Placing Controllers over Complex Wide Area SDNs Based on Clique Identification

Ling Xia Liao[1,2], Zhi Li[1], Han-Chieh Chao[2]

[1] School of Electronic Information and Automation, Guilin University of Aerospace Technology, China
[2] Department of Electric Engineering, National Dong Hwa University, Taiwan
Liaolx@guat.edu.cn, cclizhi@guat.edu.cn, hcc@gms.ndhu.edu.tw

Abstract

Wide Area Networks (WANs) form the network core that covers wide geographical areas. WANs often have complex topologies, and it is challenging to incorporate multiple controllers in the control plane to reduce the network delay in Wide Area Software Defined Networks (WASDNs). We propose a distributed controller placement problem (DCPP) for various control plane structures to address this challenge. While existing exhaustive and greedy algorithms cannot efficiently solve the DCPP over many large-scaled WASDNs, we propose a network simplification strategy based on a novel global network coefficient, polyindex, to identify all the nonoverlapped cliques in networks and characterize the topology features of such complex networks. With such strategy, the good number, organization, and placements of controllers for the DCPP over large-scaled WASDNs can be determined. Extensive evaluations demonstrate the effectiveness of the polyindex in capturing the features of sparse WANs. While applying the proposed strategy over large-scaled WANs with small and medium polyindexes can quickly find the placements for the DCPP while meeting the given delay requirement, carefully adjusting the delay requirement and threshold is the key to generate high quality frontiers while keeping the time cost low over the WANs with large scales and polyindexes.

Keywords: Complex networks, Controller placement, Network topology simplification, Clique identification

1 Introduction

Wide area networks (WANs) often have complex topologies that present the dynamic evolution of network connectivity but cannot simply describe. WANs connect terminals with various levels of intelligence and support applications with diverse requirements in a wide geographical area, and they normally should provide flexible Quality of Services (QoS) that static management strategies commonly employed by WANs cannot meet [1]. Wide area Software Defined Networks (WASDNs), which are WANs with decoupled control and data planes, can overcome such limitations by dynamically setting up flow-forwarding rules based on frequently updated global network views at controllers that monitor changes in network topology via reachability tests. These tests are often facilitated by in-band control planes that connect switches to controllers using the same infrastructure that interconnect switches [2]. This article targets WASDNs with in-band control planes. We consider to place controllers at the locations where switches are located to share the same network infrastructure for both control and data traffics. As it is difficult to optimize the location of a single controller in a large-scaled WASDN such that delays in controller-switch interactions are short enough to support delay-sensitive applications such as real-time image processing, a distributed control plane structure is commonly used. With carefully placed controllers, such a control plane can provide efficient flow and global network view management while avoiding a single point of failure.

Finding the best locations to place the controllers over the entire network has been formulated as the controller placement problem (CPP) to optimize different objectives in various scenarios [3-6]. These works found that minimizing the delay between controllers and switches (con2swi delay) and the delay among controllers (con2con delay) are significant to enhance the network performance, scalability, and availability. However, these works all address specific control plane structures that specify how controllers are organized. In contrast, this article targets joint optimization of both controller organization and placement, which has yet to be explored. Since the way controllers are organized in the control plane constrains the level of controller cooperation the control plane can support, which in turn determines the network performance and availability, this article proposes a CPP with two conflicting objectives simultaneously minimizing the con2swi delay and con2con delay to find the best controller organization and placement

*Corresponding Author: Ling Xia Liao; E-mail: liaolx@guat.edu.cn
DOI: 10.53106/160792642021092205010

over large-scaled WASDNs.

It is challenging to solve the DCPP in an acceptable time over large-scaled WASDNs. Exhaustive algorithms are not feasible, because their complexity grows significantly as the size of the network increases. Greedy heuristics algorithms [5-6] are not feasible either, because the DCPP has two conflicting objectives, while a greedy strategy typically cannot optimize one objective without degrading the other conflicted objective. Although multiple-objective problems can be converted to single-objective problems by weighing the importance of objectives or converting objectives to constraints [5], these approaches cannot generate Pareto optimal solutions for the CPPs. Pareto optimal solutions are solutions at which none of the objective functions can be improved in value without degrading the other objectives values. A set of Pareto optimal solutions form a Pareto frontier. Generating Pareto frontiers is crucial for CPPs to investigate the trade-offs of objectives and find the needed best solution given the system operator's or user's preference.

Evolution algorithms such as Non-dominated Sorting Genetic Algorithm (NSGA-II) [7], Particle Swarm Optimization (PSO) [8], and simulated anneal (SA) [9] can solve CPPs with conflicted objectives without involving greedy assignment and exhaustive searches. However, they may generate frontiers with poor qualities, because CPPs over large-scaled WASDNs often have a large solution space, while the spaces that evolution algorithms can search are very limited [9]. Also, exhaustive and evolution algorithms cannot quickly determine the number and organization of controllers, making the DCPP even harder to be solved than CPPs over large-scaled WANs.

To address these issues, this article aims to effectively reduce the solution space of the DCPP over large-scaled networks, so that the DCPP can be solved in a low time cost and high accuracy. By a thorough analysis of currently proposed topologies of WANs, we find the difference of topologies of WANs links to the number and order of the cliques existing in the networks, although WANs are often sparse and complex. Since the existing global network properties such as density and clustering coefficients [10] cannot capture such a difference among sparse topologies, this article proposes a novel global network coefficient, polyindex, and further develops a network simplification strategy based on it to significantly reduce the solution space of the DCPP over large-scaled WANs. To the best of our knowledge, this work is the first effort that applies nonoverlapped cliques identification in solving CPPs. More particularly, this article makes the following 4 contributions:

(1) The DCPP that co-optimizes the organization and placement of controllers with two conflicting objectives that minimize the worst case con2swi and con2con delays is formulated. The worst case delays instead of the average delays are considered for real-time imaging processing applications to meet their rigid delay requirements.

(2) Polyindex is proposed to capture the number of nonoverlapped cliques with orders of 3+ over a network. Nonoverlapped cliques are the cliques without other cliques inside. The order of a clique is the number of nodes that form the clique.

(3) A novel simplification strategy based on the proposed polyindex coefficient is developed to form the set of candidate locations for controller placements, so that the DCPP can be solved by the existing exhaustive algorithms over the set to achieve a high accuracy and efficiency. Layered distributed control plane with the number of controllers roughly determined by the longest path in a network for most of the WANs are suggested.

(4) Extensive evaluations over all the topologies of WANs collected by the Internet Topology Zoo [11] are conducted to demonstrate the ability of polyindex in characterizing the density of sparse and complex WANs, and to demonstrate the accuracy of the proposed simplification strategy in solving the DCPPs.

The rest of this article is organized as follows. Section 2 presents the related work and the network analysis conducted. Section 3 formulates the DCPP. While the polyindex is proposed in Section 4, the novel simplification mechanism is presented in Section 5. An extensive evaluation is provided in Section 6 followed by the conclusions in Section 7.

2 Related Works and Network Analysis

This section summarizes the related works on distributed control planes, controller placement problems, solving controller placement problems, and identification of clique over WANs.

2.1 Distributed Control Planes

Distributed control planes are logically centralized but physically distributed [12]. They maintain a global network view to enable network-wide management. They can be layered or flat distributed. Layered structure often have a root controller in the upper layer and multiple local controllers in the lower layer. Since each switch connects to a local controller, local controllers maintain the view of a network partition and process the (local) flows within the partition. The root controller maintains the global network view by collecting local network views from local controllers, and processes network-wide flows forwarded by local controllers. In a flat distributed control plane, all controllers are equal peers. Although each controller plays the same role and processes both local and network-wide flows, it may have access to other controllers to construct/retrieve the global network view. In layered distributed control plane, the root controller communicates to the local controllers, but no interaction among the local controllers is enable. However, flat distributed control plane enable interactions among any two controllers.

