面向大规模时间敏感网络的分组调度机制

邱雪松[1]，黄徐川[1]，李文萃[2]，李温静[3]，郭少勇[1]
（1. 北京邮电大学网络与交换技术国家重点实验室，北京 100876；2. 国网河南省电力公司信息通信公司，河南 郑州 450052；
3. 国网信息通信产业集团有限公司，北京 102211）

摘 要：针对大规模时间敏感网络中时间触发流量的确定性时延问题，提出了一种分组调度机制。所提机制通过拓扑修剪策略和基于谱聚类的流分组策略，避免了由于网络拓扑规模和流量规模剧增对调度响应速度的影响，提升了调度计算效率。仿真实验结果表明，所提机制在大规模调度的场景下，可以在较短时间内进行模型的求解，并且保证一定的调度成功率。

关键词：时间敏感网络；时间触发流量；拓扑修剪；谱聚类
中图分类号：TP393
文献标识码：A
doi: 10.11959/j.issn.1000−436x.2020203

Group-scheduling mechanism for large-scale time-sensitive network

QIU Xuesong[1], HUANG Xuchuan[1], LI Wencui[2], LI Wenjing[3], GUO Shaoyong[1]
1. State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China
2. Information & Telecommunication Company of State Grid Henan Electric Power Company, Zhengzhou 450052, China
3. State Grid Information and Telecommunication Group Company Limited, Beijing 102211, China

Abstract: In order to achieve the deterministic delay of time-triggered traffic in a large-scale time-sensitive network, a kind of group-scheduling mechanism was proposed. By designing a topology pruning strategy and a flow grouping strategy based on spectral clustering, the impact of the rapid increase in network topology scale and traffic scale on the speed of scheduling response was avoided, and the scheduling calculation efficiency was improved. The results of experiment show that the mechanism can solve the model in a relatively short time for large-scale scheduling problem and achieve a relatively high scheduling success rate.

Key words: time-sensitive network, time-triggered traffic, topology pruning, spectral clustering

1 引言

随着信息技术的飞速发展，在工厂自动化控制、自动驾驶和电力自动化等领域，对数据流传输的时延要求超出了传统以太网的可控范围[1]。为解决传统以太网中实时数据的可靠传输问题，IEEE 802 工作组提出了时间敏感网络（TSN, time-sensitive network）的概念。TSN 支持时间触发流量（TT, time-triggered traffic）的传输，该类流量具有周期性传输的特点，并具有确定性的低时延要求。针对 TT 的调度问题出现了大量的研究[2-10]。

文献[2]使用可满足性模理论（SMT, satisfiability modulo theory）和优化模理论（OMT, optimization modulo theory）来实现有效调度。文献[3]基于整数线性规划（ILP, integer linear programming）计算式生成门控制列表，通过计算全局调度将资源最佳地分配给时间触发流量。文献[4]提出了一种混合遗传算法，用于生成时间触发流量的静态调度

收稿日期：2020−06−18；**修回日期**：2020−09−04
基金项目：国家电网有限公司总部科技基金资助项目（No.5700-202024176A-0-0-00）
Foundation Item: State Grid Corporation of China Science and Technology Project (No.5700-202024176A-0-0-00)

表，优化了时隙的分配数量并提高了传输效率。文献[5]提出了一种基于遗传算法的启发式调度算法，使用路由和调度的联合约束来生成静态的全局调度，从而使可调度性、传输效率和资源利用率得到显著提高。文献[6-7]基于 ILP 提出了针对路由和调度联合问题的解决方案。文献[6]使用 2 个性能指标（即端到端时延和调度能力）来评估不同流量模式和网络拓扑的实验结果。文献[7]利用 ILP 解算器对参数变化很大的问题实例进行了求解，根据求解时长对算法的性能进行了评估。文献[8]基于 SMT 提出了一种联合路由和调度的迭代算法，但是算法性能对流量之间传输路径的冲突程度敏感，导致成功率较低。文献[9-10]提出了时间敏感软件定义网络（TSSDN, time-sensitive software-defined network）的概念和用于计算调度的 ILP 式，TSSDN 利用逻辑集中的控制平面来计算全局的调度方案。

上述研究通过不同方法给出了时间触发流量可行的调度机制，但是在大规模调度场景下的求解时间难以满足调度需求。为此，本文针对大规模时间敏感网络中时间触发流量的调度问题进行了研究，具体包括以下三部分。

1) 建立了分组调度模型，在每一组流量的调度中，通过时间和空间上将待调度流量分离的时分多址（TDMA, time division multiple access）思想消除了排队时延，保证了时延的确定性。

2) 针对网络拓扑规模庞大的问题，设计了一种拓扑修剪策略。通过减少不必要的约束条件简化调度模型，减少了调度的求解耗时。

3) 针对流量规模庞大的问题，设计了一种基于谱聚类的流分组策略。通过分组调度的方式减少调度求解时间，同时保证较高的成功率。

2　数学模型

2.1　时延分析

时间触发流从源主机到目的主机沿着指定路径进行周期性传输，其端到端时延包括传播时延、发送时延、处理时延，当网络中发生拥塞时，还包括排队时延。时延分析如图 1 所示。

