

基于负载均衡的大规模 TTE 消息调度表生成方法

叶帆¹, 陈银超², 王涛², 季袁冬¹, 罗懋康^{1,3}, 江秀强¹

(1. 四川大学空天科学与工程学院, 成都 610207;

2. 航空工业成都飞机设计研究所, 成都 610073;

3. 四川大学数学学院, 成都 610064)

摘要: 在复杂拓扑条件下, 机载网络存在大规模数据的高效传输需求. 此时, 传统的调度表生成方法存在高时间复杂度问题. 为了解决这个问题, 本文提出了一种基于负载均衡的时间触发以太网(Time-Triggered Ethernet, TTE)消息调度表生成方法. 该方法首先生成消息传输备选较短路径集, 综合地考虑消息长度和链路负载来选择消息路径、得到具有负载均衡特性的路径规划结果, 然后以提高时序规划效率为目标, 基于静态优先级生成调度表, 依据消息长度和周期进行消息排序, 并使用“背靠背”规则进行传输调度分配, 以保证消息的实时性和确定性. 仿真结果表明, 当可调度消息帧数超过 6000 条时, 在保证消息实时性的前提下, 该方法比传统的满足性模理论(Satisfiability Modulo Theory, SMT)时间规划方法减少 90% 以上的计算时间, 计算效率提升 10~100 倍, 可见该方法适合于求解复杂大规模数据调度表的生成问题. 本文的研究为改善机载网络的消息调度性能提供了一种可行方案.

关键词: 时间触发以太网; 路径规划; 时序规划; 负载均衡

中图分类号: O29 **文献标识码:** A **DOI:** 10.19907/j.0490-6756.2022.041001

A load-balanced TTE scheduling method for large-scale message transmission

YE Fan¹, CHEN Yin-Chao², WANG Tao², JI Yuan-Dong¹, LUO Mao-Kang^{1,3}, JIANG Xiu-Qiang¹

(1. College of Aeronautics and Astronautics, Sichuan University, Chengdu 610207, China;

2. AVIC Chengdu Aircraft Design and Research Institute, Chengdu 610073, China;

3. School of Mathematics, Sichuan University, Chengdu 610064, China)

Abstract: There are urgent requirements of large-scale message transmission in airborne networks with complex topology. When the traditional Time-Triggered Ethernet (TTE) scheduling methods are used to meet a requirement, the time complexity is very high. To solve this problem, we in this paper propose a TTE level message scheduling method based on the idea of load balancing. In this method, a set of candidate paths is firstly generated and then selected according to the message length and data flow link load. Those paths with load balancing characteristics are chosen. Then the messages are scheduled based on the static priority, sorted according to the message length and cycle, and allocated by using the ‘back to back’ rule. Finally, the method is compared to the traditional Satisfiability Modulo Theory (SMT) method to illuminate its performance. Simulation results show that, while the method can reduce the computation time by more than 90% when the schedulable message frames are more than 6000,

收稿日期: 2022-04-14

基金项目: 钱学森空间技术实验室创新工作站开放基金(GZZKFJJ2020001)

作者简介: 叶帆(1997-), 女, 河北沧州人, 硕士研究生, 主要研究方向为飞行器控制、引导与空天探测.

E-mail: 3150749480@qq.com

通讯作者: 江秀强. E-mail: jiangxiuqiang@scu.edu.cn

the computational efficiency is improved by 10~100 times on the basis of ensuring the real-time performance of the message. Therefore, the method is suitable for the complex large-scale data scheduling table generation and can feasibly improve the message scheduling performance of the airborne network.

Keywords: Time-triggered Ethernet; Routing; Scheduling; Load balancing

1 引言

分布式航电系统在航空航天领域具有广阔的应用前景^[1,2]. 此类系统要求机载网络必须具备高实时性、高容错性、高确定性及高带宽等特性. 时间触发以太网(Time-Triggered Ethernet, TTE)被认为是未来机载网络的理想解决方案^[2,3]. 时间触发以太网基于全局同步的时钟协议实现高确定性、高实时性的时间触发通信,并根据离线确定的静态通信调度表来传输时间触发消息. 当前,针对 TTE 调度表生成方法虽然已有较多的研究,但尚未形成统一标准,仍需深入研究^[4,5].

针对机载网络的大规模数据调度需求,本文研究 TTE 消息的路径规划和调度方法. 文献[6-9]设计了基于不同规则的静态优先级调度方法. 这些方法大多采用 TTE 现有的 AS6802 标准^[2]给出的星型、雪花型拓扑,消息路径唯一,且使用最短路径策略生成消息路径,规则确定、复杂度低,对大规模消息调度问题可行性好、计算时间也少,但是,这些方法没有考虑到多径及端系统规模超过 50 个^[8]的复杂拓扑等情形,在实际应用中容易出现因最短路径上的负载过大而造成消息阻塞的问题,无法进行后续的时序调度.

负载均衡在路径规划和时序规划问题中的应用越来越多^[10-13]. 文献[10]构建了一个考虑负载的路径代价函数,利用深度优先搜索算法寻找最小代价值的路由. 文献[11]基于 K 最短路径(K shortest paths, KSP)算法计算各业务的备选路径,在保证待选路径的较优性的前提下利用遗传算法优选出尽可能使全网负载均衡的业务路径组合. 文献[12]则研究了时间触发(Time-Triggered, TT)流量路径规划和时序规划对速率受限(Rate Constrained, RC)流量实时性的影响,并根据 KSP 算法和链路最大负载最小的原则进行路径规划. 但是,就我们所知,能够服务于大规模数据传输需求和负载均衡要求的机载网络的高效数据调度方法还未出现,有待进一步研究.