2.2 Controller Placement Problems

Previously proposed CPPs are formulated for both data center networks and WASDNs. CPPs for data center networks typically model the real flows in the network and consider the flow processing delay, because data center networks place their nodes closely and have much larger flow processing delays than propagation delays [13]. In contrast, CPPs over WASDNs need to consider the relatively large propagation delays between geographically widespread nodes. As listed in Table 1, most CPPs over WASDNs only consider con2swi delay or both con2swi and con2con delays given a specific control plane organization, because some objectives such as load imbalance may be meaningless [3], as varous computing and storage resources can be given to controllers according to their loads, or can be converted to a constraint based on the preference [5]. Therefore, the DCPP is formulated to simultaneously optimize both controller organization and placement without prior preference information, its solutions should simultaneously minimize the con2swi and con2con delays to produce a Pareto frontier that enables optimization given any preference information.

Table 1. The comparison of controller placement problems

CPP	Control plane	Objective	Algorithms
[3]	specific	1	Greedy
[6]	specific	1	Greedy
[5]	specific	2	Greedy
[4]	specific	2	Exhaustive
[8]	specific	2+	EA
[14]	specific	2+	EA
[15]	specific	2+	EA
DCPP	general	2	Exhaustive

2.3 Solving Controller Placement Problems

Since exhaustive algorithms [4, 14] and greedy heuristic algorithms [5-6] are not suitable for the DCPP over large-scaled WASDNs, evolution algorithms (EAs) can be applied to solve many CPPs proposed previously. However, it is a great challenge for them to provide high quality approximation solutions [16-18]. Our previous work incorporates a PSO-based mutation for NSGA-II to produce approximate solutions for the CPP with a short convergence time, but it may generate poor quality solutions over large-scaled WANs. Therefore, finding a way to simplify the complex topology of WANs and effectively reducing the solution space of CPPs over large-scaled WANs is significant in making exhaustive algorithms time feasible or improving the quality of frontiers generated by evolution algorithms.

2.4 Identification of Cliques in WANs

WANs are often complex and have wide various scales. We collect 227 networks archived by the Internet Topology Zoo and compute their basic information. We find out these WANs have 3 to 714 nodes, the length of the paths (the shortest path between two nodes) ranges from 3.4 to 2,155,564 km, and degrees of the nodes range from 1 to 53. In average, a WAN has 33.5 nodes with degrees of 2.5. The average length of paths in a WAN is 88,773 km.

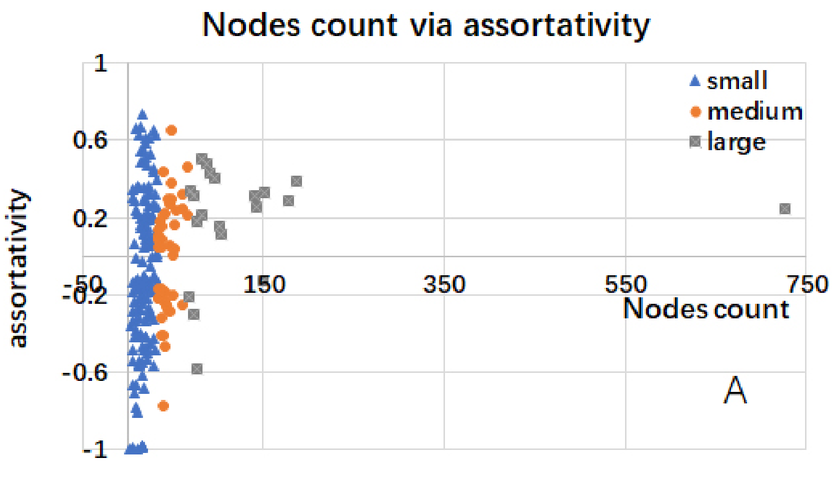
We then group all the WANs into 3 categories, as listed in Table 2, the ones with the number of nodes 1) less than the average (small scale), 2) twice of the average (medium scale), and 3) the others (large scale). We find out 160 of 227 WANs have the number of nodes smaller than the average (33.5 nodes), 47 of 227 WANs have 34 to 67 nodes, and only 20 of 227 WANs have the number of nodes greater than 67. We then further compute their typical global network properties including assortativity, clustering, density, and clique communities using built-in functions provided by Brain Connectivity Toolbox [19] of Matlab [20].

Table 2. WANs collected by the Internet Toplogy Zoo

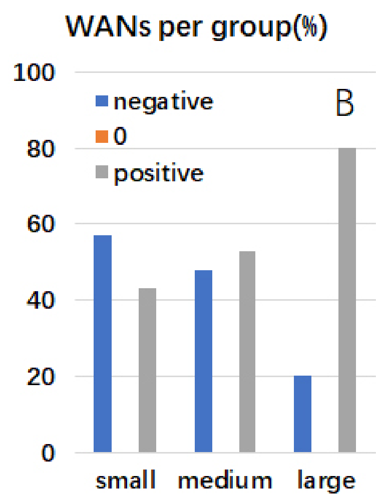
Network scales	Node counts	Network counts
Small	Less than 33.5	160
Medium	Less than 67	47
Large	Greater than 67	20

2.4.1 Assortativity

Assortativity [21] is a property that characterizes how nodes tend to be connected with each other. It is defined as the Pearson correlation between the degree of connected nodes. Commonly, a network that has a positive assortativity implies that the nodes with similar degrees tend to be connected to each other, and a negative assortativity implies that the nodes with a low degree tend to connect to the nodes with a high degree [10]. As shown in Figure 1(a), all the WANs collected have negative or positive assortativities, and no assortativity of 0 has been found. Figure 1(b) indicates that smaller-scaled WANs tend to have negative assortativities while the assortativities of larger-scaled WANs tend to be positive, implying that smaller-scaled WANs are likely to form star-based topologies while the topologies of larger-scaled WANs are likely circle- or line-based. Therefore, the common method that removes the nodes with degree of 1 from networks can significantly simplify a smaller-scaled WAN due to the star-based topology, but it may not fit larger-scaled WANs, in which very few nodes have degrees of 1 due to the circle- or line-based topology. This is why we need to develop extra strategy for topology simplification.



(a) Nodes count

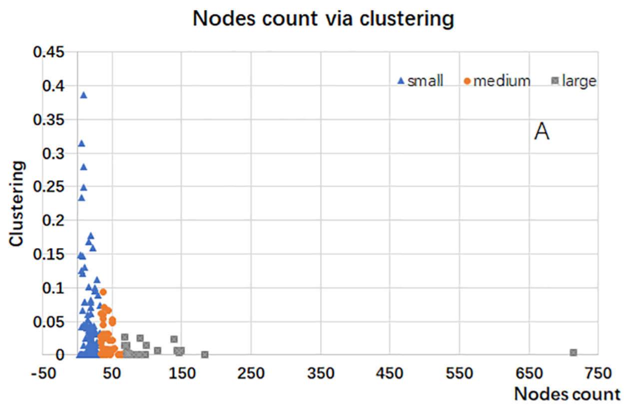


(b) WANs per group (%)

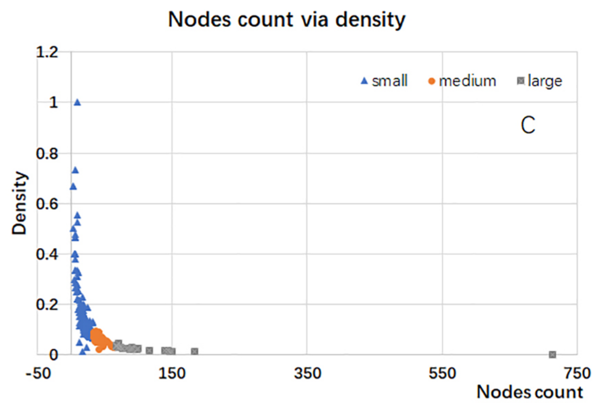
Figure 1. Assortativity of all collected topology

2.4.2 Clustering, Density, and Clique Communities

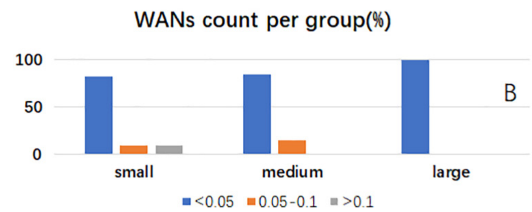
The clustering and density coefficients calculate the number of triangles and edges in a network, respectively [10, 19]. While Figure 2(a) and Figure 2(b) show the clustering coefficients for the collected WANs, the density coefficients are illustrated in Figure 2(c) and Figure 2(d). It is apparent that the clustering and density coefficients decrease as WANs scale up, implying larger-scaled WANs are sparser. Figure 2(c) demonstrates a trend that a WAN with more nodes has a lower density, although its clustering coefficient does not have to be larger than the ones with less nodes, especially for those WANs with medium or large scales, as shown in Figure 2(a). Since the clustering and density coefficients are both very small for larger-scaled WANs, it is difficult to use such properties to differentiate the features of such WANs.



