

# Economic Viability of Software Defined Networking (SDN)

Murat Karakus<sup>a,\*</sup>, Arjan Durreresi<sup>a</sup>

<sup>a</sup>*Department of Computer and Information Science  
Indiana University Purdue University Indianapolis  
Indianapolis, IN 46202, USA*

---

## Abstract

Economical 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 investigate how programmable network architectures, i.e. SDN technology, affect the network economics compared to traditional network architectures, i.e. MPLS technology. We define two metrics, *Unit Service Cost Scalability* and *Cost-to-Service*, to evaluate how SDN architecture performs compared to MPLS architecture. Also, we present mathematical models to calculate certain cost parts of a network. In addition, we compare different popular SDN control plane models, Centralized Control Plane (CCP), Distributed Control Plane (DCP), and Hierarchical Control Plane (HCP), to understand the economic impact of them with regards to the defined metrics. We use video traffic with different patterns for the comparison. This work aims at being a useful primer to providing insights regarding which technology and control plane model are appropriate for a specific service, i.e. video, for network owners to plan their investments.

*Keywords:* SDN, Economics, Network Programmability, Cost, CAPEX, OPEX

---

## 1. Introduction

Traditional networks are forcing their limits to meet the needs of today's users, enterprises and carriers due to their limited capabilities. Increasing cloud services, server virtualization, the sharp growth of mobility, and content-like video have led researchers to rethink today's network architectures. In traditional architectures, network devices and appliances are complex and challenging for (re)configuration and (re)installation since they require highly skilled persons. Adding or moving a device from a network brings extra costs. It is also time-consuming because IT people need to deal with multiple switches, routers, etc. and update ACLs, VLANs and some other mechanisms. Furthermore, as business demands or user requirements increase day by day, application developers, carriers, and enterprises need to delve into evolving new services and facilities. However, the software and the hardware in network equipment are vertically integrated and proprietary. Therefore, vendor dependency is an obstacle deterring them from developing new networking applications and services for their networks due to slow equipment production cycle, long protocol standardization process, application testing, and deployment. As a result, dynamicity in networking becomes an inevitable and crucial feature to meet the needs of today's end users.

---

\*Corresponding author

*Email addresses:* mkarakus@iupui.edu (Murat Karakus), adurreresi@iupui.edu (Arjan Durreresi)

### 1.1. Network Programmability

Network programmability has its own benefits in terms of both network management and network revenues. Today, networks are mostly configured manually for provisioning, which requires considerable time, effort, and expertise. This process is error-prone and can lead to many mistakes due to human interventions. Also, manual repetition of the same configuration across a large number of heterogeneous devices inevitably increase the possibility of having some errors in someplace. Network programmability provides automation that is a cure for large-scale repetition of common tasks that will result in saving time and making the network more error-free and available by reducing the Mean Time Between Mistakes (MTBM). Automating these tasks can drive the bottom line down by reducing the OPERational EXPenditures (OPEX) required to run the network as well.

Moreover, programmable networks help increase the network utilization. Network traffic often follows daily, hourly, weekly, and seasonal patterns. These patterns may require rapidly moving of traffic around to less utilized links. Since this rapid modification to the network is not trivial in the traditional architectures, network administrators often over provisions their network resources, which increases the CAPital EXPenditures (CAPEX) in the network. For example, bandwidth is selected to support the highest traffic across a single link. During non-peak times, this bandwidth is not used and is, therefore, a cost that has no return on investment is incurred. In a programmable network, traffic can be engineered in near real time to adjust the increasing load on the network. This dynamic nature prevents network operators from unnecessary network capacity increases and, thus, produces CAPEX savings.

