How Advantageous Is It? An Analytical Study of Controller-Assisted Path Construction in Distributed SDN

Ziyao Zhang, Liang Ma, Kin K. Leung, Franck Le, Sastry Kompella, and Leandros Tassiulas

***Abstract*— Distributed software-defined networks (SDN), consisting of multiple inter-connected network domains, each managed by one SDN controller, is an emerging networking architecture that offers balanced centralized control and distributed operations. Under such a networking paradigm, most existing works focus on designing sophisticated controller-synchronization strategies to improve joint controller-decision-making for inter-domain routing. However, there is still a lack of fundamental understanding of how the performance of distributed SDN is related to network attributes, thus it is impossible to justify the necessity of complicated strategies. In this regard, we analyze and quantify the performance enhancement of distributed SDN architectures, which is influenced by intra-/inter-domain synchronization levels and network structural properties. Based on a generic network model, we establish analytical methods for performance estimation under four canonical inter-domain synchronization scenarios. Specifically, we first derive an asymptotic expression to quantify how dominating structural and synchronization-related parameters affect the performance metric. We then provide performance analytics for an important family of networks, where all links are of equal preference for path constructions. Finally, we establish fine-grained performance metric expressions for networks with dynamically adjusted link preferences. Our theoretical results reveal how network performance is related to synchronization levels and intra-/inter-domain connections, the accuracy of which is confirmed by simulations based on both real and synthetic networks. To the best of our knowledge, this is the first work quantifying the performance of distributed SDN in terms of network structural properties and synchronization levels.**

***Index Terms*— Distributed SDN, performance analysis, inter-domain routing, controller synchronization.**

I. Introduction

SOFTWARE-DEFINED NETWORKING (SDN) [1], a newly-deployed networking architecture [2], [3], significantly improves the network performance due to its programmable network management, easy reconfiguration, and on-demand resource allocation, which has therefore attracted considerable research interests. One key attribute that differentiates SDN from classic networks is the separation of the SDN's data and control plane. Specifically, in SDN, all control functionalities are implemented and abstracted on the control plane for operational decision making, e.g., flow construction and resource allocation, while the data plane only passively executes the instructions received from the control plane. For a typical SDN architecture, all network decisions are made in the control plane by a logically centralized control entity, called *SDN controller*. Since the logically centralized SDN controller has the full knowledge of network status, it is able to make global optimal decisions. Yet, such centralized control suffers from major scalability issues. In particular, as a network grows, the number of flow requests and operational constraints are likely to increase drastically. Such high computation and communication requirements may impose substantial burden on the SDN controller, potentially resulting in significant performance degradation (e.g., delays) or even network failures [4].

In this regard, distributed SDN is proposed [5]–[9] to balance the centralized and distributed control. Specifically, a distributed SDN network is composed of a set of subnetworks, referred to as *domains*, each managed by an independent SDN controller. Moreover, each domain contains several gateways connecting to some other domains; such inter-connected domains then form the distributed SDN architecture. In the distributed SDN architecture, controllers are expected to exchange information via proactive probing or passive listening. Such additional status information at each controller, called the *synchronized information*, can assist in enhancing decision making for inter-domain tasks. In distributed SDN, network performance relies heavily on the inter-controller synchronization level. Since complete synchronization among controllers, i.e., each controller knows the network status in all other domains, will incur high synchronization overheads especially in large networks, practical distributed SDN networks can only afford partial inter-domain synchronization.

For partial synchronization, most existing works focus on promoting the inter-domain synchronization so that the final decision making approaches optimality. For instance, information sharing algorithms are proposed in [6], [7] for negotiating common traffic policies among various domains. Similarly, frameworks are designed in [8], [9], aiming to facilitate inter-domain routing selection via fine-grained network status exchanges. However, one fundamental question regarding the distributed SDN architecture has generally been ignored: *How does the network performance in distributed SDN relate to network synchronization levels and structural properties?* It is possible that under certain network conditions, the benefit of increasing the synchronization level is only marginal. Without such fundamental understanding, it is impossible to justify

Manuscript received October 17, 2018; revised March 1, 2019 and April 2, 2019; accepted June 19, 2019; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor P. Giaccone. Date of publication July 3, 2019; date of current version August 16, 2019. This work was supported in part by the U.S. Army Research Laboratory and the U.K. Ministry of Defence under Agreement W911NF-16-3-0001. *(Corresponding author: Ziyao Zhang.)*

Z. Zhang and K. K. Leung are with Imperial College London, London SW7 2AZ, U.K. (e-mail: ziyao.zhang15@imperial.ac.uk; kin.leung@imperial.ac.uk).

L. Ma and F. Le are with the IBM T. J. Watson Research Center, Yorktown Heights, NY 10598 USA (e-mail: maliang@us.ibm.com; fle@us.ibm.com).

S. Kompella is with the U.S. Naval Research Laboratory, Washington, DC 20375 USA (e-mail: sk@ieee.org).

L. Tassiulas is with Yale University, New Haven, CT 06520 USA (e-mail: leandros.tassiulas@yale.edu).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors.

Digital Object Identifier 10.1109/TNET.2019.2924616

why a complicated mechanism for information sharing or flow construction is necessary in distributed SDN. We, therefore, investigate this unsolved yet critical problem in the distributed SDN paradigm, aiming at quantifying its performance under any given network conditions.

To this end, we first propose a network topological model to capture the intra-/inter-domain connections in distributed SDN. Based on this network topological model, we further associate a preference level (see Section II-B) to each link for path constructions, which, in practice, is adjusted by SDN controllers based on the collected network information (e.g., traffic and congestion status). Such network model is generic in that it only requires node degree/link preference distributions and the number of gateways in each domain as the input parameters, i.e., they are independent of any specific graph models. Based on this network model, we then derive analytical expressions of the network performance focusing on the average cost of the constructed paths with respect to (w.r.t.) random flow requests. Such performance metric is investigated under four canonical synchronization levels, ranging from the minimum to the maximum level of synchronization (see Section II-D), i.e., between Minimum Synchronization (MS) and *Complete Synchronization (CS)*. If a given synchronization scenario cannot be described by any of these four cases, then its performance can be bounded by our analytical results corresponding to the two extreme cases (i.e., maximum/minimum synchronization).

Specifically, we first establish an asymptotic expression to highlight the relationship between the performance metric and dominant parameters. Then, we conduct detailed analysis on two families of networks - network with uniform and network with non-uniform link preference. The main difference between them is the dynamicity of link preference, where in the former case controllers do not specify any preference for links due to the lack of network status information, thus all links have equal link preference; in the later case, however, controllers assign preference to links to achieve control objectives based on up-to-date network status information collected. Analytical results reveal the relative contributions of different parameters to the performance metric. For example, the performance metric scales linearly with the average domain-wise distance; whereas it scales logarithmically with the number of nodes in each domain (see Theorem 7). Moreover, the performance gain declines with the increasing synchronization level and the number of gateways. To validate the accuracy of the derived analytical expressions, they are compared against evaluation results using both real and synthetic networks.

A. Related Work

1) Information Sharing for Routing Quality Improvement: Researchers have looked into better understanding the performance of hierarchical routing where the internal structure of each domain is not revealed to outside nodes, with both theoretical and experimental approaches. For example, [10] shows that hierarchical routing where the topologies of the clusters are hidden can lead to suboptimal routing, and forwarding loops. [11] proposes solutions for aggregating topologies with theoretical bounds. [12] analyses the effectiveness of hierarchical routing (e.g., ATM PNNI [13], Nimrod [14]), and [15] studies the performance of several different aggregation schemes in terms of network throughput, and network control load. However, most of these early works are either driven by simulations or looked at different aspects of the problem analytically. Thus, they have not tried to study the impact of synchronization and other network structural properties on the network performance from an analytical approach yet. Note that although some of the theoretical results presented in this paper could be applied to the analysis of legacy networks under certain conditions, our work is SDN-focused because many of the assumptions we have for modeling can only be realized through fine-grained control under SDN. For example, SDN's state update process enables fine-grained domain information exchange, which is crucial in our definition of controller synchronization. On the other hand, SDN also makes it possible for joint-decision making and implementation of routing policies, which is in the core of our analysis.

2) Distributed SDN: The distributed SDN architecture, which improves scalability and solves the single-point-of-failure problem, has stimulated many research efforts in this area. Specifically, controller architectures and designs, such as DISCO [16], HyperFlow [17], ONOS [18], and Kandoo [19], are proposed to realize logically centralized but physically distributed SDN architecture with their corresponding aims. In addition, these works [20]–[22] discuss some important issues when realizing the distributed SDN control plane, such as fault-tolerance and the level of control localization, which differ from our work in scope and approaches employed.

3) Performance Analytics: Since all theoretical results in this paper are obtained based on a graph model, our work is also related to the area of graphical analysis of complex networks. Most works in these areas are dedicated to the study of specific graph properties, e.g., small-world effect [23], network motif [24], scale-free [25], etc. Thus, they are substantially different from our work. Early technical reports of this paper can be found here [26], [27].

B. Summary of Contributions

Our main contributions are five-fold.

1) We propose a generic two-layer network model to capture intra-/inter-domain connections and their properties;

2) On top of the network model in *1)*, we use the average path cost (APC) of the constructed paths as the performance metric to develop the asymptotic expression of the APC under any given synchronization levels;

3) For networks with uniform link preference, we develop the analytical expression of the APC lower bound under each synchronization scenario;

4) For networks with non-uniform link preference, we integrate dynamic link preference levels and develop the fine-grained analytical expression of the APC under each synchronization scenario;

5) We evaluate our analytical results by extensive simulations using both real and synthetic networks, which confirm their high accuracy and ability in revealing insights into the actual performance under various network conditions.

In this paper, we do not intend to design improved inter-domain routing mechanisms, and thus only the basic and representative routing strategy (see Section III) is employed for theoretical analysis. To the best of our knowledge, this is the *first work* that studies distributed SDN from the graph-theoretical perspective. The significance of these results is that, given SDN controller's synchronization levels, they shed light on the relationships between network performance and SDN domains' topological properties, thus laying foundations for synchronization protocol design and optimization.

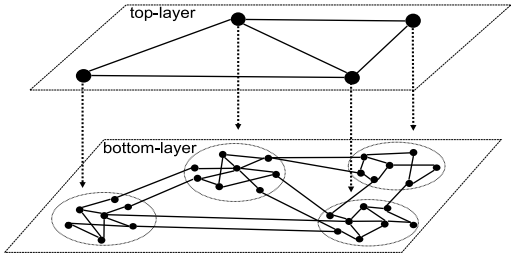


Fig. 1. Two-layer network model: Top-layer abstracts the domain-wise topology; bottom-layer determines all physical connections in the network.

The rest of the paper is organized as follows. Section II formulates the problem. Section III describes the path construction mechanism used for our analysis. Section IV establishes a compact asymptotic APC expression. Then under different synchronization scenarios, Sections V–VII discuss the APC and its performance bound in networks with uniform link preference, based on which Section VIII provides a universal expression of the APC lower bound, which is applicable to any case. Further, sections IX–XII derive the fine-grained expressions of APC in networks with non-uniform link preference under four different synchronization scenarios. Evaluations of the derived analytical expressions are conducted in Section XIII. Finally, Section XIV concludes the paper. Derivation details and explanations of lemmas, theorems and corollaries can be found in the supplementary material.

