This is the author's version of an article that has been published in the proceeding of IEEE Globecom 2017. Changes were made to this version by the publisher prior to publication. The final version is available at https://doi.org/10.1109/GLOCOM.2017.8254748 .

Cluster Leader Election Problem for Distributed Controller Placement in SDN

Genya Ishigaki†, Riti Gour†, Ashkan Yousefpour†, Norihiko Shinomiya*, and Jason P. Jue†
†The University of Texas at Dallas, Richardson, Texas 75080, USA
*Graduate School of Engineering, Soka University Tokyo, 192-8477, Japan
Email: {gishigaki, rgour, ashkan, jjue}@utdallas.edu, shinomi@ieee.org

***Abstract*—This paper discusses the intractability and heuristics of the Cluster Leader Election Problem (CLEP), which is to assign a controller to one of the switches in each cluster of a network. In order to improve scalability in Software Defined Networking (SDN), a network can be clustered into subnetworks and managed by multiple distributed controllers. CLEP deals with the optimal placement of the distributed controllers in the subnetworks considering metrics between the controllers, such as distance and connectivity. Additionally, it is shown that some metrics within each subnetwork, which are discussed in other literature can be guaranteed by selecting specific clustering methods before the placement.**

***Keywords*—Software Defined Networking (SDN), Controller placement, Clustered networks**

I. Introduction

Software Defined Networking (SDN) has ameliorated the simplicity and programmability of networks by dissociating control logic from forwarding functions. However, SDN is approaching the limitation of its centralized management strategy as the size of SDN networks increases.

SDN with distributed controllers divides a managed network into multiple subnetworks (clusters), and each controller is assigned to manage the switches in its cluster. Because topology and control information of a cluster is encapsulated by each controller, this partially centralized architecture succeeds in reducing the global information and results in providing scalability to SDN [1].

One of the most important open problems in SDN relates to the placement of a controller in a given network [2]. The initial model of SDN assumed direct connections between all the switches and their controller with southbound communication links that are dedicated only to control messages. Nonetheless, current models suppose that the control messages are sent by communication links of the data plane because of physical and financial reasons (in-band control messages). In these models, a controller is piggybacked onto one of the switches, and the connections between the controller and the dominated switches are realized by paths on the data plane instead of a physical direct link between them. The controller placement problem tries to optimize the metrics between the controller and the switches, such as communication latency [2] and survivability of the logical connections [3], by selecting an appropriate location for the controller.

When deploying multiple controllers, some metrics over clusters (over-cluster metrics) could be considered for the placement in addition to the metrics within a cluster (intra-cluster metrics). For example, the placement is conducted based on the number of disjoint paths between switches and both their initial and backup controller in order to improve the survivability in [4]. Also, the load of each controller, which is commonly defined by the number of switches, is alleviated by balanced placements of controllers [5], [6].

However, these over-cluster metrics only consider the relationship between the switches in a cluster and some controllers. Assuming cooperative operations among distributed SDN controllers, which require state synchronizations, metrics between the controllers would become another important aspect of the multiple controller placement problem.

Hence, this paper proposes a variant of the multiple controller placement problem as the Cluster Leader Election Problem (CLEP). CLEP is defined as a graph optimization problem of minimizing the distance between controllers over which in-band control messages travel. Additionally, the intractability of CLEP is proved based on reduction to the Set Cover Problem (SCP). Some greedy algorithms for CLEP are also proposed, exploiting an approximation algorithm for SCP. Our simulation compares the performance of these algorithms under different types of graph topologies and clustering methods.

II. Assumptions and Our Model

This paper assumes that a communication network is divided into multiple connected subnetworks called clusters. When dealing with the multiple controller placement problem, the clustering method used to divide a whole network into subnetworks influences the result of placement algorithms, and vice versa. When the placement determines the clusters, the solution can achieve the global optimum solution. The placement algorithm based on K-means in [7] indicates the effectiveness of clustering based on the placement of controllers. However, this type of method cannot be applied to all types of metrics discussed in the placement problem because of the non-convexity of some metrics other than distance.

