ISSN: 2414-1895 DOI: 10.6919/ICJE.202401_10(1).0051

Research on Multi-channel Delay Symmetry Technology for AC Differential Protection Services

Zhaohe Wang*, Hong Xu, Kaili Dong

State Grid Henan Information & Telecommunication company, China

*519271097@qq.com

Abstract

Electric power data, especially important generation services such as AC differential protection, have strict requirements for low latency, low jitter, etc., when transmitted in the power communication network, which plays a crucial role in maintaining the safe and stable operation of the power grid. This paper proposes a group scheduling mechanism for AC differential protection services aimed at large-scale power transmission networks. By establishing a group scheduling model, it eliminates queuing delay and ensures deterministic latency. Furthermore, it designs topological trimming strategies and flow grouping strategies to reduce model constraints and solving time, improving the scheduling success rate. In addition, this paper also presents a group scheduling strategy based on spectral clustering, achieving a balance between scheduling success rate and solving time through the introduction of the concept of path set similarity. Experimental results show that the flow grouping strategy can reduce solving time while ensuring a certain scheduling success rate.

Keywords

Power Transmission Network; Latency; Group Scheduling.

1. Introduction

Electric power data, especially important generation services such as AC differential protection, have strict requirements for low latency, low jitter, no congestion, no packet loss, and high robustness when transmitted in the power communication network, which plays a crucial role in maintaining the safe and stable operation of the smart grid. The continuous improvement in the level of informatization has led to an explosive growth in the scale of power network topology and power data, placing higher demands on the performance of power communication networks. In addition to the continuous growth in network and data scale, there is also a continuous demand for stricter real-time control of business information, leading to a contradiction in the actual power communication network between the demand for large-scale scheduling and the real-time transmission requirements.

In the case of large-scale network topology and traffic scheduling, congestion in the transmission of electric power data is inevitable. Considering that the existing data forwarding mechanisms for transmission tasks in the power communication network have difficulty meeting scheduling demands in large-scale scheduling scenarios, it is of great significance to address the data forwarding issues for latency-sensitive services in large-scale scheduling scenarios to ensure the stability, safety, and reliability of the smart grid operation.

First, we establish a group scheduling model [1]. In the scheduling of each flow group, the time and space-based time division multiple access technique is employed to eliminate queuing delays and ensure the determinacy of the latency. To address the issue of large-scale network topologies, a topology pruning strategy is designed to simplify the scheduling model by reducing unnecessary

ISSN: 2414-1895 DOI: 10.6919/ICJE.202401_10(1).0051

constraints and decreasing the scheduling solving time. Additionally, a flow grouping strategy based on spectral clustering is designed to handle large traffic scales. The proposed algorithm reduces scheduling solving time through flow scheduling and ensures a high success rate.

2. Data Forwarding Architecture for Large-scale Scheduling Demands

The designed data forwarding mechanism for large-scale scheduling demands is illustrated in Fig. 1. The control center has a global view of the network topology and real-time monitoring of the business transmission demands of terminal nodes. Furthermore, the control center can issue specific data forwarding strategies to the exchange nodes within the network, which then complete the forwarding of latency-sensitive data according to the strategy. The computing center receives the global network topology information and the set of transmission tasks to be scheduled from the control center, and calculates the data forwarding strategy with the optimization of latency range as the objective, and then issues this strategy to the control center.

