $D_{\text{down}}^{j_{\text{ont}}}$ 分别表示 j_{ont} 的时延上下界; j_{co} 为协同确定性需求业务, 需保障到达的时序逻辑, $T_{\text{in}}^{j_{\text{co}}}$ 为业务的入网时间。

传播时延 t_i 与传播距离呈正比关系, 节点排队时延主要受节点当前拥塞量、端口速率影响, 本文采用802.1Qch提出的轮询机制CQF实现Non-work-conserving模式, 完成对排队时延的控制。定义CQF队列的周期为 t_{CQF} , 节点处拥塞量为 M , 数据包在节点 i 处的等待次数为 k_i , 交换机端口速率为 v , 则

$$q_i = t_{\text{CQF}} \cdot k_i = t_{\text{CQF}} \cdot (M_i/v) \quad (5)$$

合理的CQF周期时间既不能太长也不能太短, 具体长度应由数据流状态决定。因此基于交通中Webster最佳信号周期公式^[19]可初始化CQF周期。该方法根据平均数据包的时延最小原则与稳态理论建立时延与CQF周期关联表达式

$$d_{\text{ave}} = \frac{t_{\text{CQF}}}{8(1-2q)} + \frac{x^2}{2q(1-x)} - 0.65 \left(\frac{t_{\text{CQF}}}{q^2} \right)^{\frac{1}{3}} x^{\frac{9}{2}} \quad (6)$$

其中, d_{ave} 为数据包平均时延; q 为节点实际吞吐量; x 为吞吐量饱和度, 即实际吞吐量与最大吞吐量之比。定义相邻节点间路径长度相同, 即相邻节点处理时延固定, 同一端到端节点 (x, y) 间的时延 $d_{x,y}$ 表达式可进一步表示为

$$\begin{aligned} d_{x,y} &= \sum_0^h (t_i + p_i) + \sum_0^h t_{\text{CQF}} \cdot k_i \\ &= \sum_0^h (t_i + p_i) + K \cdot t_{\text{CQF}} \end{aligned} \quad (7)$$

其中, h 为中间节点数, K 为总等待次数。将式(6)代入目标函数可得

$$\min F = \sum_{j=1}^m \sum_{x=0}^s \sum_{y=0}^s \left(\sum_0^{h_j} (t_i + p_i) + K_j \cdot t_{\text{CQF}} \right) X_{x,y}^j \quad (8)$$

其中, h_j 和 K_j 为变量且要求为整数, 采用分支定界法^[20]求解得最优解 \mathbf{X}^* , 确定各业务流的最优传播路径 h 与排队等待总次数 K

$$\mathbf{X}^* = (x_1^*, x_2^*, \dots, x_n^*)^T \quad (9)$$

$$\mathbf{x}^* = (h_1^*, K_1^* \quad h_2^*, K_2^* \dots h_m^*, K_m^*)^T \quad (10)$$

各节点处排队次数可由节点根据自身资源可利用率尽可能均匀分配, 可基本保证到达的时序逻辑类业务的正确性与及时、准时类业务的有效性。

大规模承载网络中, 存在网络资产归属或经营主体不同的应用场景, 业务报文携带的时间信息异构, 建立全网统一的时标体系实现困难, 而同步调度算法依赖于全局的时钟信息, 因此本文提出虚拟到达队列编排算法。虚拟到达队列是指业务报文按照“入口等待时间”在节点入口缓存中排列的缓存队列。虚拟时间是指业务报文不依据其自身携带的时间信息, 而采用统一的业务时间体系, 业务流到达各交换设备的线卡处时, 利用业务同步机制代替时间同步, 将时间驱动转换为事件驱动。其描述如下。