II. Problem Formulation

A. Network Model

We formulate the distributed SDN network as an undirected graph according to a two-layer network model (Fig. 1), where the top-layer abstracts the inter-domain connections, and given these cross-domain connections, the bottom-layer characterizes physical connections among all network elements. This two-layer structure captures the fact that inter- and intra-domain connections in real-world networks normally have different characteristics. For both layers, we use *degree distribution*, which refers to the distribution of the number of neighboring nodes of an arbitrary node, to capture how nodes are connected in that layer. Our two-layer network model is generic in that the input node degrees and link preference levels can be of any distributions; such distributions can be empirical or extracted from real networks of interest.

Specifically, the top-layer is a graph consisting of m vertices, where each vertex represents a domain in the distributed SDN. These m vertices are connected via undirected links according to a given *inter-domain degree distribution*, which refers to the distribution of the number of neighboring domains of an arbitrary domain. The top-layer graph, denoted by $\mathcal{G}_d = (V_d, E_d)$ (V_d/E_d: set of vertices/edges in $\mathcal{G}_d$, $|V_d| = m$), is called *domain-wise topology* in the sequel. The existence of an edge in E_d connecting two vertices $v_1, v_2 \in V_d$ in the domain-wise topology implies that the two network domains corresponding to v_1 and v_2 are connected. Based on this domain-wise topology, we next construct the physical network in the bottom-layer. In particular, each of the m domains in $\mathcal{G}_d$ corresponds to an undirected graph with n nodes in the bottom-layer; these n nodes are connected following a given *intra-domain degree distribution*, which is the distribution of the number of neighboring nodes of an arbitrary node within the same domain. We also assume that such intra-domain degrees across all domains are independently and identically distributed (i.i.d.). The graph of each domain is referred to as *intra-domain topology*. Then for each $e \in E_d$ with end-points corresponding to domains $\mathcal{A}_i$ and $\mathcal{A}_j$, we (i) randomly select two nodes w_1 from $\mathcal{A}_i$ and w_2 from $\mathcal{A}_j$ and connect these two nodes if link w_1w_2 does not exist, and (ii) repeat such link construction process between $\mathcal{A}_i$ and $\mathcal{A}_j$ β times.[1] By this link construction process, the bottom-layer network topology $\mathcal{G} = (V, E)$ is therefore formed (V/E: set of nodes/links in $\mathcal{G}$, $|V| = mn$); see Fig. 1 for illustrations. In each domain, nodes having connections to other domains are called *gateways*.

B. Link Preference and Path Cost

In the distributed SDN architecture, a routing path construction between a pair of nodes is determined by all involved controllers. To reach an optimized routing decision, controllers take into account the traffic status, load balancing, and other policy-related factors. To this end, controllers can proactively assign a weight to each link to indicate the link preference based on the collected network information, i.e., the smaller the link weight, the better the link is for path construction, so that the end-to-end accumulated weight of any path matches its corresponding path construction preference. Therefore, the goal for constructing an optimized end-to-end inter-domain path under a given network status is reduced to finding the end-to-end path with the minimum accumulated weight under the given link weight assignment. We refer to such accumulated path weight as the *path cost*.

Discussion: Under distributed SDN, link preference assignment is adjusted dynamically by the domain controller according to the current network status and the routing performance metric used. For example, when routing performance is delay (additive metric), the domain controller simply assigns as link weights the delays of intra-domain links under the given traffic levels. In another example, if the routing objective is to find the least congested path (non-additive metric), then the weight assignments should reflect the preference for links with low load levels. We assume that controllers assign such link preferences according to their control objectives; the exact mechanism of link preference assignment subject to different routing objectives is beyond the scope of this paper, and thus not discussed.

Since the link preference (weight) can be dynamic, in this paper, we conduct our analysis in two types of networks which we call *network with uniform link preference (Type-1 Network)* and *network with non-uniform link preference (Type-2 Network)*. For Type-1 Networks, all link preferences are static and equal; therefore, without loss of generality, all link weights in Type-1 Networks are set to 1. By contrast, in Type-2 Networks, random variables (r.v.) are used to capture the dynamicity of link preference. Specifically, for Type-2 Networks, we assume that intra-domain link preferences across all domains are at least 1 and i.i.d. Furthermore, in real distributed SDN environments, unlike the intra-domain links which are potentially wireless, inter-domain gateway-to-gateway links are likely to be wired with high bandwidth, thus more stable. In this regard, we characterize all inter-domain link weights by

[1]Note that the domain-wise topology is a multigraph due to multiple inter-domain links between two directly connected domains. In this paper we assume that the domain-wise path construction uses loop-detection mechanism similar to that in BGP to avoid routing loops.

a non-negative constant. Without loss of generality, we assume that the link preference levels for all inter-domain links are 1; all our theoretical results can be easily extended to other policy-based inter-domain setups, if the behaviors of such policy-based setups can be captured by random variables with certain distributions.

C. SDN Data and Control Plane

Thus far, we have only discussed the graphical properties of the distributed SDN networks. One critical aspect of SDN that differentiates it from other networks is the separation of the data and control planes, which are formulated as follows.

1) Data Plane: We exploit graph $\mathcal{G}$ generated by the two-layer network model in Section II-A to represent the data plane of the distributed SDN. Specifically, a node/link exists in $\mathcal{G}$ if and only if it can be used for data transmission in the network.

2) Control Plane: Under the two-layer network model, each domain contains one logical SDN controller that carries out control operations and facilitates information sharing. SDN controllers together with all inter/intra-domain controlling channels form the control plane.

Remark: Our network model makes no assumptions on how the control and data planes interact with each other; therefore, it is applicable for both in-band and out-of-band control.

D. Synchronization Among SDN Controllers

Since link preference (weight) captures the controller's view of the current domain, i.e., network status information, the process of controller synchronization involves the exchange of such information, which we formally define below.

Definition 1: Domain $\mathcal{A}_i$ is synchronized *with domain $\mathcal{A}_j$ if and only if the SDN controller in $\mathcal{A}_i$ knows the minimum path cost between any two nodes in $\mathcal{A}_j$.*

By Definition 1, clearly there exist a significant number of synchronization cases. Moreover, in real networks, it is usually the case that synchronization difficulty is high when two SDN controllers are far apart. In this paper, we therefore categorize inter-domain synchronizations into the following cases, sorted by their corresponding synchronization difficulties.

*a) Minimum Synchronization (**MS**)*: Under MS, no domains synchronize with any other domains. As a result, each controller only knows its own intra-domain topology and the domain-wise topology, but the controller does not assign link preference levels (all links have an equal link preference of 1) due to the lack of network status information. This scenario captures IGP routing protocols that do not take into account any link weights but select routes purely based on the hop count (e.g., Routing Information Protocol (RIPv2)). Note that MS corresponds to the minimum network knowledge that is always available, including scenarios in *b)–d)*;

*b) Self-domain Synchronization (**SS**)*: In addition to the information under MS, each controller under SS knows nothing more except for its intra-domain and out-going inter-domain link preference levels. With this additional information, one controller can find the optimal intra-domain path for any intra-domain flow requests, within its own domain;

*c) Partial Synchronization (**PS**)*: PS refers to any synchronization levels between SS and the following complete synchronization (CS), where some controllers exchange the views of their own domains gained through SS. See an example of PS in Fig. 2 where there are five domains, among which domain pairs $\{\mathcal{A}_1, \mathcal{A}_2\}$, and $\{\mathcal{A}_3, \mathcal{A}_4\}$ are respectively synchronized; $\mathcal{A}_5$ is not synchronized with the other four domains. Under PS, SDN and legacy routing policies could coexist, e.g., those synchronized domains may operate on SDN routing utilizing the synchronized information (see Section III for details), whereas those not synchronized operate on a fully distributed inter-domain routing protocol such as BGP. PS is the most realistic scenario in distributed SDN, as it balances the benefits and costs of controller synchronization;

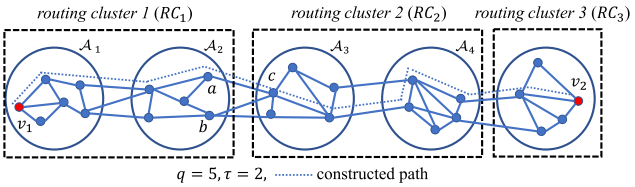


Fig. 2. Path construction w.r.t. v_1 and v_2, whose shortest domain-wise path traverses $\mathcal{A}_1$, $\mathcal{A}_2$, $\mathcal{A}_3$, $\mathcal{A}_4$, and $\mathcal{A}_5$.

*d) Complete Synchronization (**CS**)*: Under CS, every pair of domains $\mathcal{A}_i$ and $\mathcal{A}_j$ synchronize with each other. As such, there is effectively one logically centralized controller, which can make globally optimal decisions. Among all these synchronization scenarios, CS experiences the highest synchronization difficulty.

E. Problem Statement and Objective

Given the distributed SDN network model in Section II-A, our goal is to study the performance of the paths constructed by a basic and representative path construction mechanism (see Section III for details) under various synchronization scenarios. In real networks, the performance of routing can be measured by many metrics, such as delay and congestion level, depending on the goal of network management. In order to make our analytical work sufficiently generalized to capture the performance metric that is important to most network management tasks, we employ the *Average Path Cost (APC)*, measured by the average cost of the constructed path, as the performance metric. Here APC is a natural generalized performance metric, as link weights are dynamically adjusted by controllers based on the current network status to reflect time-varying link preference. Formally, our research objective is:

Objective: Suppose (i) each network realization under the two-layer network model exists with the same probability, and (ii) the source-destination node pair belonging to two different domains in a given network realization also exists with the same probability. Our goal is to derive mathematical expressions of APC under each of the four synchronization scenarios, i.e., MS, SS, PS, and CS, in both Type-1/Type-2 Networks (networks with uniform/non-uniform link preference).

Remark: In this paper, we are only interested in studying the cross-domain routing, since controllers can easily find the optimal intra-domain paths without relying on inter-controller synchronizations. Note that our two-layer network model is a random graph model, i.e., there exist multiple network realizations satisfying the same set of

input parameters. Therefore, APC is an expected value over not only random source/destination node pairs but also random network realizations. All our theoretical results on APC are based on the given network parameters (e.g., degree and weight distributions) rather than a specific network realization.

III. Path Construction Mechanism

We describe a path construction mechanism for 4 synchronization scenarios introduced in Section II-D. The intuition behind the path construction mechanism is that given a particular synchronization level, the synchronized controllers attempt to use the synchronized information and make joint decision to minimize the overall accumulated cost of the constructed path in their domains. Then the selected path segments in all participating domains between the source/destination nodes concatenate into a cross-domain, end-to-end path. Before presenting the path construction mechanism, we first introduce several definitions as follows.

Definition 2: a) *In the domain-wise topology $\mathcal{G}_d$, the vertex corresponding to domain $\mathcal{A}$ in $\mathcal{G}$ is denoted by $\vartheta(\mathcal{A})$. Given a pair of source and destination nodes v_1 and v_2 with*[2] *$v_1 \in \mathcal{A}_1$, $v_2 \in \mathcal{A}_2$, and $\mathcal{A}_1 \neq \mathcal{A}_2$, the* domain-wise path *w.r.t. v_1 and v_2 is a path in $\mathcal{G}_d$ starting at vertex $\vartheta(\mathcal{A}_1)$ and terminating at vertex $\vartheta(\mathcal{A}_2)$;*

b) *The* domain-wise distance *w.r.t. domains $\mathcal{A}_1$ and $\mathcal{A}_2$ is the length of the shortest path from the vertex corresponding to $\mathcal{A}_1$ to the vertex corresponding to $\mathcal{A}_2$ in the domain-wise topology $\mathcal{G}_d$.*

Based on Definition 2, we then define synchronization radius to capture different levels of synchronizations as follows.