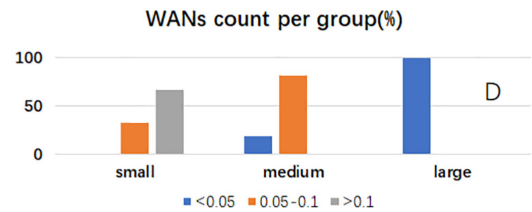
(a) Nodes count via clustering



(b) Clustering WANs



(c) Nodes count via density



(d) Density WANs

Figure 2. Clustering and density coefficient for all collected topology

Clique community coefficient [19] uncovers the overlapped cliques given the maximum size of cliques. It identifies the communities in a network but not the relationship among communities. Therefore, it is not suitable to identify the structure or hierarchy of a WAN. Clique community coefficient also has a shortage in determining the maximal size/order of the cliques. The clustering coefficients with a certain size of *n* cannot apply to identify the hierarchy of WANs, because there is no easy way to determine the value of *n*, and the clustering coefficients for the cliques with a particular order of n may still very small since larger-scaled WANs are very sparse. Therefore, it is significant to propose a novel global network coefficient to differentiate the feature of sparse WANs.

2.4.3 Summary

WANs are sparse and may consist of a surprising level of structural order. Detecting cliques can identify the hidden structure of hierarchies and communities in a network. The density, clustering [22-23], and clique community coefficients [24] proposed in current research can characterize a certain type of structures in networks. In general, density and clustering detect the nonoverlapped 2-order and 3-order cliques, respectively, while clique community coefficient catches all overlapped cliques with the order smaller than *n*. As listed in Table 3, Caldarelli et al. propose a 4-order clustering coefficient that catches the nonoverlapped 4-order communities [26], and Bianconi et al. further extend the cliques' order to *n* in growing scale-free networks for discovering the nonoverlapped n-order communities [25]. Shergin et al. identify the number of circles as a function of network size, and Chen et al. further find that a majority of the circle orders obey Weibull distribution, of which the scale parameter *A* and the shape parameter *B* have approximate power-law relationships with the Perron-Frobenius eigenvector of the samples [27]. Since community coefficient targets community discovery, and the existing density coefficients only identify the cliques, with a particular order, for instance, the density and clustering coefficients identify 2- and 3-order cliques, the 4-order and n-order clustering coefficients only identify the cliques with the order of 4 and *n*, none of them can directly differentiate the structure of WANs, which is typically quite sparse. Therefore, a novel network coefficient is needed to characterize the topology feature of WANs, and our proposed polyindex is the first effort for this purpose.

Table 3. Network properties for cliques discovery

Property	Description
Density	edges over all connections
Clustering [23]	triangles over a/all nodes
cliques [26]	polyshape with order of 4
cliques [25]	cliques with order of n
clique community [24]	cliques with order of n-
Polyindex	all nonoverlapped cliques

3 Fomulating DCPP

Given a WAN that controllers can only be placed at the locations where the switches are located, let *I* be the set of controllers in the control plane, and *J* be the set of switches in the data plane, we compute the worst case con2swi delay $D_{ctos}(I, J)$ and con2con delay $D_{ctoc}(I)$ to abstract the performance and scalability of the whole network. We consider the worst case delay scenario instead of the average delay scenario to ensure every controller and switch can meet the delay requirement for real-time image processing applications in IoT networks. We consider the following 3 constraints:

- two controllers cannot be placed at the same location (*C1*);
- each switch is connected to only one controller (*C2*);
- each switch generates one unit of load to its attached controller (*C3*).

For each controller placement, consider two different controllers $i, k \in I$, and a switch $j \in J$. We use binary variables a_{ij} to describe the attachment of switch *j* to controller *i*, and b_{ik} to describe the cooperation of controllers *i* and *k*, such that $a_{ij} = 1$ or $b_{ik} = 1$ if such an attachment or cooperation is enabled, or equal 0 otherwise. Since the DCPP targets in-band control planes, we use p_{ij} and p_{ik} to represent the propagation delays between controller *i* and switch *j*, and between controllers *i* and *k*, respectively. The largest con2swi delay and con2con delay can be respectively computed as equations (1) and (2), and the contraints can be formulated as (3), (4), and (5), respectively.

$$D_{ctos}(I, J) = Max_{i \in I,\, j \in J}\ p_{ij} a_{ij} \quad \textbf{(1)}$$

$$D_{ctoc}(I) = Max_{i \in I,\, k \in I,\, I \neq j}\ p_{ik} b_{ik} \quad \textbf{(2)}$$

$$C1: p_{ik} \neq 0, \forall i \in I, k \in I, i \neq k \quad \textbf{(3)}$$

$$C2: (a_{ij} = 1 \wedge a_{kj} = 1) \neq 1, \forall j \in J, i, k \in I, i \neq k \quad \textbf{(4)}$$

$$C3: c_i = \Sigma_{j \in J} a_{ij} 0, \forall i \in I \quad \textbf{(5)}$$

For flat distributed control planes, (1) and (2) respectively compute the worst case delays between switches and their controllers, and between two controllers; for layered distributed control planes, (1) and (2) respectively compute the longest delays between switches and their local controllers, and between local controllers and the root controller. $b_{ik} = 0$ for every *i* and *k* belong to *I* abstracts a distributed control plane with no inter-controller cooperation, over which only (1) applicable, and (2) is meaningless. Consequently, (1) and (2) in general are applicable to any distributed control planes, and the DCPP is generalized to $Min_{I,J}(D_{ctos}(I, J), D_{ctoc}(I)$ such that $C1$, $C2$, and $C3$ are satisfied for each *I* and *J* over WASDNs.

4 Polyindex

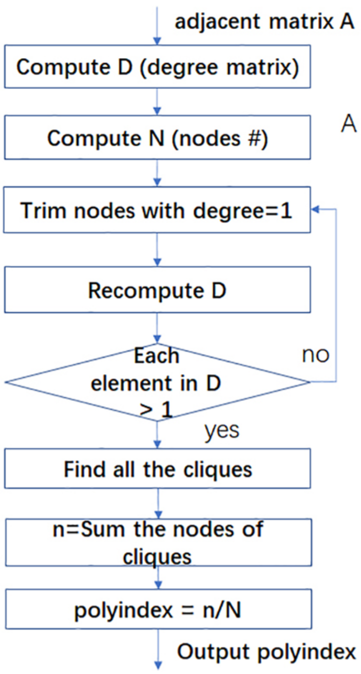
The analysis in Section 2.4 shows the majority of larger-scaled WANs have a positive assortativity with very small clustering coefficients for any specific order. This suggests that WANs have circle- or line- based topologies. We therefore develop a novel network coefficient called polyindex that counts the number of nodes attached to all nonoverlapped cliques in a network divided by the total number of nodes of the network, to capture the topology feature of such larger-scaled WANs. Particularly, WANs are abstracted to undirected graphs with the set of nodes *M*, let *C* be the set of nonoverlapped cliques, c_i be the number of nodes attached to the clique $i \in C$, the polyindex is formulated as follows:

$$polyindex = \frac{1}{|M|} \Sigma_{i \in C} c_i \quad \textbf{(6)}$$

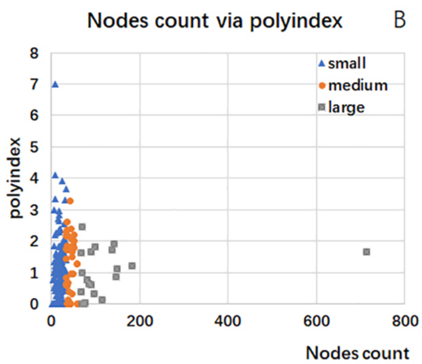
Polyindex accumulates the total number of nodes linking to all nonoverlapped cliques in a network. The value of the polyindexes of WANs can be wide various. A WAN with polyindex of 0 suggests that WAN has a star-based or line-based topology, since no clique exists. A WAN with polyindex smaller than 1 suggests that network has nodes not linking to any circles. Although a network with polyindex of 1 does not mean it should have cliques exactly linking to all the nodes in the network, since some nodes may be shared by cliques, a larger polyindex value does imply a denser topology, especially when the value of ployindex is greater than 1.