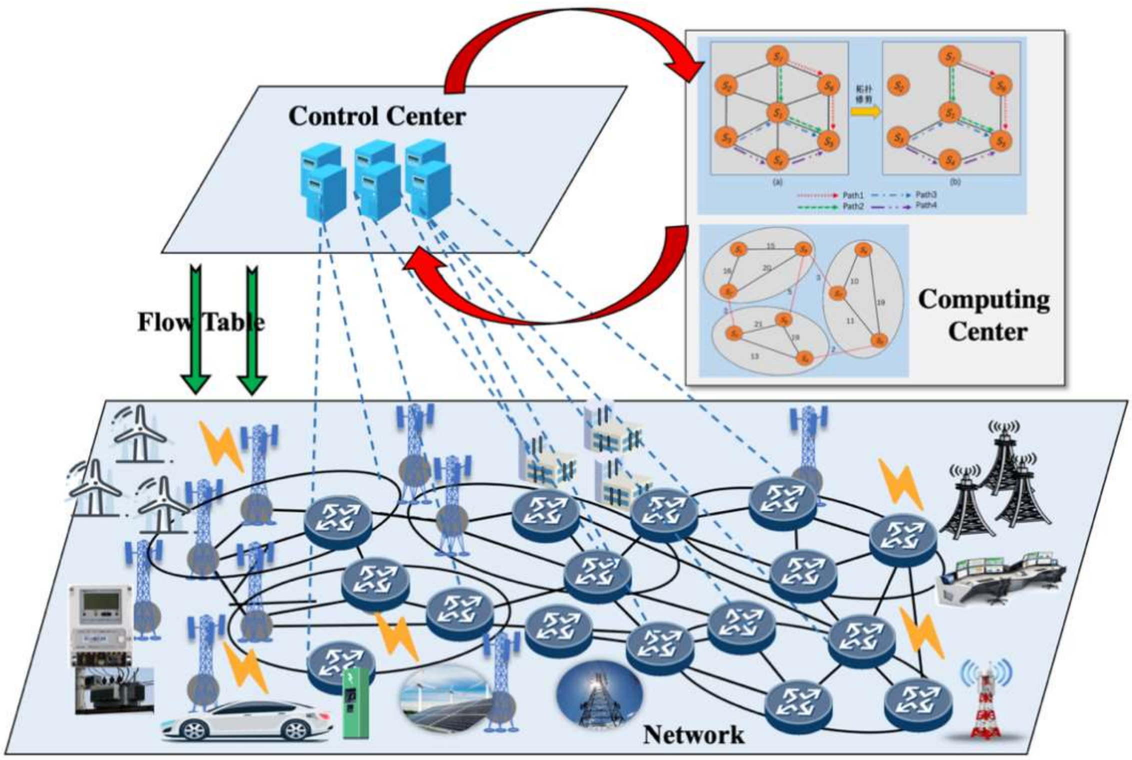

Fig. 1 Forwarding Architecture

Latency-sensitive flows transmit periodically from source hosts to destination hosts along specified paths, where the end-to-end latency comprises propagation delay, transmission delay, and processing delay, and in case of network congestion, it also includes queuing delay. The latency analysis is illustrated in Fig. 1.

In Figure 2, ti represents the sending time of node vi, dtrans, dprop, and dproc respectively represent transmission delay, propagation delay, and processing delay. When queuing occurs, the queuing delay is denoted as dqueue, and D represents the end-to-end latency between a node and the next node.

$$D = d_{\text{trans}} + d_{\text{prop}} + d_{\text{proc}} + d_{\text{queu}} \quad (1)$$

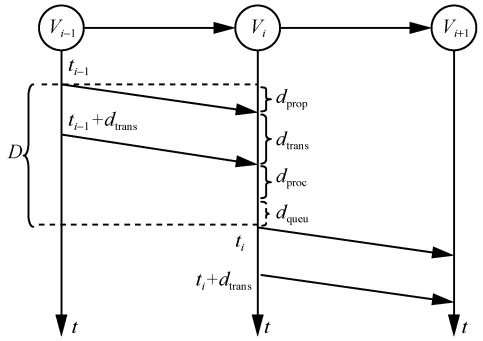

Fig. 2 Time delay analysis

When there is no queuing, D only includes transmission delay, propagation delay, and processing delay. Propagation delay is determined by the channel length and the propagation speed of the signal in the medium, and can be considered deterministic. The internal processing delay of the switch is related to its store-and-forward capability, and can be considered deterministic. The transmission delay is determined by the frame size and transmission rate, and is also deterministic. In the event of network congestion, the queuing time inside the switch is uncertain, resulting in uncertain queuing delay.

Since the uncertainty of end-to-end latency mainly comes from the potential queuing delay, it is necessary to avoid queuing situations when designing scheduling models for latency-sensitive flows, in order to ensure their transmission determinacy.

3. Data Forwarding Model for Large-scale Scheduling Demands

When multiple data packets[2] attempt to be transmitted through the same output port of a switch, queuing occurs. If specific link resources are allocated for their transmission at the time of triggering the flow transmission, queuing can be eliminated, thereby ensuring the determinacy of end-to-end latency. In terms of time, different time-triggered flows can be assigned different slots within a basic period, and in terms of space, non-conflicting link resources can be allocated for different time-triggered flows. By using the time division multiple access (TDMA)to separate different time-triggered flows in time and space, a vacant queue is always available for each time-triggered flow when it arrives at the switch, thus avoiding queuing situations.

As shown in Fig. 3, a basic period is divided into multiple time slots, each starting from 0 and numbered up to tmax, and assigned to specific time-triggered flows for transmission. The duration of each time slot is wide enough to accommodate the maximum transmission unit (MTU) data packets passing through the longest network path.