Definition 3: The synchronization radius *τ ($\tau \geq 1$) is an* integer *such that (i)* any *two domains with their domain-wise distance less than or equal to $\tau - 1$ are synchronized, and (ii)* no *two domains with their domain-wise distance greater than $\tau - 1$ are synchronized.*

According to the definition of synchronization radius, $\tau = 1$ for MS or SS, depending on link preference status; $\tau = \phi$ for CS, where ϕ is the maximum domain-wise distance between any two domains in the network. Any value of τ between 1 and ϕ falls in the category of PS. As such, we use a given τ to capture the PS scenario. Under a specified synchronization level, the synchronized controllers leverage the shared information to jointly make routing decisions on any domain-wise paths between source/destination nodes. Formally, we have the following definition.

Definition 4: The group of domain(s) on the domain-wise path where routing decisions are jointly made by their synchronized controller(s) is referred to as a routing cluster (RC). *Specifically, given a domain-wise path between the source and destination nodes, for all domains on this domain-wise path:*

a) *Under MS or SS ($\tau = 1$), each domain constitutes an RC;*

b) *Under PS ($1 < \tau < \phi$), starting from the source domain, every τ domains form an RC such that each domain belongs to one and only one RC, and only the RC including the destination domain may have less than τ domains;*

c) *Under CS ($\tau = \phi$), all domains on the domain-wise path form an RC, where ϕ is the maximum domain-wise distance between any two domains in the network.*

According to Definition 3 and 4, for any domain pairs inside an RC, there must be at least one domain-wise path connecting them s.t. all intermediate domains on the domain-wise path between them are within the same RC; otherwise, jointly optimal routing decisions cannot be guaranteed between any two nodes within the RC, due to the presence of external domain(s) en route, whose information is not known to RC domains. Based on Definition 4, let q and μ denote the number of domains and the number of RCs on the domain-wise path, respectively. For PS with synchronization radius τ, the RC that includes the destination domain has $q - \tau(\mu - 1)$ domains, whereas all other RCs have τ domains. Now, we are ready to introduce the path construction mechanism between two arbitrary nodes v_1 and v_2 in the following steps:

Step 1) Select the shortest domain-wise path w.r.t. v_1 and v_2, which consists of q domains, with ties (if any) broken arbitrarily. That is, no domain-wise path w.r.t. v_1 and v_2 traverses less than q domains.

Step 2) Based on the given synchronization status of all involved domains on the above domain-wise path, partition these domains into μ RCs ($\mu = q$ for MS and SS, $\mu = 1$ for CS, and $\mu = \lceil q/\tau \rceil$ for PS);

Step 3) For each RC_i (RCs are sequentially labeled from the source to the destination, $i = 1, 2, \ldots, \mu$), a path segment starting from the entering node (which is v_1 if $i = 1$, or is specified by RC_{i-1}) and terminating at one of the exiting nodes (which are gateways connecting to RC_{i+1}, or node v_2 if $i = \mu$) with the minimum cost is constructed.[3] Such path segment is denoted by $\mathcal{P}_i$ in RC_i. Also let $e_{i,i+1}$ be the edge leading from $\mathcal{P}_i$ in RC_i to connect to the entering node in RC_{i+1} if $i \leq \mu - 1$;

Step 4) The final v_1-to-v_2 path $\mathcal{P}$ is

$$\mathcal{P} = \mathcal{P}_1 + e_{1,2} + \mathcal{P}_2 + e_{2,3} + \ldots + \mathcal{P}_{\mu-1} + e_{\mu-1,\mu} + \mathcal{P}_\mu. \quad (1)$$

Discussion: ***Step 1)*** is similar to the BGP protocol used for inter-domain routing in the Internet. We further justify the selection of the shortest domain-wise path in Theorem 12 and Corollary 13. The path construction mechanism described above relies on routing clusters as the basic routing unit, it is therefore referred to as *routing cluster-based path construction (RCPC)* in the sequel. Fig. 2 shows a PS example with $q = 5$ and $\tau = 2$ under RCPC. After the selection of a domain-wise path which consists of domains $\mathcal{A}_1 - \mathcal{A}_5$, the domains are partitioned into 3 RCs according to ***Step 2)***, as shown in the figure. Then, by ***Step 3)***, routing decision is made jointly by controllers in each RC to minimize the corresponding path cost. For example, assume that all link preferences are 1 in Fig. 2, the controllers of $\mathcal{A}_1$ and $\mathcal{A}_2$ jointly choose node a as the exit point and thus construct a path segment between v_1 and a in RC_1. The core of RCPC is that the synchronized SDN controllers jointly decide the routing policies according to the link preferences in their domains, i.e., to minimize accumulated end-to-end link weights. For mathematical tractability, other routing-related factors, such as the LOCAL PREFERENCE in BGP, are reflected in the controller-assigned link weights (see *Discussion* in Section II-B).

Note that the intention in this paper is not to design a new routing mechanism; instead, the goal is to use a basic routing mechanism, RCPC, to understand the network performance

[2]In this paper, for graph $\mathcal{G} = (V, E)$, by abusing graph theory notations, we use vertex $v \in \mathcal{G}$ to denote $v \in V$ and edge $e \in \mathcal{G}$ to denote $e \in E$.

[3]Note that ECMP or similar schemes could be applied within RCs, since equal intra-RC costs would be incurred.

in distributed SDN. For improved routing mechanisms, our RCPC-based analytical results serve as performance bounds.

IV. Asymptotic APC Under Different Synchronization Levels

Before the discussion of fine-grained analytical results on APC, we first present the asymptotic analysis of APC (called *asymptotic APC*) under various synchronization scenarios in this section. The basic idea here is that we highlight, in the form of directly observable expressions, the interactions among different parameters in determining the overall APC.

The basic intuition behind the derivation of the asymptotic APC is that we first compute the average domain-wise distance w.r.t. two arbitrary source/destination nodes. Then, with the given synchronization level (τ), the domains on the domain-wise path form RCs according to RCPC. Finally, we calculate the APC inside individual RCs and add up these APCs to obtain the accumulated end-to-end APC. When the number of domains inside an RC is more than one, we employ a special graph, called the *Randomized Degree-Preserving Network (RDPN)*, to help us derive its APC. In essence, RDPN is obtained by aggregating the topologies of all domains inside an RC to a single graph, for which the aim is to make the derivation of APC tractable (see Definition 9 for details).

Let m and n be the number of domains and the number of nodes in each domain in the network, and γ the average number of gateways connecting two neighboring domains ($\gamma = n(1-(1-1/n)^{\beta})$). Next, within a domain $\mathcal{A}$, let z_i denote the average number of vertices that are i-hop away from a random vertex within $\mathcal{A}$. Similarly, in the top-layer $\mathcal{G}_d$ of our two-layer model, let z'_i denote the average number of vertices (here each vertex represents a domain) that are i-hop away from a random vertex in $\mathcal{G}_d$. In addition, let ζ_i be the average number of vertices which are i-hop away from a random vertex in an RDPN. Main notations and abbreviations used in this paper are summarized in Table I. Under all above definitions and path construction mechanisms, we present the asymptotic APC in the following theorem.

Theorem 5: Given the synchronization radius τ, the asymptotic APC (denoted by $\mathcal{L}$) in the two-layer network model is

$$\mathcal{L}=\begin{cases} O\Big(\dfrac{(\Delta-1)\log(\frac{n\tau'}{\zeta_1\gamma})}{\tau\log(\zeta_2/\zeta_1)}+\dfrac{\log(n\tau'/\zeta_1)}{\log(\zeta_2/\zeta_1)}\Big) & \text{if } \gamma\le\dfrac{n\tau'+1}{\zeta_1+1},\\ O\Big(\dfrac{\Delta-1}{\tau}+\dfrac{\log(n\tau'/\zeta_1)}{\log(\zeta_2/\zeta_1)}\Big) & \text{otherwise,}\end{cases} \tag{2}$$

where $\tau'=\min\{\tau,\Delta+1\}$; see Table I for other notations.

Theorem 5 directly shows how the synchronization level (τ) affects the APC. Specifically, when τ is small, there are two dominant terms, which are both logarithmic functions, in (2). However, with the increase of τ, when the network achieves CS, only the second logarithmic function is dominant, and the two cases under different values of γ in (2) are merged into one unified expression, i.e., $\mathcal{L}=O\Big(\frac{\log(n\tau'/\zeta_1)}{\log(\zeta_2/\zeta_1)}\Big)$, with $\tau'\approx\Delta+1$. To better observe these trends, we consider a sample two-layer network with the *Erdös-Rényi (ER)* model (see Section XIII-A.2) as the graph model in each layer with the following parameters: $m=200$, $n=500$, $p=2/199$ for the domain-wise topology, and $p=3/499$ for the intra-domain topology (see Section XIII-A.2 for parameter p) and visualize the corresponding expression of (2) in Fig. 3. Clearly, $\mathcal{L}$ steadily descends with a diminishing amount every time τ increases by 1, thus implying the declining benefit of the increased synchronization level. In addition, we also observe that having more gateways (larger β) results in a smaller $\mathcal{L}$. However, the performance gain of larger β also gradually diminishes as the synchronization level grows. Thus, there is a cost/benefit trade-off that needs to be considered in practical network design. The asymptotic APC's ability to reveal the relationship between APC and other parameters are validated in Section XIII.

TABLE I
Main Notations and Abbreviation

Notation	Meaning
m	number of domains in the network
n	number of nodes in each domain
β	inter-domain connection parameter
γ	$\gamma=n(1-(1-1/n)^{\beta})$, average number of gateways in a domain connecting to a neighbouring domain
z_1, z_2	average number of nodes that are 1-/2-hop away from a randomly chosen node within a domain
z'_1, z'_2	average number of domains that are 1-/2-hop away from a randomly chosen domain in the domain-wise topology
τ	synchronization radius
ζ_i	average number of vertices which are i-hop away from a random vertex in a RDPN (Definition 9)
Δ	$\Delta=\frac{\log(m/z'_1)}{\log(z'_2/z'_1)}+1$ is the average domain-wise distance w.r.t. two arbitrary domains (Section V)
MS	minimum synchronization
SS	self-domain synchronization
PS	partial synchronization
CS	complete synchronization

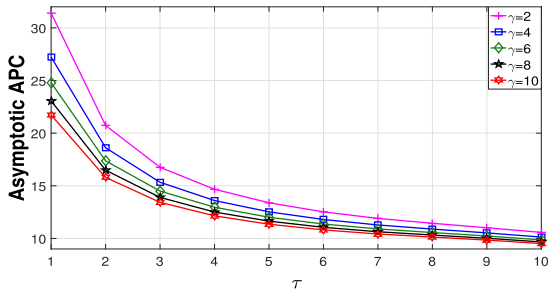


Fig. 3. $\mathcal{L}$ in (2) in a sample network with varying τ and γ.

V. APC Under MS in Type-1 Networks

In this section, we study the APC under MS in Type-1 Networks (all links are of equal preference, i.e., link preference levels are 1 for all links) based on the path constructions mechanism RCPC introduced in Section III. To this end, we first present the results in the existing work [28] to assist our mathematical analysis.