针对复杂拓扑条件下机载网络的大规模数据高效传输问题,本文提出了一种基于负载均衡思想

的 TTE 消息调度表生成方法. 该方法主要有以下几个特点:

(i) 考虑了复杂网络拓扑. 与星型网络及雪花型网络等相比,本方法中的消息路径并不唯一;

(ii) 考虑了消息长度的影响. 相比单个时隙传输,本方法具有更好的实时性;

(iii) 设计了基于负载均衡和静态优先级的调度分配方法.

此外,本方法具有较低的算法复杂度,适用于对大规模消息数据进行高效地时序规划.

2 模型

2.1 TTE 消息传输

TTE 支持三种类型的流量在网络中传输,即 TT 流量, RC 流量和尽力而为(Best-Effort, BE)流量^[2-6]. 针对机载网络的实际需求,本文仅考虑 TT 流量的传输. TT 流量根据各网络节点所加载的静态调度表中预定义的时间进行传输,由全局同步时间触发,具有确定的延迟和抖动. 该流量常被用于具有严格确定性和实时性要求的情形,能够满足未来机载网络对大规模数据传输确定性和时效性的需求.

时间触发以太网在空间和时间维度上调度网络流量,使得网络具有确定性,即确定数据包何时在何链路上传输,以及何时最终到达其目的端口. 这种调度在空间维度上体现为路径规划,需要为每个数据包找到一条合适的路由,即由链路和交换机组成的网络路径;在时间维度上是时序规划,需要沿着每个数据包路由的链路分配时隙(slot),每个时隙仅保留用于所配置帧的传输. 具体的传输过程如图 1 所示.

路径规划需要确定消息传输所经过的网络节点. 在图 1 中, Sender 为发送端, Switch 为交换机, Receiver 为接收端. 此外,还有两条消息,其中浅灰色的 TT 消息周期为 2ms,从 Sender1 传输到 Receiver,传输路径为

Sender1→Switch1→Switch3→Receiver,

深灰色的 TT 消息周期为 3ms,从 Sender2 传输到 Receiver,传输路径为

Sender2→Switch2→Switch3→Receiver.

消息在数据流链路上传输过程中的时序规划就是确定消息间的先后顺序。在图 1 中, 两个 TT 消息(深灰色和浅灰色所示)同时在 Switch3 到 Receiver 的链路上传输, 消息间的先后关系以它们的集群周期为基准。因为浅灰色 TT 消息和深灰

色 TT 消息的周期分别为 3 ms 和 2 ms, 其集群周期, 即消息周期的最小公倍数(least common multiple, LCM)为 6 ms。因此, 可以根据消息传输的开始时刻与集群周期的开始时刻间的时间间隔的长短来衡量消息的先后。

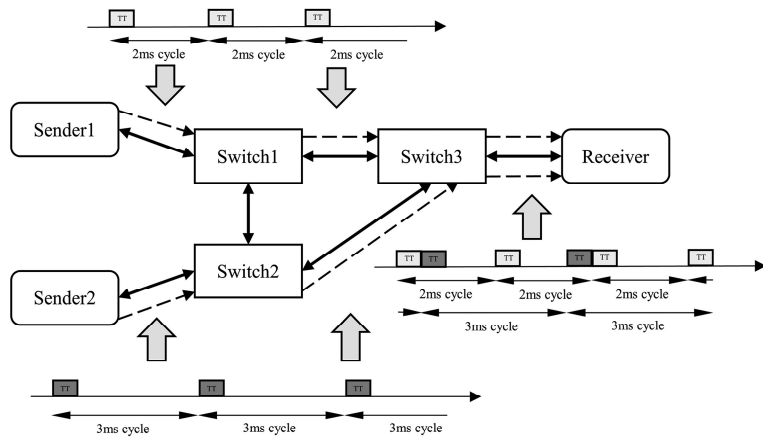


图 1 TT 消息传输示意图
Fig. 1 A scheme for the TT message transmission

2.2 TTE 消息调度模型

与传统的星型及雪花型网络拓扑不同, 实际的机载网络规模大、拓扑结构复杂。为了验证大规模、非单一路径网络拓扑下的算法性能, 本文以 TTE 多跳网络拓扑^[13]为基础, 将三组多跳网络拓扑互联组合形成如图 2 所示的 TTE 多跳互联网络。该网络中有终端系统(End System, ES)及网络交换机(Network Switch, NS)这两种节点角色, 各节点之间通过全双工链路进行连接。在图 2 中, ES 与 NS 之间的双向箭头代表矩形中各 ES 与 NS 之间均存在全双工链路, 只不过为了表示清晰在此简化为一条双向箭头。在各个节点中, 算法将加载离线生成调度表。根据调度表中的派发时刻和接收时刻, 算法对 TT 消息进行传输, 转发行为只允许发生在网络交换机中, 全双工链路允许双向通信。

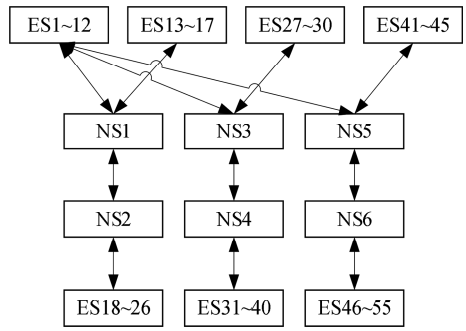


图 2 TTE 多跳互联网络拓扑
Fig. 2 Topology of the TTE network connected in multi-hop

将网络模型建模为有向图 $G=[V,L]$, 其中 V 为网络节点集合, $V=ES\cup NS$, L 为网络中数据流链路的集合。数据流链路是两个节点间的定向通信连接, $l_{pk}=[v_p,v_k]\in L, v_p,v_k\in V$ 。