There is an upper limit on the number of slots that a basic period can provide. For example, for a 1 Gbit/s link, assuming an MTU of 1,500 B and the maximum allowed network path of 8 hops, the total transmission delay of an MTU data packet passing the longest network path is 1,500 B × 8 ÷ 1 Gbit/s × 8 = 0.096 ms. Considering only the transmission delay, a 5 ms basic period can provide a maximum of 5 ms ÷ 0.096 ms ≈ 52 time slot partitions. Considering factors such as transmission distance and switch processing capacity, the actual upper limit on the number of time slot partitions is slightly smaller.

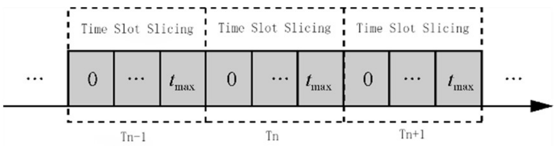

Fig. 3 Time slots slicing of basic cycle

ISSN: 2414-1895 DOI: 10.6919/ICJE.202401_10(1).0051

The network topology is modeled as a directed graph G=(V, L), where V=S∪H is the set of all nodes in the network, with S representing switches and H representing host terminals. L⊆V×V denotes a set of directed full-duplex links connecting pairs of nodes in the network. The ordered pairs (Vi, Vj) and (Vj, Vi) represent two independent transmission links, with the first element indicating the transmitting node and the second element indicating the destination node. The following sets are defined.

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

Where F is the set of time-triggered flows in the network. All time-triggered flows are grouped according to a specific scheduling strategy: G1, G2, …, Gmax−1, Gmax, where max is the number of groups, and each group of time-triggered flows is scheduled in each iteration of the scheduling process. The set of all possible transmission paths for all flow in a set of time-triggered flow sets Gm is:

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

Where fj = (s, d) represents a flow in the set Gm, where s and d respectively represent the source and destination hosts of fj, Pfj represents the set of possible paths that the flow fj may take. In the model, Pfj is set to be the set of shortest paths for the flow fj, so PGm includes all the shortest paths from the source to the destination for each time-triggered flow fj∈Gm. The set of time slots T and the set of all links L are represented as (4) and (5).

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

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

The following are defined:

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

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

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

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

FP is a mapping between time-triggered flows and paths, where if flow f passes through path p from the source to the destination, then X(f, p)=1; otherwise Y(p,l)=0. PL is a mapping between paths and links, where if path p contains link l, then Y(p,l)=1; otherwise Y(p,l)=0. LT is a mapping between links and time slots, where if link l is already occupied in the previous iteration of the scheduling process on time slot t, then M(l,t)=1; otherwise M(l,t)=0. PT is a mapping between paths and time

ISSN: 2414-1895 DOI: 10.6919/ICJE.202401_10(1).0051

slots, where if the time-triggered flow transmits along path p in time slot t, then N(p,t)=1; otherwise N(p,t)=0.

The following constraints are defined:

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

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

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

Constraint (10) states that each path can be assigned to at most one time slot. Each delay-sensitive flow can be assigned to at most one time slot, as a flow can ultimately choose only one of its possible shortest paths for transmission, therefore, for each delay-sensitive flow, at most one possible shortest path can be assigned a time slot, which leads to constraint (11). If two paths with the same link should be assigned to the same time slot, it may cause conflicts, therefore, on a certain link, at most one delay-sensitive flow can be transmitted in a time slot, resulting in constraint (12).

Given the set of delay-sensitive traffic and the known network topology, to improve the success rate of scheduling, the scheduling mechanism should maximize the number of delay-sensitive flows that can be accommodated in the network while ensuring delay determinism. Additionally, since a delay-sensitive flow corresponds to multiple possible shortest paths for transmission but ultimately only selects one for transmission, the problem of assigning delay-sensitive flows to time slots can be transformed into assigning paths to time slots, and the optimization objective is to maximize the number of paths assigned to time slots. The scheduling model for a flow group Gm can be formally described as:

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

4. Optimization Mechanism for Large-Scale Scheduling Demands

1) Topology Pruning Strategy

The establishment of model constraints is closely related to the network's topology. As indicated by Equation (13), each link in the network topology adds a constraint to the model for each time slot. However, as the scale of the network topology increases, the number of constraints in the model will rapidly increase, leading to a significant increase in problem complexity and ultimately unacceptable solution times.