图 1 中，t_i 表示节点 v_i 的发送时刻，d_{trans}、d_{prop} 和 d_{proc} 分别表示发送时延、传播时延和处理时延。当发生排队时，排队时延为 d_{queu}，D 表示某个节点和下一节点之间的端到端时延，如式(1)所示。

$$D = d_{\text{trans}} + d_{\text{prop}} + d_{\text{proc}} + d_{\text{queu}} \tag{1}$$

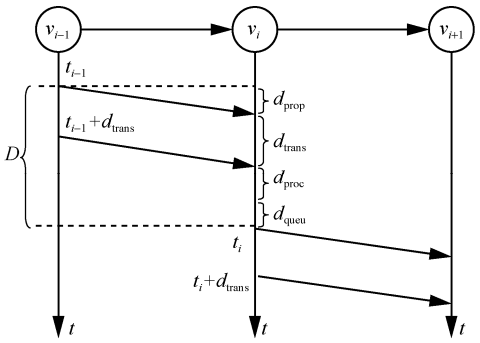

图 1　时延分析

当不发生排队时，D 只包含发送时延、传播时延和处理时延。传播时延由信道长度和介质中信号的传播速率决定，可以认为具有确定性。交换机内部的处理时延与其存储转发的能力有关，可以认为具有确定性。发送时延由帧大小和发送速率决定，也具有确定性。当网络中发生拥塞时，在交换机内部排队时间无法确定，导致产生的排队时延具有不确定性。

由于端到端时延的不确定性主要来源于可能包含的排队时延，因此在设计时间触发流的调度模型时，需要避免排队情况的出现，从而保证其传输的确定性。

2.2　调度模型

当多个数据分组试图在交换机的同一个输出端口进行传输时，会发生排队的情况，若在时间触发流传输时为其分配特定的链路资源仅供其传输，就可以消除排队的情况从而保证端到端时延的确定性。在时间上，可以在一个基本周期内为不同的时间触发流分配不同的时隙；在空间上，可以为不同的时间触发流分配不冲突的链路资源。通过这种在时间和空间上将不同时间触发流分离的时分多址思想，可以实现每个时间触发流到达交换机时，始终有一个空状态的队列为其开放，从而避免了排队情况的出现。

如图 2 所示，一个基本周期被分为多个时隙，每个时隙从 0～$t_{\max}$ 编号，被分给特定的时间触发流进行传输，且时隙的长度足够宽，足以使最大传输单元（MTU, maximum transmission unit）的数据分组穿过最长的网络路径。

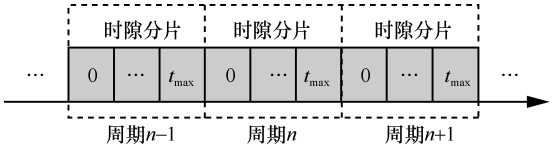

图 2　基本周期的时隙分片

基本周期能提供的时隙分片数量有一个上限值。例如，对于 1 Gbit/s 的链路，假设 MTU 为 1 500 B，数据分组允许穿越的最长网络路径为 8 跳，则一个 MTU 的数据分组穿越最长网络路径的总发送时延为 1 500 B×8 ÷ 1 Gbit/s×8 = 0.096 ms。在仅考虑发送时延的情况下，5 ms 的基本周期最多可以提供 5 ÷ 0.096 ≈ 52 个时隙分片，考虑到传输距离及交换机处理能力等影响因素，实际时隙分片数量的上限要更小一点。

网络的拓扑被建模为有向图 $G=(V,L)$，其中 $V=S\cup H$ 是网络中所有节点的集合，S 表示网络中的交换机，H 表示终端的主机。$L\subseteq V\times V$ 表示网络中相互连接的 2 个节点之间的一条有向全双工链路。有序对(V_i,V_j)和(V_j,V_i)分别表示 2 条独立的传输链路，有序对的第一个元素表示发送节点，第二个元素表示目的节点。定义网络中时间触发流的集合 F 为

$$F=\bigcup_{m=1}^{\max} G^m=\{f_i\} \tag{2}$$

所有的时间触发流按照特定的分组策略被分组为：$G^1, G^2, \cdots, G^{\max-1}, G^{\max}$，其中 max 表示分组的数量，在调度环节的每次迭代中完成一组时间触发流集合的调度。一组时间触发流集合 G^m 中所有流可能的传输路径的集合为

$$P_{G^m}=\bigcup_{\forall f_j\in G^m} P_{f_j}=\{p_i\} \tag{3}$$

其中，f_j = (s, d)表示集合 G^m 中的一个流，s 和 d 分别表示 f_j 的源主机和目的主机，P_{f_j} 表示流 f_j 可能通过的路径的集合。在本文模型中，P_{f_j} 被设定为流 f_j 最短路径的集合，于是 P_{G^m} 包括每个时间触发流 $f_j\in G^m$ 从源主机到目的主机的所有最短路径。时隙分片的集合 T 和所有链路的集合 L 分别如式(4)和式(5)所示。