### 1.2. Paper Organization

In this article, we investigate how programmable networking, i.e. SDN technology ([1, 2]), affects the network economics compared to traditional networking, i.e. MPLS technology. We choose the MPLS technology as the traditional architecture for comparison because it has the concept of flows similar to SDN architecture although MPLS-based flows (FEC+LSP) are not as generic and flexible as the SDN flow abstraction in terms of the match definitions and forwarding actions. Also, MPLS is the most implemented and accepted architecture by service providers to provide QoS among others named earlier. To this end, we define two metrics *Unit Service Cost Scalability* and *Cost-to-Service* to evaluate how SDN architecture performs compared to MPLS architecture. Also, we present mathematical models to calculate certain cost parts of a network. In addition, we compare different popular SDN control plane models, Centralized Control Plane (CCP), Distributed Control Plane (DCP), and Hierarchical Control Plane (HCP), to understand the economic impact of them with regards to the defined metrics. We use video, as the service, with 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), for the comparison due to its QoS requirements and the facts explained earlier. This work aims at being a useful primer to providing insights regarding which technology and control plane model are appropriate for a specific service, i.e. video, for network owners to plan their investments. To the best of our knowledge, this study is the first attempt to explore SDN with different control plane models and MPLS technologies in terms of their service cost economic viability. Furthermore, we should note that the economic analysis framework that we propose in this study is independent of an architecture. SDN and MPLS architectures are the ones we have considered in this study. It can be applied to any architecture in order to evaluate its economic promises.

In the rest of the paper, Section 2 presents a snapshot of papers that study cost models and expenses for network providers. While 3 gives an overview of SDN technology, Section 4 discusses value proposition of SDN over network expenditures. Section 5 presents MPLS in general, and Section 6 discusses the economics of MPLS technology. Section 7 describes drivers for CAPEX and OPEX in a network. In Section 8, we analyze network costs and present two metrics: *Unit Service Cost Scalability* and *Cost-to-Service*. After discussing the experiment results in Section 9, we summarize the paper with concluding remarks in Section 10.

## 2. Related Work

The studies related to network expenditures can be mainly classified into three general categories: 1) identification of CAPEX and OPEX for a network in general, 2) economic analysis of mobile networks, and 3) cost determination using different methods for a network.

The first category of network expenditures related studies considers identification of CAPEX and OPEX for a network in general. Verbrugge *et al.* [3] 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 into four categories and OPEX into three general parts with various subparts, respectively. They also discuss activity-based descriptions of identified operational processes for telecom networks. [4] proposes an operational cost model to calculate actual OPEX cost for telecom operators. Their identified CAPEX and OPEX parts are the same. In their cost model, rented infrastructure (e.g. building and equipment) costs do not contribute to CAPEX but OPEX. However, Swisscom *et al.* [5] state that all costs related to infrastructure should be considered as CAPEX. In [6], the authors study the impact of the resilience schemes on both CAPEX and OPEX for a network operator using process-based approach. The authors in [7, 8] evaluate how GMPLS technologies impact network operators' processes using a quantitative technique, and provide a calculation of the expected OPEX savings.

The second category of network expenses related studies mostly concerns with economic analysis of mobile networks. Naudts *et al.* [9] perform a techno-economic analysis of SDN for mobile networks in different architecture cases: a classic 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 through FlowVisor [10] controller. The authors state that the benefits of SDN outweigh its extra costs according to the quantitative analysis conducted. In [11], the authors present a general qualitative study on how SDN/NFV (Network Function Virtualization) 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. Zhang and Hammainen [12] also study the SDN impact on network expenditures using a Finnish LTE reference network. Their findings show that SDN reduces the network related annual CAPEX by 7.72% and OPEX by 0.31% compared to non-SDN LTE. Knoll [13, 14] uses a model called "Life-cycle Cost (LCC)" to investigate a detailed techno-economic structure of SDN/NFV based mobile networks. The LCC-based model considers the life-cycle phases of a network from the idea to set up a certain product or service, followed by the installation and operation of the network up to the decommissioning of the equipment. Bouras *et al.* [15] also present a cost model to estimate the CAPEX, OPEX, and TCO (Total Cost of Ownership) of SDN/NFV based mobile 5G networks and compare them with a traditional network architecture. In [16], the authors study the determination of unit cost for a service with QoS parameters over various SDN network topologies. They characterize the unit cost for a service with respect to CAPEX, OPEX, and workload of a network in a certain time period.

