

Economic Impact Analysis of Control Plane Architectures in Software Defined Networking (SDN)

Murat Karakus

Department of Computer and Information Science
Indiana University-Purdue University Indianapolis
Indianapolis, IN 46202
Email: mkarakus@iupui.edu

Arjan Durrezi

Department of Computer and Information Science
Indiana University-Purdue University Indianapolis
Indianapolis, IN 46202
Email: adurrezi@iupui.edu

Abstract—Economic and operational facets of networks drive the necessity for significant changes towards fundamentals of networking architectures. Recently, the momentum of programmable networking attempts illustrates the significance of economic aspects of network technologies. Software Defined Networking (SDN) has got the attention of researchers from both academia and industry as a means to decrease network costs and generate revenue for service providers due to features it promises in networking. In this article, we perform an economic analysis of SDN about different popular SDN control plane architectures: Centralized Control Plane (CCP), Distributed Control Plane with Local View (DCP_LV), and Hierarchical Control Plane (HCP) model. In particular, we investigate the economic impact of these control plane architectures about the *unit cost for a service* with bandwidth QoS parameter as well as Total Cost of Ownership (TCO) and network revenue for network owners under different traffic patterns. We characterize the unit cost for a service concerning CAPEX, OPEX, and workload of a network in a certain time period and apply the calculation methods in different SDN control plane models. Our experiments and analysis show that CCP model shows the highest TCO while DCP_LV model results in lowest amount among them. In addition, HCP model shows the lowest unit cost for a service among all models while CCP gives the highest cost for the same service tier. This work aims at being a useful primer to providing insights regarding the economic impact of control plane architectures in SDN for network researchers and owners to plan their investments.

Index Terms—SDN, Economics, Price, Cost, CAPEX, OPEX

I. INTRODUCTION

Software Defined Networking (SDN) [1] has emerged as a new way to architect networks by providing network programmability through exposing network APIs, recently. SDN has got the attention of researchers from both academia and industry as a means to be leveraged to decrease network costs and generate revenue for service providers due to features it promises in networking. SDN paradigm has several key attributes that have an impact on the CAPEX (CAPital EXpenditures) and OPEX (OPerational EXpenditures) equations of a network. Some of the main attributes of SDN are network programmability, hardware and software independence, virtualized software infrastructure, multi-tenancy, and resource pooling.

Verbrugge *et al.* [2] introduce a cost model to identify expenditures of telecom operators. They discuss the relation

between CAPEX and OPEX for telecom networks. The authors mainly split CAPEX in four categories and OPEX into three general parts, respectively. [3] proposes an operational cost model to calculate actual OPEX cost for telecom operators. In their cost model, rented infrastructure (e.g., building and equipment) costs do not contribute to CAPEX but OPEX.

Naudts *et al.* [4] perform a techno-economic analysis of software defined networking for mobile networks in different architecture cases: a classical scenario in which a distributed network is considered, an SDN scenario with centralized network architecture and a network architecture shared based on SDN among several network operators.

Furthermore, Valencia *et al.* [5] present a general qualitative study on how SDN/NFV affects OPEX for service provider networks. The authors summarize that SDN/NFV is expected to reduce service provider OPEX due to consolidating and optimizing the network and surrounding operating model.

In this article, we perform an economic analysis of SDN about different popular SDN control plane architectures: Centralized Control Plane (CCP), Distributed Control Plane with Local View (DCP_LV), and Hierarchical Control Plane (HCP) model. In particular, we investigate the economic impact of these control plane architectures with regard to the *unit cost for a service* with bandwidth QoS parameter as well as Total Cost of Ownership (TCO) and network revenue for network owners under different traffic patterns: (1) 20% (inter-domain) - 80% (intra-domain), 2) 50% (inter-domain) - 50% (intra-domain), and 3) 80% (inter-domain) - 20% (intra-domain). We characterize the unit cost for service concerning CAPEX, OPEX, and workload of a network in a certain period and apply the calculation methods in different SDN control plane models. To the best of our knowledge, this work is the first attempt for such a purpose and, therefore, can be a useful primer to providing insights regarding economic impact of control plane architectures in SDN for network researchers and owners to plan their investments.