$$T=\{t_i\} \tag{4}$$

$$L=\{l_i\} \tag{5}$$

定义如式(6)～式(9)所示映射。

$$\mathrm{FP}=\{X_{(f,p)}\},\forall f\in G^m,\forall p\in P_{G^m} \tag{6}$$

$$\mathrm{PL}=\{Y_{(p,l)}\},\forall p\in P_{G^m},\forall l\in L \tag{7}$$

$$\mathrm{LT}=\{M_{(l,t)}\},\forall l\in L,\forall t\in T \tag{8}$$

$$\mathrm{PT}=\{N_{(p,t)}\},\forall p\in P_{G^m},\forall t\in T \tag{9}$$

FP 为时间触发流和路径之间的映射，如果流 f 通过路径 p 从源主机到达目的主机，则 $X_{(f,p)}$=1；否则 $X_{(f,p)}$=0。PL 为路径和链路之间的映射，如果路径 p 包含链路 l，则 $Y_{(p,l)}$=1，否则 $Y_{(p,l)}$=0。LT 为链路和时隙之间的映射，如果时隙 t 上的链路 l 在先前的迭代调度过程中已经被占用，则 $M_{(l,t)}$=1，否则 $M_{(l,t)}$=0。PT 为路径与时隙之间的映射，如果时间触发流沿着路径 p 在时隙 t 上进行了传输，则 $N_{(p,t)}$=1，否则 $N_{(p,t)}$=0。

定义如式(10)～式(12)所示约束。

$$\forall p\in P_{G^m}:\ \sum_{\forall t\in T} N_{(p,t)}\leqslant 1 \tag{10}$$

$$\forall f\in G^m:\ \sum_{\forall p\in P_{G^m}}\sum_{\forall t\in T} N_{(p,t)}X_{(f,p)}\leqslant 1 \tag{11}$$

$$\forall t\in T,\forall l\in L:\ M_{(l,t)}+\sum_{\forall p\in P_{G^m}} N_{(p,t)}Y_{(p,l)}\leqslant 1 \tag{12}$$

约束条件(10)表示每一条路径最多只能分配到一个时隙上。每一个时间触发流最多只能分配到一个时隙上，由于该流最终只能选择一条最短路径进行传输，因此对于每个时间触发流而言，最多只能为其一条可能的最短路径分配时隙，则有约束条件(11)。如果具有相同链路的 2 条路径应被分配到同一个时隙时，可能引起冲突，因此某条链路的某个时隙上最多只允许一个时间触发流进行传输，则有约束条件(12)。

在时间触发流量集合和网络拓扑已知的情况下，为了提高调度的成功率，调度机制应在保证时延确定性的前提下，尽可能地提高网络中能够容纳的时间触发流的数量。另外，由于一个时间触发流对应多个其可能传输的最短路径，但是最终只会选择其中一条进行传输，因此可以把时间触发流分配到时隙上的问题转化为把路径分配到时隙上的问题，则优化的目标是尽可能地提高分配到时隙中的路径的数量。某个流组 G^m 的调度模型可以形式化描述为

$$\begin{aligned}&\max \sum_{\forall p\in P_{G^m}}\sum_{\forall t\in T} N_{(p,t)}\\ &\text{s.t. } \forall p\in P_{G^m}:\ \sum_{\forall t\in T} N_{(p,t)}\leqslant 1\\ &\forall f\in G^m:\ \sum_{\forall p\in P_{G^m}}\sum_{\forall t\in T} N_{(p,t)}X_{(f,p)}\leqslant 1\\ &\forall t\in T,\forall l\in L:\ M_{(l,t)}+\sum_{\forall p\in P_{G^m}} N_{(p,t)}Y_{(p,l)}\leqslant 1\end{aligned} \tag{13}$$

3　拓扑修剪策略与流分组策略

3.1　拓扑修剪策略

模型约束条件的建立与网络的拓扑具有密切关系。根据式(12)可知，拓扑结构中每一条链路在每一个时隙上都会为模型添加一条约束条件。而当拓扑规模更加庞大时，模型的约束条件数目会快速增加，从而导致问题的复杂程度急剧增加，最终导致求解时间不能被接受。

然而，对于一组待调度的时间触发流而言，每一个时间触发流最终只会选择最短路径集合中的一条路径进行传输，该组时间触发流可能占用的链路资源为最短路径集合中所有路径所包含的链路。因此，整个拓扑结构中剩余的链路在本组调度中不会有流通过，不可能发生同时传输的冲突情况。通过拓扑修剪的方式将本组时间触发流调度时的网络拓扑进行修剪，删除不会进行传输的链路，从而达到减少模型约束条件的目的。