However, for a group of time-triggered flows to be scheduled, each flow will ultimately only select one path from the set of shortest paths for transmission. The group of time-triggered flows may occupy the link resources contained in all paths of the shortest path set. Therefore, the remaining links

ISSN: 2414-1895 DOI: 10.6919/ICJE.202401_10(1).0051

in the entire topology will not have flows passing through them for this group of scheduling, and simultaneous transmission conflicts cannot occur. By pruning the network topology for the scheduling of this group of time-triggered flows, non-transmitting links are removed, thus reducing the number of model constraints.

Fig. 4 provides an example of the topological pruning. Fig. 4(a) shows the original network topology with 2 traffic flows to be scheduled, flow1=(S7, S5) and flow2=(S3, S5). Flow1 has two shortest paths from the source node to the target node, path1 and path2, while flow2 has two shortest paths from the source node to the target node, path3 and path4. The links contained in these four paths represent the link resources that may be occupied by this group's scheduling, and each link adds a constraint to the model. After topological pruning, as shown in Fig. 4(b), the reduction in links reduces the number of constraints in the model, thereby reducing the scale of the problem. Furthermore, since the topological pruning strategy does not affect the final scheduling result, the scheduling success rate remains unchanged.

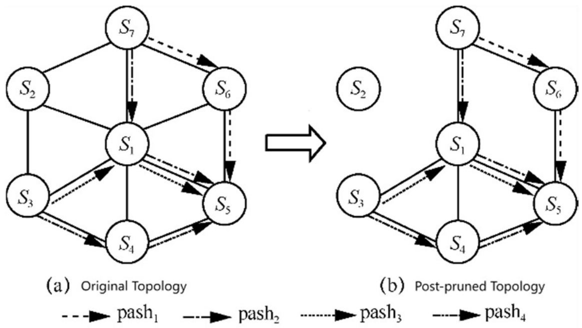

Fig. 4 An example of the topological pruning

2) Flows grouping strategy

The scheduling model described earlier is an ILP problem, which is NP-Hard. As the number of flows to be scheduled increases, the solving time of the model will sharply increase, becoming unacceptable. Grouping all time-triggered flows and separately scheduling each group, that is, the ILP model of each flow group is solved, and the obtained scheduling result is input as the constraint condition of the ILP model of the next flow group. This grouping scheduling method can divide the original large-scale solution problem into smaller-scale ILP problems, thus significantly reducing the final total time consumption when the number of flows is large.

However, this method of grouped scheduling[3] results in a locally optimal solution for each group of time-triggered flows, which may lead to deviations from the original results and decrease the overall scheduling success rate. Therefore, ensuring a certain scheduling success rate is a primary consideration when conducting grouped scheduling.

Spectral clustering (SC) is a graph-based clustering method that partitions a weighted undirected graph into two or more optimal subgraphs, aiming for high similarity within subgraphs and large dissimilarity between subgraphs, resulting in a uniform partition with similar numbers of nodes in each partition. The idea of spectral clustering is introduced into the research of grouping strategy. For two flows to be scheduled, if there is a considerable overlap in the potential transmission paths for each, their simultaneous transmission would be challenging. Likewise, for two flow groups, if there is a significant overlap in the potential transmission path sets for all flows, simultaneous transmission for these two flow groups would also be challenging. To maximize the scheduling success rate after grouped scheduling, it is crucial to minimize the overlapping paths between the path sets of each flow group.

The concept of path set similarity is used to measure the degree of overlap between two sets of paths, and the path set similarity between two flows is defined as:

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

where w(fi, fj) represents the degree of overlap between the transmission path sets of flows fi and fj. For a partition of flow sets F, namely G1, G2, …, Gmax−1, Gmax, the total path set similarity is given by:

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

An undirected graph ξ=(F,W) is built based on the similarity of path sets between flows, where the vertex set F consists of all delay-sensitive flow groups, and the edge set W=F×F with weights representing the similarity of path sets between two flows fi and fj, denoted as w(fi, fj). Two vertices are connected if the corresponding flow path sets have some overlapping paths, and are not connected if the path sets have no overlapping paths (i.e. w(fi, fj) = 0). The following figure gives a division of the flow set F={ f1, f2…f8, f9} to be scheduled, namely G1={f1, f2, f8}, G2={f3, f4, f9}, G3={f5, f6, f7}. Solid lines connect flows within the same group, while dashed lines connect flows from different groups. According to equation (15), the sum of weights of dashed lines in the figure represents the total path set similarity of the partition.