Proposition 6 [28]: In an undirected connected graph $\mathcal{H}$ with n_0 vertices and the vertex degree satisfying a given distribution, let x_i be the average number of vertices that are i-hop away from a random vertex in $\mathcal{H}$. Suppose all edge weights are 1, and $x_2 \gg x_1$ [4]. Then

$$\text{a)} \qquad x_i=(x_2/x_1)^{i-1}x_1; \tag{3}$$

[4]This is a valid assumption because, according to our observations of real network datasets, the number of two-hops nodes is (exponentially) larger than the number of immediate neighbor nodes in most cases, i.e., $x_2 > x_1^2$, in both intra-domain and domain-wise topologies.

b) *APC in $\mathcal{H}$ is*

$$\frac{\log(n_0/x_1)}{\log(x_2/x_1)}+1. \tag{4}$$

In our two-layer model, the top-layer graph $\mathcal{G}_d$ (domain-wise topology with m vertices) itself is a random graph following a given domain-wise degree distribution. Therefore, similar to [28], let z_i' denote the average number of vertices that are i-hop away from a random vertex in $\mathcal{G}_d$. For two arbitrary nodes v_1 and v_2 with $v_1 \in \mathcal{A}_1$, $v_2 \in \mathcal{A}_q$, and $\mathcal{A}_1 \neq \mathcal{A}_q$, let Δ denote the average distance of the shortest domain-wise path from domain $\mathcal{A}_1$ to domain $\mathcal{A}_q$. Assuming $z_2' \gg z_1'$, then according to (4), we have

$$\Delta = \frac{\log(m/z_1')}{\log(z_2'/z_1')}+1. \tag{5}$$

With (5), we know that the average number of domains for MS under RCPC is $\Delta+1$. If we further know the average cost of $\mathcal{P}_i$ associated with the traversed domain $\mathcal{A}_i$, then we can estimate the average cost of $\mathcal{P}$. To this end, let $|\mathcal{P}|$ denote the *number of hops* on path $\mathcal{P}$. Then, $|\mathcal{P}| = |\mathcal{P}_1| + |\mathcal{P}_2| + \ldots + |\mathcal{P}_{\Delta+1}| + \Delta$ according to (1), where $|\mathcal{P}_i|$ is a r.v. The expectation of $|\mathcal{P}|$ is:

$$\begin{aligned}\mathbb{E}[|\mathcal{P}|] &= \mathbb{E}[|\mathcal{P}_1| + |\mathcal{P}_2| + \ldots + |\mathcal{P}_{\Delta+1}|] + \Delta \\ &= \mathbb{E}[|\mathcal{P}_1|] + \mathbb{E}[|\mathcal{P}_2|] + \ldots + \mathbb{E}[|\mathcal{P}_{\Delta+1}|] + \Delta. \end{aligned} \tag{6}$$

According to the path construction procedure for MS, $\mathbb{E}[|\mathcal{P}_1|] = \mathbb{E}[|\mathcal{P}_2|] = \ldots = \mathbb{E}[|\mathcal{P}_\Delta|]$ for two reasons. First, all domains have the same statistical properties. Second, in each domain $\mathcal{A}_i$ ($i \leq \Delta$), the routing mechanism selects a gateway (from a set of gateway options) that is closest to the ingress node. By contrast, in domain $\mathcal{A}_{\Delta+1}$, the routing mechanism only selects the minimum-cost path from the ingress node to the destination node v_2. Thus, (6) is simplified as

$$\mathbb{E}[|\mathcal{P}|] = \Delta \cdot \mathbb{E}[|\mathcal{P}_1|] + \mathbb{E}[|\mathcal{P}_{\Delta+1}|] + \Delta. \tag{7}$$

In a domain $\mathcal{A}$ with n nodes, let z_i denote the average number of intra-domain nodes that are i-hop away from an arbitrary node v ($v \in \mathcal{A}$). Then again by (4), we have

$$\mathbb{E}[|\mathcal{P}_{\Delta+1}|] = \frac{\log(n/z_1)}{\log(z_2/z_1)}+1, \tag{8}$$

assuming $z_2 \gg z_1$. Hence, to compute $\mathbb{E}[|\mathcal{P}|]$ in (7), it suffices to consider only $\mathbb{E}[|\mathcal{P}_1|]$ associated with domain $\mathcal{A}_1$.

In $\mathcal{A}_1$, on average, there are $\gamma = n(1-(1-1/n)^\beta)$ gateways connecting to $\mathcal{A}_2$. Suppose $\mathcal{A}_1$ contains exactly γ gateways, denoted by set S. Then regarding path $\mathcal{P}_1$ from the starting point v_1 in $\mathcal{A}_1$ to set S, there are two cases. First, $v_1 \in S$, then $\mathcal{P}_1$ is a degenerate path containing only one node v_1, i.e., $|\mathcal{P}_1| = 0$. Second, $v_1 \notin S$, which complicates the computation of $|\mathcal{P}_1|$. For the second case, let $l := \mathbb{E}[|\mathcal{P}_1| \ |v_1 \notin S]$, i.e., the expectation of $|\mathcal{P}_1|$ conditioned on $v_1 \notin S$. Regarding the gateway set S, there are up to γz_i non-gateways that are i-hop away from the closest gateways. Let $l_{\max} := \arg\max_i z_i$ s.t. $\gamma + \sum_{j\leq i} z_j \leq n$. According to (3), z_i increases exponentially with i. In other words, the majority of non-gateways are $l_{\max}$-hop away from the closest gateways; therefore, we use $l_{\max}$ to approximate l. Thus, $z_l \approx z_{l_{\max}} \approx n-\gamma \approx n+1-\gamma$ when n is large. By solving $z_l = n+1-\gamma$, we obtain

$$l = \frac{\log(\frac{n+1-\gamma}{z_1\gamma})}{\log(z_2/z_1)}+1, \tag{9}$$

where $\gamma = n(1-(1-1/n)^\beta)$. By close examination of (9), we notice that it is also needed to guarantee $l \geq 1$. Hence, (9) can be calibrated as follows.

$$l = \begin{cases} \dfrac{\log(\frac{n+1-\gamma}{z_1\gamma})}{\log(z_2/z_1)}+1 & \text{if } \gamma \leq \dfrac{n+1}{z_1+1}, \\ 1 & \text{otherwise.} \end{cases} \tag{10}$$

We can verify that when $\gamma = 1$, (10) reduces to (8) as expected. A key threshold $\gamma_0 = (n+1)/(z_1+1)$ is revealed in (10). When $\gamma \leq \gamma_0$, the distance from an arbitrary non-gateway to the closest gateway is relatively large; when $\gamma > \gamma_0$, there are sufficiently many gateways randomly distributed in one domain, causing each non-gateway to have a gateway neighbor with high probability. Hence,

$$\begin{aligned}\mathbb{E}[|\mathcal{P}_1|] &= \mathbb{E}[|\mathcal{P}_1| \ |v_1 \notin S]\mathsf{Pr}(v_1 \notin S) \\ &\quad + \mathbb{E}[|\mathcal{P}_1| \ |v_1 \in S]\mathsf{Pr}(v_1 \in S) + 1 = (\frac{n-\gamma}{n})l + 1, \end{aligned} \tag{11}$$

where $\frac{n-\gamma}{n}$ is the percentage of non-gateway nodes in a domain. Putting (5), (8), and (11) into (7), the final expression of APC under MS is summarized in the following theorem.

Theorem 7: The APC in Type-1 Networks under MS (denoted by $L_{MS}^{\text{Type-1}}$) is

$$L_{MS}^{\text{Type-1}} = \begin{cases} \Delta\Big((\dfrac{n-\gamma}{n})(\dfrac{\log(\frac{n+1-\gamma}{z_1\gamma})}{\log(z_2/z_1)}+1)+2\Big) & \\ +\dfrac{\log(n/z_1)}{\log(z_2/z_1)}+1 & \textit{if } \gamma \leq \dfrac{n+1}{z_1+1}, \\ \Delta(\dfrac{n-\gamma}{n}+2)+\dfrac{\log(n/z_1)}{\log(z_2/z_1)}+1 & \textit{otherwise}, \end{cases} \tag{12}$$

see Table I for notations.

It can be observed that the domain-wise distance (Δ) and the number of gateways (γ) in domains are the most influential factors in shaping the APC for MS. Specifically, $L_{\text{MS}}^{\text{Type-1}}$ is logarithmic in domain structural parameters n, γ, z_1, and z_2, and it is near linear in Δ.

Since SS coincides with MS in Type-1 Networks, we therefore discuss synchronization scenario PS in the next section.

VI. APC Under PS in Type-1 Networks

In this section, we consider the partial synchronization (PS) model[5] as defined in Definition 3. RCs are created as basic routing units according to Definition 4 under PS. Note that the network graph of an RC is no longer a random graph, because multiple domains are connected via inter-domain connections in a specific way as dictated by the network model. As such, we cannot directly apply the results obtained in Section V for the APC expression under MS in Type-1 Networks. Regarding such difficulties, in this section, we instead derive the APC lower bound for PS with the assistance of an auxiliary network called the *Randomized Degree-Preserving Network (RDPN)* (see Definition 8). Here is the sketch of our methodology.

Sketch of Analytical Methodology:

[5]Such PS model enables an efficient analytical method for understanding the routing performance under different partial synchronization levels (quantified by the synchronization radius). Other PS models are left for future work.

a) Given a domain-wise path, we identify all RCs along the path according to Definition 4;

b) We construct the RDPN associated with each RC;

c) We compute the path cost incurred in RDPNs, and prove it is a lower bound of the actual path cost incurred in its original RC;

d) Adding up RDPN path costs and the number of inter-RC connections, we get the lower bound of APC under PS.

Based on this methodology, we next describe the details on how the APC lower bound under PS is derived.

A. The Line Network and Its Randomized Degree-Preserving Network (RDPN)

We first formally define the following terms: (i) the *line network* that generalizes RCs; and (ii) the Randomized Degree-Preserving Network (RDPN) of a line network. These concepts are also used in the analysis of Type-2 Networks.

Definition 8: A line network *with k domains is a special graph generated via the two-layer network model, consists of k domains, where its domain-wise topology is a connected linear graph (i.e., a connected tree where no vertex has degree 3 or more). The domains with inter-domain degree being 1 and 2 in a line network are called* end-domains *and* transit-domains*, respectively.*

Definition 9: For a line network (denoted by $\mathcal{F}$) with k domains and n nodes in each domain, the corresponding Randomized Degree-Preserving Network (RDPN) of $\mathcal{F}$, denoted by $\mathcal{F}_R$, is a randomly generated network with kn nodes such that $\mathcal{F}_R$ and $\mathcal{F}$ have the same degree distribution.

Discussion: Although $\mathcal{F}$ and $\mathcal{F}_R$ have the same degree distribution and the number of nodes, they differ significantly from the perspective of randomness. In particular, $\mathcal{F}$, as a line network, is constrained to certain structural properties, i.e., the domain-wise topology must be a linear graph with k vertices. The RDPN $\mathcal{F}_R$, however, is purely random without such constrains. Thus, let $S_{\mathcal{F}}$ and $S_{\mathcal{F}_R}$ be the sets of all graph instances of $\mathcal{F}$ and $\mathcal{F}_R$, respectively. Then, $S_{\mathcal{F}} \subseteq S_{\mathcal{F}_R}$.

B. Path Cost in RDPN

With the concept of RDPN, we now show the relationships between path costs in the line network and its corresponding RDPN. Specifically, we discuss the minimum path cost between a randomly chosen vertex and a vertex set in a line network and its corresponding RDPN. To this end, we first derive the following theorem.