In contrast, when a set of clusters are given as an input for the placement problem, the topological characteristics of each cluster, such as distance and connectivity, can be guaranteed, though it could be possible that a different clustering provides a better solution for an objective function in terms of the global view. Therefore, in our work, a set of clusters is given as an input in order to simplify the dependency between clustering

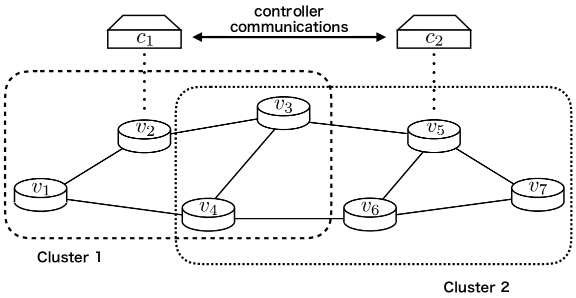

Fig. 1. In-band SDN model with multiple clusters.

and placement. Section IV shows that the placement problem becomes intractable even with fixed clusters.

The communication nodes in a cluster collaborate with each other through communications within the cluster. When information exchange among clusters is required, a leader node of a cluster on which the controller is piggybacked gathers necessary information in its cluster and broadcasts the information to the adjacent clusters, which are defined as clusters sharing some node. Note that this information exchange is conducted using logical connections between leader nodes that are realized by in-band control messages on paths in a data-plane network.

The key assumption is that a leader node cannot directly send its information to non-adjacent clusters. The communications with such clusters are conducted by repeated information exchanges between adjacent clusters, although another work [8] that considers latency among controllers adopts the longest or average distance between all pairs of the controllers. When the size of networks expand, this relay strategy helps leader nodes to reduce the information that they need to maintain. Additionally, the update for a newly joining cluster or leader node becomes simpler because it only requires local arrangements with neighboring leaders.

In our model, some nodes are included in multiple clusters so that all the edges are covered by clusters. However, each node selects one cluster from all the clusters containing the node. When the node receives in-band control messages from clusters other than its own cluster, it relays the messages to appropriate neighbor nodes. For example, in Figure 1, v_3 and v_4 are included in both Cluster 1 and 2. These vertices can select to become a member of either of these two clusters. Suppose v_3 chooses Cluster 1. When receiving a message labeled as Cluster 2, v_3 simply relays the message to v_4 or v_5.

III. Problem Formulation

A. Placement Metric

The single controller placement problem is first discussed by Heller et al. in [2] as a problem to minimize distance between switches and a controller. This latency metric is the most common objective function for the controller placement problem and appears in many other works [9], [10]. In contrast, our CLEP problem is defined as the problem of minimizing latency between controllers, while other works try to reduce the latency between switches and a controller.

B. Problem Statement

A communication network is represented as a graph G consisting of vertices V (communication nodes) and edges E that represent connections between vertices. Also, the node groups are named vertex clusters $\mathcal{C} = \{C \subseteq V\}$.

Then, the cluster leader election problem is finding a leader set $H \subseteq V$ that minimizes the distance-based delay metric satisfying the correctness of the leader election defined as follows.

Definition 1. *Correct Leader Set:* When the following is satisfied, the elected leader set $H \subseteq V$ is said to be correct. For every vertex cluster $C \subseteq V$, at least one vertex $\eta \in C$ that is included in the leader set H exists:

$$\forall C \in \mathcal{C}, \exists \eta \in H \subseteq V \text{ such that } \eta \in C. \tag{1}$$

This statement means that every cluster needs to have its leader; otherwise, the leader election is not correctly completed. This correctness always holds when a mapping function from a set of leaders to a set of clusters is surjective: $\forall y \in Y, \exists x \in X$ such that $f(x) = y$.

Definition 2. *Cluster Adjacency function:* A cluster adjacency function $\alpha : \mathcal{C} \times \mathcal{C} \longrightarrow \{0, 1\}$ gives 0 iff a given pair of vertex clusters $C \in \mathcal{C}$ and $C' \in \mathcal{C}$ does not share any vertices between them; otherwise, it returns 1.

$$\alpha(C, C') = \begin{cases} 0 & \text{if } C \cap C' = \varnothing \\ 1 & \text{if } C \cap C' \neq \varnothing \end{cases}. \tag{2}$$

From the assumption in Section II, the distance based delay for communications among clusters is represented as the sum of distances between each leader $\eta \in H$ and the leaders of the clusters adjacent to the cluster whose leader is η. When $\eta, \eta' \in H$ are the leaders of $C, C' \in \mathcal{C}$ respectively, the delay metric is represented as

$$\sum_{C \in \mathcal{C}} \sum_{C' \in \mathcal{C}} d(\eta, \eta')\alpha(C, C'), \tag{3}$$

where $d(v, v')$ $(v, v' \in V)$ is the distance between two vertices. When this metric is minimized by selecting an appropriate set of leaders H^*, the set of leaders is said to be minimum. Thus, the cluster leader election problem is finding the minimum correct leader set in a given weighted graph.