定义网络拓扑矩阵 G ,

$$G_{ij} = \begin{cases} 0, & l_{ij} \notin L, \\ 1, & l_{ij} \in L \end{cases} \quad (1)$$

其中, 存在 $v_i, v_j \in V$, 使得当数据流链路 l_{ij} 存在时网络拓扑矩阵元素 G_{ij} 值为 1, 当数据流链路 l_{ij} 不存在时网络拓扑矩阵元素 G_{ij} 值为 0。

定义集群周期为消息周期的最小公倍数。由于 TT 消息均为周期性消息, 调度过程随 LCM 呈现周期性重复。因此, 本文所讨论的调度表生成问题只考虑一个 LCM 内的消息调度即可。

TT 消息帧格式符合 ARINC 664p7 规范, 消息通过帧的有效载荷进行传输。本文假设所有 TT 消息为周期性消息, 且不考虑打包和分段。将 TT 消息建模为如下的 6 元组形式, 通过该 6 元组能够完整描述 TT 消息的传输行为:

$$m_i = \{m_{i, \text{period}}, m_{i, \text{length}}, m_{i, \text{source}}, m_{i, \text{dest}}, m_{i, \text{path}}, m_{i, \text{offset}}\} \quad (2)$$

其中 $m_{i, \text{period}}$ 表示消息周期对应的时间长度, $m_{i, \text{length}}$ 表示消息长度引发的传输时间对应的时间长度, $m_{i, \text{source}}$ 表示消息传输源节点, $m_{i, \text{dest}}$ 表示消息传输目的节点, $m_{i, \text{path}}$ 表示消息传输路径, 为数据流链路

的有序序列, $m_{i, offset}$ 表示消息相对集群周期起始时间的偏移量. 为了将时序规划问题简化为整数规划问题, 我们使用时隙^[5]作为调度表的时间单位对消息的周期和消息传输时间进行整数化处理, 使得 $m_{i, period}$ 为消息周期对应的时隙数, $m_{i, length}$ 为消息长度引发的传输时间对应的时隙数.

本文主要考虑如何以确定性的优化规则分配大规模消息数据的传输路径和传输时隙. 为了研究方便, 并结合实际应用^[2-6], 我们作如下的假设:

- (i) 网络中所有消息都为周期性的 TT 消息;
- (ii) 网络中所有链路资源都可用, 交换机采用存储转发方式;
- (iii) 只考虑 TT 消息单播的情形;
- (iv) 调度表设计时以时隙为单位对消息周期、消息传输时间进行化整;
- (v) 假设网络中每次发送的 TT 消息都能封装在一个帧里, 且一个帧只传输一条消息, 不考虑帧的打包和分段.

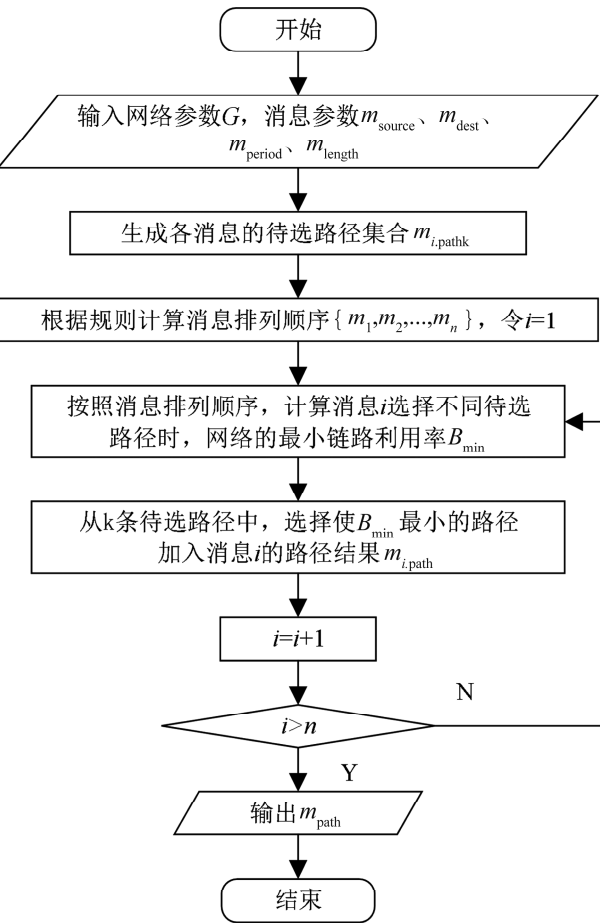


图 3 路径规划流程图
Fig. 3 Flow diagram of the routing method

3 基于负载均衡的调度表生成方法

3.1 路径规划

网络消息的路径规划过程根据消息的源节点 $m_{i, source}$, 目的节点 $m_{i, dest}$ 和网络拓扑 G 确定各个消息的传输路径 $m_{i, path}$. 该路径为数据流链路的有序序列 $\{l_1, l_2, \dots, l_n\}$, 具体实现过程如图 3 所示. 首先, 我们使用 Yen's KSP 算法^[14]生成各个消息的待选路径集合, 然后根据规则对消息进行排序, 最后按照排序顺序依次为消息从待选路径集合中选择网络链路最小负载最小的路径, 加入规划路径集合.