Theorem 10: For a line network (denoted by $\mathcal{F}$) consisting of k domains sequentially labelled as $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_k$, let $\mathcal{F}_R$ denote the RDPN of $\mathcal{F}$. Let ρ be the average path cost of the minimum-cost path between a random node μ ($\mu \in \mathcal{A}_1$) and a random node set M ($\mu \notin M, M \subseteq \mathcal{A}_k$), and ρ_R the average path cost of the minimum-cost path between a random node μ_R ($\mu_R \in \mathcal{F}_R$) and a random node set M_R ($\mu_R \notin M_R, M_R \subseteq \mathcal{F}_R$) such that $|M| = |M_R|$. Then, $\rho_R \leq \rho$ holds.

With Theorem 10, the APC lower bound under PS can be obtained by combining the path costs of RDPNs of all associated RCs. Therefore, we only need to focus on the computation of path cost in each RDPN. Viewing each RDPN of RCs as a random graph following a certain degree distribution, we reapply the results in Section V. Specifically, the first step of path cost calculation in a random network is to determine the number of 1-hop and 2-hop vertices from a randomly selected vertex. As such, we present the following lemma.

Lemma 11: In the RDPN of a line network $\mathcal{F}$ with k domains and the inter-domain connection parameter β, let ζ_1 and ζ_2 denote the number of vertices that are 1-hop and 2-hop away from a random vertex, respectively. Then, the following holds:$\zeta_1 \approx z_1 + \frac{2\beta(k-1)}{nk}$, $\zeta_2 \approx z_2 + z_1\frac{4\beta(k-1)}{nk}$, if $\beta \ll n$, where z_1 and z_2 are the average number of 1-hop and 2-hop nodes from a randomly chosen node within a domain in $\mathcal{F}$, respectively.

By applying (8), which gives an estimation of the path cost between two random nodes within a domain, and substituting relevant parameters of the RDPN, we can express the path cost between two random nodes in an RDPN as $g(k)$, a function of the number of domains (k) in the RDPN :

$$g(k) = \frac{\log(nk/\zeta_1)}{\log(\zeta_2/\zeta_1)} + 1, \tag{13}$$

where ζ_1 and ζ_2 are defined in Lemma 11. Equation (13) estimates the path cost between two random nodes. However, as discussed in Section V, path construction needs to consider the gateway selection in domains that are not the destination domain. Similarly, in an RC that does not contain the destination node, the constructed path in its RDPN is the minimum-cost path from a random vertex to a random vertex set with the cardinality γ. Therefore, by applying (10) and considering the probability of a random vertex not belonging to the random vertex set, the path cost in an RDPN that does not include the destination node is

$$h(k) = \begin{cases} \frac{nk-\gamma}{nk}\Big(\frac{\log(\frac{nk+1-\gamma}{\zeta_1\gamma})}{\log(\zeta_2/\zeta_1)} + 1\Big) & \text{if } \gamma \leq \frac{nk+1}{\zeta_1+1}, \\ \frac{nk-\gamma}{nk} & \text{otherwise.} \end{cases} \tag{14}$$

where k is the number of domains in this RDPN.

C. APC Lower Bound for PS

For PS of the synchronization radius τ, again, we use a line network with $\Delta+1$ domains to compute the APC lower bound under PS. Such line network is divided into $\eta_1 + 1$ (when $(\Delta+1) \mod \tau' = 0$) or η_1+2 (when $(\Delta+1) \mod \tau' > 0$) RCs, where $\tau' = \min\{\tau, \Delta+1\}$ and $\eta_1 = \lfloor \frac{\Delta+1}{\tau'} \rfloor - 1$. Moreover, the number of domains in the RC that does not include the destination node is always τ', whereas the number of domains in the RC that includes the destination node is $\eta' = (\Delta+1) \mod \tau'$ when $\eta' \neq 0$, or τ' when $\eta' = 0$.

In a line network, the path cost in all RCs, excluding the one with the destination node, is estimated by (14), whereas the path cost in the RC that includes the destination node is estimated by (13). Thus, the APC lower bound under PS is

$$L_{\text{PS}}^{\text{lower}} = \begin{cases} \eta_1(h(\tau')+1) + g(\tau') & \text{if } \eta' = 0, \\ (\eta_1+1)(h(\tau')+1) + g(\eta') & \text{if } \eta' > 0. \end{cases} \tag{15}$$

Hence, when $\eta' = 0$,

$$L_{\text{PS}}^{\text{lower}\ \eta'=0} = \begin{cases} \eta_1\Big(\frac{\xi\log(\frac{n\tau'+1-\gamma}{\zeta_1\gamma})}{\log(\zeta_2/\zeta_1)} + \xi + 1\Big) \\ \quad + \frac{\log(n\tau'/\zeta_1)}{\log(\zeta_2/\zeta_1)} + 1 & \text{if } \gamma \leq \frac{n\tau+1}{\zeta_1+1}, \\ \eta_1(\xi+1) + \frac{\log(n\tau'/\zeta_1)}{\log(\zeta_2/\zeta_1)} + 1 & \text{otherwise,} \end{cases} \tag{16}$$

where $\xi = 1 - \frac{\gamma}{n\tau'}$. When $\eta' > 0$, we have $\eta_2 = (\Delta + 1) \bmod \tau' = \eta'$; therefore,

$$L_{\text{PS}}^{\text{lower}\ \eta'>0} = \begin{cases} (\eta_1+1)\Big(\dfrac{\xi\log(\frac{n\tau'+1-\gamma}{\zeta_1\gamma})}{\log(\zeta_2/\zeta_1)}+\xi+1\Big) \\ +\dfrac{\log(n\eta'/\zeta_1)}{\log(\zeta_2/\zeta_1)}+1 & \text{if } \gamma \le \dfrac{n\tau+1}{\zeta_1+1}, \\ (\eta_1+1)(\xi+1)+\dfrac{\log(n\eta'/\zeta_1)}{\log(\zeta_2/\zeta_1)}+1 & \text{otherwise.} \end{cases} \tag{17}$$

Clearly, $L_{\text{PS}}^{\text{lower}}$ is linear in the number of RCs, and is logarithmic in network structural parameters such as n and γ. This suggests that enlarging the synchronization radius to reduce the number of RCs on the domain-wise path results in near linear reduction in APC.

VII. APC Under CS in Type-1 Networks

For complete synchronization (CS), since all SDN domains are synchronized, controllers can make the global optimal decisions that generate the end-to-end path with minimum path cost. In this regard, we first study whether RCPC can construct such a global optimal path, and then establish the APC expression under CS in Type-1 Networks.

Given two arbitrary nodes v_1 and v_2, suppose the shortest domain-wise path $\mathcal{P}^*$ w.r.t. v_1 and v_2 contains k vertices in the domain-wise topology. If $\mathcal{P}^*$ (selected by RCPC) corresponds to the minimum-cost path between v_1 and v_2, then the APC lower bound under CS can be easily obtained by calculating the APC between two random nodes in the end-domains of a line network consisting of k domains. However, the global minimum-cost path may visit more than k domains to yield the minimum end-to-end path cost. We, therefore, examine how the domain-wise shortest path $\mathcal{P}^*$ is related to the global minimum-cost path between v_1 and v_2 in the following.

Theorem 12: Let $L_k(\beta)$ be the APC between two random nodes in the two end-domains of a line network, which consists of k domains and all inter-domain connections are governed by parameter β. Then, $L_k(\beta) < L_{k+1}(\beta)$ when $k \ge 3$.

Theorem 12 reveals an important property of $L_k(\beta)$, i.e., a longer domain-wise path incurs higher end-to-end path cost if the shortest domain-wise path between two nodes contains at least three vertices. See analysis and discussions on the two uncovered cases ($k = 1, 2$) in the supplementary material.

An implicit assumption for Theorem 12 is that the domain-wise path associated with the constructed path is a *simple path*, i.e., a path without repeated vertices. To show that visiting more domains cannot construct a shorter end-to-end path, we still need to prove that visiting one domain more than once is also disadvantageous. To this end, we define $L'_k(\beta)$ which is the same as $L_k(\beta)$ except that the corresponding domain-wise path contains repeated vertices.

Corollary 13: For the two-layer network model, $L_k(\beta) < L'_{k'}(\beta)$ for $3 \le k \le k'$.

Theorem 12 together with Corollary 13 suggest the following corollary.

Corollary 14: For any source-destination node pair residing in different domains, on average, the optimal path between them traverses the minimum number of domains.

Recall that the average number of domains on the shortest domain-wise path between two random domains is $\Delta + 1 = \frac{\log(m/z'_1)}{\log(z'_2/z'_1)} + 2$. Therefore, we compute the APC under CS based on a domain-wise path traversing $\Delta+1$ domains. Under CS, the path construction in each domain is independent of other domains' structures, thus complicating the mathematical analysis. We, therefore, leverage RDPN of a line network with $\Delta + 1$ domains to estimate the APC for CS, which is a lower bound according to Theorem 10. Thus, reapplying (13) with $\Delta + 1$ as the input, we obtain the APC lower bound for CS, denoted by $L_{\text{CS}}^{\text{lower}}$:

$$L_{\text{CS}}^{\text{lower}} = g(\Delta+1) = \frac{\log\left(\frac{n\log(m/z'_1)}{\zeta_1\log(z'_2/z'_1)} + \frac{2n}{\zeta_1}\right)}{\log(\zeta_2/\zeta_1)} + 1. \tag{18}$$

The expression of $L_{\text{CS}}^{\text{lower}}$ shows a function that bounds the APC under the best-case scenario, i.e., CS, which therefore is also a lower bound under other synchronization scenarios. Since (18) is a logarithmic function of a logarithmic function, it suggests that the routing efficiency can be significant if CS is achieved in the network. Moreover, under CS, (18) is of the form of $\log(n\log(m))$, showing that the number of nodes n has a stronger impact than the number of domains m on the value of $L_{\text{CS}}^{\text{lower}}$, i.e., intra-domain routing is more critical.

VIII. Universal APC Lower Bound

In this section, we present the *Universal APC lower bound*, which provides an estimation of APC under any synchronization levels for both Type-1 and Type-2 Networks. The phrase *lower bound* carries two separate meanings. First, it summarizes the APC obtained for MS and the APC lower bounds obtained for PS and CS in Type-1 Networks. Second, since link preference is at least 1 for Type-2 Networks, this universal lower bound derived for Type-1 Networks also applies to Type-2 Networks.

Theorem 15: Universal APC lower bound: Given the synchronization radius τ, the lower bound of APC (denoted by L^{lower}) in the two-layer network model is

$$L^{lower} = \begin{cases} \dfrac{\eta_1\xi\log(\frac{n\tau'+1-\gamma}{\zeta_1\gamma})}{\log(\zeta_2/\zeta_1)} + \dfrac{\log(n\eta_2/\zeta_1)}{\log(\zeta_2/\zeta_1)} \\ +\eta_1(\xi+1)+1 & \textit{if } \gamma \le \dfrac{n\tau'+1}{\zeta_1+1}, \\ \dfrac{\log(n\eta_2/\zeta_1)}{\log(\zeta_2/\zeta_1)} + \eta_1(\xi+1)+1 & \textit{otherwise}, \end{cases} \tag{19}$$

where $\tau' = \min\{\tau, \Delta+1\}$, $\eta_1 = \lfloor(\Delta+1)/\tau'\rfloor - 1$, $\eta_2 = (\Delta \bmod \tau') + 1$, and $\xi = 1 - \frac{\gamma}{n\tau'}$.

In (19), L^{lower}, requiring the network topologies and synchronization levels as inputs, is a logarithmic function non-increasing with τ. Moreover, when the number of gateways in each domain is sufficiently large (i.e., large γ), the expression of L^{lower} is significantly simplified due to easier inter-domain routing. In addition, the synchronization radius τ, representing different levels of inter-domain synchronizations, is instrumental in determining the APC lower bound L^{lower}. For example, (19) reduces to (12) for MS in Type-1 Networks when $\tau = 1$; (19) reduces to (16) for PS when $\eta' = 0$.