Finally, the last category consists of the studies concerned about cost determination using different methods for a network. Kwak *et al.* [17] propose a cost estimation method based on ABC (Activity-Based Costing) procedure to reduce network OPEX and, thus, the general cost. The authors also present several useful use cases of the suggested cost estimation method and describe expected effects. [18] proposes a method to analyze the implementation of network design optimization by validating network cost models. The authors' key validation technique is the balance between the total network cost calculated from traffic and which of summing the cost of modules. In [19], the authors use a cost mode named "Total Element based Long Run Incremental Cost (TELRIC)" to calculate and distribute the cost of a network element according to the usage that each user type makes of in Next Generation Networks (NGNs). In [20], the authors present a technique for estimating and comparing the costs of different data center architectures and analyze costs of these architectures based on the proposed methodology. Casier *et al.* [21] propose a cost allocation model based on a combination of resource usage and peak capacity to understand how the different CAPEX and OPEX cost parts for a service

provider can be allocated to the services. Bailey [22] discusses the economic realities of migrating to the cloud and virtualized networking. His conclusion is that once the network was a service, now the data center is the new network for the purpose of economic modeling and business insights.

As can be seen from the studies presented above, this study differs from them in a way that we define and present two metrics to investigate how programmable networking, i.e. SDN technology, impact the unit cost for a service and service introduction cost compared to traditional networking, i.e. MPLS technology. We calculate CAPEX and OPEX as a means to use in calculation of unit service cost metric. We also include network performance with regards to total number of satisfied requests (i.e. workload) in the model proposed unlike aforementioned studies.

### 3. SDN Overview

SDN allows managing flows in a finer-granular way based on more attributes of packet headers by means of a Controller-Data Plane Interface (C-DPI) such as OpenFlow protocol [23]. As shown in Fig. 1, Open Networking Foundation (ONF)<sup>1</sup> vertically splits SDN architecture into three main planes [24]:

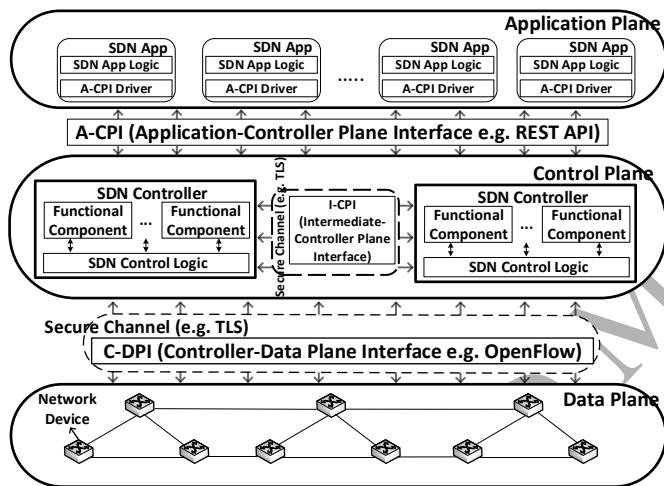


Figure 1: An Overview of an SDN Control Plane. Main components in a control plane of an SDN network are a controller(s) and interfaces (e.g. A-CPI, C-DPI, and I-CPI).

i.e. A-CPI).

**Application Plane:** An SDN application plane consists of one or more end-user applications (security, visualization, etc.) that interact with a controller(s) to utilize an abstract view of the network for their internal decision making process. These applications communicate with a controller(s) via an open A-CPI (e.g. REST API). An SDN application comprises an SDN App Logic and A-CPI Driver.

Fig. 2 illustrates some of the popular control plane models used in SDN technology: CCP, DCP and HCP. These models have their own intrinsic advantages and disadvantages with respect to the different concepts such as control plane scalability [26], resiliency [27], better manageability and so on. These are the control plane models in SDN that we have considered while conducting analysis to understand their impact over network economics. However, we should note that they are not the only control plane models in SDN.

**Centralized (Single) Controller Plane Model (CCP).** This type of settings revolves around a single central controller [28, 29] with a global network view. The model is simple and it is easy to manage the network. This design may meet the needs of small to mid-size networks.

**Data Plane:** The data plane is the bottom plane and consists of network devices such as routers, physical/virtual switches, access points, etc. These devices are accessible and managed through C-DPIs by SDN controller(s). The network elements and controller(s) may communicate through secure connections such as the TLS connection.