Problem 1. *Cluster Leader Election Problem (CLEP):* Given a graph $G = (V, E)$, a weight function on edges $w : E \longrightarrow \mathbb{R}_+$, a set of vertex clusters $\mathcal{C} = \{C \subseteq V\}$ and a threshold $K \in \mathbb{R}_+$, is there a subset of vertices $H = \{\eta\}$ whose leader function $l : V \supseteq H \longrightarrow \mathcal{C}$ is surjective and the cost defined as follows is less than K?

$$c(H) := \sum_{\eta \in H} \sum_{C \in l(\eta)} \sum_{\eta' \in H} \sum_{C' \in l(\eta')} d(\eta, \eta')\alpha(C, C'), \tag{4}$$

where $d(s, t)$ $(s, t \in V)$ is the distance between a pair of vertices s, t in terms of the given edge weight w.

Eq. (4) is interpreted as the formal version of Eq. (3). The first two summations indicate the sum over all clusters. The first summation provides the total distance metric for every leader vertex in the leader set H. Because it is possible that some clusters designate the same vertex as their leader, the second summation collects the metric for all the clusters whose leader is the given $\eta \in H$. The other two summations represent the sum for all adjacent clusters. The third summation considers all possible destinations for the communication from η, and the forth summation takes the shared leader into account.

IV. Intractability of CLEP

In this section, the intractability of CLEP is stated by demonstrating the equivalence of the problem and a well-known $\mathcal{NP}$-complete problem called the *Weighted Minimum Set Cover Problem.*

Problem 2. *Weighted Minimum Set Cover Problem (WMSCP):* Given a set U, a finite family of subsets of U, namely $\mathcal{S} = \{S \subseteq U\}$, a weight function on the subsets $w : \mathcal{S} \longrightarrow \mathbb{R}_+$ and a threshold $K \in \mathbb{R}_+$, is there a subfamily $\mathcal{S}^* \subseteq \mathcal{S}$ such that $\bigcup_{S \in \mathcal{S}^*} S = U$ and the cost defined as follows is less than K?

$$c(\mathcal{S}^*) := \sum_{S \in \mathcal{S}^*} w(S) \; . \tag{5}$$

Note that for any cost function w on subsets $S \in \mathcal{S}$, WMSCP is known to be NP-complete [11].

Theorem 1. The Cluster Leader Election Problem is $\mathcal{NP}$-complete.

Proof. Let R_i be a set of clusters such that a cluster C in R_i contains vertex v_i: $R_i := \{C \in \mathcal{C} \mid v_i \in C\}$. Also, $\mathcal{R}$ denotes the collection of all the sets: $\{R_i\}$ $(v_i \in V)$ since R_i is defined for each vertex in V.

Then, with the definition of R_i, the Cluster Leader Election Problem is converted to the following.

Problem 3. *CLEP':* Given a set of vertex clusters $\mathcal{C} = \{C \subseteq V\}$, a finite family of subsets of $\mathcal{C}$, namely $\mathcal{R} = \{R_i \subseteq \mathcal{C}\}$, a weight function on the subsets $w : \mathcal{R} \longrightarrow \mathbb{R}_+$ and a threshold $K \in \mathbb{R}_+$, is there $\mathcal{R}^* \subseteq \mathcal{R}$ that satisfies $\bigcup_{R_i \in \mathcal{R}^*} R_i = \mathcal{C}$ and the cost defined as follows is less than K?

$$c(\mathcal{R}^*) := \sum_{R_i \in \mathcal{R}^*} w(R_i), \tag{6}$$

where

$$w(R_i) := \sum_{C \in l(v_i)} \sum_{v_j : R_j \in \mathcal{R}^*} \sum_{C' \in l(v_j)} d(v_i, v_j) \alpha(C, C'). \tag{7}$$

Therefore, the equivalence of CLEP and WMSCP is straightforward.