3.1.1 负载计算公式 定义数据流链路 l_{kp} 上的负载 B 的计算公式为

$$B_{kp} = \sum_{l_{kp} \in m_{i, path}} \frac{LCM}{m_{i, period}} \frac{m_{i, length}}{LCM} = \sum_{l_{kp} \in m_{i, path}} \frac{m_{i, length}}{m_{i, period}} \tag{3}$$

其中, $m_{i, path}$ 为消息传输路径, $m_{i, period}$ 为消息周期对应的时隙个数, $m_{i, length}$ 为消息长度引发的传输时间对应的时隙个数. 网络最小链路利用率 B_{min} 为

$$B_{min} = \min_{k, p \in V} B_{kp} \tag{4}$$

其中 k, p 为网络节点 v_k, v_p .

3.1.2 排序规则 生成待选路径后, 为了选出负载均衡的路径结果, 我们首先需要根据规则对消息进行排序. 本文考虑消息长度带来的影响, 定义规则为消息长度越长, 排序越靠前, 优先进行路径选择; 消息长度越短越靠后, 在已选路径的负载基础上进行路径选择, 相同长度的消息按照原本顺序进行排序.

3.1.3 选择过程 为了达到负载均衡的目的, 本文选择使得消息路径中链路最小负载最小的路径作为路径规划的结果. 由于某些消息的传输任务路径唯一, 导致一些主干链路的负载很高且难以缓解, 因而设计链路最小负载作为路径选择的条件能够关注到较小负载的均衡状况, 以达到负载均衡的目的.

3.2 时序规划方法

针对大规模消息的调度问题, 本文设计了一种静态优先级调度方法. 与智能优化算法相比, 该方法的计算时间更少、效率更高. 此外, 该方法基于规则, 因而具有确定性. 对于已知传输路径的消息时序规划问题, 调度过程需要确定各个消息在其路径的各个数据流链路中的首个数据帧的偏移量 $offset_{ij}$ (第 i 个消息在路径第 j 个数据流链路上

输时首个消息帧偏移量), 同时还要确定该消息帧传输的结束时刻 end_{ij} , 具体过程为: 首先根据优先级评估计算公式计算消息优先级, 然后按照从小到大的顺序对消息进行优先级排序, 评估值越小时排序越靠前、优先级越高, 最后根据优先级排序依次按照规则对消息进行调度分配。

(i) 优先级评估值计算. 由于优先级较高的消息将先进行时序规划, 较低优先级消息的排序会受到已规划消息的影响, 因而应使得较难调度的消息先进行时序规划, 较易调度的消息靠后. 鉴于单调速率算法^[15]已证明, 根据消息周期进行优先级排序具有最优性, 消息周期越短, 本文综合考虑消息周期与消息长度对优先级排序的影响, 消息周期越短, 优先级越高, 消息长度越长, 优先级越高; 反之,

消息周期越长, 消息长度越长, 优先级越低. 优先级评估值 p_i 的计算公式为

$$p_i = \frac{m_{i, \text{period}}}{m_{i, \text{length}}}$$

(5)

这样, 优先级评估值 p_i 越小, 消息周期越短、长度越长, 优先级越高. 当消息优先级评估值相同时, 按消息原始顺序进行排序。

(ii) 调度规则. 优先级排序后, 我们将按照优先级顺序对消息进行调度. 若该消息为链路中首个被调度的消息, 则令其集群周期偏移为 0, 该链路中后续消息将与前一个消息“背靠背”排布进行调度. 若产生冲突, 则令该消息后移一个 slot, 直到无冲突. 若无法消除该冲突, 则无法生成调度表. 具体规则如(6)~(8)式所示.

$$end_{i,j} = offset_{i,j} + m_{i, \text{length}}$$

(6)

$$offset_{i,1} = \begin{cases} 0, & \text{if } cont(m_{i, \text{pathl}}) = 0; \\ end_{q,n}, & \text{if } cont(m_{i, \text{pathl}}) > 0, \exists m_{i, \text{pathl}} = m_{q, \text{pathm}}, cont(m_{q, \text{pathm}}) = cont(m_{i, \text{pathl}}) - 1 \end{cases}$$

(7)

$$offset_{i,j} = \begin{cases} end_{i,j-1} + delay_{i,j-1}, & \text{if } cont(m_{i, \text{pathj}}) = 0; \\ \max(end_{i,j-1} + delay_{i,j-1}, end_{p,m}), & \text{if } cont(m_{i, \text{pathj}}) > 0, \\ \exists m_{i, \text{pathj}} = m_{p, \text{pathm}}, cont(m_{p, \text{pathm}}) = cont(m_{i, \text{pathj}}) - 1. \end{cases}$$

(8)

其中 $offset_{i,j}$ 表示消息 i 在其路径中第 j 个数据流链路上的传输时刻距离集群周期 LCM 起始的偏移量, $cont(m_{i, \text{pathj}})$ 表示数据流链路 $m_{i, \text{pathj}}$ 上已规划的消息偏移个数, $end_{i,j}$ 表示消息 i 在其路径中第 j 个数据流链路上传输完毕时刻距离集群周期 LCM 起始的偏移量, $delay_{i,j}$ 表示消息 i 在其路径中第 j 个数据流链路上传输的硬件延迟和排队抖动, 该数据依据实际的网络物理性能而离线设定。

4 仿真与分析

4.1 仿真设置

路径和时序规划仿真实验的硬件配置 CPU 为 Intel(R) Core(TM) i7-10510U、内存 16GB, 采用 MATLAB R2016a 进行编程计算, 并利用 PyCharm Community Edition 2021. 1. 1 x64 开发平台进行 SMT 对比仿真。