**Control Plane:** An SDN control plane comprises a set of software-based SDN controller(s) to provide control functionality in order to supervise the network forwarding behavior through C-DPI. It has interfaces to enable communication among controllers in a control plane (Intermediate-Controller Plane Interface, i.e. I-CPI [25], optionally secured using the TLS), between controllers and network devices (C-DPI), and also between controllers and applications (Application-Controller Plane Interface,

<sup>1</sup><https://www.opennetworking.org/>

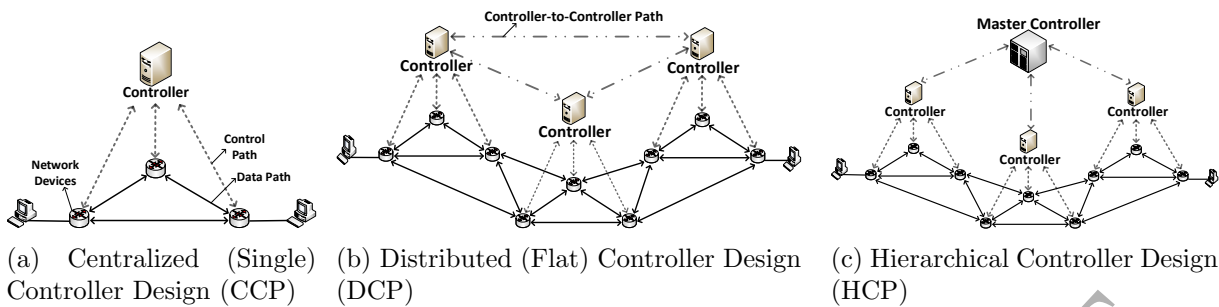


Figure 2: An overview of popular SDN control plane models. The two-sided solid, dashed, and dashed-dotted arrows represent two-way data path among network devices, control path between controller and data plane devices, and controller-to-controller path among controllers, respectively. In 2a (CCP), there is one main controller with global network state. In 2b (DCP), every controller is responsible for different sites/parts of the network(s) with partially shared network view. In 2c (HCP), there are levels where controllers reside and are responsible for different sites (sub-domains) and a master controller on top with global network view for global applications like routing.

*Distributed Controller Plane Model (DCP)*. This model [30, 31] consists of distributed controllers associated with switches. Each controller manages a subnetwork/domain of the whole network and has its own local network view, which is, in turn, abstracted as a logical node to its neighboring controllers. These controllers communicate with each other (i.e. connected neighbors) when they receive a packet destined out of its domain in order to set up an end-to-end path.

*Hierarchical Controller Plane Model (HCP)*. An HCP design [32, 33] consists of two control plane layers minimum: The lower-layer, consisting of local domain controllers, and the upper-layer where another controller, usually called “master”, resides. The domain controllers manage their own domains with full control and are not connected to each other but the master controller. However, 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.

## 4. SDN Value Proposition

In this section, we discuss the values of SDN architecture that are results of programmable networking. It is important for a network owner to identify and understand these values while evaluating economic position of an architecture before investing money on the architecture.

### 4.1. SDN Impacts over CAPEX

Virtualization and flexible placement of network functions, fine-grain network traffic optimization, and efficient resource utilization through orchestration associated with SDN provide an intuitive indication of potential CAPEX reduction [11]. SDN can influence CAPEX of a network in different ways. Some of the key factors that affect the potential CAPEX changes include:

- **Simpler Network Devices:** In an SDN network, each network device will be simpler because complex features such as proprietary software implemented by vendors are not needed. The devices will be simpler and cheaper white-boxes.
- **Extra Components:** In an SDN scenario, the network will have other elements that a traditional network does not have. These elements include a controller(s) hardware and controller software licenses (if not used an open-source software).
- **Network Dimensioning:** Since network controller(s) can have global network view in SDN case, this leads to better network resource utilization by means of some methods such as load balancing. Therefore, there may not be a need for overprovisioning the network, which can reduce the capital expenditures.