To compute the value of polyindex, we develop an algorithm based on the class of polyshape in Matlab. As shown in Figure 3(a), the algorithm takes the adjacent matrix *A* of a WAN as the input and starts to calculate the degree of each node. It then keeps trimming the nodes with degrees of 1 until all the nodes in the trimmed network have degrees greater than 1. By finding all the nonoverlapped cliques in the network after trimming and summing the number of nodes linking to those cliques divided by the total number of nodes in the original network, the algorithm outputs the value of polyindex.

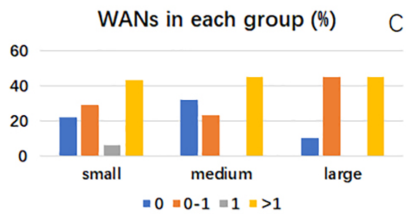
We further compute the polyindex of all the WANs collected by the Internet Topology Zoo. As shown in Figure 3(b), the polyindexs of the WANs in the groups of small, medium, and large scales range from 0 to 7, 0 to 4, and 0 to 3, respectively. The polyindexs of WANs in each group spread all over the ranges, no increase or decline trend has been suggested. As shown in Figure 3(c), a certain portion of the WANs in each group has polyindexes equaling 0, implying that a certain portion of WANs in each group has star- or line-based topologies, although the portion in the group of larger scales is smaller. Although almost every WAN collected has very small dense and clustering coefficients, nearly half of the WANs in each group has polyindexs greater than 1, suggesting that forming cliques with high orders (higher than 3) is one of the major features of WANs. Since WANs often cover a wide geographic area, cliques with higher order often suggest a coverage of larger geographic areas. Carefully dealing with the nodes located at the boundary of high-order nonoverlapped cliques will be the key for network simplification.



(a) The algorithm computing polyindex



(b) Nodes count via polyindex



(c) WANs in each group (%)

Figure 3. The algorithm that computes the polyindex for WANs and the value of polyindex computed

5 Network Simplification for the DCPP

This section introduces the method that simplifies the topologies of WANs to determine the number and the organization of controllers for DCPP.

5.1 Candidate Location Set for Controllers

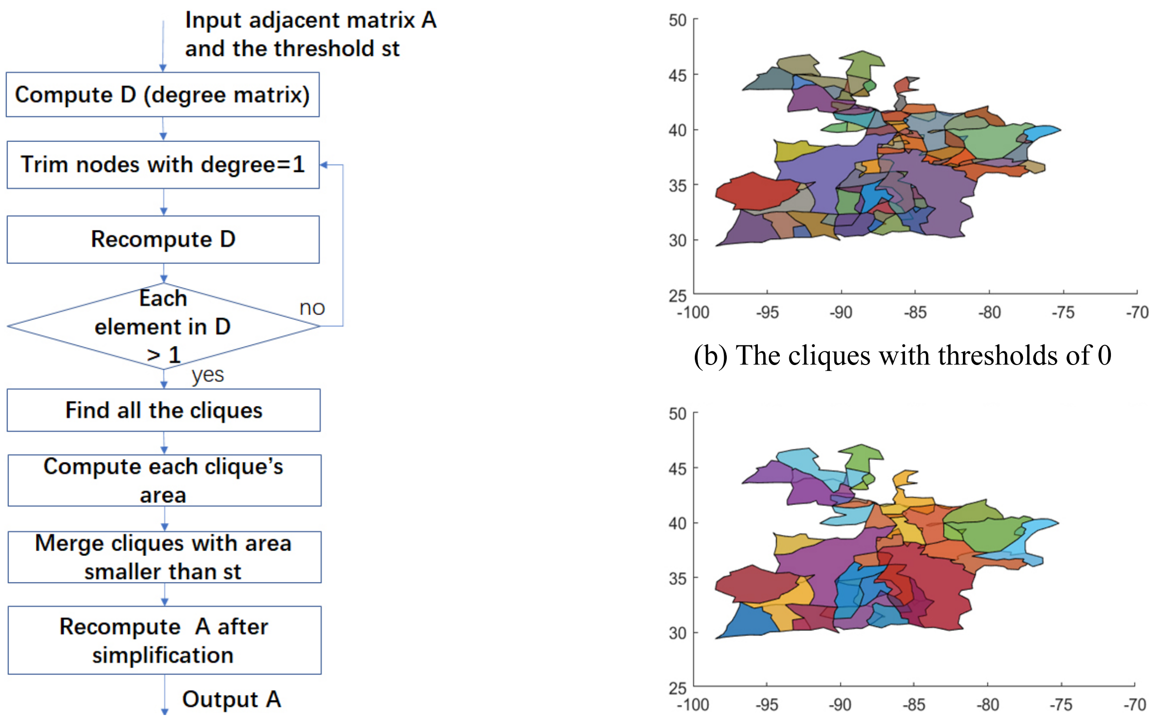
Simplifying WANs such that the solution space of the DCPP over large-scaled WANs can be effectively reduced is crucial to generate high quality frontiers while reducing the computation complexity, no matter exhaustive or evolution algorithms are applied. The common method that only trims the nodes with degrees of 1 over WANs is not efficient, while simply trimming all the nodes with degrees of 2 may ruin the hierarchy of topology, because a majority of nodes linking to cliques and they have degrees of 2.

As shown in Figure 3, although WANs have a big difference in topologies, larger-scaled WANs often have positive assortativity and polyindex greater than 1, suggesting high-order cliques widely exist and they may cover the entire networks. Accordingly, carefully dealing with cliques without changing the structure of networks is therefore the key for network simplification.

Since the DCPP is to minimize the worst case con2swi and con2con delays, we propose to generates a candidate location set I_{can} for the placement of controllers by jointly considering the longest path, the nodes with long link distance, and merging the small nonoverlapped cliques in networks.

Particularly, given a WAN, let I_{can1} be the set consisting of all the nodes along its longest path, and I_{can2} be the set consisting of all the nodes with link distance greater than a given L that is typically related to a delay requirement that the DCPP should meet. Let I_{can3} be the set consisting of all the nodes with degrees greater than 2 and shared by at least two cliques. Considering some large-scaled WANs may have many nonoverlapped cliques, leading to a huge number of nodes in I_{can3}, we can merge those cliques that are neighbors and have areas smaller than a given threshold to reduce the size of I_{can3}. The candidate location set I_{can} is generated by unioning the sets I_{can1}, I_{can2}, and I_{can3} for the controller placements of the DCPP.

Merging nonoverlapped cliques may only works on the WAN with large polyindex, since the WANs with small polyindex have no clique or only several separated cliques in the network. For those WANs with large polyindex, the value of the threshold determines the size of set T_{can3}, which directly affects the quality of frontier generated by the exhaustive algorithms that solve the DCPP. Choosing the value of the threshold should consider the coverage of cliques. Let *aveS* be the average area of cliques of a WAN, Figure 4(a) illustrates the algorithm merging cliques. Figure 4(b) and Figure 4(c) present the cliques with thresholds of *0* (no clique has been merged) and *4aveS*, respectively. Each color presents a clique.



(a) Simplification algorithm

(b) The cliques with thresholds of 0

(c) The cliques with thresholds of *4aveS*

Figure 4. Simplification algorithm and the topology after simplification with variou thresholds:

5.2 Number of Controllers

Determining the number of controllers in the control plane for the DCPP is very challenging. The easiest way is to do an exhaustive search starting from 1 controller up, but it is too time consuming. Current literature also proposed to use clustering approaches such as hierarchical clustering or Self-Organized Map (SOM) to quickly determine the number of clusters in WANs [28]. However, as the proposed DCPP is to find the controller placement for real-time image processing applications that often have a requirement in network delays, we can roughly determine the number of controllers according to such a delay requirement.