In the rest of the paper, in Section II, we study calculation of CAPEX, OPEX, and unit cost for service tier with bandwidth QoS parameter. After discussing the experiments results in Section III, we summarize the paper with concluding remarks in Section IV.

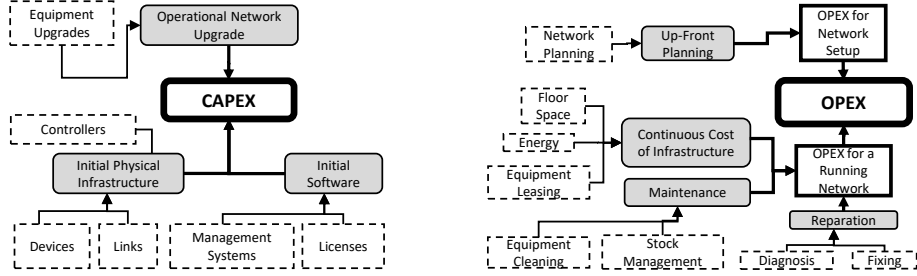


Fig. 1: Expenditure structure of a network. The dashed rectangles represent the input costs and actions to the corresponding expenditure groups represented as rounded rectangles. These are the cost groups that are most-affected by SDN and considered in this study.

II. CAPEX, OPEX AND UNIT SERVICE COST ANALYSIS IN SDN CASE

In this section, we study calculation of unit cost for a service with QoS parameters as well as the determination of CAPEX and OPEX for a network in SDN case. In this study, we only consider bandwidth QoS parameter for service requests along with multiple numerical service tiers. We note that the cost calculation scheme proposed in this paper reflects the minimum cost that a network should charge for the service to compensate its expenditures. Therefore, it neither leads to an economic loss nor a profit for a network. Keeping cost at a higher or lower amount for users depends on network's market strategy and is beyond the scope of this paper.

As seen in Fig. 1, total expenses of a network are mainly classified into two categories [2]: CAPEX and OPEX. Although identifying CAPEX may be easier than OPEX they have an entwined relationship. The dashed rectangles represent the input costs and actions to the corresponding expenditure groups represented as rounded rectangles. These are the cost groups that are most-affected by SDN and considered in this study. The arrows point to the direction of the input. The line rectangles are the two general expenditures categories of the OPEX.

We characterize the unit cost for a service request from a tier as a function of network *CAPEX*, *OPEX*, and *workload* over a certain period. We refer *workload* to service requests of all tiers coming from users/customers to and *satisfied* by the network. The general unit cost framework for a service request with one QoS parameter (bandwidth) from a tier is shown in Eq. 1. This formula implies that the unit cost of a service request from a tier is the ratio of TCO over workload in a given period.

$$C_{bw_j} = f(\mathbb{C}, \mathbb{O}, W) = \begin{cases} \frac{\mathbb{C} + \mathbb{O}}{\sum_{j=1} w_j \cdot bw_j} \cdot |bw_j| & \text{before } \delta \\ \frac{\mathbb{C} + \mathbb{C}_\delta + \mathbb{O} + \mathbb{O}_\delta}{\sum_{j=1} (w_j + w_{\delta_j}) \cdot bw_j} \cdot |bw_j| & \text{after } \delta \end{cases} \quad (1)$$

where bw_j , $|bw_j|$, C_{bw_j} , \mathbb{C} , \mathbb{O} represent type of (i.e. bandwidth) service tier j , numerical value of the service tier bw_j , the unit cost of the service bw_j , CAPEX, and OPEX in a time period (e.g. month, year), respectively. w_j and w_{δ_j} represent the workload and possible additional workload of service bw_j and $W = \sum_{j=1} w_j$ and $W_\delta = \sum_{j=1} w_{\delta_j}$. \mathbb{C}_δ , \mathbb{O}_δ , and W_δ represent possible extra CAPEX, OPEX, and workload, respectively,

incurred after introducing different kinds of changes/upgrades (represented as δ) in the network.