仿真采用的网络拓扑如图 2 所示, 包含 55 个端系统和 6 个交换机, 共 61 个网络节点, 且转发行为只发生在交换机上. 在仿真程序中, 该网络拓扑表示为矩阵 $G_{61 \times 61}$ 的形式, 如式(1)所示, 矩阵元素 G_{ij} 为 1 时节点 i 到节点 j 间存在数据流链路 l_{ij} , 矩

阵元素为 0 则表示不存在数据流链路。

表 1 消息传输需求

Tab. 1 Task of the message transmission

序号	源节点	目的节点	长度/slot	周期/slot
1	ES1	ES13	30	2
2	ES13	ES18	30	1
3	ES17	ES1	60	3
4	ES44	ES49	150	1
5	ES12	ES14	30	2

表 2 仿真实验消息设置

Tab. 2 Setup of the experimental messages

消息组别	消息数量/个	单个 LCM 内的帧数量/条
1	50	470
2	100	1319
3	124	1535
4	248	3070
5	496	6140

由文献[11], 我们设置网络链路的传输速率为 100 MB/s, 并根据 TTE 帧格式(符合 ARINC 664p7)设置帧长为 64~1518 B. 由文献[2], 我们设置消息周期为 15、25、30、50、75、150 ms, 时隙长度为 0. 5 ms, 并据此对消息周期和帧长进行取整

计算,可得帧长传输时隙为 1~3 slot,消息周期时隙为 30、50、60、100、150、300 slot. 在可达的前提下,令消息端口均匀分布,消息传输需求示例如表 1 所示.

为了验证本文所提方法的有效性和在时间复杂度方面的性能提升,我们采用传统的迪杰斯特拉(Dijkstra, DIJ)路径规划算法^[6,16]和经典的基于满足性模理论(Satisfiability Modulo Theory, SMT)的时序规划方法^[5],并与本文所设计的基于 K 短路径的负载均衡路径规划方法及基于规则的静态优先级调度(Static Priority Scheduling, SPS)方法综合,得到四种方法组合,并在不同消息规模下进行对比仿真实验. 这四种方法分别为:DIJ 最短路径算法+SMT 时序规划方法(DIJ+SMT),基于 K 短路径的负载均衡路径规划方法+SMT 时序规划方法(KSP+SMT),DIJ 最短路径算法+基于规则的静态优先级调度方法(DIJ+SPS),以及基于 K 短路径的负载均衡路径规划方法+基于规则的静态优先级调度方法(KSP+SPS). 对四个方法组合在不同消息规模下进行仿真实验时的具体消息设置参见表 2.

4.2 仿真结果与分析

4.2.1 仿真结果 我们以表 2 中第二组消息为例展示本文方法仿真结果,DIJ 算法与 KSP 方法的路径规划结果分别如图 4a 和 4b 所示,两种方法的时序规划结果如图 5、6 所示. 图 4 展示了部分路径结果矩阵 Path_{100×5},其中矩阵行元素为调度消息序号,列元素为路径结果中数据流链路排列序号,矩阵中的元素代表了传输路径中数据流链路的序号,0 代表无链路即无路径.

可以看出,当消息为 1~10 时,两种路径方法结果相同. 这验证了 KSP 方法的有效性和正确性. 当消息为 90~100 时,KSP 方法的数据流链路路径与 DIJ 算法不同. 我们以第 90 个消息为例进行分析. DIJ 算法的路径结果为数据流链路 1 传输至数据流链路 97,KSP 算法的路径结果为数据流链路 53 传输至数据流链路 149. 由于两个算法第 1 个消息的路径结果均为数据流链路 1 传输至数据流链路 98,因而对于第 90 条消息而言,KSP 算法缓解了数据流链路 1 上的负载. 由此,我们可以推断其他消息的链路负载情况,尤其当网络中链路负载不均匀时,DIJ 算法中最短路径的负载难以缓解,更容易出现路径阻塞的现象.

图 5 和 6 展示了 TT 消息调度表,表的横坐标

以时隙为单位,表示一个 LCM 内的消息帧排布情况,纵坐标表示网络中的数据流链路序号,图中的黑色小矩形代表各消息已被分配的时隙,通过调度表能够确定消息在网络中传输的路径和时间顺序.

由于消息路径中数据流链路上的时隙偏移 *offset* 决定了调度表中消息被分配的时隙的位置,因而通过图 7 和 8 中的时隙偏移 *offset* 分布直方图和分布曲线图可以看出,本文设计的 SPS 方法因采用了“背靠背”的排布规则,比 SMT 方法时隙偏移 *offset* 分布更加集中,更靠近 LCM 前端,体现在调度表中消息排列更为紧凑,更好地保证了 TT 消息传输的确定性和实时性.

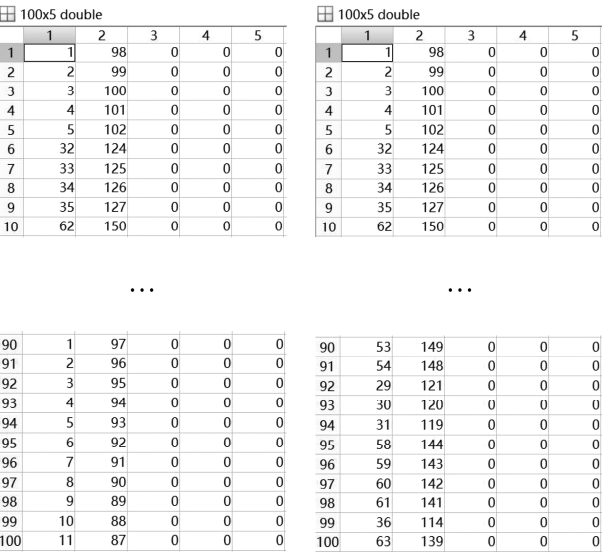


图 4 路径规划结果:(a) DIJ 算法路径矩阵;(b) KSP 方法路径矩阵

Fig. 4 Result of the routing methods; (a) DIJ algorithm routing matrix; (b) KSP algorithm routing matrix

4.2.2 评估指标与分析 调度方法的关键评估指标包括:计算时间指标、延时指标和负载指标.