Since the DCPP considers the worst case delays, we propose to use the length of the longest path in a WAN ($L_{longest}$) to determine the least number of controllers I_{min} that the network have to involve, given the delay requirement as T_{stoc}. It should be noticed that I_{min} may not exactly the number of controllers for the DCPP. In many cases, for instance when the WANs with several very long paths spreading the whole network, I_{min} works as the starting point to find the final result to reduce the time cost. Let $L_{longest}$ and L be the length of the longest path of a WAN and the distance a controller can manage, respectively, then $L = 3\times10^8 T_{stoc}$, and the least number of controllers in the control plane should be $I_{min} = \frac{L_{longest}}{2L}$.

5.3 Organization of Controllers

Although the organization of controllers is typically highly related to the topology of WANs, we have conducted extensive evaluations and found that a layered distributed control plane typically maintained shorter con2swi and con2con delays than a flat distributed one. It is more suitable to support real-time image processing application over large-scaled WANs for IoTs. Layered organization is also easy to enforce edge computing and learning-based applications while maintaining the existing administration features in IoTs. We will use the layered organization in the DCPP.

6 Evaluation

We compute the $L_{longest}$ for each WAN collected by the Internet Topology Zoo, and find that such WANs has longest pathes ranging from 4.4×10^6 to 2155.6×10^6 meters. Among 227 collected WANs, 117 of them have the length of longest pathes greater than 30×10^6 meters, 147 of them have the length of longest pathes greater than 60×10^6 meters. Since the network latency of WANs are normally the level of ms, we let T_{stoc} be 100ms (it can be considered as the con2swi delay that the DCPP should roughly meet), then L be 30×10^6 meters ($L = 3\times10^8\times100\times10^{-3} = 3\times10^6$). Since 1 controller should be placed every *2L* along the longest path to roughly meet the requirement of 100ms, then 147 of such WANs has to have at least 2 controllers to meet the delay requirement.

We categorize WANs collected by the Internet Topology Zoo into 3 types: 1) WANs with small ([0, 0.5)), 2) medium ([0.5, 1)), and 3) big ([1, -)) polyindexes, as listed in Table 4. In the rest of this section, we firstly discuss the network simplification for each type of WANs, and then solve the DCPP over the selected WANs. All the algorithms used for evaluations are programmed in Matlab and run on a laptop with i5-8250U CPU (1.6GHz) and 8G RAM.

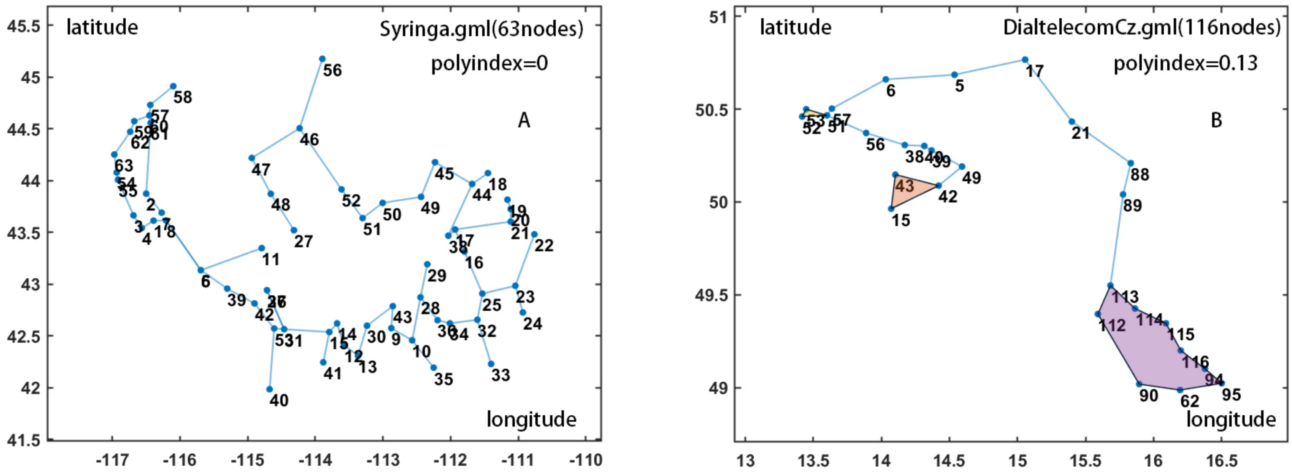
Table 4. WANs in the Internet Topology Zoo (MLP-max longest path, LC-local controllers)

Cat.	Count	Sca.	Top.	MLP	LC
[0, 0.5)	44	S	Line+	14L	7+
	8	M		6L	3+
	5	L		6L	3+
[0. 5,1)	27	S	Several cliques	62L	31+
	18	M		18.9L	10+
	7	L		26.3L	14+
[1, -)	85	S	Many cliques	55.6L	28+
	21	M		44.3L	23+
	8	L		8.7L	5+

6.1 Number and Organization of Controllers

57 of 227 WANs have polyindexes in [0, 0.5), suggesting that these WANs have a line-based topology with no or very limited number of nonoverlapped cliques. Line-based topologies may have a long path together wih multiple short paths. Pure line-based topologies remain nothing after trimming. Therefore, for such type of WANs, their $|I_{can3}|$ is 0 or a small interger if polyindex is 0 or otherwise. Line-based topologies may have positive or negative assortativities. While a line-based topology with a positive assortativity suggests that nodes with similar degrees in the network are connected with each other, a line-based topology with a negative assortativity implies that although many nodes with low degrees connect to some nodes with high degrees, the nodes with high degrees form a line topology. Figure 5(a) shows the topology of Syringa. The length of its longest path is 84.12×10^6. Since this path is greater than *2L* but smaller than *3L*, 2+ local controllers have to involve to meet the requirement of 100ms, 1 more controller can be added as the root controller to form a layered structure. The number of controllers for the DCPP over Syinga is 3+ and they should be layered distributed. Figure 5(b) shows the topology of DialtelecomCz, which consists of 116

nodes. Its polyindex is 0.13, implying a line-based topology with very limited areas covered by cliques. Since the length of its longest shortest path is $38.11\times10^{6}m$ that is smaller than *2L*, 1 controller can manage the whole network while meeting the delay requirement of 100ms. No layered organization is needed.



(a) Syringa (b) DialtelecomCz

Figure 5. The topology of Syringa and DialtelecomCz

52 of such 227 WANs have polyindexes ranging from 0.5 to 1. These WANs are likely to construct several big cliques. Figure 6 shows the topology of Vtlwavenet2011, which consists of 91 nodes. Vtlwavenet2011 has two big nonoverlapped cliques and its longest shortest path is $135.57\times10^{6}m$, which is greater than *4L* but smaller than *6L*. It implies that 3+ local controllers should be placed, and 1 more controller can be added as the root controller to form a layered organization.

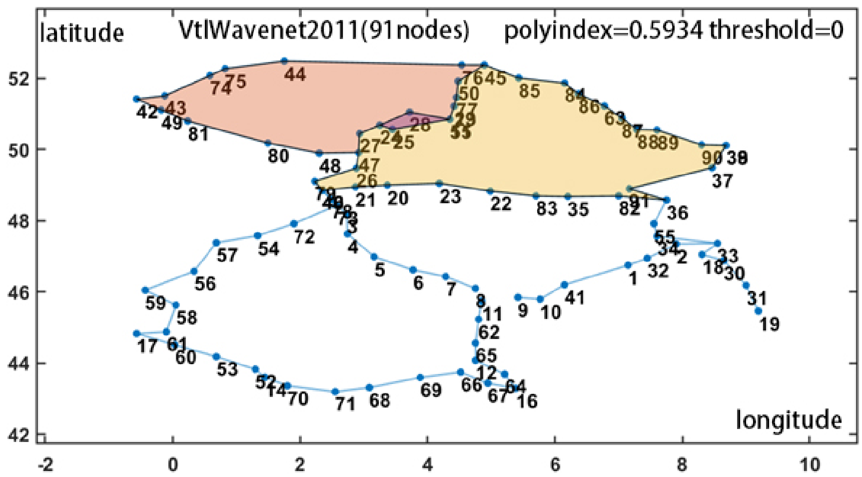


Figure 6. The topology of Vtlwavenet2011

114 of 227 WANs have polyindexes greater than 1, suggesting those WANs construct many high-order cliques, the larger the polyindex is, the denser the WAN is. For such type of WANs, the proposed network simplification strategy should be applied. For instance, network Kdl consists of 714 nodes. Its polyindex is 1.14. It is the largest network among the WANs collected by the Internet Topology Zoo. The length of its longest path is $191.43\times10^{6}m$, which is greater than *6L*, implying 4+ local controllers should be placed, and 1 more controller can be added to as the root to form a layered control plane. Since network Kdl is dense based on its polyindex, we calculate the average area (*aveS*) of all the nonoverlapped cliques, and use 0 and *4aveS* as the threshold to merge the nonoverlapped cliques. The topologies after merging are shown in Figure 4(b) and Figure 4(c), respectively, and the number of nodes in I_{can3} is 186 and 42, respectively. This suggests that adjusting the value of the threshold can change the number of cliques in the network Kdl, and hence the size of set I_{can}. Exhaustive algorithms can be applied to solve the DCPP if the