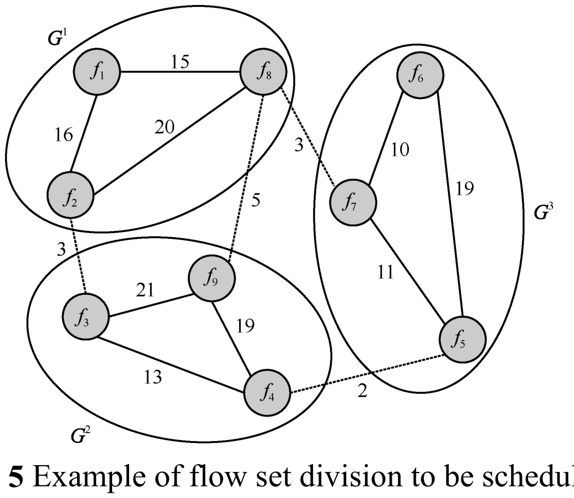

Fig. 5 Example of flow set division to be scheduled

In order to address the issue of model solving time, it is also important to ensure that the sizes of each group are as balanced as possible to avoid the situation where one group becomes excessively large, which would lead to a less than ideal total solving time. The characteristic of even partitioning in spectral clustering can meet this requirement.

Therefore, the problem of finding the best grouping of flows is transformed into the problem of finding the optimal partition of the undirected graph ξ based on the similarity of path sets. In practical implementation, the SpectralClustering module of sklearn.cluster in Python integrates spectral clustering methods, allowing for the input of the similarity matrix M (where Mi,j represents the similarity of path sets between flow fi and flow fj) and the number of groups to obtain the grouping results. This result is the optimal partition with the smallest total similarity of path sets and uniformity, and the resulting groups G1, G2, …, Gmax−1, Gmax will be used for grouping scheduling mechanisms.

3) Data forwarding time slot scheduling algorithm

The overall process of the grouping scheduling mechanism [4] is as follows. The overall input for Algorithm 1 is the set of all time-triggered flows F, the network's topology G=(V, L), and the maximum number of groups max, and the output is the mapping of paths and time slots N*(p,t). The obtained N*(p,t) is the final scheduling result. In the initialization step (lines 1) to 3) of the algorithm), F is grouped according to the spectral clustering-based grouping strategy, resulting in the groups G1, G2,…, Gmax−1, Gmax. In addition, the variables N*(p,t)and the constraints M*(l,t) brought by the scheduling results of the previous flow group should also be initialized. During the scheduling process of each group, topological pruning is performed first to avoid unnecessary constraints. Subsequently, the ILP model is solved, and the obtained scheduling result is updated to N*(p,t), while the constraints brought by the scheduling of this group to the subsequent scheduling are updated to M*(l,t) from M(l,t).

Algorithm 1:

Input: *F*, *max*

Output: $N^*_{(p,t)}$

1 Construct undirected graph ξ according to the similarity of path sets between every two streams of flow group F

2 For any flow $f_i, f_j \in F$, calculate the path set similarity $w(f_i, f_j)$ based on equation (13)

3 *F* is processed according to the flow grouping strategy, and the grouping results are obtained: $G^m|_{m=1,2\ldots,max-1,max}$.

4 Initialize $N^*_{(p,t)}$ and $M^*_{(l,t)}$

5 **for** i∈*max* **do:**

6 Trim the network topology scheduled by this group

7 According to the given G^m and $M^*_{(l,t)}$ to generate constraints

8 Solve the ILP problem of formula (12)

9 Update $N^*_{(p,t)}$ and $M^*_{(l,t)}$: $N^*_{(p,t)}= N^*_{(p,t)}+ N_{(p,t)}$, $M^*_{(l,t)}= M^*_{(l,t)}+ M_{(l,t)}$

10 **end**

5. Experiment and Evaluation

The scheduling model proposed is an ILP problem and can be directly solved using solvers such as CPLEX or Lingo. Simulation experiments were conducted using Python 3.7 and IBM's CPLEX (version V12.10). The network topology was generated using the networkx package in Python, and the scheduling of traffic was grouped using the SpectralClustering module from sklearn.cluster. The experimental approach involved verifying the effectiveness of the algorithm by simulating its performance under different scheduling scales.

Simulation conditions included aspects of the network topology scale and the quantity scale. The former was represented by the number of switches in the network, while the latter was represented by the number of traffic flows to be scheduled. Model parameters included whether topological pruning was performed (represented by 1 for yes and 0 for no), the number of groups, and the number of time slot fragments in the basic period.

The experimental simulation conditions and parameters are shown in Table 1.