#### 4.2. SDN Impacts over OPEX

[34] reports that OPEX for service providers are up to 5 times higher than CAPEX according to the financial analysis conducted. This increased cost leads service providers to pay more attention to the OPEX part of their expenses. SDN promises to lower some of the main OPEX components for service providers using its various features:

- **Energy-Related Costs:** In an SDN network, switches will not have an embedded control plane, which consumes the most of the total energy that a switch needs. Also, since SDN allows more efficient traffic optimization over the network devices, this reduces the total number of needed devices.
- **Maintenance Costs:** SDN creates a homogeneous network environment on hardware and software. There is no case in which different vendor-dependent devices need to be managed and maintained independently.
- **Reparation Costs:** SDN provides better testing opportunities, identifying bugs, and so on before reaching out the actual production traffic. Software related issues can be remotely fixed without touching network devices since these devices are simple and SDN is software-centric.
- **Service Provisioning Costs:** Service provisioning cost in SDN scenario is expected to be lower due to automated configuration of network devices, less personnel need for network tasks due to automation, reduced manual configuration and so on.

### 5. MPLS Overview

Although the original idea behind the development of MPLS was to facilitate fast packet switching, currently its main goal is to support traffic engineering and provide quality of service. The goal of traffic engineering is to facilitate efficient and reliable network operations and at the same time optimize the utilization of network resources. Routers that support MPLS are known as *Label Switching Routers (LSRs)*. When an LSR identifies the *Forwarding Equivalent Class (FEC)*<sup>2</sup> associated with the packet, it selects a label from a pool of free labels, and makes an entry in a table referred to as the *Label Forward Information Base (LFIB)* [35]. This table contains information regarding the incoming and outgoing labels associated with an FEC and the output interface, i.e. the FEC's next hop router. The LSR also saves the label in its FIB in the entry associated with the FEC. A Label Switched Path (LSP) is referred to as a path from the ingress node to the egress node of an MPLS domain followed by packets with the same label.

#### 5.1. MPLS-TE

MPLS-TE model mainly consists of Path Management, Traffic Assignment, Network State Information Dissemination, and Network Management components [36]. Path Management is a mechanism by which MPLS network manages the packet forwarding, which includes choosing the right path for the specific packet, maintaining the existing path and finding new paths due to failure or addition of links. Traffic Assignment performs the assignment of traffic to the established tunnel by path management to do load distribution. Network State Information Dissemination component conducts the advertisement of the topological and state information of the network to all the nodes of the MPLS network. The final component, Network Management, is responsible for configuration and fault management functions.

---

<sup>2</sup>A FEC is a class which comprises the group of packets which are treated in the same manner by the LSR.

## 5.2. Signaling Protocols in MPLS-TE

There are two main signaling protocols that support TE in MPLS networks: *Constraint-based Label Distribution Protocol (CR-LDP)* [37] and *Resource Reservation Protocol-Traffic Engineering (RSVP-TE)* [38].

CR-LDP is the extension of the signaling protocol LDP. It is extended from LDP with the additional support to explicitly route the information about the traffic parameters for the reservation of the resources along the LSPs. CR-LDP is a hard state protocol as it sends the signaling messages only once without refreshing.

RSVP uses the direct routes to set up the *Constraint-based Routed LSPs (CR-LSPs)* (also known as ER-LSPs). It uses UDP for resource reservation and label allocation. RSVP supports Integrated Service (IntServ) model of QoS. The TE supported RSVP, which is an extended version of RSVP and known as RSVP-TE, supports loop detection, periodization, reordering of a path, and strict/loose CR-LSPs.

## 6. Economic Position of MPLS

Service providers continue to use MPLS to transparently carry legacy services as part of their evolutionary service strategy. This factor results in an opportunity for both service providers and customers to exploit MPLS as a new service opportunity. In terms of the service provider, MPLS can speed the service delivery window for customers who subscribe to these services. For the customers, MPLS can reduce WAN costs or offer services internally to various departments or subsidiaries. On the other hand, enterprise organizations are using MPLS to develop virtualized architectures to scale WAN/LAN, campus, and data center resources [39].

MPLS attracts service providers as a business opportunity due to cost savings and revenue generation factors it can provide. This business opportunity is ultimately translated to deploying a global ubiquitous network and to developing services that are based on this technology. Moreover, MPLS-based faster TTM service delivery windows of new services are critical for the service providers. In addition, MPLS can provide any-to-any service constructs that customers can utilize.