定义 T_{Ei}^j 表示业务报文 j_{con} 在节点 i 处的入口等待时间, T_{Ai}^j 表示 j 在节点 i 处, 在业务时间体系下获得的虚拟到达时间, $T_{j_{\text{con}}}$ 表示 j_{con} 到达目的节点的传输路径时延。根据入口等待时间的定义描述可知

$$T_{Ei}^{j_{\text{con}}} = T_{Ai}^{j_{\text{con}}} - T_{j_{\text{con}}}, i \geq 1 \quad (11)$$

其中, $T_{Ai}^{j_{\text{con}}}$ 可以是满足业务报文时序逻辑要求的任意合理时间。如有报文 $j_{\text{con}1}$ 和 $j_{\text{con}2}$, 到达目的节点的时序要求为 $j_{\text{con}2}$ 在 $j_{\text{con}1}$ 之前, 则应满足 $T_{Ai}^{j_{\text{con}1}} > T_{Ai}^{j_{\text{con}2}}$ 。

以入口等待时间为指标, 在业务入网前, 即应用层中建立虚拟到达队列VRQ(i), 假设 $T_E^{\text{VRQ}(i)}$ 表示VRQ(i)的入口等待时间, 其值等于VRQ(i)中任意业务报文的入口等待时间, $T_G^{\text{VRQ}(i,i+1)}$ 表示VRQ(i)与VRQ($i+1$)的保护间隔, 建立VRQ的原则如下: 相同 $T_{Ei}^{j_{\text{con}}}$ 的业务报文放在同一VRQ(i)中, 且 $T_E^{\text{VRQ}(i)}$ 越小的VRQ(i)越靠近入口缓存队列头部, 优先得到调度机会。设置VRQ(i)与VRQ($i+1$)的保护间隔

$$T_G^{\text{VRQ}(i,i+1)} \geq T_E^{\text{VRQ}(i)}, i \geq 1 \quad (12)$$

按式(11)设置VRQ之间的保护间隔, 可以保障业务报文按照指定的时序逻辑到达目的节点。全路径时间协调算法的伪代码可如表2所示, 输入 S 为松弛问题集合, 输出的 \mathbf{X}^* 为各业务流传播路径与等待次数最优解集合。

3.2 基于紧急度的IntServ类节点业务流调度算法

针对进入承载网中的不同类型业务流, 在各节点处为实现其确定的等待时延, 本文提出基于紧急度的IntServ类节点业务流调度算法UIS, 以周期协同方式确定调度周期, 实现时间驱动型调度, 其示

意图如图4所示。

首先根据业务映射结果将业务流进行分类，获取流的特征信息，如速率、周期等，假定网络中 n 个确定性需求流 f_1, f_2, \dots, f_n ，对于每一个流 $f_i (i = 1, 2, \dots, n)$ ，其分组长度为 L_i ，发送周期为 T_i ，流速为 R_i ，则存在 $T_i = L_i/R_i$ ，定义 T_s 为一次调度周期

$$T_s = \text{lcm}(T_1, T_2, \dots, T_n) \quad (13)$$

分组在调度周期中的顺序和位置决定了其延迟抖动情况，考虑存在流的突发、拥塞，时钟同步误差等，相同流到达不同节点，其特征可能发生改变。若沿用固定优先级，即固定调度顺序与位置，不能保障及时与准时性需求，因此定义紧急度

表 2 全路径时间协调算法伪代码

全路径时间协调算法伪代码
(1) IF 不需要应用层编排 THEN
(2) WHILE $S \neq \emptyset$
(3) REMOVE P FROM S
(4) Solve LP(P)
(5) IF LP(P)具有可行解 THEN
(6) X^* 成为LP(P)的最优基础解
(7) IF X^* 满足约束条件 THEN
(8) IF $\text{cost}(X^*) < F$ THEN
(9) 保留 X^* ，更新 F
(10) ELSE
(11) IF $\text{cost}(\text{LP}(P)) \geq F$ THEN
(12) 进行剪枝
(13) ELSE 划分子问题
(14) ELSE
(15) IF Q.count > 1 THEN
(16) 对Q快速排序
(17) ELSE Return
(18) FOR i 0 to Q.count - 1 BY 1 DO
(19) $T_{Ai}^{\text{con}} = T_{Pi} + \text{Random}$, $T_{Ei}^{P_i} = T_{Ai}^{P_i} - T_{Pi}$
(20) END