Discussion: Since all link preference levels in Type-2 Networks are at least 1, this universal APC lower bound still holds in Type-2 Networks, thus providing insights into the routing performance under any synchronization and network scenarios. In Sections IX–XII, we derive fine-grained APC expressions under different synchronization scenarios in Type-2 Networks. More importantly, these fine-grained APC expressions can also be applied to Type-1 Networks by setting all edge weights to 1.

IX. APC Under MS in Type-2 Networkss

In this section, we present the APC expression under MS in Type-2 Networks, denoted by $L_{\text{MS}}^{\text{Type-2}}$. Though edges in Type-2 Networks exhibit various edge weights, such weight information is not available to any controllers under MS. Thus, the path construction from the source to the destination is independent of the edge weight distributions. Recall that in our two-layer network model, all intra-domain link weights are modeled as a given i.i.d. r.v., denoted by W, and all inter-domain edges are of weight 1. Hence,

$$\begin{aligned} L_{\text{MS}}^{\text{Type-2}} &= \Delta \cdot (\mathbb{E}[|\mathcal{P}_1|] \cdot \mathbb{E}[W] + 1) + \mathbb{E}[|\mathcal{P}_{\Delta+1}|] \cdot \mathbb{E}[W] + \Delta \\ &= \left(\frac{(n-\gamma) l \Delta}{n} + \frac{\log(n/z_1)}{\log(z_2/z_1)} + 1 \right) \mathbb{E}[W] + \Delta, \end{aligned} \tag{20}$$

where $\mathcal{P}_i$ and l are defined in (1) and (10), respectively. Substituting the expressions of l and Δ into (20), we obtain the full expression of $L_{\text{MS}}^{\text{Type-2}}$ under MS in Type-2 Networks:

$$L_{\text{MS}}^{\text{Type-2}} = \begin{cases} \left(\frac{n-\gamma}{n} \left(\frac{\log(\frac{n+1-\gamma}{z_1 \gamma})}{\log(z_2/z_1)} + 1 \right) \left(\frac{\log(m/z_1')}{\log(z_2'/z_1')} + 1 \right) \right. \\ \left. + \frac{\log(n/z_1)}{\log(z_2/z_1)} + 1 \right) \mathbb{E}[W] \\ + \frac{\log(m/z_1')}{\log(z_2'/z_1')} + 1 & \text{if } \gamma \le \frac{n+1}{z_1+1}, \\ \left(\frac{n-\gamma}{n} \left(\frac{\log(m/z_1')}{\log(z_2'/z_1')} + 1 \right) + \frac{\log(n/z_1)}{\log(z_2/z_1)} + 1 \right) \\ \mathbb{E}[W] + \frac{\log(m/z_1')}{\log(z_2'/z_1')} + 1 & \text{otherwise.} \end{cases} \tag{21}$$

It is verifiable that (21) is same as (12) when $\mathbb{E}[W] = 1$, i.e., the Type-2 Network is reduced to the Type-1 Network.

X. APC Under SS in Type-2 Networks

SS is a special synchronization scenario that only exists in Type-2 Networks. Similar to MS, under SS, no two domains synchronize. To analyze APC under SS, we first introduce a new concept, named *path cost distribution*, as the basis for further analysis. Here is the sketch of our analytical methodology.

Sketch of Analytical Methodology:

a) We compute the distribution of the path cost (in terms of accumulated link preferences) between two random intra-domain nodes, called *intra-domain path cost distribution*;

b) By (1), we need to determine the average cost of $\mathcal{P}_i$ for $i = 1, 2, \ldots, \mu$. Since the total number of RCs is the same as MS, we have that the expected value of μ in (1) is $\Delta + 1$;

c) As all controllers involved in the path construction process follow the same procedure, similar to (7), it suffices to only quantify the average cost of $\mathcal{P}_1$ and $\mathcal{P}_{\Delta+1}$ using the intra-domain distance distribution derived in *a)*.

A. Intra-Domain Path Cost Distribution

In one domain, consider a path with λ links. Let $W_1, W_2, \ldots, W_\lambda$ be i.i.d. r.v. of link weights on this path with the probability density functions (pdf) being $f_{W_1}(x) = f_{W_2}(x) = \ldots = f_{W_\lambda}(x)$. Define r.v. $\mathcal{W}_\lambda := \sum_{i=1}^{\lambda} W_i$ as the accumulated weight on this path. Then the pdf of $\mathcal{W}_\lambda$ is the convolution of the pdfs of $W_1, W_2, \ldots, W_\lambda$, i.e., $f_{\mathcal{W}_\lambda}(x) = f_{W_1}(x) * f_{W_2}(x) * \ldots * f_{W_\lambda}(x)$. By the principle in mixture distribution [29], we still need to know the probability $p_{\mathcal{W}_\lambda}$ that the minimum-cost path between two random nodes contains λ links. By the concept of z_i defined in the analysis of MS (Section V), we know that $p_{\mathcal{W}_\lambda}$ is determined by z_λ, i.e., $p_{\mathcal{W}_\lambda} = z_\lambda / n$ (since link weights are i.i.d.). Note that when $\lambda = 0$, $z_0 = 1$ and the cumulative distribution function (cdf) of $\mathcal{W}_0$ is a unit step function. Let r.v. $\mathcal{D}$ be the minimum path cost (in term of accumulated link weights) between two random nodes in one domain, with the pdf being $f_{\mathcal{D}}(x)$, i.e., intra-domain distance distribution. Then by mixture distribution, $f_{\mathcal{D}}(x)$ can be estimated as follows

$$f_{\mathcal{D}}(x) = \sum_{i=0}^{h_{\max}} p_{\mathcal{W}_i} f_{\mathcal{W}_i}(x) = \sum_{i=0}^{h_{\max}} \frac{z_i}{n} \cdot f_{\mathcal{W}_i}(x), \tag{22}$$

where $h_{\max} := \arg\max_i z_i$ s.t. $\sum_{i=0}^{h_{\max}} z_i \le n$. Hence, the APC between two nodes in one domain $\mathbb{E}[\mathcal{D}]$ can be computed using (22).

B. Domain-Wise Path

Though SS and MS represent different synchronization levels, the corresponding domain-wise paths are exactly the same w.r.t. a pair of source and destination nodes in a given network. Thus, by (5), again, we have that μ in (1) equals $\Delta + 1$. Let $L(\mathcal{P})$ be the end-to-end accumulated link preferential levels (i.e., cost) of path $\mathcal{P}$, which is a random variable. Then the expectation of $L(\mathcal{P})$, i.e., the APC for SS in Type-2 Networks, denoted by $L_{\text{SS}}^{\text{Type-2}}$, is

$$\begin{aligned} L_{\text{SS}}^{\text{Type-2}} &= \mathbb{E}[L(\mathcal{P})] \\ &= \mathbb{E}[L(\mathcal{P}_1) + L(\mathcal{P}_2) + \ldots + L(\mathcal{P}_{\Delta+1})] + \Delta \\ &= \Delta \cdot \mathbb{E}[L(\mathcal{P}_1)] + \mathbb{E}[L(\mathcal{P}_{\Delta+1})] + \Delta \\ &= \Delta \cdot \mathbb{E}[L(\mathcal{P}_1)] + \mathbb{E}[\mathcal{D}] + \Delta. \end{aligned} \tag{23}$$

The reason for the last row in (23) is that $\mathbb{E}[L(\mathcal{P}_{\Delta+1})]$ essentially is the path cost between two nodes in one domain. Thus, it suffices to determine $\mathbb{E}[L(\mathcal{P}_1)]$ next, i.e., the minimum path cost between a random node and the closest gateway in one domain connecting to the next domain on the domain-wise path.

C. Minimum Path Cost Between an Arbitrary Node and Gateways

Section X-A provides the estimation of path cost between two arbitrary nodes in Type-2 Networks. Based on (22), we quantify the path cost between an arbitrary node and the gateway that incurs the minimum path cost, which is formally presented in the following theorem.

Theorem 16: Let r.v. $M^{(\beta)}$ denote the path cost between an arbitrary node and the gateway in the candidate gateway set

(established with parameter β) that incurs the minimum path cost. Then, the pdf of $M^{(\beta)}$ is:

$$f_{M^{(\beta)}}(x)=\begin{cases}(1-F_{\mathcal{D}}(x-1))^{\beta} \\ -(1-F_{\mathcal{D}}(x))^{\beta} & \text{for } x\geq 1,\\ 1-(1-F_{\mathcal{D}}(0))^{\beta} & \text{for } x=0.\end{cases} \tag{24}$$

With Theorem 16, we can derive $\mathbb{E}[L(\mathcal{P}_1)]=\mathbb{E}[M^{(\beta)}]$, using the pdf expression in Theorem 16. Then, substituting (5), $\mathbb{E}[L(\mathcal{P}_1)]$, and $\mathbb{E}[\mathcal{D}]$ into (23), we get the expression of the APC under SS in Type-2 Networks, denoted by $L_{\text{SS}}^{\text{Type-2}}$:

$$L_{\text{SS}}^{\text{Type-2}}=\Delta\cdot\int_{x=0}^{+\infty}xf_{M^{(\beta)}}(x)+1+\int_{x=0}^{+\infty}xf_{\mathcal{D}}(x)+\Delta. \tag{25}$$

Comparing to $L_{\text{MS}}^{\text{Type-2}}$, the expression of $L_{\text{SS}}^{\text{Type-2}}$ is more complicated as we do not impose any constraint on the distributions of link preference levels. Nevertheless, it is verifiable that $L_{\text{SS}}^{\text{Type-2}}$ is smaller than $L_{\text{MS}}^{\text{Type-2}}$, thus bounded by $L_{\text{MS}}^{\text{Type-2}}$.

XI. APC Under PS in Type-2 Networks

To compute the corresponding APC under PS in Type-2 Networks, denoted by $L_{\text{PS}}^{\text{Type-2}}$, we first present the APC w.r.t. two nodes with their shortest domain-wise path containing exactly q vertices, denoted by L_q, in the following theorem.

Theorem 17: Let L_q denote the APC between two arbitrary nodes under PS with synchronization radius τ in Type-2 Networks where there are q domains on the domain-wise path. Then, we have

$$L_q=\begin{cases}(q/\tau-1)\cdot(\mathbb{E}[M_{\tau}^{(\beta)}]+1)+\mathbb{E}[D_{\tau}^{(\beta)}] & \text{if } \theta=0;\\ (\lfloor q/\tau\rfloor-1)\cdot(\mathbb{E}[M_{\tau}^{(\beta)}]+1)+\mathbb{E}[D_{\theta}^{(\beta)}] & \text{if } \theta>0,\end{cases} \tag{26}$$

where $\theta=q \mod \tau$, $M_i^{(\beta)}$ is the r.v. of the minimum path cost incurred in an RC with i non-destination domains, and $D_i^{(\beta)}$ is the r.v. of the minimum path cost incurred in the RC with $i-1$ non-destination domains and the destination domain.