A. CAPEX Calculation

CAPEX is mainly determined by the total of initial physical infrastructure expenses (\mathcal{H}), initial software expenses (\mathcal{S}), and operational network upgrade costs (\mathcal{A}). Therefore, CAPEX is a function of these expenses and can be written as in Eq. 2:

$$\mathbb{C} = f(\mathcal{H}, \mathcal{S}, \mathcal{A}) = \mathcal{H} + \mathcal{S} + \mathcal{A} \quad (2)$$

1) *Initial Physical Infrastructure Expenses (\mathcal{H}):* The initial physical infrastructure (\mathcal{H}) of a network primarily consists of links, network devices, and also hardware, such as a server, that controller installed on. So, \mathcal{H} can be written as in Eq. 3:

$$\mathcal{H} = \sum_{i=1}^{|\mathcal{l}|} C_{l_i} + \sum_{j=1}^{|\mathcal{d}|} C_{d_j} + \sum_{k=1}^{|\mathcal{c}|} C_{c_k} \quad (3)$$

where $|\mathcal{l}|$, $|\mathcal{d}|$, $|\mathcal{c}|$ represent the total number of links, network devices, controller hardware and C_{l_i} , C_{d_j} , C_{c_k} represent the cost of the corresponding link, network device, and controller hardware, respectively.

2) *Initial Software Expenses (\mathcal{S}):* Similarly, the initial software expenses (\mathcal{S}) such as the purchase of management systems, licenses for proprietary controllers can be calculated as in Eq. 4:

$$\mathcal{S} = \sum_{m=1}^{|\mathcal{s}|} C_{s_m} \quad (4)$$

where $|\mathcal{s}|$ and C_{s_i} represent the total number of paid software used in the network and the cost of corresponding software, respectively.

3) *Operational Network Upgrade Costs (\mathcal{A}):* These expenses are incurred from the ongoing network upgrade activities, represented as δ , such as adding/deleting/upgrading controller(s), network device(s), link(s), and so on in the network. These expenses correspond the \mathbb{C}_δ and \mathbb{O}_δ in the Eq. 1. They are considered because, after foregoing modifications, additional CAPEX, therefore additional OPEX, may be incurred in the network. In a single-centralized controller model (CCP), for example, if new data plane devices are added to the network and the controller does not have any more free

ports to connect with the new devices, a controller upgrade (i.e. replacement) may be needed in this case. However, this upgrade process will make the network non-operational for a while, which in turn results in a reduction of network revenue because the network cannot serve upcoming requests, which essentially compose the revenue of an ISP, during network down time (i.e. upgrade period). On the DCP_LV and HCP model cases, current controllers do not stay non-operational during upgrade period because a new controller is just added to the network for new devices. Therefore, there is no revenue loss during this upgrade period.

Substituting Eq. 3 and Eq. 4 in Eq. 2, the new CAPEX equation becomes as in Eq. 5:

$$\begin{aligned} \mathbb{C} &= f(\mathcal{H}, \mathcal{S}, \mathcal{A}) \\ &= \sum_{i=1}^{|\mathcal{I}|} \mathcal{C}_{l_i} + \sum_{j=1}^{|\mathcal{D}|} \mathcal{C}_{d_j} + \sum_{k=1}^{|\mathcal{C}|} \mathcal{C}_{c_k} + \sum_{m=1}^{|\mathcal{S}|} \mathcal{C}_{s_m} + \mathcal{A} \end{aligned} \quad (5)$$

We consider only the initial physical infrastructure (\mathcal{H}), the initial software (\mathcal{S}) expenses, and the operational network upgrade costs (\mathcal{A}) as the main drivers for CAPEX.

B. OPEX Calculation

OPEX is more complicated to calculate than CAPEX because it requires more information about internal network dynamics. However, such information is proprietary and highly hidden by network owners. Main drivers for OPEX are (i) expenses for a running network (\mathcal{T}) and (ii) expenses for network setup (\mathcal{N}) in a network. Therefore, we can state, in general, that OPEX is a function of these expenses and can be written as in Eq. 6:

$$\mathbb{O} = f(\mathcal{T}, \mathcal{N}) = \mathcal{T} + \mathcal{N} \quad (6)$$