$C_i(t)$ ，即非同步动态优先级，动态编排每一节点中流的调度顺序

$$C_i(t) = \begin{cases} \left(k_1 \cdot \frac{L_i(t)}{L_d}\right) \cdot \left(k_2 \cdot \frac{D_i(t)}{D_d}\right), & D_i(t) \leq D_d \\ \text{drop}, & D_i(t) > D_d \end{cases} \quad (14)$$

其中， k_1 和 k_2 分别为业务流对丢包率与时延的敏感系数； $L_i(t)$ 和 L_d 分别为当前一段时间内该业务流的丢包率与丢包率门限； $D_i(t)$ 为 t 时刻业务流的时延， D_d 为该节点 d 处期望时延值，可具体表示为

$$D_d = \sum_{i=0}^d q_i + T_d \quad (15)$$

其中， q_i 为全路径时间协调算法所要求各节点处的排队时延， T_d 为到达节点 d 的链路时延之和。定义一个调度周期内，分组传输的开始与结束时刻之间的时间段成为该分组的传输窗口，窗口宽度为 W_i

$$W_i = (L_i + L_{\text{over}}) / R \quad (16)$$

其中， L_{over} 为发送分组所需要的固定开销， R 为输出链路速率。假定在一个周期内，流 f_i 有 k 个分组发送，对于序列中的第 j 分组，其传输窗口记为 $W_{tx}(i, j)$ ，相邻传输窗口的发送间隔为 $TD_i(j)$

$$TD_i(j) = \begin{cases} W_{tx}(i, j+1)_{\text{start}} - W_{tx}(i, j)_{\text{start}}, & j < k - 1 \\ W_{tx}(i, 0)_{\text{start}} + T_s - W_{tx}(i, k-1)_{\text{start}}, & j = k - 1 \end{cases} \quad (17)$$

其中， $W_{tx}(i, j)$ 为相对于调度周期开始时刻的起始时间。定义 f_i 的抖动特性PDV(f_i)为

$$\text{PDV}(f_i) = \max\{TD_i(m)\} - \min\{TD_i(n)\} \quad (18)$$

根据整体编排调度顺序，进而根据抖动特性、实际时延与期望时延差值等进行微调，寻得最优排列方案，再将调度周期内剩余时间窗口分配给非确定性流。

4 仿真

本文使用omnet++平台验证算法的有效性，

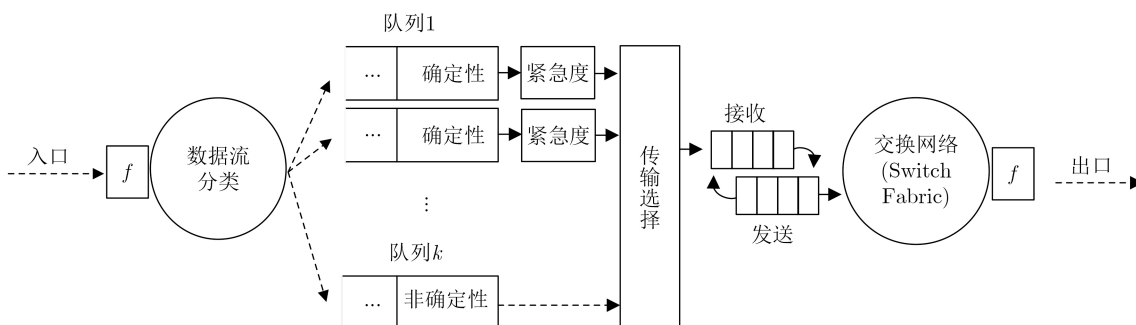


图 4 节点业务流调度示意图

根据电力数据采集网络的网状拓扑模型，设网络中节点的数目 $n(50, 100)$ ，其中源节点数目不超过10%，源目节点的比例为(1, 3)，边的密度为(0, 1)，以此生成随机测试拓扑，设置相邻接节点间的传播时延为2 ms，链路带宽为10 Gbps。本文分别在时间基准一致与不一致场景下，采用业务到达顺序验证算法对业务协同可达性的影响；采用时延与抖动验证算法对业务及时与准时可达性的影响。