Furthermore, service providers expect operational savings by deploying new IP/MPLS-based services. Applications such as voice, once implemented in circuit-based networks, are perceived by service providers to be less expensive to deploy over IP. These cost savings come from the opportunity to consolidate multiple infrastructures (PSTN for voice and video, and data over IP). The consolidation can be facilitated by such mechanisms as a differentiated class of service (CoS).

Finally, controlling costs while supporting existing and new services, and transitioning multiple networks to a consolidated packet-based service-oriented technology, such as MPLS, are indeed requirements for service providers.

## 7. CAPEX and OPEX for a Network

Recent studies show that it is a non-trivial task to clearly define the cost structure of a network due to its complicated nature and many contributing factors involved. As seen in Fig. 3, total expenses of a network are mainly classified into two categories [3]: 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. The gray ones represent the cost groups that are considered in this study. The arrows point to the direction of the input. The solid rectangles are the three general expenditures categories of the OPEX.

### 7.1. CAPEX for a Network

CAPEX consists of expenses that are made for acquiring or upgrading fixed, physical and non-consumable assets of a company. They are needed to expand the services to the customers. CAPEX for a network is usually split into five major parts. They include:

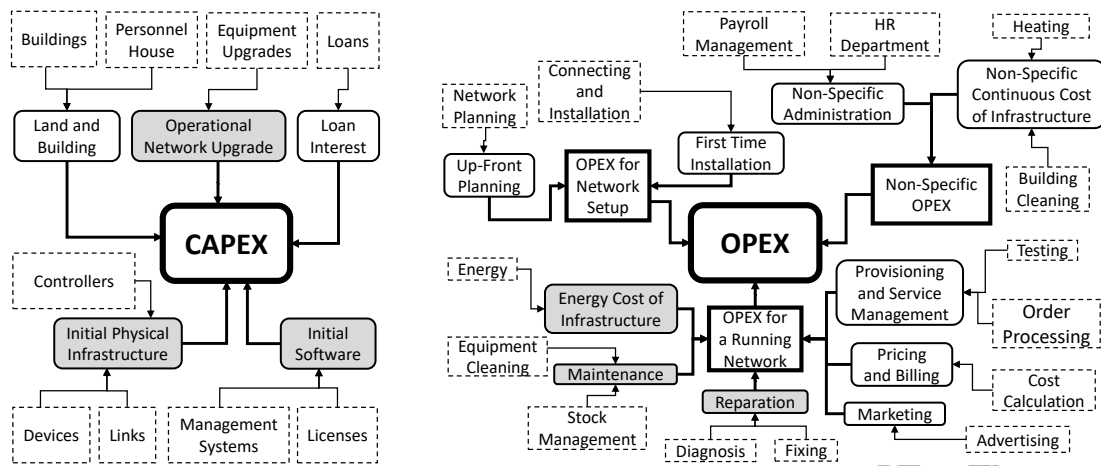


Figure 3: Expenditure structure of a network. The dashed rectangles represent the input costs and actions to the corresponding expenditure groups represented as rounded rectangles. The gray rounded rectangles represent the cost groups that are considered in this study. The arrows point to the direction of the input. The solid rectangles are the three general expenditures categories of the OPEX.

1. *Land and Building Costs*: These assets are stationary and used for network's residency such as network headquarters building, house to the personnel, building for a data center and so on.
2. *Initial Physical Infrastructure Costs*: These expenses are related to hardware and the infrastructure of a network. They include the purchase of optical fibers, routers, switches, and so on.
3. *Initial Software Costs*: These costs include the purchase of software management systems, licenses (e.g. controller software license), and so on.
4. *Operational Network Upgrade Cost*: These costs are incurred from the ongoing network planning activities such as day-to-day planning, re-optimization, and planning upgrades.
5. *Loan Interests*: Some cost models, as in [4], also proposed to include interests to be paid for a loan in CAPEX.