1) *OPEX for a Running Network (\mathcal{T}):* Expenses for a running network (\mathcal{T}) are primarily characterized by continuous infrastructure expenses (\mathcal{K}), maintenance expenses (\mathcal{M}), and reparation expenses (\mathcal{R}). Therefore, running network expenses are a function of these foregoing expenses and can be written as in Eq. 7:

$$\begin{aligned} \mathcal{T} &= f(\mathcal{K}, \mathcal{M}, \mathcal{R}) \\ &= \mathcal{K} + \mathcal{M} + \mathcal{R} \end{aligned} \quad (7)$$

a) *Infrastructure (\mathcal{K}), Maintenance (\mathcal{M}), and Reparation (\mathcal{R}) Costs:* It is difficult to simulate these expenses in an artificial simulation environment without real and accurate parameters from networks. In order to model them without data from a real network, we assume that they have a relation with the total messages handled in the network. These messages are internal overhead messages (\mathcal{O}) generated in the network and service requests (\mathcal{W}) (i.e. workload) entering the network. The idea behind this assumption is that the more messages handled in the network result in the more continuous infrastructure expenses (e.g. due to more power and energy consumption) and require the more maintenance and reparation, which brings more expenses as well, in the network. In order to monetize continuous infrastructure, maintenance and reparation expenses for OPEX, we assume that every single message processed in

the network brings $\$ \varepsilon$ cost to the network. Therefore, the total of these expenses becomes as in Eq. 8:

$$\mathcal{K} + \mathcal{M} + \mathcal{R} = \varepsilon \left(\sum_{j=1} w_j + \mathcal{O} \right) = \varepsilon(\mathcal{W} + \mathcal{O}) \quad (8)$$

2) *OPEX for Network Setup (\mathcal{N}):* Network setup expenses (\mathcal{N}) mainly come from up-front planning expenses (\mathcal{X}). Therefore, network setup expenses are function of the up-front costs and can be written as in Eq. 9:

$$\mathcal{N} = f(\mathcal{X}) = \mathcal{X} \quad (9)$$

a) *Up-Front Planning Cost (\mathcal{X}):* In this category, we consider service creation/introduction process in networks. Network administrators may decide to introduce new services for their users for different purposes such as generating new revenue opportunities. However, this process includes various steps to offer a fully operational service: Service Design/Implementation (\mathcal{I}), Service Testing (\mathcal{E}), and Service Tuning Up (\mathcal{T}). These steps are called *Time-to-Market* or *Time-to-Revenue* period. Each step brings its own expenses to the overall network OPEX based on a number of employees (e), employee pay rate (p), and time spent by employees (t) in each step. Therefore, this cost is a function of the factors above and can be written as in Eq. 10:

$$\begin{aligned} \mathcal{X} &= f(e, p, t) \\ &= \mathcal{C}^{\mathcal{I}} + \mathcal{C}^{\mathcal{E}} + \mathcal{C}^{\mathcal{T}} \\ &= \sum_{i=1} e_i^{\mathcal{I}} p_i^{\mathcal{I}} t_i^{\mathcal{I}} + \sum_{k=1} e_k^{\mathcal{E}} p_k^{\mathcal{E}} t_k^{\mathcal{E}} + \sum_{m=1} e_m^{\mathcal{T}} p_m^{\mathcal{T}} t_m^{\mathcal{T}} \end{aligned} \quad (10)$$

where subscript i , j , k , and m represent the corresponding employee IDs, their pay rates, and time spent by each in each corresponding steps.

Finally, substituting Eq. 8, and Eq. 10 in corresponding equations and Eq. 5 and then Eq. 6 in Eq. 1 gives the final unit service cost formula for a service request.

C. Overhead Messages

As we defined in our previous work [6], we refer overhead messages to the messages processed in the control plane by a controller(s). In SDN, when the first packet of a new flow enters a network through a switch, the switch starts a flow initiation request if there is no rule entry matching the packet in switch's flow table. This flow initiation request is then sent to the controller. The controller processes it and installs rule(s) for the flow in switches over the path calculated by the controller. Therefore, a rule-missing flow results in various control messages that are created, processed, and sent by the controller. Also, a controller may deal with some other periodic messages, such as statistics, generated in the network but not related to rule installation process. We categorize these types of messages as an overhead message. In an SDN network with OpenFlow protocol, there are three different types of messages between a controller and data plane device: *Controller-to-Switch*, *Asynchronous*, and *Symmetric* messages. Each of these message types has its sub-types as well.