图 3 是上述拓扑修剪的一个示例。图 3 (a)为网络原拓扑结构，网络中存在 2 个待调度的流量，分别为 $flow_1=(S_7, S_5)$和 $flow_2=(S_3, S_5)$。$flow_1$ 从源节点到目标节点有 2 条最短路径，分别为 $path_1$ 和 $path_2$，而 $flow_2$ 从源节点到目标节点有 2 条最短路径，分别为 $path_3$ 和 $path_4$。这 4 条路径所包含的链路是本组调度可能占用的链路资源，每条链路都会为模型的求解添加约束条件。拓扑修剪之后的结果如图 3(b)所示，链路的减少使模型的约束条件减少，从而降低了问题的求解规模。另外，由于拓扑修剪策略不会影响最终的调度结果，因此调度成功率保持不变。

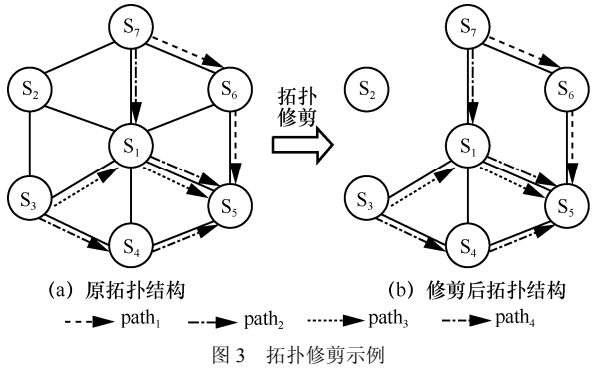

图 3　拓扑修剪示例

3.2　基于谱聚类的流分组策略

前文所述的调度模型是一个 ILP 问题，该问题是一个 NP-hard 问题。随着待调度流量数量的增加，模型的求解时间会急剧增加而变得不能被接受。首先对所有的时间触发流进行分组，然后对各个流组分别进行调度，即对每个流组的 ILP 模型进行求解，并将所得的调度结果作为下一流组 ILP 模型的约束条件进行输入，这种分组调度的方式可以将原来大规模的求解问题划分为较小规模的 ILP 问题，从而在流的数量较多时，显著的减少最终的总耗时。

然而，这种分组调度的方式在对每组时间触发流进行调度时所获得的结果是一个局部最优的结果，最终的调度结果与原先的结果可能具有一定的偏差，从而导致调度成功率的降低。因此，在进行分组调度时，如何保证一定的调度成功率是首要考虑的问题。

谱聚类（SC, spectral clustering）[11]是一种基于图论的聚类方法，将带权无向图划分为 2 个或 2 个以上的最优子图，使子图内部尽量相似，而子图之间距离尽量远，所得结果是一个均匀划分，即各个划分所包含的节点数目相近。本文将谱聚类的思想引入分组策略的研究中。对于 2 个待调度的流而言，如果它们各自可能传输的路径之间具有较多重叠的链路，则它们同时进行传输的难度也较大。同理，对于 2 个流组而言，如果它们中所有的流可能传输的路径集合之间具有较多重叠的链路，则这 2 个流组同时进行传输的难度也较大。因此，为了使分组调度之后的调度成功率尽可能高，各个流组之间路径集合的重叠链路应尽可能少。

用路径集相似度的概念来表示 2 个路径集合之间的重叠程度，定义 2 个流之间路径集相似度为

$$w_{(f_i,f_j)}=\sum_{\substack{\forall p_m\in P_{f_i}\\ \forall p_n\in P_{f_j}}}\frac{1}{\left|P_{f_i}\right|\left|P_{f_j}\right|}\left|p_m\cap p_n\right| \tag{14}$$

其中，$w(f_i, f_j)$是流 f_i 和流 f_j 的传输路径集合之间链路重叠程度的表征。对于流集 F 的一个划分，即 $G^1, G^2, \cdots, G^{\max-1}, G^{\max}$，其总路径集相似度为

$$w(G^1,G^2,\cdots,G^{\max-1},G^{\max})=\sum_{\substack{\forall f_1\in G^i,\forall f_2\in G^j\\ i\neq j}}w(f_1,f_2) \tag{15}$$

基于流之间的路径集相似度来建立无向图 $\xi=(F,W)$，其中顶点集 F 由所有时间触发流组成，边集 $W=F\times F$ 的权重表示 2 个流 f_i 和流 f_j 之间的路径集相似度 $w(f_i, f_j)$。2 个顶点之间相连表示对应的 2 个流之间的路径集合存在一定的重叠链路，如果这 2 个流之间的路径集合不存在重复链路，即路径

集相似度为 0，则对应的顶点之间不连接。图 4 给出了待调度流量集合 $F=\{f_1, f_2, \cdots, f_8, f_9\}$的一个划分，即 $G^1=\{f_1, f_2, f_8\}$, $G^2=\{f_3, f_4, f_9\}$, $G^3=\{f_5, f_6, f_7\}$。根据式(15)可知，图 4 中虚线的权重之和表示该划分的总路径集相似度。