表 3 计算时间的对比(单位:s)
Tab. 3 Comparison of the calculation times(Unit: s)

消息数量/个	50	100	124	248	496
方法名称					
DIJ+SMT	0.229	0.349	0.436	0.788	1.565
	7.710	16.446	24.017	150.125	—
KSP+SMT	1.059	2.378	2.644	5.945	12.789
	3.137	14.899	19.315	120.600	—
DIJ+SPS	0.229	0.349	0.436	0.788	1.565
	0.028	0.026	0.027	0.053	0.143
KSP+SPS	1.059	2.378	2.644	5.945	12.789
	0.022	0.027	0.026	0.050	0.118

注:“—”表示实验运行时间超过 1 h,停止实验.

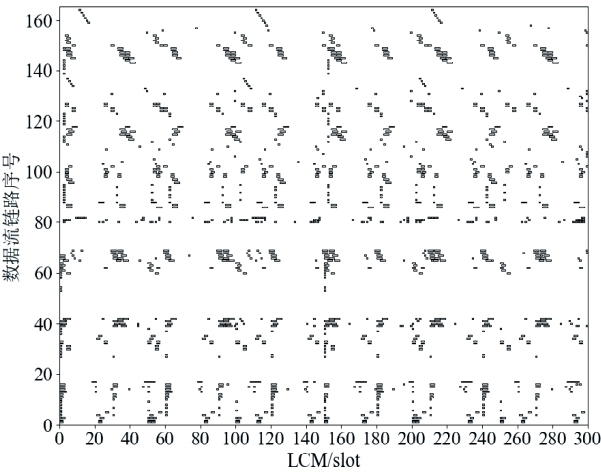


图 5 KSP+SMT 时序规划结果
Fig. 5 Scheduling result of the KSP+SMT method

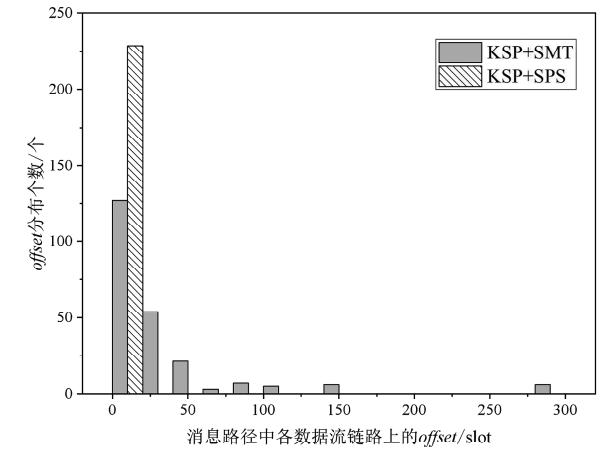


图 7 时隙偏移分布直方图
Fig. 7 Histogram of the time slot offsets distributions

上述四种方法在不同规模的消息数据下的计算时间的比较结果如表 3 所示,其中每个单元格里的上下两个数据分别是路径规划时间和时序规划时间. 通过比较可知,路径规划方法的计算时间都是多项式级别的,而 SMT 方法的计算时间呈现指数型增长. 当消息规模为 496 时,SMT 方法很难得出结果. SPS 方法的计算时间大多较短,且在消息规模为 496 时计算时间仍远小于 1 s. 因此,相比于 DIJ+SMT 和 KSP+SMT 方法,KSP+SPS 方法能够有效减少大规模消息的计算时间. 虽然 DIJ 算法耗时比 KSP 方法少,但当方法流量不均匀时,DIJ 算法更容易出现路径阻塞的现象,导致调度表生成失败. 因此,为了保证调度表生成方法的适应性,KSP+SPS 方法比 DIJ+SPS 方法更适用于复杂拓扑下大规模消息数据传输的场景.

延时指标包括平均延时 D_a 和平均延时百分比 D_{ap} . 假设各个消息的传输截止时间为其周期.