We specify the overhead in a control plane as the function of these three message types of OpenFlow protocol and synchronization messages among controllers. As explained in [6], we quantify the number of overhead messages as in Eq. 11:

$$O = f(\alpha, \beta, \gamma, \omega) = \sum_{j=1}^{|d|} (\alpha_j + \beta_j + \gamma_j) + \omega \quad (11)$$

where α_j , β_j , γ_j , ω represent the number of *Controller-to-Switch* messages sent from a controller to a data plane device j , the number of *Asynchronous* messages sent from device j to a controller, the number of *Symmetric* messages sent between a controller and device j , and the number of synchronization messages sent from a controller to other controller(s). The total number of overhead messages in a network model such as a single centralized controller, distributed controllers, and hierarchical controllers, can be different due to the way they operate. Therefore, the OPEX in each corresponding model can be different. The details of overhead messages calculation and discussion are omitted to conserve the page limit. For details, we refer the readers to our previous work [6].

III. EVALUATION

In this section, we present some numerical results to provide some insight on SDN's value proposition concerning different network topology models.

A. Experimental Setup

In the experiments, all requests made by users require 1 *Mbps* bandwidth. Therefore, there is only one service tier and $|bw| = 1$ for all requests. We have applied the proposed scheme in three different SDN network models: CCP model, DCP_LV model, and HCP model. We have used Mininet emulator with POX controller for network service request setup. Other numerical calculations have been done using the Matlab platform. Also, we have used a modified version of Waxman random topology generator [7], defined by Erdos-Renyi random graph model, to randomly create the networks while preserving connectivity degrees of nodes (i.e., switches) as three in all switch cases and models. Furthermore, we have used a heuristics, i.e., A*Prune Algorithm, to find a feasible path through the network. A*Prune algorithm can be used to solve finding the K shortest paths subject to multiple constraints (KMCSP). Finally, all experiments were performed on Ubuntu 14.04 in Oracle VirtualBox using an Intel Core i7-5500 system with 12GB RAM.

CCP: CCP setting revolves around a single centralized controller with a global network view.

DCP_LV: This design consists of distributed controllers associated with switches. Each controller manages a sub-network/domain of the whole network. In the DCP_LV design, every controller has its local network view, which is, in turn, abstracted as a logical node to its neighboring controllers. In DCP_LV setting, we divided the whole network into four fully-connected subnetworks at both physical layer and control plane layer levels.

HCP: An HCP design consists of two layers: The lower layer

consisting of local domain controllers and the upper layer in which a master controller resides. The domain controllers manage their domains with full control and are not connected to each other but are connected to the master controller. However, as in the DCP_LV design, a local controller does not maintain a global view of the whole network. Instead, the master controller has a full global view of the entire network by abstracting all domains as logical nodes. As in DCP_LV, we divided the entire network into four fully-connected sub-networks at physical layer level.

We have used the scheme explained in our previous work [6] to get overhead messages concerning some requests in each network setting. In all network models, each switch is connected to one host. In the experiments, we have generated three different traffic patterns: 1) 20% (inter-domain) - 80% (intra-domain), 2) Random, and 3) 80% (inter-domain) - 20% (intra-domain). In each pattern, the source and destination hosts are chosen randomly while preserving the traffic pattern condition. Also, the network settings are mesh-connected in all switch cases in all models. Furthermore, all calculations are based on one year period. Finally, we have averaged 15 runs for each experiment to achieve and exceed 95% statistical significance.

Parameter	Value
$ d $	4, 8, 12, 16, 20, 24, 28, 32
\mathcal{C}_d	\$1000
$ l $	6, 12, 18, 24, 30, 36, 42, 48
\mathcal{C}_l	\$500
$ c $	1, 4, 4 + 1
\mathcal{C}_c	$\frac{\$500 * d }{r}$
$ s $	1, 4, 4 + 1
\mathcal{C}_s	$\frac{\$500 * d }{r}$
ε	$\$10^{-8}$

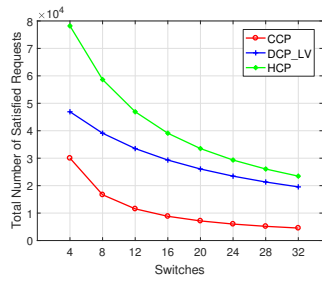
TABLE I: List of parameters and their values used in calculation of CAPEX and OPEX.