Because WMSCP is known to be $\mathcal{NP}$-complete for any weight function on the given subsets, CLEP is also $\mathcal{NP}$-complete. $\square$

In the proof of the converted CLEP (Problem 3), the vertex set derived by indices of $R_i \in \mathcal{R}^*$ forms the required leader set that is minimum and correct. The combination of Eq. (6) and Eq. (7) corresponds to the cost function for the leader set H in Eq. (4). Therefore, the threshold K for the cost function $c(\mathcal{R}^*)$ is equivalent to the minimization of the cost of the leader set $c(H)$. This realizes the minimum condition of the leader election. On the other hand, the condition that $\bigcup_{R_i \in \mathcal{R}^*} R_i = \mathcal{C}$ implies the correctness of a leader set. When this condition is satisfied, at least one vertex exists in the leader set derived by R^* for each cluster in $\mathcal{C}$.

Algorithm 1 WMSCP-based greedy algorithm for CLEP

Input: a set of vertex clusters: $\mathcal{C} = \{C \subseteq V\}$, a finite family of subsets of $\mathcal{C}$: $\mathcal{R} = \{R_i \subseteq \mathcal{C}\}$, a weight function on the subsets $w : \mathcal{R} \longrightarrow \mathbb{R}_+$

$T \leftarrow \varnothing,\ \mathcal{I} \leftarrow \varnothing$
while $T \neq \mathcal{C}$ **do**
  Find $R_i \subseteq \mathcal{C}$ with minimal $\frac{w(R_i)}{|R_i \setminus T|}$
  $T \leftarrow T \cup R_i$
  $\mathcal{I} \leftarrow \mathcal{I} \cup i$
end while

Output: a set of leaders $H = \{v_i \in V\}_{\mathcal{I}} \quad (i \in \mathcal{I})$

V. Heuristic Algorithms

In this section, heuristics for CLEP are proposed based on the greedy approximation algorithm for WMSCP [12]. The approximation algorithm for WMSCP selects a subset whose cost efficiency, defined as $\frac{w(R_i)}{|R_i \setminus T|}$, is minimal at each iteration until all the clusters are covered.

A. WMSCP-based Greedy Algorithm for CLEP

Algorithm 1 describes the steps to elect leaders using the WMSCP algorithm. In order to obtain the set of leaders, the indices of the selected subsets T are also stored in $\mathcal{I}$ during the set covering execution. This greedy algorithm outputs a set of leaders indexed by $\mathcal{I}$ with the input of a set of clusters $\mathcal{C}$, a family $\mathcal{R}$ of the subsets of $\mathcal{C}$, and a weight function w.

As will be understood, the cost efficiency is determined by the weight function w on the subsets R_i of $\mathcal{C}$ and the number of uncovered elements in the subset. However, the weight function w on R_i defined in Eq. (7) requires an exponential number of calculations for all possible combinations of leaders when the number of clusters increases. In general, it is not practical to calculate the exact weight values in an arbitrary network topology. Therefore, our methods try to approximate the values by assuming the worst case scenario and obtaining expected values of the distances between leaders. Note that the computation complexity of the proposed greedy algorithm can be shown by a similar analysis on the complexity of the WMSCP approximation algorithm in [12].

1) Constant Weight w_{con}: The first type of weight function is a constant weight. This function returns a fixed constant value c regardless of any subset.

$$w_{\text{con}}(R_i) = c. \tag{8}$$

2) Worst Weight w_{wst}: The worst weight function weights R_i with the maximum distance between the leader v_i and all other vertices in all clusters in R_i. This value shows the cost of choosing v_i when one of the clusters in R_i designates a vertex furthest from v_i as its leader.

$$w_{\text{wst}}(R_i) = \max_{C \in R_i} \max_{v_j \in C} d(v_i, v_j). \quad (9)$$

3) Average Weight w_{avg}: Similarly, the average weight function determines a weight for a subset R_i as the average of all the distances from the leader v_i to all other vertices in R_i. This weight is the expected cost to select v_i as a leader vertex.

$$w_{\text{avg}}(R_i) = \frac{1}{\sum_{C \in R_i} |C|} \sum_{C \in R_i} \sum_{v_j \in C} d(v_i, v_j). \quad (10)$$

B. Mapping between Leaders and Clusters

Though the heuristics based on the WMSCP algorithm output a set of vertices H that covers all the clusters with minimal cost, they do not decide which vertex becomes a leader for which cluster when one cluster contains more than one vertex from the set H. Thus, it is necessary to map a cluster to one of the vertices in H to complete the leader election problem.