Recall that the probability that two arbitrary nodes with their domain-wise path containing q domains is $z'_{q-1}/(m-1)\approx z'_{q-1}/m$. Therefore, the APC under PS in Type-2 Networks, $L_{\text{PS}}^{\text{Type-2}}$, is

$$L_{\text{PS}}^{\text{Type-2}}=\sum_{q=2}^{h'_{\max}+1}L_q z'_{q-1}/m, \tag{27}$$

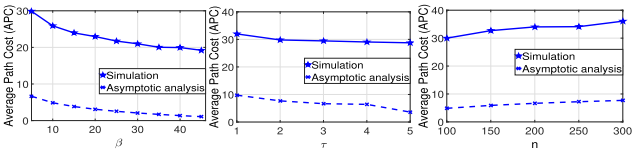
where $h'_{\max}:=\arg\max_i z'_i$ s.t. $1+\sum_{i=1}^{h_{\max}}z'_i\leq m$. The accuracy of $L_{\text{PS}}^{\text{Type-2}}$ is evaluated in Section XIII.

XII. APC Under CS in Type-2 Networks

For complete synchronization (CS), globally optimal routing decisions are made in Type-2 Networks. Here, let $L_k(\beta):=\mathbb{E}[D_k^{(\beta)}]$, where $D_k^{(\beta)}$ is the r.v. of the minimum path cost incurred in the RC with $k-1$ non-destination domains and the destination domain. By close examination of $L_k(\beta)$, some additional conclusions are made in the following corollaries.

Corollary 18: For the two-layer network model, $L_{k+1}(1)-L_k(1)=\mathbb{E}[\mathcal{D}]+1$.

Corollary 19: For the two-layer network model, $\lim_{\beta\to\infty}\left(L_{k+1}(\beta)-L_k(\beta)\right)=1$.



(a) (b) (c)

Fig. 4. Evaluation 1: APCs collected from simulations and asymptotic analysis under varying parameters. (a) APCs for varying β. (b) APCs for varying τ. (c) APCs for varying n.

Note that Theorem 12 and Corollary 13 remain valid for the Type-2 Network scenario, which suggest that for any source-destination node pair residing in different domains, on average, the optimal path between them traverses the minimum number of domains. Therefore, when the shortest domain-wise path between two nodes contains k vertices, we can use $L_k(\beta)$ to approximate the corresponding optimal APC. Thus, let $L_{\text{CS}}^{\text{Type-2}}$ denote the APC under CS in Type-2 Networks. We have

$$L_{\text{CS}}^{\text{Type-2}}\approx\sum_{k=2}^{h'_{\max}+1}L_k(\beta)z'_{k-1}/m=\sum_{k=2}^{h'_{\max}+1}\mathbb{E}[D_k^{(\beta)}]z'_{k-1}/m, \tag{28}$$

where $h'_{\max}$ is defined in (27). Though experiencing high complexity due to global cross-domain routing optimality, $L_{\text{CS}}^{\text{Type-2}}$ is shown in Section XIII to have high accuracy in estimating APC under CS.

XIII. Evaluations

To evaluate our analytical results of distributed SDN for various synchronization scenarios, we conduct two sets of experiments (called *Evaluation 1* and *Evaluation 2*), with different focuses, on network topologies generated from both real and synthetic datasets. In Evaluation 1, we test the accuracy of the asymptotic analysis presented in Theorem 5, which, in its concise form, demonstrates the interplay of different parameters in determining the overall APC. Second, we validate the accuracy of the derived fine-grained expressions for $L_{\text{MS}}^{\text{Type-2}}$, $L_{\text{SS}}^{\text{Type-2}}$, $L_{\text{PS}}^{\text{Type-2}}$, and $L_{\text{CS}}^{\text{Type-2}}$ in Type-2 Networks in Evaluation 2. We compare these theoretical results with the actual APCs collected from the above networks. Based on these evaluation results, we can validate the accuracy of our theoretical results and observe to what extent synchronization levels and network structural properties affect APCs.

A. Network Realizations

1) Network Topologies Based on Real Datasets: To generate network topologies based on real datasets, we need the degree distributions as the input. Specifically, we use the real datasets collected by the University of Oregon Route Views Project (Routeview project) [30], the Rocketfuel project [31], and the CAIDA project [32] for input degree distributions.

Given a specific degree distribution, one graph realization is generated in the following way: We assign each vertex (the total number of vertices is given) a target degree according to the degree distribution. We then select two vertices randomly and add an edge between them; the number of edges added w.r.t. each vertex is then recorded. If the degree target w.r.t. a vertex is met, this vertex will not be selected again to connect with other vertices. This process repeats until all vertices reach their degree targets.

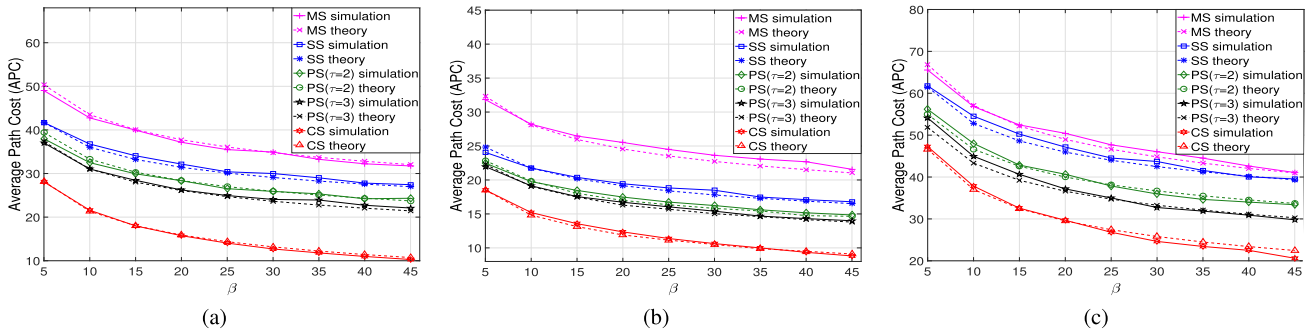


Fig. 5. Evaluation 2: APC under different simulation cases. (a) Case 1: Intra-domain and inter-domain degree distributions are derived from as20000102, Oregon1010331 datasets, respectively. (b) Case 2: Intra-domain and inter-domain degree distributions are derived from Rocketfuel AS-1239, CAIDA AS-27524, respectively. (c) Case 3: Intra-domain and inter-domain degree distributions are derived from BA and ER model, respectively.

2) Network Topologies Based on Synthetic Models: We select Barabási-Albert [33] and Erdös-Rényi [34] models to generate network topologies.

a) Barabási-Albert (BA) model: BA model starts with a small connected graph of a few nodes/edges. Then, we sequentially add new nodes in the following way: For each new node v, we connect v to ϱ existing nodes such that the probability of connecting to node w is proportional to the degree of w. If the number of existing nodes is smaller than ϱ, then v connects to all existing nodes. Vertex degree for the BA model follows a near power-law distribution. BA graphs can be used to model some naturally occurring networks, e.g., social networks.

b) Erdös-Rényi (ER) model: For the ER model, the graph is generated by independently adding an edge between two nodes with a fixed probability p. The result is a purely random topology where all graphs with an equal number of links are equally likely to be selected. Vertex degree under ER follows a binomial distribution.

Then, intra-/inter-domain topologies are generated based on the above network realization methods; see Section XIII-B.2 for details. Next, on top of the generated inter-domain topologies, gateway connections are constructed according to parameter β, and intra-domain links are associated with link preference no less than 1.

Remark: It should be noted that the above network realizations are only for the evaluation purpose. Our developed analytical results are generic and do not require specific topological conditions.

B. Evaluation Settings

1) Evaluation 1: We conduct three experiments in networks with varying gateway connection parameter β, varying synchronization radius τ, and varying number of nodes in each domain n, respectively. The APCs collected from these simulated networks are compared with the predictions made by the asymptotic expressions. Intra-domain degree distributions for three experiments are all derived from Rocketfuel "AS 1239", in which $z_1 = 6.165$, and $z_2 = 41.835$. We configure the domain-wise topologies to have an average domain-wise distance of 10, using statistics collected from CAIDA "AS 27524". Unless otherwise specified, the default parameter settings for three experiments are: $\beta = 5$, $\tau = 2$, and $n = 200$.

2) Evaluation 2: Three evaluation cases are studied to validate the derived fine-grained APC expressions. In particular, Case 1 and 2 use topologies generated based on degree distributions extracted from real network datasets downloaded from Stanford SNAP [35]; their names can be found in captions of Fig. 5. As for Case 3, synthetic data are used where all intra-domain topologies are BA graphs and the inter-domain topology is a ER graph (we pick $p = 0.015$ for ER graphs, $\varrho = 1$ for BA graphs). In all three cases, the distribution of link preference levels (weight) is derived from Rocketfuel topologies, i.e., the intra-domain link preference ranges from 1 to 16 with the expectation and variance being 3.2505 and 4.5779, respectively. For each case, the two-layer network consists of 100 domains, each containing 200 nodes, i.e., $m = 100$ and $n = 200$. In addition, for PS, two special cases, i.e., $\tau = 2$ and $\tau = 3$, are studied to compare against other synchronization scenarios. It should be noted that these settings are determined arbitrarily, as our analytical model does not require the input degree distributions to have any patterns/properties. In addition to the evaluation results presented for the three cases above, we conduct extensive evaluations using other randomly chosen datasets from Rocketfuel and CAIDA, for which similar results are generated. Thus, we select the three cases as representatives; others are omitted to save space and to avoid repetitive results.

C. Evaluation Results

The simulated APCs and the APCs estimated by the asymptotic analysis are presented in Fig. 4(a)-4(c) for Evaluation 1. For Evaluation 2, the simulated APC averaged over all network realizations and source-destination node pairs are reported in Fig. 5(a)-5(c), for the three simulation cases, respectively. It should be noted that the plotted simulated APCs are the average results of multiple network graph realizations sharing the same given network settings, as the simulation results of a single network instance cannot indicate the characteristics of the network with the given set of settings. Specifically, every curve plotted is the average of results of 30 topology realizations with 50 random source-destination pairs (in different domains) per topology realization.

1) Accuracy of the Theoretical Results: The asymptotic analysis is conducted to enable direct and clear observations of the relationships between APC and parameters related to synchronization levels and network structural properties. The asymptotic analysis' ability to reveal these relationships is confirmed in Evaluation 1. From three figures in Fig. 4, we can see that the trends in APC changes with varying parameters are closely captured by the curves obtained using expressions of the asymptotic analysis, as the simulation and analysis curves have common shapes. The presence of the gap between two curves is due to the fact that the asymptotic

analysis is only intended to highlight the relative relationship among different parameters in simple expressions, and thus it is not meant to be employed as an exact estimation. In comparison, the evaluations of various real/synthetic networks in Evaluation 2 demonstrated in Fig. 5 confirm the high accuracy of our fine-grained theoretical results in predicting the performance metric APC in distributed SDN networks. Specifically, the simulation curves can be closely approximated by the theoretical results for all values of β and synchronization scenarios. Moreover, the theoretical results for PS and CS are obtained by an efficient computation method, which reduces calculation complexity. See details in the supplementary material.