Table I lists the parameters and their values used in the calculation of CAPEX and OPEX. These values have been accumulated and determined from the literature review ([2], [8]–[10]) and our discussions. Although we should note that these input numbers may not seem very accurate, they do not impact the nature of the calculation framework since these values are very relative for every network. We have assumed the same cost for all links (\mathcal{C}_l) and devices (\mathcal{C}_d) in all topology models.

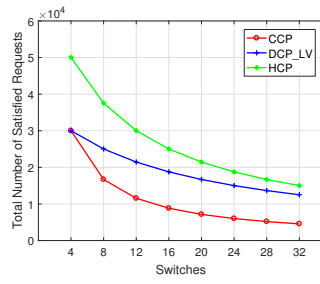
We have also assumed the same cost, which is proportional to the number of network devices ($|d|$) and that of the controller (r), for controller hardware (\mathcal{C}_c) and software (\mathcal{C}_s) in all models. The controller hardware and software number are 1, 4, and 4+1 (depending on the number of domain controllers as well as a master controller in HCP) in CCP, DCP_LV, and HCP models, respectively.

B. Experimental Results

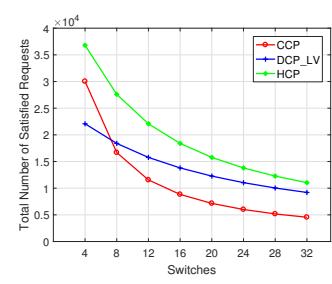
Fig. 2 shows the relation between the total number of satisfied requests with QoS concerning different switch numbers in CCP, DCP_LV, and HCP models under different traffic patterns. In this experiment, we have provided enough network resources in the models so that there is no service request rejection due to network resource limitations. In each switch number case, the corresponding satisfied request numbers represent the maximum number of requests that controller can handle before



(a) 20% (inter-domain) - 80% (intra-domain) Traffic Pattern

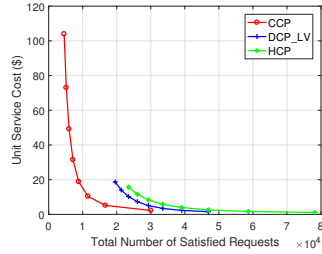


(b) 50% (inter-domain) - 50% (intra-domain) Traffic Pattern

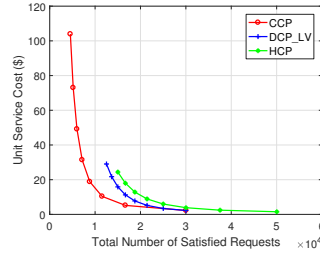


(c) 80% (inter-domain) - 20% (intra-domain) Traffic Pattern

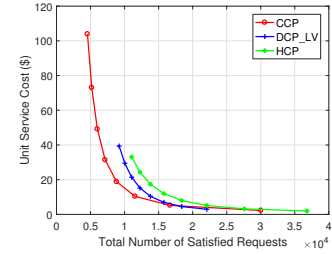
Fig. 2: Total number of satisfied (controller(s)) requests (i.e. Workload) with QoS with respect to different switch numbers in CCP, DCP_LV, and HCP architectures under different traffic patterns.



(a) 20% (inter-domain) - 80% (intra-domain) Traffic Pattern

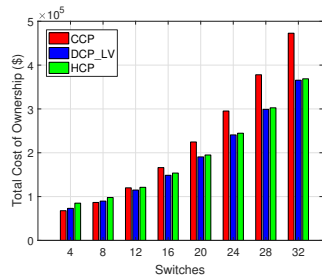


(b) 50% (inter-domain) - 50% (intra-domain) Traffic Pattern

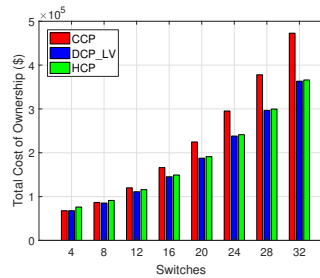


(c) 80% (inter-domain) - 20% (intra-domain) Traffic Pattern

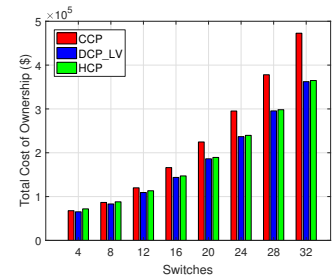
Fig. 3: Unit service cost with respect to a total number of satisfied requests (i.e. Workload) in CCP, DCP_LV, and HCP models under different traffic patterns.