For simplicity in our mapping process, a leader vertex for a cluster C is decided by uniform random selection from the list of all the possible leader vertices in H. As a result, each vertex becomes a member of one cluster whose leader gives control information for the vertex and participates in other clusters as a relay vertex. Some improvements in results could be realized by a mapping method that considers topology information.

VI. Simulation

The proposed greedy algorithms and a centroid algorithm are compared in terms of their performance with respect to the distance between leaders in three types of graph topologies clustered by two clustering methods. The centroid algorithm determines a leader of each cluster so that it can minimize the longest distance from the leader to any other vertices in the cluster. Thus, the centroid algorithm is similar to the common approaches such as K-means [7] for latency reduction except that the clustering is given as an input.

A. Network Models

The algorithms are compared in their performances in three kinds of graph topologies. For the sake of cycle clustering, each graph is augmented so that it has at least 2-edge connectivity. The assurance of 2-edge connectivity is conducted exploiting Tarjan's bridge detection algorithm [13]. Suppose that the endpoints of a bridge found by the Tarjan's algorithm are v_1 and v_2, and $L(v_i)$ is the set of vertices adjacent to v_i. A new edge is added between a pair of randomly selected vertices from $L(v_1) \cup \{v_1\} \setminus \{v_2\}$ and $L(v_2) \cup \{v_2\} \setminus \{v_1\}$ for each bridge. When the edge creates a self-loop or multi-edge in the graph, the random selection is tried again.

The following types of graph topologies are considered:

1) Bi-bridged Barbell graph: A bi-bridged barbell graph consists of two complete graphs with n_1 and n_2 vertices and two paths P_1 and P_2 of length l connecting the two complete graphs. It is guaranteed that a pair of vertices of v_j^1 and v_j^2 is adjacent with one edge of K_{n_j} denoting a lead vertex of path P_i connecting to complete graph K_{n_j} by v_j^i.

2) Newman Watts Strogatz (NWS) random graph: A Newman Watts Strogatz (NWS) random graph shows the scale-free property and high clustering coefficient as well as the small-world property. The degree distribution of a scale-free random graph follows the power law. Additionally, the high clustering coefficient means that a connected triple of vertices tends to have three edges among them. This graph model first composes a cycle covering all the n vertices and connects each vertex with k nearest neighbors in the cycle. Furthermore, additional edges are spanned between two vertices with the probability parameter p.

3) Real-world Network Topology: A real-world network topology, called UUNET, is used for original graphs in our simulation. The UUNET is a network of IP layer with 49 vertices in the United States. The topology information is retrieved from *Internet Topology Zoo* [14].

B. Cluster Settings

1) Diam-k Tree (DkT) Clustering: Diam-k Tree (DkT) clustering is based on tree structures with diameter k. Each cluster is induced by the vertices within k hops from a randomly selected root vertex.

2) Fundamental Cycle (FC) Clustering: Fundamental Cycle (FC) clustering divides a graph into multiple cycle structures using a spanning tree on it. Our simulator first composes a spanning tree from a randomly selected root vertex based on the Breath First Search (BFS) algorithm. The addition of one non-tree edge to the spanning tree formulates a cycle, and this cycle becomes a cluster in this method.

C. Results

Figure 2 and 3 respectively indicate the average distance from a leader to its adjacent leaders in bi-bridged barbell graphs clustered by D2T and FC clustering. In this simulation, the size of two complete graphs (n_1, n_2) is equally set to $20, 30, 40, 50, 60$, and the length of the paths l between the two complete graphs is $6, 10, 14, 17, 20$, respectively.

The average distances in the NWS random graphs with D1T and FC clustering are shown in Figure 4 and 5, respectively. The number of vertices n increases from 20 to 100 with the parameters $k = 2$, $p = 0.1$.