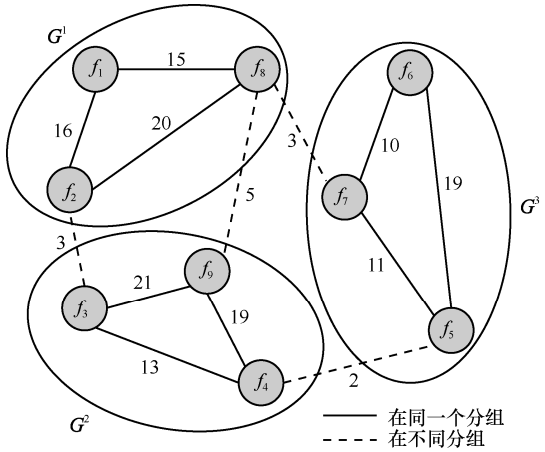

图 4 待调度流量集合划分示例

考虑到模型求解耗时的问题，还应使各个分组的大小尽可能一致，避免出现某个分组过大的情况，否则会导致模型求解的总时间变得不再理想，而谱聚类均分划分的特点可以满足该需求。

因此，求解流的最佳分组问题被转化为求解基于路径集相似度构建的无向图 ξ 的最佳划分问题。在具体实现上，由于 Python 中 sklearn.cluster 的 SpectralClustering 模块集成了谱聚类方法，将由路径集相似度构建的相似度矩阵 $\boldsymbol{M}$（$\boldsymbol{M}_{i,j}$ 表示流 f_i 和流 f_j 之间的路径集相似度）和要分组的数目作为参数输入，即可获得分组结果。该结果是一个总路径集相似度最小且均匀的最佳划分，分组的结果 G^1, G^2, …, $G^{\max-1}$, $G^{\max}$ 将被用于分组调度机制。

3.3 基于拓扑修剪策略和流分组策略的分组调度机制

分组调度机制的整体流程如算法 1 所示。算法 1 的整体输入为所有时间触发流的集合 F、网络的拓扑结构 $G=(V, L)$、分组数 max，输出为路径和时隙的映射 $N^*_{(p,t)}$。最终获得的 $N^*_{(p,t)}$即为最终的调度结果。在初始化步骤中(算法的 1)～3)行)，对 F 按照基于谱聚类的分组策略进行分组，从而得到分组结果 $G^1, G^2, \cdots, G^{\max-1}, G^{\max}$。另外还要对变量 $N^*_{(p,t)}$以及前一组流量的调度结果带来的约束 $M^*_{(l,t)}$进行初始化。在每一组的调度过程中，首先进行拓扑修剪，避免产生不必要的约束条件；然后进行 ILP 模型求解，将得出的调度结果更新到 $N^*_{(p,t)}$中，并将本组调度带来的对后续调度的约束 $M_{(l,t)}$更新到 $M^*_{(l,t)}$中。

算法 1 分组调度机制

输入 F，$G=(V, L)$，max

输出 $N^*_{(p,t)}$

1) 对任意流 $f_i, f_j \in F$，基于式(13)计算路径集相似度 $w(f_i, f_j)$

2) 对 F 按照流分组策略处理，将由路径集相似度 $w(f_i, f_j)$构成的相似度矩阵 $\boldsymbol{M}$ 输入 sklearn.cluster.SpectralClustering，以获得分组结果 $G^m|_{m=1, 2, \cdots, \max-1, \max}$

3) 初始化 $N^*_{(p,t)}$和 $M^*_{(l,t)}$

4) for $i \in$ max do

5) 对本组流量集合调度时的网络拓扑进行修剪

6) 根据给定的 G^m 和 $M^*_{(l,t)}$产生约束条件

7) 对式(12)的 ILP 问题进行求解

8) 更新 $N^*_{(p,t)}$和 $M^*_{(l,t)}$

$N^*_{(p,t)}= N^*_{(p,t)}+N_{(p,t)}$

$M^*_{(l,t)}= M^*_{(l,t)}+ M_{(l,t)}$

9) end for

4 仿真实验和结果分析

4.1 实验环境

所提调度模型是一个 ILP 问题，可直接采用 CPLEX 或 Lingo 等解算器进行求解，本文基于 Python3.7 和 IBM 的 CPLEX（版本 V12.10）进行了仿真实验。通过 Python 的 networkx 包生成网络拓扑，基于 sklearn.cluster 的 SpectralClustering 模块完成对调度流量的分组。实验思路是，通过不同调度规模下算法的仿真表现来验证其有效性。

仿真条件包括网络的拓扑规模和数量规模两方面，前者表现为网络中交换机的数目，后者表现为待调度流量的数目。模型的参数包括是否进行拓扑修剪（分别用 1 和 0 表示）、分组的数量、基本周期的时隙分片数量。

实验的仿真条件及参数如表 1 所示。

4.2 仿真结果和分析

本节将分组调度机制与文献[9-10]调度方法的仿真实验结果进行对比，重点分析拓扑修剪以及流分组 2 种策略在大规模调度场景下的有效性。算法性能的评价指标是求解时间与调度成功率。

表 1　　仿真条件及参数的设置

实验	交换机数目/个	流的数目/个	拓扑修剪	分组数/个	时隙分片数/个
实验 1	50	100～550	0 或 1	10	7
实验 2	40～110	150	0 或 1	10	5
实验 3	50	250	1	1、10～100	5
实验 4	50	250	1	10	5～15