4.1 时间基准一致条件下时序逻辑分析

在测试拓扑中选择任意一对源目节点进行端到端测试，在源节点中生成30个大小相同且具有一定时序逻辑需求的业务报文依次发往对应的目的节点，系统各节点时钟精确同步，时钟体系相同，CQF周期约为0.5 ms。仿真采用OpenDayLight项目中提供的两种服务功能链编排作为对比：Dijkstra和负载均衡算法^[21]。随机生成100个测试拓扑并重复上述实验，对实验结果进行概率统计。

通过检测到达目的节点的报文中的时间戳可判断时序逻辑的正确性。经过对业务报文到达时间的统计分析，使用对比算法发送的业务报文按序到达概率随着网络拥塞程度的增加显著下降，且由于最短路径算法不考虑节点负载率而导致节点产生更严重的“忙碌”，大量请求长时间未得到响应后进行重传，使得成功率降低程度更明显。而经过事先编排后按照指定路由与等待时延发送，能够基本保障时序逻辑类业务需求。当网络拥塞程度大于70%时，成功率的下降可能由网络长时间拥塞，源节点中Socket连接认定业务不可达，进行了重新调整业务连接的规避时间操作导致的。根据实验所统计的结果，可得出结论：在全局时间同步的基础上，全路径时间协调算法可有效提高业务到达时时序逻辑的正确性，如图5所示。

4.2 时间基准不同条件下时序逻辑分析

虚拟到达队列编排算法需提前固定转发的路径，

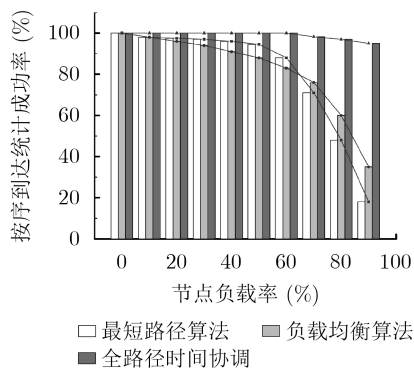


图5 业务到达时序逻辑图

获取承载网中传输时延范围。在每个测试拓扑中选定源节点和目的节点后，再以MPLS技术固定源目节点的传输路径，图6所示为一次随机拓扑中截取的1个源节点到3个不同目的节点的3条不同路径 S_1, S_2, S_3 的情况。

每个源节点根据不同的目的节点分发 P_1, P_2, P_3 3类业务报文，分别沿着3条由MPLS确定路径发送到目的节点。设业务报文虚拟到达时间变量观测窗口，精度为 10^{-6} s；同时，在发送报文时将源节点当前时间作为封装的数据传输并在网络接口eth0中检测，记录到达的当前时间，并计算其与业务报文携带发送时间的差值。实验分别在每次测试拓扑中模拟了网络无拥塞与丢包，负载率为20%与存在拥塞、丢包，负载率为120%时VRQ编排后业务报文的到达情况，实验结果如表3和表4所示，粗体标记表示错误的达到时序。

实验中产生误差的主要因素包括：业务报文获得的时间戳精度不够高；传输路径时延抖动等。实验结果1表明 P_1, P_2, P_3 按照指定的虚拟到达时间到达了目的节点，且时序逻辑正确。保持实验1中参数设置不变，重复大量实验(大于100组)。对比实验结果1，实验结果2误差 $T_{\Delta W}^{P_i}$ 偏大，主要是由网络拥塞造成的。重复大量实验，结果为 P_1, P_2, P_3 不能按照指定的虚拟到达时间到达目的节点，且时序不固定，25%的情况下出现了乱序。动态调整业务报