Regardless of the difference in diameters between the graph models, the average weight w_{avg} and constant weight w_{con} demonstrate better performance in reducing the average distance for all the graphs. Because of the paths connecting two complete graphs, the diameter of bi-bridged barbell graphs is always equal to $l + 2$. In contrast, NWS graphs have shorter diameters: $E[\text{diam}(G)] \approx 3.0$, which is caused by the random addition of edges based on p.

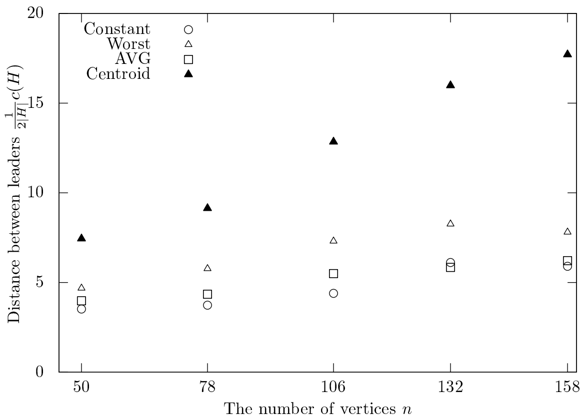

Fig. 2. Average distance from a leader to its adjacent leaders in bi-bridged barbell graph clustered by D2T clustering.

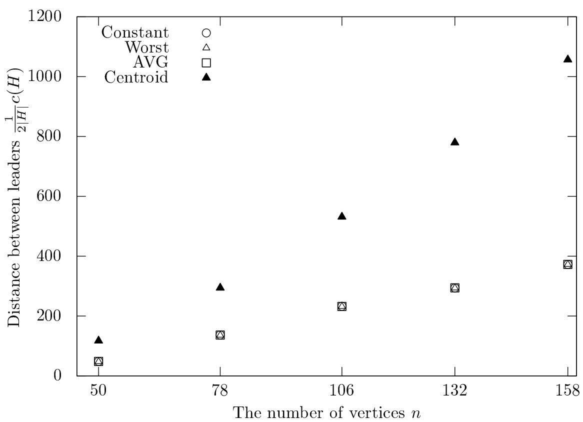

Fig. 3. Average distance from a leader to its adjacent leaders in bi-bridged barbell graph clustered by FC clustering.

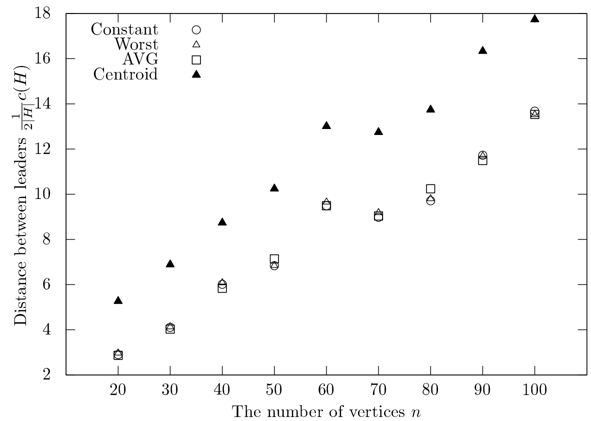

Fig. 4. Average distance from a leader to its adjacent leaders in NWS random graph clustered by D1T clustering.

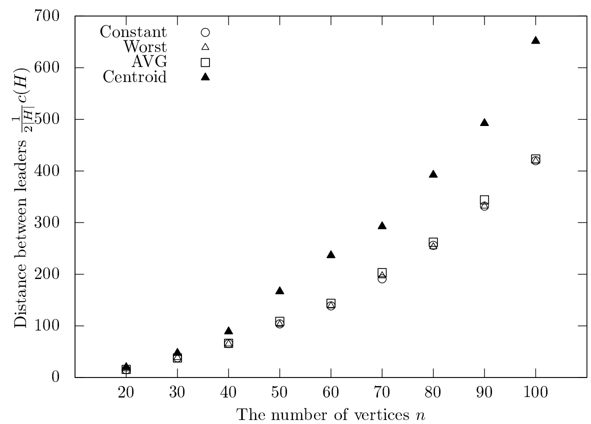

Fig. 5. Average distance from a leader to its adjacent leaders in NWS random graph clustered by FC clustering.