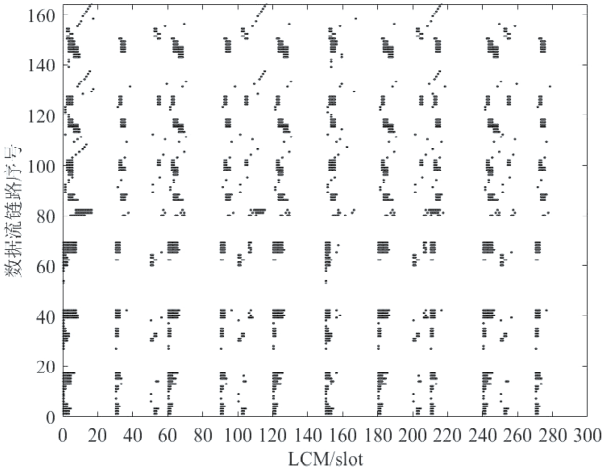


图 6 KSP+SPS 时序规划结果
Fig. 6 Scheduling result of the KSP+SPS method

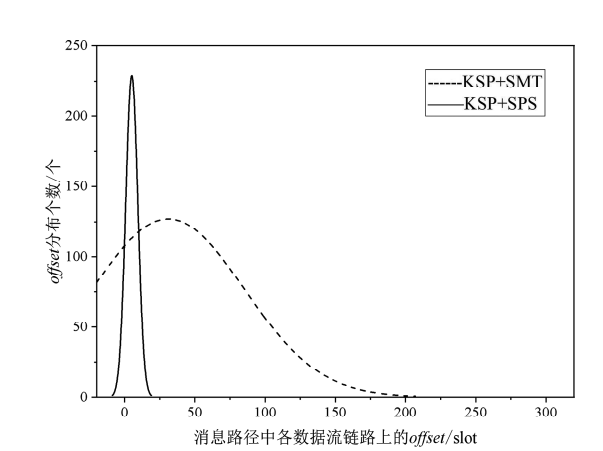


图 8 时隙偏移分布曲线
Fig. 8 Distribution curves of the time slot offsets

则该指标可通过下面两式进行计算:

$$D_a = \frac{1}{N} \sum_{i=1}^N (offset_{i, last} - offset_{i, 1} + m_{i, length}) \tag{9}$$
$$D_{ap} = \frac{1}{N} \sum_{i=1}^N \frac{offset_{i, last} - offset_{i, 1} + m_{i, length}}{m_{i, period}} \tag{10}$$

其中,消息个数为 N , $offset_{i, last}$ 为消息传输最后一条路径上的偏移时隙. 四种方法在不同规模消息数据下的延时指标比较结果参见表 4, 可以看到,四种方法的延时百分比相差不大, KSP+SPS 方法的表现好于两种 SMT 方法,与 DIJ+SPS 方法基本持平. 因此, KSP+SPS 方法能够较好地满足消息的确定性和实时性需求.

消息负载指标包括网络链路最大负载 BD_{max} 、平均负载 BD_{avg} 、负载标准差 BD_{sign} 和网络链路负载分布直方图, 由于网络负载之和不变, 若链路数

量相同则平均负载结果相同,为了对比方法效果平均负载和负载标准差的计算剔除了负载为 0 的链路. 两种路径规划方法在不同消息规模下的网络链路最大负载、平均负载和负载标准差如表 5 所示,在消息规模 248 和 496 下的链路负载分布直方图对比结果如图 9 和图 10 所示.

表 4 延时指标对比

Tab. 4 Comparison of the message delays

延时指标	消息数量/个	DIJ+SMT	KSP+SMT	DIJ+SPS	KSP+SPS
平均延时 D_a /slot	50	4.68	4.24	4.28	3.32
	100	5.26	5.35	5.15	5.36
	124	5.54	5.29	5.44	5.65
	248	5.72	5.74	6.99	7.15
	496	—	—	10.09	10.04
平均延时百分比 D_{ap} /%	50	7.11	6.57	6.45	5.09
	100	10.83	11.03	9.82	9.90
	124	10.59	10.29	9.77	9.94
	248	10.98	10.96	11.72	12.04
	496	—	—	15.63	15.29

表 5 网络链路负载指标对比

Tab. 5 Comparison of network dataflow link load

负载指标	消息数量/个	50	100	124	248	496
BD_{max} /%	DIJ	17.7	19.3	19.3	38.7	77.3
	KSP	8.7	19.3	19.3	38.7	77.3
BD_{avg} /%	DIJ	9.0	7.5	8.6	17.1	34.3
	KSP	3.1	6.4	7.1	14.1	28.2
BD_{sigm} /%	DIJ	3.8	5.7	6.0	12.0	24.0
	KSP	2.1	5.6	5.4	10.7	21.2

由表 5 中的 BD_{max} 数据可知,对于本文研究的拓扑而言,最短路径方法和负载均衡方法对于网络链路最大负载的影响都很小. 这是由于某些消息调度任务的路径唯一导致一些主干链路上的负载难以缓解. 但是,通过负载均衡方法能够缓解路径不唯一的消息调度任务带来的负载影响. 由表 5 中的 BD_{avg} , BD_{sigm} 可以看出, KSP 方法的负载平均值和负载标准差均小于 DIJ 算法. 这表明 KSP 方法平均负载更小且负载分布更为均衡. 进一步,由图 9 和图 10 可以看出,绿色的 KSP 方法较小负载的频率分布高于红色的 DIJ 算法,红色的 DIJ 算法

较大负载的频率分布高于或持平绿色的 KSP 方法,且 DIJ 算法中负载为 0 的链路也多于 KSP 方法. 可见,本文所设计的基于负载均衡的路径规划方法均衡负载的效果较为显著,能够较好满足机载网络中非 TT 消息传输的负载需求.