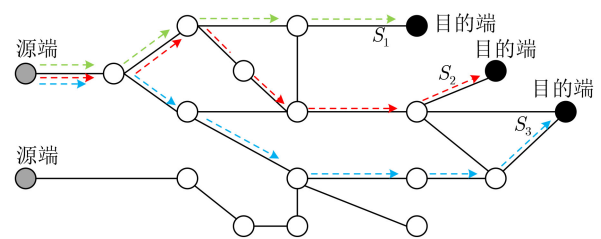


图6 虚拟到达队列编排模型仿真拓扑

表3 实验结果1

P	虚拟到达(ms)	观测窗口(ms)	误差(ms)
P_1	12.384206	13.021408	0.637202
P_2	11.253912	11.687653	0.433741
P_3	10.897265	11.327356	0.430091

表4 实验结果2

P	虚拟到达(ms)	观测窗口(ms)	误差(ms)
P_1	15.725614	27.684932	11.959318
P_2	11.687306	24.394108	12.706802
P_3	10.433697	25.870495	15.436798

文在源节点中形成的VRQ之间的保护间隔，重新完成实验2，97%的情况下 P_1, P_2, P_3 能按照指定的虚拟到达时间到达目的节点，不会出现乱序。出现3%的错误情况也为重新调整业务连接的规避时间操作所导致。

通过以上实验结果，可得出结论：(1)在没有网络拥塞和丢包的情况下，VRQ可以保障业务报文按照指定的虚拟到达时间到达目的节点，即在应用层保障业务报文传输的时序逻辑；(2)在有网络拥塞和丢包的情况下，可通过动态调整源端VRQ队列之间的保护间隔，重新编排业务报文的入网时序，提高保障业务报文按时序逻辑传输的能力。

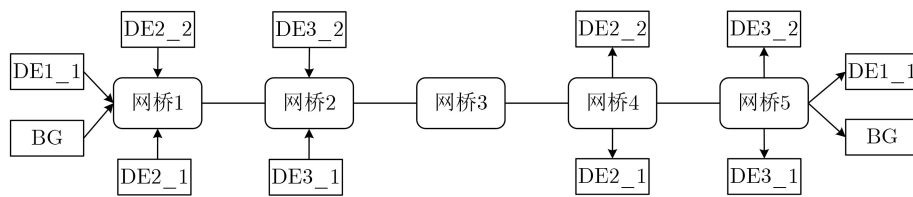


图7 UIS仿真拓扑图

确定性流DE采用CBR流量模型，背景流BG采用均匀分布流量模型，其参数如表5所示。

本文以常用的严格优先级调度算法SP^[22]作为对比实验，为直观显示结果，忽略传播时延与交换机内部处理时延，如表6、表7所示。并以DE1_1流为例，截取300 s仿真时间内分别采用SP与UIS算法的报文时延图进一步对比，如图8所示。

实验结果表明：SP算法相比于UIS算法，虽高优先级业务报文的最小时延表现较好，但时延抖动严重，不利于实时控制系统的预测需求。UIS算法

表5 流量参数

	DE	BG
流量类型	CBR	VBR
流速率(Gbps)	1.5	1.2~2.8
数据包载荷(byte)	1500	1000~2000
发送间隔(ns)	8000	5500~7500

表6 SP算法实验结果

流	最小时延(ns)	最大时延(ns)	抖动(ns)
DE1_1	5999	41599	35600
DE2_1	4799	24899	20100
DE2_2	4799	26233	21234
DE3_1	4799	33899	29100
DE3_2	4799	35233	30434
BG	17999	1829770	1811771