Due to the high clustering coefficient of complete graphs and NWS graphs, the size of fundamental cycles remains about the same $(E[|C|] \approx 4)$ for any graph models. This topological characteristic induces the increase in the number of adjacent cycles along with the growth of the graph size. As a result, the FC clustering has the tendency to increase the average distance to the adjacent leaders. Contrarily, the average size of DkT clusters increases along with the increase in the graph size. Thus, the average distance is moderately augmented in the case of DkT clustering.

Table I describes the results in the UUNET clustered with both D2T and FC clusterings. In terms of the optimization on the distance, the similar discussion holds for this real-world topology. Since the diameter of real-world communication networks tends to be larger than NWS, the results are similar to the case of bi-bridged barbell graphs.

Figure 6 illustrates the comparison of the algorithms and the optimum solution obtained by an exhaustive search in lesser bi-bridged barbell graphs with 16, 18, 20, and 22 vertices. This result implies that the proposed algorithms do not provide solutions that are exceptionally divergent from the optimum solution.

VII. Discussions

A. Guarantee on the Bound of Intra-metrics

Our greedy algorithms to decide a set of leaders do not consider the metrics within each cluster. However, some metrics such as latency and survivability could be guaranteed based on the property of clustering methods rather than their leader election process.

The latency, which is defined by the distance to a controller from switches, could be upper bounded by analyzing the diameter of clusters. It is obvious that the Diam-k Tree clustering always provides the upper bound for the latency between switches and their controller with length $2k$. When a network is partitioned by the fundamental cycle clustering with BFS, the size of the largest cycle becomes $2 \times \text{diam}(G)$.

In addition to latency, survivability is also determined by clustering methods. Because the fundamental cycle clustering divides the entire network into 2-connected clusters, the number of disjoint paths between a controller and switches is equal to 2. In contrast, the Diam-k Tree clustering cannot provide any backup paths for single link failures in a cluster.

TABLE I
AVERAGE DISTANCE FROM A LEADER TO ITS ADJACENT LEADERS IN UUNET WITH CLUSTERED BY D2T AND FC CLUSTERING.

Algorithm	D2T	FC
Constant	7.97	8.53
Worst	8.34	9.22
AVG	**7.45**	**8.16**
Centroid	13.33	20.85

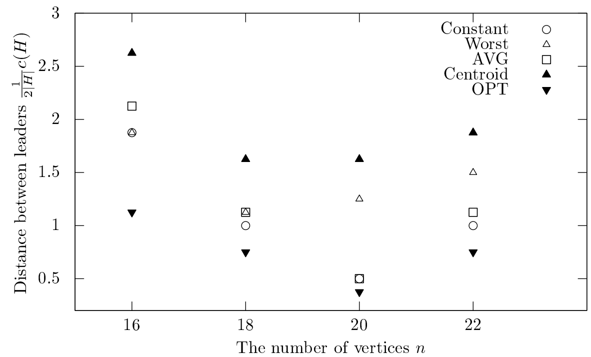

Fig. 6. Average distance from a leader to its adjacent leaders in bi-bridged barbell graph clustered by D1T clustering.

B. Combining Multiple Clusters

As discussed in Section II, the clustering is predetermined as the input before the placement occurs. When the number of clusters is required to be smaller than K in the input, like the work in [7], combining some clusters makes our algorithms satisfy the requirement.

In order to avoid forming a giant cluster, it is better to select the smallest cluster and combine it with the smallest adjacent cluster. After repeating this procedure until the number of clusters becomes less than and equal to K, the proposed algorithm can be used to decide the placement. Even in this case, the discussion on the bound of intra-metrics above holds because the sum of the upper bounds on intra-metrics of combined clusters gives the upper bound for the new cluster at each combining step. It is obvious that the diameter of the new cluster is the sum of diameters of two clusters in the worst case.

C. $\mathcal{NP}$-completeness under Other Assumptions

Though the objective of our Cluster Leader Election problem is the minimization of the distance between controllers, any CLEP aiming at optimizing different metrics between controllers becomes $\mathcal{NP}$-complete. In the proof discussed in Section IV, the only part depending on the distance metric is the weight function on the leader set of Eq. (6). However, the $\mathcal{NP}$-completeness still holds with other functions because WMSCP is known to be $\mathcal{NP}$-complete for any weight functions.