4.2.1　算法随流量规模增长的表现

为了探究不同流量规模下算法性能的表现，利用表 1 中实验 1 的仿真条件和参数进行仿真实验，网络中交换机的数量为 50 个，待调度流量的数量为 100～550 个，实验结果分别如图 5 和图 6 所示。

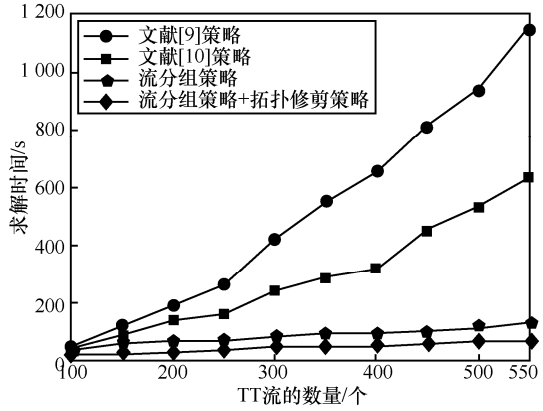

图 5　求解时间随 TT 流数量的变化

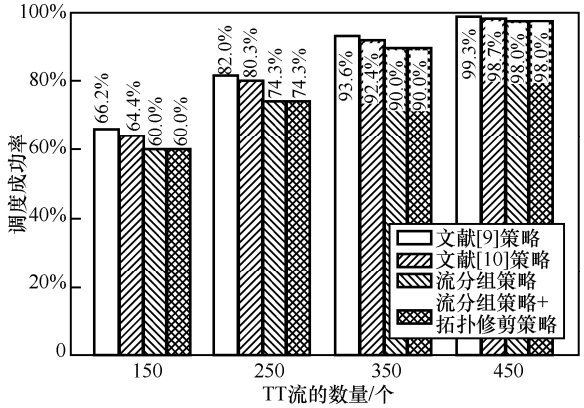

图 6　调度成功率随 TT 流数量的变化

由图 5 可知，当采用流分组策略时，本文算法求解时间相对于文献[9-10]方法有一定下降；并且随着流量规模的增加（流量数目达到 300 个以上时），求解时间降低显著。在此基础上，如果采用拓扑修剪策略，求解时间又有一定的降低。

由图 6 可知，拓扑修剪策略在减少求解时间的同时，可以保证成功率不变，这与 3.1 节的理论分析是一致的。对比文献[9-10]方法和采用流分组策略的结果，当待调度流量的数量为 150 时，流分组策略的调度成功率与文献[9-10]方法相近。而随着待调度流量数量的增加，流分组策略会造成调度成功率出现一定程度降低，但是降低的程度是可以接受的（图 6 中成功率的偏差在 8%之内），同时，显著减少了求解时间。因此，综合来看分组调度机制是有效的。

4.2.2　算法随网络拓扑规模增长的表现

为了探究不同网络拓扑规模下算法性能的表现，利用表 1 中实验 2 的仿真条件和参数进行了仿真实验，待调度流量的数量为 150，网络中交换机的数量为 40～110 个，实验结果分别如图 7 和图 8 所示。

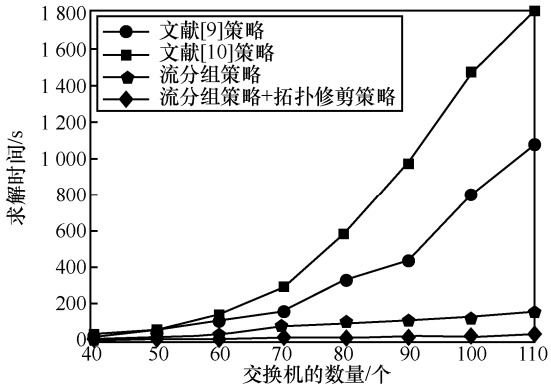

图 7　求解时间随交换机数量的变化

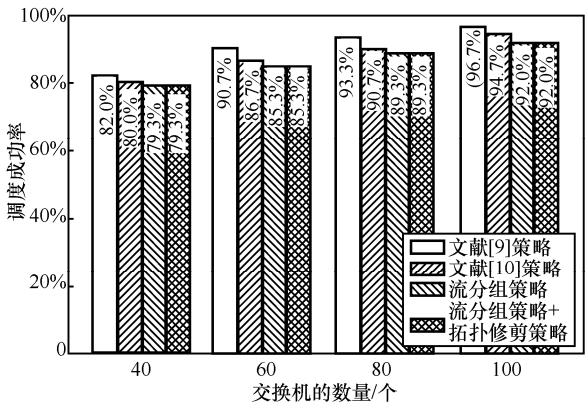

图 8　调度成功率随交换机数量的变化

几种方案的求解时间如图 7 所示。当采用流分组策略时，求解时间相对于文献[9-10]方法有了一定的下降，并且随着网络拓扑规模的增加（交换机数量超过 70 时），求解时间降低愈加显著。在此基础上，如果采用拓扑修剪策略，求解时间又有一定的降低。