(a) 20% (inter-domain) - 80% (intra-domain) Traffic Pattern

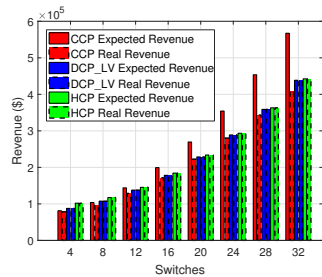


(b) 50% (inter-domain) - 50% (intra-domain) Traffic Pattern

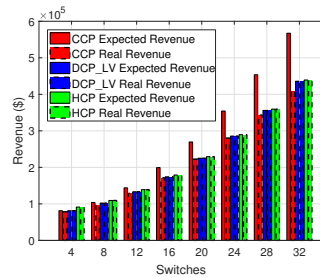


(c) 80% (inter-domain) - 20% (intra-domain) Traffic Pattern

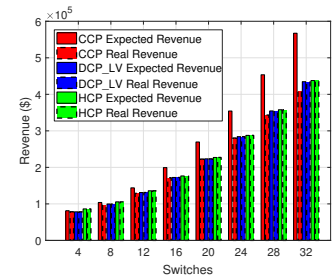
Fig. 4: Total Cost of Ownership (TCO) with respect to different switch numbers in CCP, DCP_LV, and HCP models under different traffic patterns.



(a) 20% (inter-domain) - 80% (intra-domain) Traffic Pattern



(b) 50% (inter-domain) - 50% (intra-domain) Traffic Pattern



(c) 80% (inter-domain) - 20% (intra-domain) Traffic Pattern

Fig. 5: Network revenue before and after a network (i.e. controller upgrade in this case) upgrade with respect to different switch numbers in CCP, DCP_LV, and HCP models under different traffic patterns.

it rejects a request. This rejection happens due to controller message handling capacity. In each traffic pattern, the total number of satisfied requests in all models show reduction while switch number increases because the network paths that are set up by the controllers become longer. Therefore, controllers need to handle more overhead messages per path setup. In CCP, total satisfied requests show similar numbers in each traffic pattern because each path setup is dealt within the same model. However, in DCP_LV and HCP, total satisfied request number for 20% (inter-domain) - 80% (intra-domain) Traffic Pattern (see Fig. 2a) is more than 80% (inter-domain) - 20% (intra-domain) Traffic Pattern (see Fig. 2c) case because most of the requests are local (i.e. intra-domain) and requires shorter paths, therefore, less overhead is generated.

Fig 3 shows the relation between the unit service cost concerning a total number of satisfied requests (i.e., workload) in CCP, DCP_LV, and HCP models under different traffic patterns. These total satisfied request numbers are based on different switch numbers, i.e., each point on the plots correspond to the associated switch numbers. Based on a one-to-one comparison of each point, while CCP shows the highest unit service cost among all models, DCP_LV gives higher unit service cost than HCP. As the total number of satisfied requests increase the unit service cost decreases as the Eq. 1 implies. They all reach a stable pattern while workload increases. Furthermore, all models result in lower unit service costs as the traffic becomes more local (i.e., intra-domain).

Fig. 4 shows the relation between the TCO concerning different switch numbers in CCP, DCP_LV, and HCP models under different traffic patterns. In this experiment, when controllers reach their maximum throughput point, we have added more switches to the network and restarted sending traffic. We have assumed that a controller's port number is the same as the number of switches it manages. Therefore, when new switches are added to the network, the current controller either needs to be upgraded (i.e., replaced with a new one with enough ports) or a new controller needs to be added, depending on the model, in the network. Since CCP model has only one controller, then the current controller needs to be upgraded. However, this upgrade brings additive CAPEX since the previous controller is not used anymore. On the other hand, in DCP_LV and HCP models, current controllers can still be used while adding new controllers in the network. This brings fewer expenses compared to CCP model case. This fact is the reason for a fast increase in TCO of CCP model under all traffic types, which is also an example for C_{Δ_s} discussed in Subsection II-A3. Furthermore, TCO of HCP models is more than DCP_LV model because HCP model handles more workload (W) than DCP_LV model, see Eq. 8, and there is extra master controller cost in HCP model.