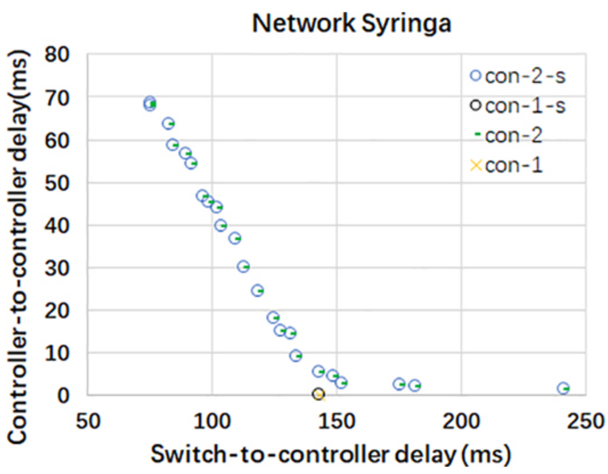
number of nodes in I_{can} can be reduced to 50-.

6.2 Solving the DCPP

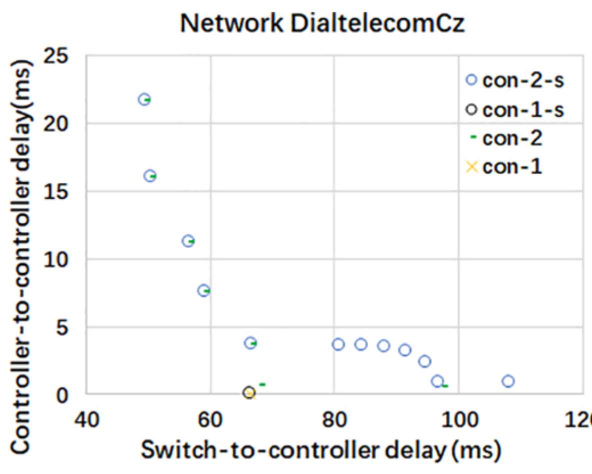
As listed in Table 4, exhaustive algorithms can be directly applied to quickly solve the DCPP when the number of controllers is small and the scale of WANs is not large. For the WANs with more nodes or more controllers, exhaustive algorithms cannot solve the DCPP in a low time cost unless the network simplification strategy is enforced. We evaluate all the WANs collected by the Internet Topology of Zoo to show how the proposed strategy can help to solve the DCPP over large-scaled WANs.

6.2.1 Over WANs with Small Polyindex

Large-scaled WANs that have small polyindexes often have line-based topologies. For such type of WANs to generate the set I_{can} for controller placements, sets I_{can1} and I_{can2} are mainly considered, because I_{can3} is very small. For instance, I_{can3} for both networks Syringa and DialtelecomCz is 0. Since the length of the longest paths in networks Syringa and DialtelecomCz is *2.8L* and *1.3L*, respectively, 2+ and 1+ local controllers should be placed to roughly meet the delay requirement of 100ms, respectively. As shown in Figure 7(a) and Figure 7(b), while 2 local controller can make the con2swi delay of the DCPP over network Syringa smaller than 100ms, 1 local controller is good enough for the network DialtelecomCz. Having more controllers in the control plane can always reduce the con2swi delay. We also apply the exhaustive algorithm to solve the DCPP under $I_{can} = I$. It is apparent that the frontier generated in both situations are roughly the same, showing the proposed simplification strategy does not reduce the quality of frontiers for this type of WANs.



(a) Syringa



(b) DialtelecomCz

Figure 7. Solving networks, frontiers with labels con-2-s and con-1-s are over I_{can}, frontiers with labels con-2 and con-1 are over I

6.2.2 Over WANs with Medium Polyindex

This type of WANs often has several big overlapped cliques if their polyindexes are slightly greater than 0.5. If the WANs have polyindexes close to 1, they may also consist of a number of small cliques besides those big cliques. Regarding the WANs with large scales, it is practical and reasonable to deploy 14+ local controllers over 68+ switches. We choose network Vtlwavenet2011, which consists of 91 nodes and has polyindex of 0.5934. It is time consuming to directly applying exhaustive algorithms due to its network scale. We apply the proposed network simplification strategy to form the set I_{can}. We let threshold=0 and compute I_{can3}, which only consists of 4 nodes. We further compute I_{can2}, which is ϕ (no node has link distance greater than L), and I_{can1}, which consists of 43 nodes. Since the length of its longest path is *4.5L*, and no nodes in I_{can2}, 3+ local controllers may need to roughly meet the delay requirement of 100ms. We let I_{can} be the union of I_{can2} and I_{can3}, and solve the DCPP with controllers placed over I_{can}.

As shown in Figure 8, having 2 local controllers in the control plane is able to reduce the con2swi delay smaller than 100ms. Keeping increasing the number of local controllers does not really change the frontiers a lot. If we let $I_{can} = I$, the exhaustive algorithms can find higher quality frontiers, in which placements have slightly shorter con2swi but much shorter con2con delays. Since we aim to find placements with con2swi delays that meet the requirement of 100ms, the proposed simplification strategy does find the placements that meet the requirement while reducing the time cost.

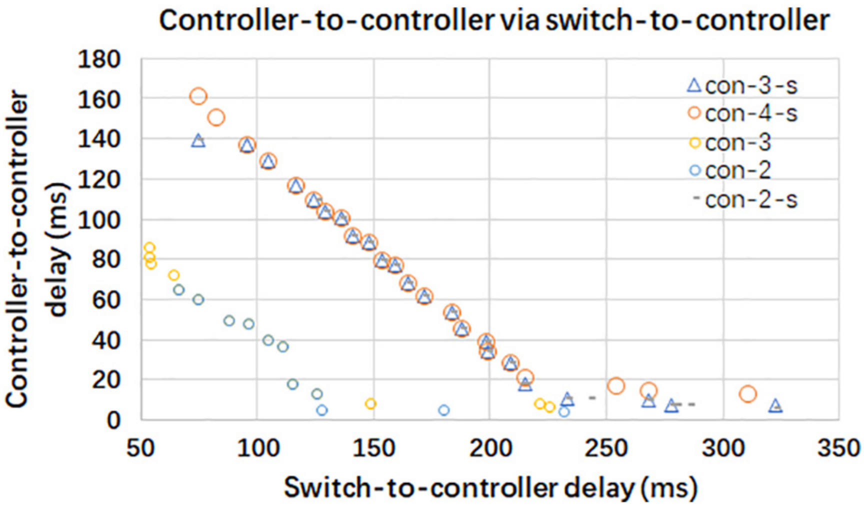
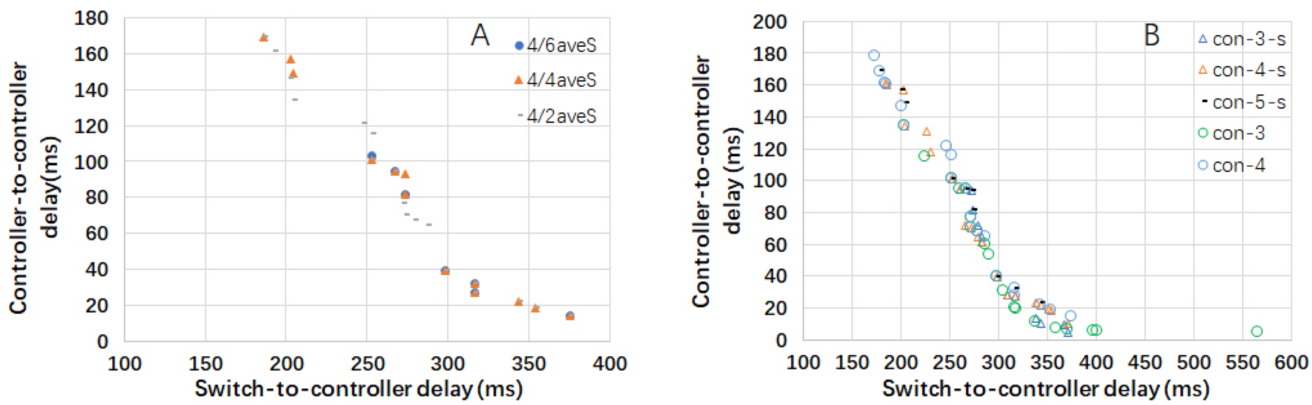