4.3 业务时延与抖动分析

对本文所提基于紧急度的IntServ类节点业务流调度算法UIS进行验证，仿真拓扑采用802.1CM示例拓扑，如图7所示。网络支持802.1Q协议，拓扑中包含：(1)确定性业务流，为高优先级恒定比特率型流量，主要为及时性与准时性需求流DE；(2)非确定性业务流，为低优先级可变比特率型流量，主要为背景流BG。拓扑中共有6条流，其中DE1_1, BG流从网桥1发送至网桥5节点，DE2_1, DE2_2流从网桥1发送至网桥4节点，DE3_1, DE3_2从网桥2发送至网桥5节点，报文优先级随DE1_1~DE3_2递减。

虽然整体时延值高于SP算法，但其时延抖动在理想情况下可降至为节点时钟自身偏移所带来的抖动值，本文中由于考虑时钟同步精度、硬件和软件处理时间误差影响，故时延抖动在1000 ns内。

在流量特征不发生变化的情况下，UIS算法性能等同于采用TAS算法，但考虑存在流的突发、拥

表7 UIS算法实验结果

流	最小时延(ns)	最大时延(ns)	抖动(ns)
DE1_1	42299	43299	1000
DE2_1	26299	27299	1000
DE2_2	27633	28633	1000
DE3_1	36299	37299	1000
DE3_2	37633	38633	1000
BG	72568	1829770	1757202

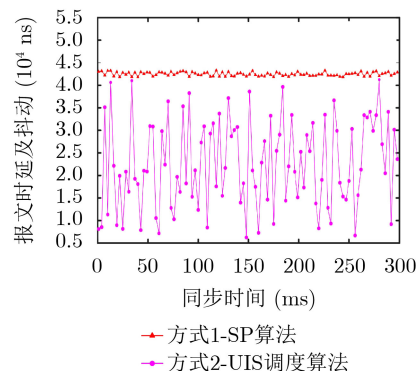


图8 DE1_1流时延及抖动图

塞, 时钟同步误差等, 相同流到达不同节点, 其特征发生改变。模拟该场景, 定义DE1_1流的时延需求为50 μs 内, DE2_1流的时延需求为30 μs 内, 两种业务流对丢包敏感。在DE2_1流发送时增加额外的突发流, 使得DE2_1流在网桥1中的调度受到影响, 延误一次调度周期发送, 采用TAS与UIS算法分别仿真DE1_1与DE2_1, 如图9所示。

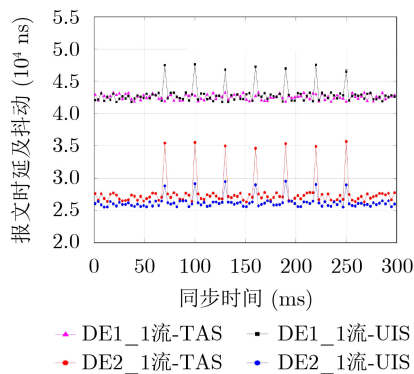


图9 TAS与UIS算法抖动性能对比图

DE2_1流的时延在不允许丢包的情形下采用固定优先级分配时隙的TAS调度, 由于存在突发流影响使其时延增加一个调度周期, 虽整体时延抖动平稳, 但最大时延增大且不满足30 μs 内时延需求, 而采用基于紧急度为依据的UIS算法可在网桥2~5中动态分配调度周期内的时隙, 交换DE1_1与DE2_1流的调度顺序, 结果表明: 虽增加了DE1_1流的最大时延值, 但UIS算法可使DE1_1与DE2_1流最终都满足其时延需求。综上可得出结论: UIS算法可有效保障业务流的及时与准时可达性。