调度成功率的结果如图 8 所示。随着交换机数

量的增加，可用的链路传输资源越来越多，因此各种方法的调度成功率逐渐增加。此外，通过与文献[9-10]方法的结果进行对比可以看出，随着网络拓扑规模的增加，流分组策略可以在减少求解时间的同时，保证一定的调度成功率（图 8 中成功率的偏差在 5%以内）。此外，对比采用拓扑修剪策略前后的结果可以看出，拓扑修剪策略在实现降低求解耗时的同时，可以保证调度成功率保持不变。

4.2.3 模型参数对于算法性能的影响

1) 分组数目

通过上述实验分析，可以看出流分组策略针对大规模调度场景具有一定的有效性。为了进一步探究分组的数目对于算法性能的影响，本节利用表 1 中实验 3 的仿真条件和参数进行仿真实验。网络交换机数量为 50，待调度流量的数量为 250，分组的数量分别为一个（表示不使用流分组策略）、10～100 个，实验结果分别如图 9 和图 10 所示。求解时间随着分组数目增加的变化曲线如图 9 所示，可以看出，当分组的数量增加时，求解时间会逐渐降低，但是降低的幅度逐渐变缓。调度成功率如图 10 所示，随着分组数量的增加，调度的成功率逐渐降低。

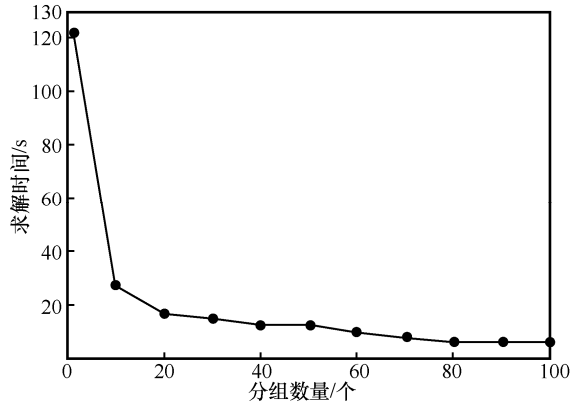

图 9 求解耗时随分组数量的变化

流分组策略可以实现调度求解时间的降低，然而一直增大分组的数量也是不可取的，还应当考虑调度成功率的因素。分组数量为 10 时，调度成功率为 77.2%；不分组（即分组数量为 1）时，调度成功率为 82.8%，2 种情况下的成功率相差 5%左右。随着分组数量的增加，求解时间降低的幅度逐渐变缓，但是调度成功率与不分组时成功率（82.8%）相差逐渐增大。因此，分组数目应综合考虑求解时间和调度成功率两方面因素进行选择。

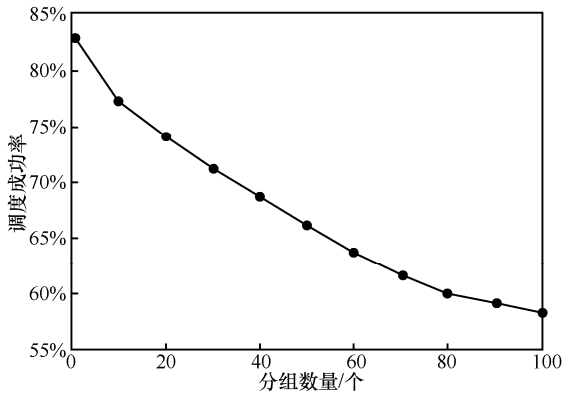

图 10 调度成功率随分组数量的变化

2) 基本周期的时隙分片数目

为了探究时隙分片的数目对于算法性能的影响，本节利用表 1 中实验 4 的仿真条件和参数进行仿真实验，网络交换机数量为 50，待调度流量的数量为 250，流分组的数量为 10，基本周期的时隙分片数目为 5～15，实验结果分别如图 11 和图 12 所示。求解时间随时隙分片数目增加的变化曲线如图 11 所示，可以看出，随着时隙分片数目的增加，模型逐渐复杂导致求解时间逐渐增加。算法的调度成功率随时隙分片数目增加的变化曲线如图 12 所示，可以看出，时隙分片的数目越多，调度成功率越高，当时隙分片的数目超过 11 时，调度成功率达到 100%，此时所有的待调度流量均可以在网络中传输。

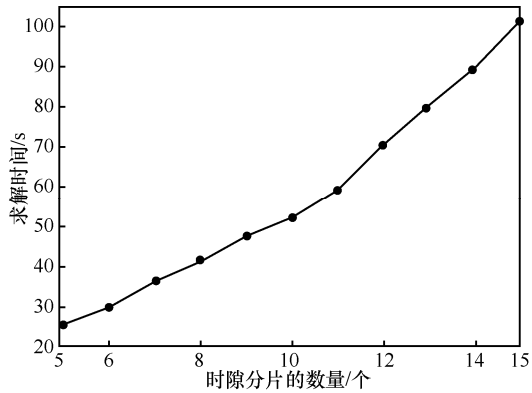

图 11 求解耗时随时隙分片数量的变化

时隙分片的数目代表了网络在时间维度上容纳时间触发流的能力。时隙分片的数目越多，网络中能够同时进行传输的时间触发流也越多，图 12 的仿真结果印证了这一点。然而，随着时隙分片数目的增加，求解时间也会增加，这是因为每增加一个时隙，模型的变量数目（即路径与时隙的映射变量）以及约束条件（即每条链路在每个时隙上的传输约束）都会线性增加，这将导致问题规模增大，