Furthermore, CLEP stays $\mathcal{NP}$-complete even after relaxing the assumption on the multiple membership of a communication node to some clusters. In other works on the clustering methodology, the membership of a node can be restricted to exactly one cluster in order to assure the disjointness of clusters. Though the size of a set R_i remains equal to 1 under this assumption, the completeness proof holds.

VIII. CONCLUSION

In this paper, we formulated the Cluster Leader Election Problem (CLEP), generalizing the multiple controller placement problem of allocating a controller to each cluster while minimizing the distance metric between the controllers. CLEP is proven to be $\mathcal{NP}$-complete by reducing it to Weighted Minimum Set Cover Problem (WMSCP). Exploiting the well-known approximation algorithm for WMSCP, algorithms with three kinds of weights for greedy choices have been proposed. The simulation results indicate the effectiveness of the proposed greedy algorithms using the average distance weight and constant weight.

REFERENCES

[1] T. Koponen, M. Casado, N. Gude, J. Stribling, L. Poutievski, M. Zhu, R. Ramanathan, Y. Iwata, H. Inoue, T. Hama, and S. Shenker, "Onix: A distributed control platform for large-scale production networks," in *Proceedings of the 9th USENIX Conf. on Operating Systems Design and Implementation*, pp. 351–364, 2010.

[2] B. Heller, R. Sherwood, and N. McKeown, "The controller placement problem," in *Proceedings of the First Workshop on Hot Topics in Software Defined Networks*, HotSDN '12, pp. 7–12, ACM, 2012.

[3] Y. Hu, W. Wendong, X. Gong, X. Que, and C. Shiduan, "Reliability-aware controller placement for software-defined networks," in *Integrated Network Management, 2013 IFIP/IEEE Int. Symposium on*, pp. 672–675, May 2013.

[4] L. Muller, R. Oliveira, M. Luizelli, L. Gaspary, and M. Barcellos, "Survivor: An enhanced controller placement strategy for improving sdn survivability," in *Global Communications Conf. (GLOBECOM), 2014 IEEE*, pp. 1909–1915, Dec 2014.

[5] G. Yao, J. Bi, Y. Li, and L. Guo, "On the capacitated controller placement problem in software defined networks," *Communications Letters, IEEE*, vol. 18, pp. 1339–1342, Aug 2014.

[6] L. Yao, P. Hong, W. Zhang, J. Li, and D. Ni, "Controller placement and flow based dynamic management problem towards sdn," in *Communications (ICC), 2015 IEEE Int. Conf. on*, pp. 369–374, June 2015.

[7] G. Wang, Y. Zhao, J. Huang, Q. Duan, and J. Li, "A k-means-based network partition algorithm for controller placement in software defined network," in *2016 IEEE Int. Conf. on Communications (ICC)*, pp. 1–6, May 2016.

[8] S. Lange, S. Gebert, T. Zinner, P. Tran-Gia, D. Hock, M. Jarschel, and M. Hoffmann, "Heuristic approaches to the controller placement problem in large scale sdn networks," *IEEE Trans. on Network and Service Management*, vol. 12, pp. 4–17, March 2015.

[9] Y. Zhang, N. Beheshti, and M. Tatipamula, "On resilience of split-architecture networks," in *Global Telecommunications Conf. (GLOBECOM), 2011 IEEE*, pp. 1–6, Dec 2011.

[10] M. Bari, A. Roy, S. Chowdhury, Q. Zhang, M. Zhani, R. Ahmed, and R. Boutaba, "Dynamic controller provisioning in software defined networks," in *Network and Service Management (CNSM), 2013 9th Int. Conf. on*, pp. 18–25, Oct 2013.

[11] B. Korte, J. Vygen, B. Korte, and J. Vygen, *Combinatorial optimization*, vol. 2. Springer, 2012.

[12] V. V. Vazirani, *Approximation algorithms*. Springer Science & Business Media, 2013.

[13] R. E. Tarjan, "A note on finding the bridges of a graph," *Information Processing Letters*, vol. 2, no. 6, pp. 160–161, 1974.

[14] S. Knight, H. Nguyen, N. Falkner, R. Bowden, and M. Roughan, "The internet topology zoo," *Selected Areas in Communications, IEEE Journal on*, vol. 29, pp. 1765 –1775, october 2011.