Fig 5 shows the network revenue before and after a controller upgrade concerning different switch numbers in CCP, DCP_LV, and HCP models under different traffic patterns. In this experiment, we have simulated the revenue reaction explained in the example given in the Subsection II-A3. We have assumed and used upgrade periods that are proportional to the number of switches in each case to make it more

realistic because we do not have real data for it. As seen in the figure, the network loses some expected revenue during this upgrade period in CCP case since it cannot serve to new upcoming flows due to the nonoperational controller. On the DCP_LV and HCP model cases, current controllers do not stay non-operational. Therefore, there is no change in network revenue. Also, networks make more revenue in case of HCP than DCP_LV case than CCP case as well as the traffic becomes more local.

IV. CONCLUSIONS

In this article, we performed an economic analysis of SDN about different popular SDN control plane architectures. In particular, we have investigated the economic impact of these control plane architectures about the unit cost for a service with bandwidth QoS parameter as well as Total Cost of Ownership (TCO) and network revenue for network owners under different traffic patterns. We have characterized the unit cost of service with respect to CAPEX, OPEX, and workload of a network in a certain time period and apply the calculation methods in different SDN control plane models. Our experiments and analysis have shown that CCP model shows the highest Total Cost of Ownership (TCO) while DCP_LV model results in lowest amount among them. In addition, HCP model shows the lowest unit cost for a service among all models while CCP gives the highest cost for the same service tier.

REFERENCES

- [1] M. Karakus and A. Durresi, "A Survey: Control Plane Scalability Issues and Approaches in Software-Defined Networking (SDN)," *Computer Networks*, vol. 112, pp. 279 – 293, 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S138912861630411X>
- [2] S. Verbrugge, D. Colle, M. Pickavet, P. Demeester, S. Pasqualini, A. Iselt, A. Kirstädter, R. Hülsermann, F.-J. Westphal, and M. Jäger, "Methodology and input availability parameters for calculating opex and capex costs for realistic network scenarios," *J. Opt. Netw.*, vol. 5, no. 6, pp. 509–520, Jun 2006.
- [3] S. Verbrugge, S. Pasqualini, F. J. Westphal, M. Jager, A. Iselt, A. Kirstädter, R. Chahine, D. Colle, M. Pickavet, and P. Demeester, "Modeling Operational Expenditures for Telecom Operators," in *Conference on Optical Network Design and Modeling*, 2005., 2005.
- [4] B. Naudts, M. Kind, F. J. Westphal, S. Verbrugge, D. Colle, and M. Pickavet, "Techno-economic Analysis of Software Defined Networking as Architecture for the Virtualization of a Mobile Network," in *2012 European Workshop on Software Defined Networking*, 2012.
- [5] E. Hernandez-Valencia, S. Izzo, and B. Polonsky, "How will NFV/SDN Transform Service Provider Opex?" *IEEE Network*, 2015.
- [6] M. Karakus and A. Durresi, "A Scalability Metric for Control Planes in Software Defined Networks (SDNs)," in *2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*, March 2016, pp. 282–289.
- [7] B. M. Waxman, "Routing of multipoint connections," *IEEE Journal on Selected Areas in Communications*, vol. 6, no. 9, pp. 1617–1622, Dec 1988.
- [8] N. Zhang and H. Hmminen, "Cost efficiency of sdn in lte-based mobile networks: Case finland," in *2015 International Conference and Workshops on Networked Systems (NetSys)*, March 2015, pp. 1–5.
- [9] C. Bouras, P. Ntarzanos, and A. Papazois, "Cost modeling for sdn/nfv based mobile 5g networks," in *2016 8th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, Oct 2016, pp. 56–61.
- [10] J. Lhteenmki, H. Hmminen, N. Zhang, and M. Swan, "Cost modeling of a network service provider cloud platform," in *2016 IEEE International Conference on Cloud Engineering Workshop (IC2EW)*, April 2016, pp. 148–153.