Figure 8. Solving the DCPP over network VtlWavenet2011, frontiers with labels con-2-s, con-3-s, and con-4-s are generated with 2, 3, and 4 local controllers over I_{can}, respectively, frontiers with labels con-2 and con-3 are generated by 2 and 3 local controllers over I

6.2.3 Over WANs with Large Polyindex

Among all the 227 WANs collected by the Internet Topology Zoo, 114 WANs have polyindexs greater than 1.5, implying such WANs have a relatively denser topology than other types of WANs. We choose network Kdl consisting of 714 nodes (the largest network collected by the Internet Topology Zoo) and solve the DCPP over it. Since its longest path is *6.4L* and $I_{can2} = \phi$, 4+ local controllers are needed to roughly meet the delay requirement of 100ms. It is time consuming to directly applying exhaustive algorithms over network Kdl due to 714 nodes consisted in the network. We apply our proposed network simplification strategy to form the set I_{can}. We compute I_{can1}, which consists of 62 nodes. Figure 9(a) shows the number of nodes in set I_{can3} using various thresholds. It is apparent that the number of nodes in I_{can3} decreases as the value of threshold increases. Since I_{can} is the union of I_{can1} and I_{can3}, and the number of nodes in both sets can be large, we need to form a I_{can} with reasonable number of nodes without sacrificing too much accuracy.

As shown in Figure 9(a), as the value of threshold increases from *2aveS* to *4aveS*, the number of nodes in I_{set3} decreases from 67 to 41, while the frontier generated by the DCPP with controllers placed over I_{can3} is not really changed. We let the threshold be *4aveS*. Since $I_{can1} = 62$, we form a I'_{can1} by randomly selecting 1 4th nodes from I_{can1}, and let $I_{can} = I_{can3} \cup I'_{can1}$ and solve the DCPP. As shown in Figure 9(b), as the number of controllers increases, the placements with shorter con2swi and con2con delays are found and the quality of frontiers generated improved. However, the smallest con2swi delay is still greater than 100ms, keeping increasing the number of controllers may achieve the goal but it is too time consuming.



(a) Various thresholds

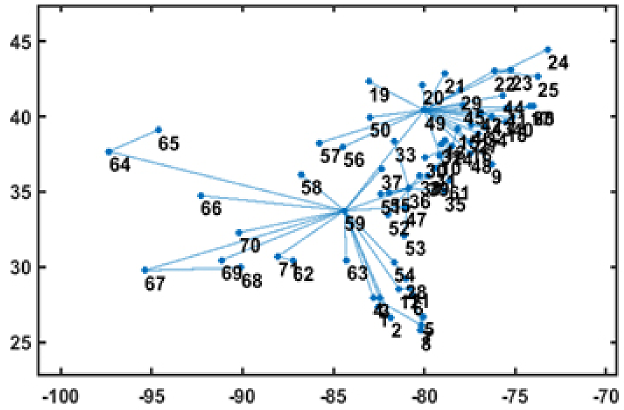
(b) Various number of Controllers

Figure 9. Solving the GCPP over network Kdl, frontiers with labels con-3-s, con-4-s, and con-5-s generated by exhaustive algorithms over I_{can}, frontiers with labels con-3 and con-4 generated over I

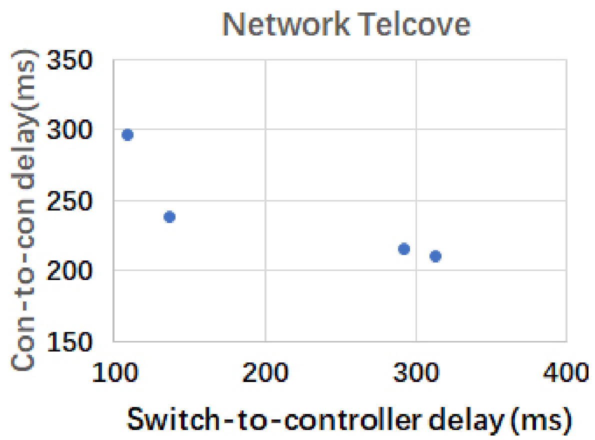
6.3 Discussion

6.3.1 Adjustment of Delay Requirement

Although in many cases, the number of controllers can be roughly determined by the length of the longest path, the size of set I_{can2} should also be considered if I_{can2} is greater than the ceiling of $\frac{L_{longest}}{2L}$. As shown in Figure 10(a), network Telcove consists of 71 nodes and has the polyindex of 0. The length of its longest path is roughly *6L*, and 14 nodes have link distances greater than L ($|I_{can2}|=14$). While placing 3/4/5 local controllers in the control plane is only able to reduce the con2swi delay to roughly 290ms, as shown Figure 10(b), increasing the number of local controllers to 14 is able to reduce the con2swi delay to roughly 100ms. This is because it is impossible to reduce the con2swi delay to be shorter than 100ms unless each node in set I_{can2} is placed a controller if those nodes do not have other nodes to connect to. In this case, the delay requirement of 100ms should be increased to make the number and placement of controllers more reasonable.



(a) Topology of network Telcove



(b) The delays under various number of controllers

Figure 10. Solving GCPP over Network Telcove

For some WANs that have small scales but very long paths, using the ceiling of $\frac{L_{longest}}{2L}$ to roughly determine the number of controller can lead to a situation in which the number of controllers is comparable with the number of nodes in networks. In this case, the delay requirement of 100ms should also be increased to make the number and placement of controllers more reasonable and practical.

6.3.2 Frontiers' Quality after Simplification

The proposed simplification strategy can bring inaccuracy in generated frontiers, since controllers can only be placed on the selected locations. If WANs only need 2 or 3 local controllers according to their longest path, exhaustive algorithms can be directly applied over the original networks without simplification. If WANs have larger scales or need more controllers, the simplification strategy should be applied. The WANs with small polyindexes can have higher quality frontiers after simplification, because the candidate locations for controllers include the nodes along the longest path, which typically forms the structure of such WANs. The simplification strategy does reduce the quality of frontiers generated for the DCPP over WANs with large polyindexes. Since the value of threshold for cliques merging affects the size of the candidate locations, there is a trend that the more candidate locations are provides, the higher quality frontiers can be generated. However, decreasing the value of threshold does not always increase the quality of generated frontiers. Choosing the right value for the threshold is crucial to generate high quality frontiers while reducing the time cost. To further increase the quality of frontiers, comprehensive approaches [29] should be applied, but more research should be done in our future work.

6.3.3 Layered and Flat Distributed Control Planes

We have applied layered and flat distributed control plane to all the WANs collected by the Internet Topology Zoo, but cannot present all the results in detail for the sake of pages. We have found that without inter-controller cooperation, distributed control planes can achieve the shortest con2swi delay, but do not support network-wide management. Flat distributed control planes can support network-wide management, but often place controllers closely to reduce the con2con delay. Shorter con2swi and con2con delay cannot be achieved simultaneously. We have found that layered distributed control planes in which isolated local controllers cooperate through a root controller can balance the con2swi and con2swi delays and provide efficient network-wide management for large-scaled WASDNs.

6.3.4 Placing Controllers over Networks with Big Cliques

For in-band SDN systems, it is difficult to place controllers over the entire network to reduce the con2swi and con2con delays if the network consists of big loops. Such networks, in our case, will have polyindex ranging from 0.5 to 1. Placing controllers in the middle of the loops is an effective way to simultaneously reduce the con2swi and con2con delays if extra infrastructure can be constructed.