Table 1. The simulation parameters

ISSN: 2414-1895 DOI: 10.6919/ICJE.202401_10(1).0051

Simulation parameters	List 1	List 2	List 3	List 4
Number of switches	50	30-75	50	50
Number of flows	50-150	150	250	250
Topological pruning	0or1	0or1	1	1
Number of groups	10	10	1, 10-100	10
Time slots number	5	5	5	5-15

1) Performance of the algorithm as traffic scale increases

To investigate the performance of the algorithm under different traffic scales, simulation experiments were conducted using the conditions and parameters from Experiment 1 in Table 1. The number of switches in the network was set to 50, and the number of traffic flows to be scheduled varied from 100 to 550. The experimental results are shown in Figs. 6 and 7.

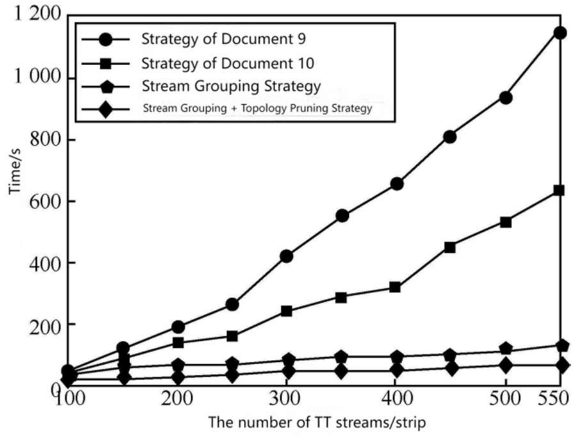

Fig. 6 The change of solution time with the number of TT streams

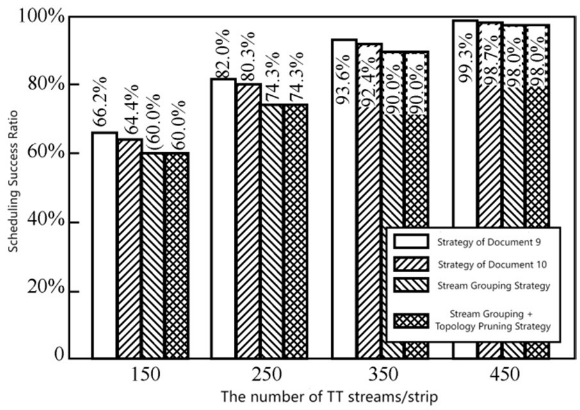

Fig. 7 The change of scheduling success rate with TT flow number

Fig. 6 shows that when using the flow grouping strategy, the algorithm's solving time is relatively lower compared to the methods in literature [9-10], and the solving time significantly decreases as the traffic scale increases (reaching 300 flows or more). Additionally, when using topological pruning, there is a further reduction in the solving time.

Fig. 7 indicates that the topological pruning strategy reduces solving time while maintaining a consistent success rate, consistent with previous theoretical analysis. Comparing the results of the flow grouping strategy with the methods in literature [9-10], when the number of traffic flows to be scheduled is 150, the success rate of the flow grouping strategy is similar to that of the literature methods. With an increase in the number of traffic flows to be scheduled, the flow grouping strategy leads to a certain degree of reduction in the success rate, but the reduction is acceptable (deviation in success rate is within 8%), while significantly reducing the solving time. Therefore, the group scheduling mechanism is effective overall.

2) Performance of the algorithm as network topology scale increases

To investigate the performance of the algorithm under different network topology scales, simulation experiments were conducted using the conditions and parameters from Experiment 2 in Table 1. The number of traffic flows to be scheduled was 150, and the range of switches in the network was from 40 to 110. The results of the simulation experiment are shown in Figs. 8 and 9.

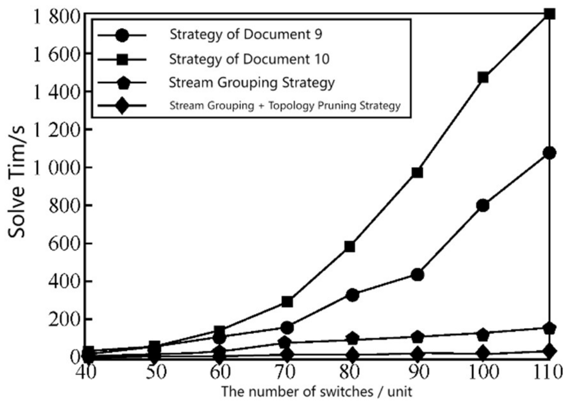

Fig. 8 The solution time varies with the number of switches.