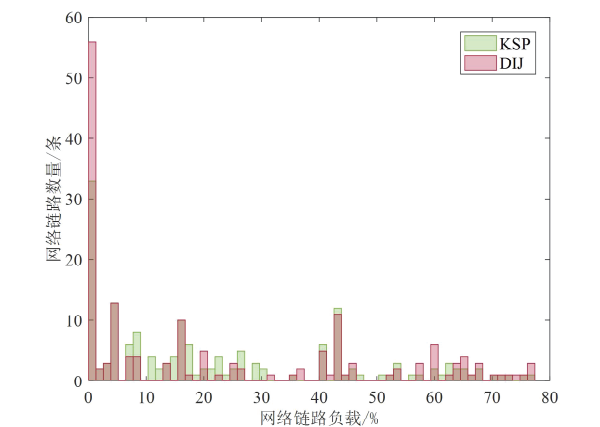


图 9 链路负载频率分布直方图(消息数量=496)
Fig. 9 Histogram of the probability distributions of data-flow link load with 496 messages

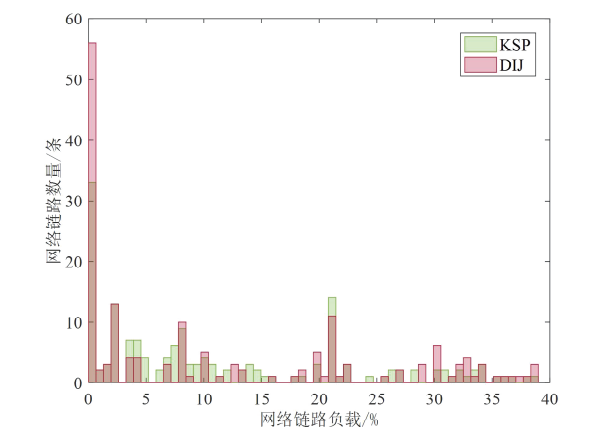


图 10 链路负载频率分布直方图(消息数量=248)
Fig. 10 Histogram of the probability distributions of data-flow link load with 248 messages

5 结 论

未来的机载网络拓扑趋于更加复杂、数据量更大,对调度表生成速度及负载可靠性要求更高. 本文提出了一种基于负载均衡的 TTE 消息调度表生成方法,该方法结合消息长度和负载的影响选取负载均衡的路径规划结果,根据优先级评估值以“背靠背”原则为基础计算消息的时间偏移,实现了对负载均衡的路径规划与时序规划问题的联合求解. 在平均负载、负载标准差方面比传统的 DIJ 算法提升了 10% 以上,改善了链路的负载均衡状况. 同时,计算效率比 SMT 方法提升 2 到 3 个数量

级,而延时指标与传统方法则在同一个量级,能保证消息传输的实时性.

综上,该方法在改善链路负载的同时提高了调度表生成的效率,能够满足更高的 TT 消息实时性和确定性需求,为机载 TTE 网络负载均衡和调度表生成提供了一种可行的解决方案.

参考文献:

[1] 刘美山,王勃. 分布式航电系统通用开放式软件架构设计研究[J]. 电子测试, 2022, 36: 123.

[2] 屈静. 航电机载网络的业务调度与性能评价技术研究[D]. 西安: 西安电子科技大学, 2020.

[3] Zhang X Y, Zhao X D. Architecture design of distributed redundant flight control computer based on time-triggered buses for UAVs [J]. IEEE Sens J, 2020, 21: 3944.

[4] Zhao R, Qin G H, Yan J, *et al.* Schedule optimization for TT Ethernet-based time-triggered automotive systems [J]. Int J Auto Tech, 2020, 21: 1483.

[5] Jian J, Wang L D, Chen H, *et al.* Scheduling optimization of time-triggered cyber-physical systems based on fuzzy-controlled QPSO and SMT solver [J]. Energies, 2020, 13: 668.

[6] 段沅廷. 时间触发网络的流量调度软件[D]. 成都: 电子科技大学, 2020.

[7] Gao W, Zhao B R, Mao X. Research on incremental scheduling backtracking algorithm for time-triggered ethernet [C]// 2nd International Conference on Advances in Computer Technology, Information Science and Communications. Piscataway: IEEE,

2020.

[8] Lu J, Xiong H G, He F, *et al.* Enhancing real-time and determinacy for network-level schedule in distributed mixed-critical system [J]. IEEE Access, 2020, 8: 23720.

[9] Wei A D, Zhang G Q, Zhang T. Research on time triggered Ethernet scheduling planning method [J]. J Phys Conf Ser, 2020, 1518: 012007.

[10] 刘国辉,祁志民. 一种基于路径代价的 TTE 调度算法[J]. 兵器装备工程学报, 2021, 42: 163.

[11] 赵俊龙. 时间触发以太网综合业务调度算法与管控平台研究[D]. 成都: 电子科技大学, 2018.

[12] 郑重,何锋,李浩若,等. 基于贪婪随机自适应搜索法的 TTE 通信调度算法[J]. 北京航空航天大学学报, 2021, 47: 2268.

[13] Zhao L L, Ni M, Cui X D, *et al.* The research of scheduling algorithm based on time-triggered Ethernet [C]// 22nd International Conference on Advanced Communication Technology. Piscataway: IEEE, 2020.

[14] Guo A P, Yuan C H. Network intelligent control and traffic optimization based on SDN and artificial intelligence [J]. Electronics, 2021, 10: 700.

[15] Sharma R, Nitin N, Alshehri M, *et al.* Priority-based joint EDF-RM scheduling algorithm for individual real-time task on distributed systems[J]. J Supercomput, 2021, 77: 890.

[16] 邓希,胡晓兵,江代渝,等. 基于混合遗传算法的柔性作业车间机器人和 AGV 规划[J]. 四川大学学报: 自然科学版, 2021, 58: 67.

引用本文格式:

中文: 叶帆,陈银超,王涛,等. 基于负载均衡的大规模 TTE 消息调度表生成方法[J]. 四川大学学报: 自然科学版, 2022, 59: 041001.

英文: Ye F, Chen Y C, Wang T, *et al.* A load-balanced TTE scheduling method for large-scale message transmission [J]. J Sichuan Univ: Nat Sci Ed, 2022, 59: 041001.