7 Conclusion

We have formulated a DCPP to investigate how the controller organization and placement can be jointly optimized with respect to the performance and scalability of a WASDN. We have proposed polyindex to characterize the feature of topologies of WANs. We have introduced the network simplification strategy based on polyindex to capture the structure of topologies, to determine the number and organization of controllers, and to form the candidate location set for controller placements, so that the DCPP over large-scaled WANs can be solved using exhaustive algorithms in a low time cost without sacrificing two much high accuracy. We have computed the polyindexes of all the WANs collected by the Internet Topology Zoo, and shown that polyindex coefficient can distinguish sparse WANs. Extensive evaluations have shown that the proposed network simplification strategy can quickly determine the number and organization of controllers and solve the DCPP with high quality frontiers over the WANs with small and medium polyindexes. Regarding the WANs with large polyindexes, the quality of frontiers increases as the threshold used by the simplification strategy to merge cliques decreases. Carefully choosing the threshold is the key to provide high quality frontiers while maintaining a low time cost.

Acknowledgements

This work was supported by the National Nature Science Foundation of China (61962016).

References

[1] I. D. Priya, S. Silas, A survey on research challenges and applications in empowering the SDN-Based Internet of Things, *in: J. Peter, A. Alavi, B. Javadi (Eds.), Advances in Big Data and Cloud Computing*, Springer, Singapore, 2019, pp. 457-467.

[2] Open Networking Foundation, *Software-defined networking: The new norm for networks*, White Paper, April, 2012.

[3] B. Heller, R. Sherwood, N. McKeown, The controller placement problem, *ACM SIGCOMM Computer Communication Review*, Vol. 42, No. 4, pp. 473-478, October, 2012.

[4] T. Zhang, P. Giaccone, A. Bianco, S. De Domenico, The role of the inter-controller consensus in the placement of distributed SDN controllers, *Computer Communications*, Vol. 113, pp. 1-13, November, 2017.

[5] S. Lange, S. Gebert, J. Spoerhase, P. Rygielski, T. Zinner, S. Kounev, P. Tran-Gia, Specialized heuristics for the controller placement problem in large scale SDN networks, *IEEE 27th International Teletraffic Congress*, Ghent, Belgium, 2015, pp. 210-218.

[6] F. J. Ros, P. M. Ruiz, On reliable controller placements in software-defined networks, *Computer Communications*, Vol. 77, pp. 41-51, March, 2016.

[7] K. Deb, A. Pratap, S. Agarwal, T. A. M. T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE transactions on evolutionary computation*, Vol. 6, No. 2, pp. 182-197, April, 2002.

[8] L. Liao, V. C. Leung, Genetic algorithms with particle swarm optimization based mutation for distributed controller placement in SDNs, *2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, Berlin, Germany, 2017, pp. 1-6.

[9] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, C. A. C. Coello, A survey of multiobjective evolutionary algorithms for data mining: Part I, *IEEE Transactions on Evolutionary Computation*, Vol. 18, No. 1, pp. 4-19, February, 2014.

[10] S. Oldham, B. Fulcher, L. Parkes, A. Arnatkeviciute, C. Suo, A. Fornito, Consistency and differences between centrality measures across distinct classes of networks, *PloS one*, Vol. 14, No. 7, Article No. e0220061, July, 2019.

[11] S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, M. Roughan, The internet topology zoo, *IEEE Journal on Selected Areas in Communications*, Vol. 29, No. 9, pp. 1765-1775, October, 2011.

[12] F. Bannour, S. Souihi, A. Mellouk, Distributed SDN control: Survey, taxonomy, and challenges, *IEEE Communications Surveys & Tutorials*, Vol. 20, No. 1, pp. 333-354, First Quarter, 2018.

[13] G. Wang, Y. Zhao, J. Huang, W. Wang, The controller placement problem in software defined networking: A survey, *IEEE Network*, Vol. 31, No. 5, pp. 21-27, September-October, 2017.

[14] D. Hock, M. Hartmann, S. Gebert, M. Jarschel, T. Zinner, P. Tran-Gia, Pareto-optimal resilient controller placement in SDN-based core networks, *2013 25th IEEE International Teletraffic Congress*, Shanghai, China, 2013, pp. 1-9.

[15] S. Lange, S. Gebert, T. Zinner, P. Tran-Gia, D. Hock, M. Jarschel, M. Hoffmann, Heuristic approaches to the controller placement problem in large scale SDN networks, *IEEE Transactions on Network and Service Management*, Vol. 12, No. 1, pp. 4-17, March, 2015.

[16] A. Jalili, V. Ahmadi, M. Keshtgari, M. Kazemi, Controller

placement in Software-defined WAN using multi objective genetic algorithm, *2nd IEEE International Conference on Knowledge-based Engineering and Innovation*, Tehran, Iran, 2015, pp. 656-662.

[17] S. Liu, H. Wang, S. Yi, F. Zhu, NCPSO: a solution of the controller placement problem in software defined networks, *International Conference on Algorithms and Architectures for Parallel Processing*, Zhangjiajie, China, 2015, pp. 213-225.

[18] C. Gao, H. Wang, F. Zhu, L. Zhai, S. Yi, A particle swarm optimization algorithm for controller placement problem in software defined network, *International Conference on Algorithms and Architectures for Parallel Processing*, Zhangjiajie, China, 2015, pp. 44-54.

[19] M. Rubinov, O. Sporns, Complex network measures of brain connectivity: uses and interpretations, *Neuroimage*, Vol. 52, No. 3, pp. 1059-1069, September, 2010.

[20] MathWorks official website, https://www.mathworks.com/productsmatlab.html.

[21] B. Müller, J. Reinhardt, M. T. Strickland, *Neural networks: an introduction*, Springer Science & Business Media, 2012.

[22] J. P. Onnela, J. Saramäki, J. Kertész, K. Kaski, Intensity and coherence of motifs in weighted complex networks, *Physical Review E*, Vol. 71, No. 6, Article No. 065103, June, 2005.

[23] D. J. Watts, S. H. Strogatz, Collective dynamics of ‘small-world’ networks, *nature*, Vol. 393, No. 6684, pp. 440-442, June, 1998.

[24] G. Palla, I. Derényi, I. Farkas, T. Vicsek, Uncovering the overlapping community structure of complex networks in nature and society, *nature*, Vol. 435, No. 7043, pp. 814-818, June, 2005.

[25] V. Shergin, L. Chala, S. Udovenko, Number of Loops in Elastic Scale-Free Networks, *2018 IEEE International Scientific-Practical Conference Problems of Infocommunications. Science and Technology*, Kharkiv, Ukraine, 2018, pp. 323-326.

[26] T. Fan, L. Lü, D. Shi, T. Zhou, Characterizing cycle structure in complex networks, *arXiv preprint arXiv:2001.08541*, January, 2020.

[27] L. Chen, Y. Kou, Z. Li, A. Xu, Y. Chang, Relationships between Perron–Frobenius eigenvalue and measurements of loops in networks, *Physica A: Statistical Mechanics and its Applications*, Vol. 501, pp. 153-163, July, 2018.

[28] S. Balakrishna, V. K. Solanki, R. Kumar, M. Thirumaran, Survey on Machine Learning-Based Clustering Algorithms for IoT Data Cluster Analysis, in: V. Solanki, M. Hoang, Z. Lu, P. Pattnaik (Eds.), *Intelligent Computing in Engineering*, Springer, Singapore, 2020, pp. 1195-1204.

[29] G. Schütz, J. A. Martins, A comprehensive approach for optimizing controller placement in Software-Defined Networks, *Computer Communications*, Vol. 159, pp. 198-205, June, 2020.

Biographies

Ling Xia Liao is currently a Professor with the School of Electronic Information and Automation, Guilin University of Aerospace Technology, China. Her research interests consist of intelligent network management and optimization, distributed learning algorithms, and edging computing.

Zhi Li is currently a Professor with the School of Electronic Information and Automation, Guilin University of Aerospace Technology, China. His research interests include intelligent instrumentation and measurement.

Han-Chieh Chao is a Professor and Chair of the Department of Electrical Engineering, National Dong Hwa University, Taiwan, R.O.C. His research interests include High Speed Networks, Wireless Networks, IPv6 based Networks and Digital Divide.