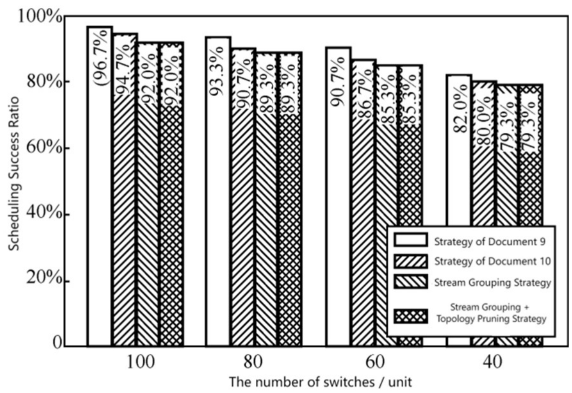

Fig. 9 The change of scheduling success rate with the number of switches

The solution times for several approaches are shown in Fig. 8. When using the flow grouping strategy, the solution time decreases compared to the methods in [9-10]. Additionally, as the network topology scale increases (with more than 70 switches), the decrease in solution time becomes more significant. Furthermore, if the topology pruning strategy is employed, there is a further decrease in solution time.

The scheduling success rates are depicted in Fig. 9. With an increase in the number of switches, there are more available link transmission resources, leading to a gradual increase in the scheduling success rates for various methods. Moreover, in comparison with the methods in [9-10], the flow grouping strategy can reduce solution time while ensuring a certain scheduling success rate as the network topology scale increases (the deviation in success rate in Fig. 9 is within 5%). Additionally, comparing the results before and after employing the topology pruning strategy indicates that the strategy can reduce solution time while maintaining the scheduling success rate.

3) Impact of model parameters on algorithm performance

ISSN: 2414-1895 DOI: 10.6919/ICJE.202401_10(1).0051

Number of groups.

From the above experimental analysis, it can be observed that the flows grouping strategy is effective for large-scale scheduling scenarios. To further explore the impact of the number of groups on algorithm performance, the conditions and parameters from Experiment 3 in Table 1 were used for simulation experiments. With 50 network switches and 250 traffic flows to be scheduled, the number of groups varied from 1 (indicating no use of the flow grouping strategy) to 10 to 100. The results of the experiments are shown in Figs. 10 and 11. The change in solution time as the number of groups increases, as depicted in Fig. 10, shows that as the number of groups increases, the solution time gradually decreases, but the rate of decrease gradually diminishes. The scheduling success rate, as shown in Fig. 11, gradually decreases as the number of groups increases.

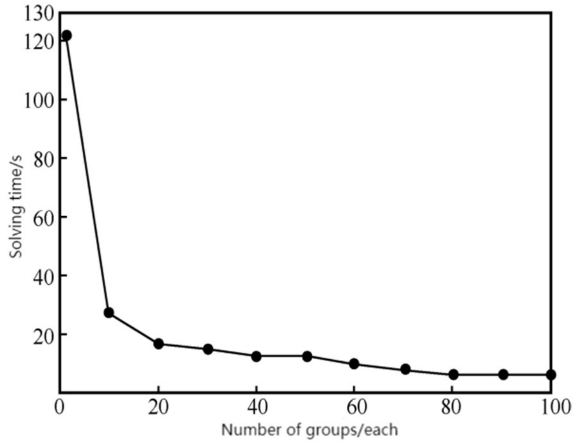

Fig. 10 The change curve of time-consuming solution with the increase of the number of groups.

While the flows grouping strategy can reduce scheduling solution time, increasing the number of groups indiscriminately should be avoided, and the scheduling success rate should also be considered. For example, with 10 groups, the success rate is 77.2%, while with no grouping (i.e., 1 group), the success rate is 82.8%, showing a difference of around 5%. As the number of groups increases, the decrease in solution time gradually diminishes. However, the difference in scheduling success rate compared to when there is no grouping (82.8%) gradually increases. Therefore, the number of groups should be chosen considering both solution time and scheduling success rate.

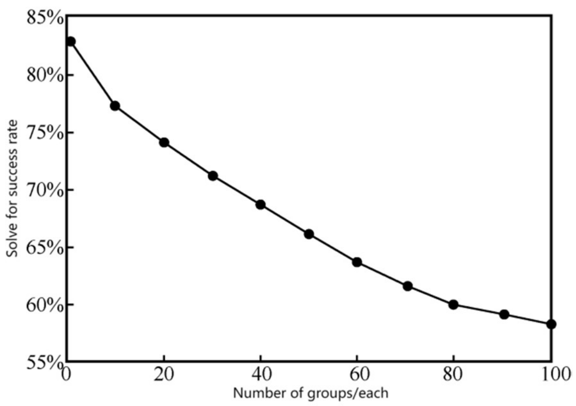

Fig. 11 The change of scheduling success rate with the number of packets

Number of time slots fragments in basic cycle.