5 结论

针对小规模、确定的工业互联网业务与大规模、异构的承载网适配不合理, 传统网络“尽力而为”的网络传送服务无法保证及时、准时和协同的业务可达性需求问题, 本文提出了确定性协调调度模型, 对业务流进行事先编排、事中控制与事后调整。仿真结果表明: 在时间基准一致时, 协同性需求业务流可在上述算法规划下按照正确逻辑顺序到达; 在基准不一致时, 将DetNet层的不确定性时延抖动反压到应用层中进行重排序, 且在网络拥塞时可调整VRQ保护间隔来保障协同性需求业务流的按序到达, 保证业务性能。此外, 相比较SP与TAS算法, 本文所提出的UIS算法能更好保证及时性与准时性业务流需求, 在理想环境下抖动为节点时钟自身偏移所带来的影响值。

参考文献

[1] 王俊文. 未来工业互联网发展的技术需求[J]. 电信科学, 2019,

35(8): 26–38. doi: 10.11959/j.issn.1000-0801.2019201.

WANG Junwen. Technical requirement of future industrial internet[J]. *Telecommunications Science*, 2019, 35(8): 26–38. doi: 10.11959/j.issn.1000-0801.2019201.

[2] 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. doi: 10.11959/j.issn.1000-436x.2019119.

HUANG Tao, WANG Shuo, HUANG Yudong, et al. Survey of the deterministic network[J]. *Journal on Communications*, 2019, 40(6): 160–176. doi: 10.11959/j.issn.1000-436x.2019119.

[3] 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. doi: 10.1109/COMST.2018.2869350.

[4] CAO Jiuyue, ZHANG Yan, AN Wei, et al. VNF-FG design and VNF placement for 5G mobile networks[J]. *Science China Information Sciences*, 2017, 60(4): 040302. doi: 10.1007/s11432-016-9031-x.

[5] YE Zilong, CAO Xiaojun, WANG Jianpin, et al. Joint topology design and mapping of service function chains for efficient, scalable, and reliable network functions virtualization[J]. *IEEE Network*, 2016, 30(3): 81–87. doi: 10.1109/MNET.2016.7474348.

[6] BECK M T, BOTERO J F, and SAMELIN K. Resilient allocation of service function chains[C]. 2016 IEEE Conference on Network Function Virtualization and Software Defined Networks, Palo Alto, USA, 2016: 128–133. doi: 10.1109/NFV-SDN.2016.7919487.

[7] MAXIM D and SONG Yeqiong. Delay analysis of AVB traffic in time-sensitive networks (TSN)[C]. The 25th International Conference on Real-Time Networks and Systems, Grenoble, France, 2017: 18–27. doi: 10.1145/3139258.3139283.

[8] CAO Jingyue, CUIJPERS P J L, BRIL R J, et al. Tight worst-case response-time analysis for Ethernet AVB using eligible intervals[C]. 2016 IEEE World Conference on Factory Communication Systems, Aveiro, Portugal, 2016: 1–8. doi: 10.1109/WFCS.2016.7496507.

[9] MOHAMMADPOUR E, STAI E, MOHIUDDIN M, et al. Latency and backlog bounds in time-sensitive networking with credit based shapers and asynchronous traffic shaping[C]. The 30th International Teletraffic Congress, Vienna, Austria, 2018: 1–6. doi: 10.1109/ITC30.2018.10053.

[10] THIELE D, ERNST R, and DIEMER J. Formal worst-case timing analysis of Ethernet TSN's time-aware and peristaltic shapers[C]. Proceedings of 2015 IEEE Vehicular Networking Conference, Kyoto, Japan, 2015: 251–258. doi:

- 10.1109/VNC.2015.7385584.
- [11] CRACIUNAS S S, OLIVER R S, CHMELÍK M, *et al.* Scheduling real-time communication in IEEE 802.1Qbv time sensitive networks[C]. The 24th International Conference on Real-Time Networks and Systems, Brest, France, 2016: 183–192. doi: 10.1145/2997465.2997470.
- [12] NAYAK N G, DÜRR F, and ROTHERMEL K. Incremental flow scheduling and routing in time-sensitive software-defined networks[J]. *IEEE Transactions on Industrial Informatics*, 2018, 14(5): 2066–2075. doi: 10.1109/TII.2017.2782235.
- [13] NOVAK A, SUCHA P, and HANZALEK Z. Efficient algorithm for jitter minimization in time-triggered periodic mixed-criticality message scheduling problem[C]. The 24th International Conference on Real-Time Networks and Systems, Brest, France, 2016: 23–31. doi: 10.1145/2997465.2997481.
- [14] WAN Tao and ASHWOOD-SMITH P. A performance study of CPRI over Ethernet with IEEE 802.1Qbu and 802.1Qbv enhancements[C]. 2015 IEEE Global Communications Conference, San Diego, USA, 2015: 1–6. doi: 10.1109/GLOCOM.2015.7417599.
- [15] CHITIMALLA D, KONDEPU K, VALCARENGHI L, *et al.* 5G fronthaul-latency and jitter studies of CPRI over Ethernet[J]. *Journal of Optical Communications and Networking*, 2017, 9(2): 172–182. doi: 10.1364/JOCN.9.000172.
- [16] LIEBEHERR J and YILMAZ E. Workconserving vs. non-workconserving packet scheduling: An issue revisited[C]. 1999 Seventh International Workshop on Quality of Service. IWQoS'99. (Cat. No.98EX354), London, UK, 1999: 248–256. doi: 10.1109/IWQOS.1999.766500.
- [17] HAN K E, SONG J, KIM D U, *et al.* Grant-aware scheduling algorithm for VOQ-based input-buffered packet switches[J]. *ETRI Journal*, 2018, 40(3): 337–346. doi: 10.4218/etrij.2017-0057.
- [18] MEI Lichun, QIAO Lufeng, CHEN Qinghua, *et al.* A Packet Dispatching Scheme with Load Balancing Based on iSLIP for Satellite Onboard CIOQ Switches[M]. LIANG Qilian, MU Jiasong, WANG Wei, *et al.* Communications, Signal Processing, and Systems. Singapore: Springer, 2016: 77–85. doi: 10.1007/978-981-10-3229-5_9.
- [19] AKGÜNGÖR A P and KORKMAZ E. Investigating parameter interactions with the factorial design method: Webster's optimal cycle length model[J]. *Tehnički Vjesnik*, 2018, 25(S2): 391–395. doi: 10.17559/TV-20170908185847.
- [20] KOLPAKOV R M and POSYPKIN M A. On the best choice of a branching variable in the subset sum problem[J]. *Discrete Mathematics and Applications*, 2018, 28(1): 29–34. doi: 10.1515/dma-2018-0004.
- [21] MEDHAT A M, CARELLA G, LÜCK C, *et al.* Near optimal service function path instantiation in a multi-datacenter environment[C]. The 11th International Conference on Network and Service Management, Barcelona, Spain, 2015: 336–341. doi: 10.1109/CNSM.2015.7367379.
- [22] DIEMER J, THIELE D, and ERNST R. Formal worst-case timing analysis of Ethernet topologies with strict-priority and AVB switching[C]. The 7th IEEE International Symposium on Industrial Embedded Systems, Karlsruhe, Germany, 2012: 1–10. doi: 10.1109/SIES.2012.6356564.
- 胡致远: 男, 1965年生, 博士, 教授, 研究方向为智能电网、无线接入网。
- 胡文前: 男, 1996年生, 硕士生, 研究方向为智能电网通信网。
- 李 香: 女, 1997年生, 硕士生, 研究方向为智能电网通信网。
- 马 志: 男, 1996年生, 硕士, 研究方向为智能电网通信网。
- 王文莉: 女, 1992年生, 硕士, 研究方向为无线接入网。
- 王旭东: 男, 1996年生, 硕士, 研究方向为智能电网通信网。
- 李春阳: 男, 1995年生, 硕士, 研究方向为智能电网通信网。
- 黄天聪: 男, 1971年生, 博士, 副研究员, 研究方向为新一代宽带无线移动通信技术。

责任编辑: 余 蓉