从而造成求解时间的增加。此外，由于基本周期能提供的时隙分片数目是有上限的，因此时隙分片的数目应当在允许的范围内综合考虑求解时间和调度成功率两方面因素进行选择。

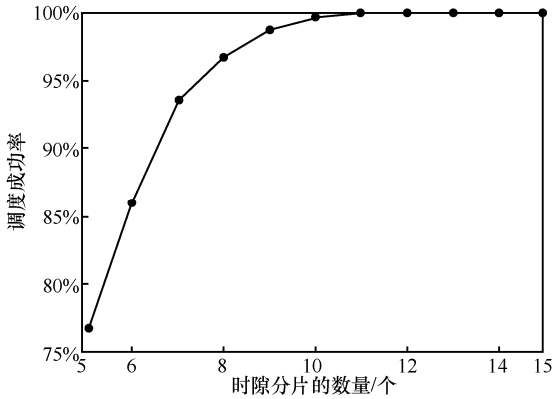

图 12　调度成功率随时隙分片数量的变化

5　结束语

针对大规模时间敏感网络中时间触发流量的调度问题，本文提出的分组调度机制通过消除排队时延保证了时延的确定性。同时，所提拓扑修剪策略解决了由于网络拓扑规模增加而导致的求解时间剧增的问题。所提基于谱聚类的流分组策略解决了由于流量规模增加而导致的求解时间剧增的问题。仿真实验表明，该机制在大规模调度场景下可以在较短时间内求出调度结果并保证一定的调度成功率。

本文所做的工作仍有许多的不足，例如在路由策略中选择了所有最短路径的集合作为可能传输的路径集。下一步研究将综合考虑路由和调度两方面的因素来进行时间触发流量的调度研究。

参考文献：

[1] 丛培壮, 田野, 龚向阳, 等. 时间敏感网络的关键协议及应用场景综述[J]. 电信科学, 2019, 35(10): 31-42.
CONG P Z, TIAN Y, GONG X Y, et al. A survey of key protocol and application scenario of time-sensitive network[J]. Telecommunications Science, 2019, 35(10): 31-42.

[2] CRACIUNAS S S, OLIVER R S, CHMELÍK M, et al. Scheduling real-time communication in IEEE 802.1 QBV time sensitive networks[C]// Proceedings of the 24th International Conference on Real-Time Networks and Systems. New York: ACM Press, 2016: 83-192.

[3] POP P, RAAGAARD M L, CRACIUNAS S S, et al. Design optimisation of cyber-physical distributed systems using IEEE time-sensitive networks[J]. IET Cyber-Physical Systems: Theory & Applications, 2016, 1(1): 86-94.

[4] LI B Q, WANG Y. Hybrid-GA based static schedule generation for time-triggered ethernet[C]// 2016 8th IEEE International Conference on Communication Software and Networks. Piscataway: IEEE Press, 2016: 423-427.

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

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

[7] FALK J, DÜRR F, ROTHERMEL K. Exploring practical limitations of joint routing and scheduling for TSN with ILP[C]//2018 IEEE 24th International Conference on Embedded and Real-Time Computing Systems and Applications. Piscataway: IEEE Press, 2018: 136-146.

[8] MAHFOUZI R, AMINIFAR A, SAMII S, et al. Stability-aware integrated routing and scheduling for control applications in Ethernet networks[C]//2018 Design, Automation & Test in Europe Conference & Exhibition. Piscataway: IEEE Press, 2018: 682-687.

[9] NAYAK N G, DÜRR F, ROTHERMEL F, et al. Time-sensitive software-defined network (TSSDN) for real-time applications[C]// Proceedings of the 24th International Conference on Real-Time Networks and Systems. New York: ACM Press, 2016: 193-202.

[10] 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.

[11] 蔡晓妍, 戴冠中, 杨黎斌. 谱聚类算法综述[J]. 计算机科学, 2008, 35(7): 14-18.
CAI X Y, DAI G Z, YANG L B. Survey on spectral clustering algorithms[J]. Computer Science, 2008, 35(7): 14-18.

[作者简介]

邱雪松（1973− ），男，江西上饶人，博士，北京邮电大学教授、博士生导师，主要研究方向为网络与业务管理、物联网与区块链。

黄徐川（1997− ），男，安徽合肥人，北京邮电大学硕士生，主要研究方向为时间敏感网络。

李文萃（1984− ），女，河南许昌人，博士，国网河南电力公司高级工程师，主要研究方向为电力通信网络传输及安全。

李温静（1984− ），女，山西太谷人，博士，国网信息通信产业集团有限公司高级工程师，主要研究方向为电力系统边缘计算、电力自动化及终端技术。

郭少勇（1985− ），男，河北邢台人，博士，北京邮电大学副教授，主要研究方向为物联网与区块链。