In order to investigate the impact of the number of time slot fragments [5] on the algorithm's performance, this section conducts simulation experiments using the simulation conditions and parameters from Experiment 4 in Table 1. The number of network switches is 50, the number of traffic flow to be scheduled is 250, the number of flow groups is 10, and the range of the number of time slot fragments in the basic cycle is 5 to 15. The experimental results are shown in Figs. 12 and 13. The change curve of solution time with the increase of the number of time slot fragments is shown in Fig. 12. It can be seen that as the number of time slot fragments increases, the model becomes gradually more complex, leading to a gradual increase in solution time. The change curve of the scheduling success rate of the algorithm with the increase of the number of time slot fragments is shown in Fig. 13. It can be seen that the more time slot fragments there are, the higher the scheduling success rate. When the number of time slot fragments reaches 11 and above, the scheduling success rate reaches 100%, at which point all the traffic to be scheduled can be transmitted in the network.

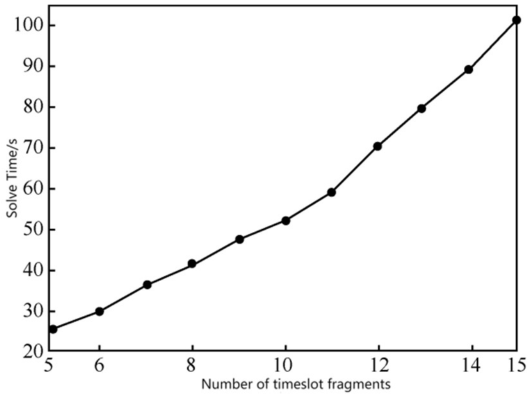

Fig. 12 The change of time-consuming solution with the number of time slots

The number of time slot fragments represents the network's ability to accommodate time-triggered flows in the time dimension. The more time slot fragments there are, the more time-triggered flows can be transmitted in the network simultaneously. The simulation results in Fig. 13 confirm this. However, as the number of time slot fragments increases, the solution time will also increase because with each additional time slot, the number of variables in the model (i.e., the mapping variables between paths and time slots) and the constraint conditions (i.e., the transmission constraints of each link in each time slot) will increase linearly, leading to an increase in problem scale and thus an increase in solution time. Additionally, since there is a limit to the number of time slot fragments that the basic cycle can provide, the selection of the number of time slot fragments should consider both solution time and scheduling success rate within the allowable range.

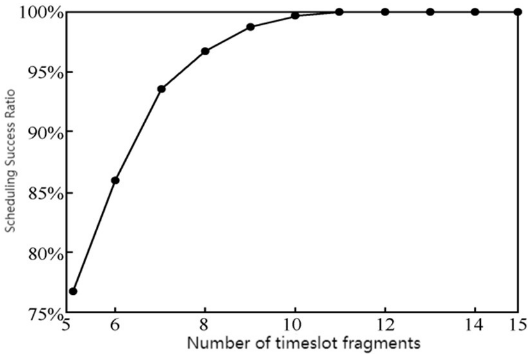

Fig. 13 The change of the scheduling success rate with the number of time slot fragments.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202401_10(1).0051

References

[1] Xu Siya, Zhang Guoyi, Guo Shaoyong, Qiu Xuesong, Meng Luoming. A Packet Scheduling Mechanism on Cognitive Radio Networks for Smart Grid [J]. Journal of Beijing University of Posts and Telecommunications, 2015(06).

[2] Zhang Maosen and Qiu Zhiliang are elegant. Combined Circuit /Packet Switching Fabric and Dispatching Scheme [J]. *Journal of Beijing University of Posts and Telecommunications*,2014(02).

[3] Liu Sai, Scheduling Game Based on Packet Switching Technology and Slack Delivery Time Window[M]. 2016(12).

[4] Zhang Hongyuan. Research on Time-Determined Network Packet Forwarding Queue Scheduling Mechanism[M]. 2021(06).

[5] Qiu Xuesong, Huang Xuchuan, Li Wencui, Li Wenjing, Guo Shaoyong. Group-scheduling mechanism for large-scale time-sensitive network [J]. *Journal on Communications*,2020(11).