2) APC Variations for Different Synchronization Levels and Structural Parameters: Both Fig. 4 and Fig. 5 confirm that the APC in distributed SDN is related to the amount of information available to the controllers, i.e., synchronization levels. As expected, higher synchronization levels are superior in reducing APCs. This can be observed in Fig. 4(b), where APC decreases when the synchronization radius τ gets larger. For Evaluation 2, Fig. 5 shows that APC for CS corresponds to the minimum APC that is achievable in all cases, i.e., a lower bound. By contrast, the results for MS act as an upper bound due to the minimum intra-/inter-domain information availability. Since the APC for MS is expressed as a logarithmic function (21), Fig. 5 shows that even with the minimum synchronization level, APC is still relatively small given the network size ($20,000$ nodes in total) when link preference levels are at least 1. Fig. 5 shows that comparing to MS, the APC reduction for CS can be up to 70%. Moreover, comparing to MS, only intra-domain link preference information is available to SS. Nevertheless, such additional information is able to reduce APC by up to 30%. However, when more synchronized information is available, the reduction in APC starts to degrade (i.e., diminishing return). In particular, for PS, comparing against the case of $\tau = 2$, the APC reduction for $\tau = 3$ is rather small, especially when β is small. This observation is also confirmed by Fig. 4(b) where the most significant decrease in APC takes place when τ changes from 1 to 2. Consequently, it is expected that with the increase of τ, the benefit to cost ratio declines sharply.

In addition, we observe that the network performance improves when β increases. This is intuitive as a large β directly renders higher probability of finding a shorter path, as there exist more inter-domain connections. In fact, Fig. 4(a) and Fig. 4(b) show that on average, increasing β is more effective in reducing APC than increasing the synchronization radius. Furthermore, Fig. 4(a) and Fig. 5 also demonstrate that APC converges to a certain value when β is large, which can be explained by Corollaries 18–19.

Finally, Fig. 4(c) reveals that the size of the network does not have a significant impact on APC. Specifically, when the number of nodes triples from 100 to 300 in each domain, APC only marginally increases by 6. Moreover, given that in Evaluation 1 there are on average 10 domains on the domain-wise path, this gives an average increase of APC by 0.3 in each domain.

In summary, these evaluation results reveal that in distributed SDN, the performance improvement space is only marginal when domains synchronize with other domains in an increasingly larger radius, or when each domain adds more gateways while the number of existing gateways is already large. Such constraints need to be addressed in practical network design and optimizations.

XIV. Conclusions

We have studied the performance of distributed SDN networks for different inter-domain synchronization levels and network structural properties from the analytical perspective. For this goal, a generic network model is first proposed to capture key attributes in distributed SDN. Based on this model, we have developed analytical results (the asymptotic expression, the universal lower bound, and fine-grained expressions of the performance metric-average path cost) to quantify the performance of the constructed paths for four canonical synchronization scenarios in two types of networks where the link preference levels are uniform and non-uniform, respectively. Extensive simulations on both real and synthetic networks show that our developed analytical results exhibit high accuracy while also providing significant insights into the relationship between the network performance and operational trade-offs, which are vital to future network architecture and protocol design.

Acknowledgment

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

References

[1] N. Feamster, J. Rexford, and E. Zegura, "The road to SDN: An intellectual history of programmable networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 2, pp. 87–98, Apr. 2014.

[2] S. Jain *et al.*, "B4: Experience with a globally-deployed software defined WAN," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, pp. 3–14, Aug. 2013.

[3] M. Casado *et al.*, "Ethane: Taking control of the enterprise," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, pp. 1–12, 2007.

[4] S. H. Yeganeh, A. Tootoonchian, and Y. Ganjali, "On scalability of software-defined networking," *IEEE Commun. Mag.*, vol. 51, no. 2, pp. 136–141, Feb. 2013.

[5] V. Kotronis, X. Dimitropoulos, and B. Ager, "Outsourcing the routing control logic: Better Internet routing based on SDN principles," in *Proc. 11th ACM Workshop Hot Topics Netw.*, Oct. 2012, pp. 55–60.

[6] V. Kotronis *et al.*, "Stitching inter-domain paths over IXPs," in *Proc. Symp. SDN Res.*, Mar. 2016, p. 17.

[7] G. Petropoulos, F. Sardis, S. Spirou, and T. Mahmoodi, "Software-defined inter-networking: Enabling coordinated QoS control across the Internet," in *Proc. 23rd Int. Conf. Telecommun. (ICT)*, May 2016, pp. 1–5.

[8] Z. Chen, J. Bi, Y. Fu, Y. Wang, and A. Xu, "MLV: A multi-dimension routing information exchange mechanism for inter-domain SDN," in *Proc. IEEE 23rd Int. Conf. Netw. Protocols (ICNP)*, Nov. 2015, pp. 438–445.

[9] P. Thai and J. C. de Oliveira, "Decoupling policy from routing with software defined interdomain management: Interdomain routing for SDN-based networks," in *Proc. 22nd Int. Conf. Comput. Commun. Netw. (ICCCN)*, Jul./Aug. 2013, pp. 1–6.

[10] Y. Lai and W. S. Lai, "A graph-theoretic model of routing hierarchies," in *Proc. Int. Conf. Adv. Inf. Netw. Appl. Workshops*, May 2009, pp. 1118–1123.

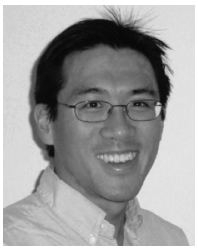
[11] B. Awerbuch and Y. Shavitt, "Topology aggregation for directed graphs," *IEEE/ACM Trans. Netw.*, vol. 9, no. 1, pp. 82–90, Feb. 2001.

[12] B. Awerbuch, Y. Du, and Y. Shavitt, "The effect of network hierarchy structure on performance of ATM PNNI hierarchical routing," *Comput. Commun.*, vol. 23, no. 10, pp. 980–986, May 2000.
[13] R. Cherukuri, D. Dykeman, and M. Goguen, "PNNI draft specification," *ATM Forum*, pp. 94–0471, May 1995.
[14] I. Castineyra, N. Chiappa, and M. Steenstrup, "The nimrod routing architecture," Raytheon BBN Technol., Cambridge, MA, USA, Tech. Rep. RFC 1992, 1996.
[15] B. Awerbuch, Y. Du, B. Khan, and Y. Shavitt, "Routing through networks with hierarchical topology aggregation," *J. High Speed Netw.*, vol. 7, no. 1, pp. 57–73, 1998.
[16] K. Phemius, M. Bouet, and J. Leguay, "Disco: Distributed multi-domain SDN controllers," in *Proc. IEEE Netw. Oper. Manage. Symp. (NOMS)*, May 2014, pp. 1–4.
[17] A. Tootoonchian and Y. Ganjali, "Hyperflow: A distributed control plane for OpenFlow," in *Proc. Internet Netw. Manage. Conf. Res. Enterprise Netw.*, Apr. 2010, pp. 1–6.
[18] P. Berde *et al.*, "ONOS: Towards an open, distributed SDN OS," in *Proc. 3rd Workshop Hot Topics Softw. Defined Netw.*, Aug. 2014, pp. 1–6.
[19] S. H. Yeganeh and Y. Ganjali, "Kandoo: A framework for efficient and scalable offloading of control applications," in *Proc. 1st Workshop Hot Topics Softw. Defined Netw.*, Aug. 2012, pp. 19–24.
[20] M. Canini, P. Kuznetsov, D. Levin, and S. Schmid, "A distributed and robust SDN control plane for transactional network updates," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr./May 2015, pp. 190–198.
[21] S. Schmid and J. Suomela, "Exploiting locality in distributed SDN control," in *Proc. 2nd ACM SIGCOMM Workshop Hot Topics Softw. Defined Netw.*, Aug. 2013, pp. 121–126.
[22] N. Katta, H. Zhang, M. Freedman, and J. Rexford, "Ravana: Controller fault-tolerance in software-defined networking," in *Proc. 1st ACM SIGCOMM Symp. Softw. Defined Netw. Res.*, Jun. 2015, p. 4.
[23] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, Jun. 1998.
[24] R. Milo *et al.*, "Network motifs: Simple building blocks of complex networks," *Science*, vol. 298, no. 5594, pp. 824–827, Oct. 2002.
[25] A.-L. Barabási, "Scale-free networks: A decade and beyond," *Science*, vol. 325, no. 5939, pp. 412–413, Jul. 2009.
[26] Z. Zhang *et al.*, "How better is distributed SDN? An analytical approach," 2017, *arXiv:1712.04161*. [Online]. Available: https://arxiv.org/abs/1712.04161
[27] Z. Zhang *et al.*, "Routing performance in distributed SDN under synchronization constraint," DAIS-ITA Project, New York, NY, USA, Tech. Rep. 2485, 2018. [Online]. Available: https://dais-ita.org/sites/default/files/2485.pdf
[28] M. E. J. Newman, S. H. Strogatz, and D. J. Watts, "Random graphs with arbitrary degree distributions and their applications," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 64, Jul. 2001, Art. no. 026118.
[29] B. S. Everitt, *Mixture Distributions*. Hoboken, NJ, USA: Wiley, 1985.
[30] Univ. Oregon. (2005). *University of Oregon Route Views Project*. [Online]. Available: http://www.routeviews.org/
[31] Univ. Washington. (2002). *Rocketfuel: An Isp Topology Mapping Engine*. [Online]. Available: http://www.cs.washington.edu/research/networking/rocketfuel/interactive/
[32] CAIDA. (2017). *Center for Applied Internet Data Analysis (CAIDA)*. [Online]. Available: http://www.caida.org/home/
[33] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.
[34] P. Erdős and A. Rényi, "On the evolution of random graphs," *Publications Math. Inst. Hung. Acad. Sci.*, vol. 5, no. 1, pp. 17–60, 1960.
[35] Standford Univ. (2017). *Stanford Network Analysis Platform (SNAP)*. [Online]. Available: https://snap.stanford.edu/snap/

Ziyao Zhang received the B.E. degree in electronics and information engineering from Northwestern Polytechnical University, Xi'an, China, in 2015, and the M.Sc. degree in communication and signal processing from Imperial College London, U.K., in 2016, where he is currently pursuing the Ph.D. degree with the Department of Electrical and Electronic Engineering. His research interests include network performance analysis, and reinforcement learning techniques for network optimization.

Liang Ma received the B.Sc. and M.Sc. degrees (Hons.) from the Beijing University of Posts and Telecommunications (BUPT), China, and the Ph.D. degree from Imperial College London, U.K. He is currently a Research Staff Member with the IBM T. J. Watson Research Center, Yorktown Heights, NY, USA. His current research is focusing on network measurement, social networks, graph theory, optimization, and mobility control in wireless networks.

Kin K. Leung received the B.S. degree from The Chinese University of Hong Kong in 1980, and the M.S. and Ph.D. degrees in computer science from the University of California, Los Angeles, CA, USA, in 1982 and 1985, respectively. He was with Bell Labs (AT&T and later Lucent Technologies) from 1986 to 2004. Since then, he has been the Tanaka Chair Professor in internet technology with Imperial College London. His research interests include networking, protocols, optimization, and modeling issues for wireless broadband, sensor, and ad hoc networks.



Franck Le received the Diplome dIngnieur from the cole Nationale Suprieure des Telecommunications de Bretagne, Brest, France, in 2000, and the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, USA, in 2010. He is currently a Research Scientist with the IBM T. J. Watson Research Center, Hawthorne, NY, USA. His research interests primarily include network management and formal verification of network protocols.

Sastry Kompella is currently the Section Head of the Wireless Network Research Section under the Information Technology Division, U.S. Naval Research Laboratory, Washington, DC, USA. His research interests include various aspects of wireless networks, from mobile ad hoc to underwater acoustic networks, with the specific focus towards cognitive and cooperative network optimization and programmable networking.

Leandros Tassiulas is currently the John C. Malone Professor of Electrical Engineering and a member of the Institute for Network Science, Yale University. His research interests are in the field of computer and communication networks with emphasis on fundamental mathematical models and algorithms of complex networks, architectures and protocols of wireless systems, sensor networks, novel internet architectures, and experimental platforms for network research.