## 7.2. OPEX for a Network

OPEX are the ongoing costs and contribute to the operational costs of a company to keep its operations (e.g. technical, commercial, and administrative) running on a daily basis. These expenses widely vary depending on the industry. OPEX are mainly split into three general categories for a service provider network: *OPEX for a running network*, *OPEX for network setup*, and *non-specific OPEX*.

1. *OPEX for a Running Network*: This category consists of the costs to operate and maintain a running and up network. It has six sub-parts as well:
  - (a) *Energy Costs of Infrastructure*: This category includes costs to keep the network operational in a failure free situation and includes costs such as power and energy-related cost.
  - (b) *Maintenance Costs*: This is the cost of preventative measures such as monitoring and maintaining the network against possible failures that may happen in the network. Some of the example actions include, but not limited to, (in)direct polling of component, logging status information, cleaning of equipment, stock management (e.g. ordering equipment), software management (e.g. tracking software versions and update installations).
  - (c) *Reparation Costs*: These are expenses incurred for repairing a failure (which may lead service interruptions) in the network. Actions include diagnosis and analysis, travel by technicians to the place of the failure, fixing the failure, testing to verify the repair, etc.



- (d) *Provisioning and Service Management Costs*: Service provisioning involves a request from a potential customer to start, move and/or stop a service, processing of the order by the network administrator, and performing a necessary test for the service to be provided.
  - (e) *Pricing and Billing Costs*: This category is about sending bills to customers and ensuring payments to receive from them. Possible actions in this category include service usage information collection per customer, per customer cost calculation, and calculating penalties to be paid by the operator for not fulfilling the promised service.
  - (f) *Marketing Costs*: The marketing part includes costs for acquiring new customers for a specific service of the network. Some actions in this category include promoting new service and providing information concerning pricing.
2. *OPEX for Network Setup*: The second general category of OPEX for a network is related to setting up a network. These costs contribute to the cost to be incurred before connecting the first customer. It has two sub-parts as well:
- (a) *First Time Installation Costs*: These costs are related to first time installation of the equipment after buying it. The actions would be actual connecting and installation of the new component into the network and the necessary testing of the component.
  - (b) *Up-Front Planning Costs*: This category denotes all planning done before the decision "let's go for this approach" is taken. Some of the example actions include planning studies to evaluate the building of a new network, changing the network topology, replacing network equipment with another brand/model, etc.
3. *Non-Specific OPEX*: The third category of OPEX is not specific to a service provider network. It may be seen in every company. It has two sub-parts as well:
- (a) *Non-Specific Continuous Costs of Infrastructure*: It denotes the costs such as heating, cleaning of buildings, etc. that are not related to the network itself.
  - (b) *Non-Specific Administration Costs*: This category includes administrative actions such as payment administration for employees, the human resources department and so on.

## 8. Network Costs Analysis

In this section, we analyze network costs and present two metrics: *unit service cost scalability* to evaluate unit cost performance of a network technology for service requests with respect to increasing service request workload and *Cost-to-Service* to economically quantify the cost of introducing a new service in a networking technology. We also present CAPEX and OPEX calculations, which is used in determination of unit cost for a service with QoS parameters in a network. We only consider bandwidth QoS parameter for service requests along with multiple numerical service tiers. We should note that the unit 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 a financial loss nor a profit for a network. Keeping cost at a higher or lower amount depends on network's market strategy and it is out of this paper's scope.

### 8.1. Unit Service Cost Scalability Metric

Scalability is a frequently-claimed attribute of various systems. It is a multi-dimensional topic. While the basic notion is intuitive, the term *scalability* does not evoke the same concept to everybody. While some people may refer to scalability as optimization of processing power to CPUs, others may define it as a measure of parallelization of applications across different machines [40]-[46]. Therefore, there is no general, precise agreement on neither its definition nor content. However, regardless of its meaning to someone, it is a desired property indicating positive sense for a system, algorithm, network and so on.

Scalability means not just the ability to operate, but to operate efficiently and with adequately quality of service, over the given range of configurations. There are certain questions that need answers

from researchers. One interesting example for such questions is that whether the cost of the system to provide service affect the system scalability. Jogalekar and Woodside [46] state that increased capacity should be in proportion to the cost of the system, and quality of service should be maintained. Moreover, when discussing network scalability, a large number of influencing factors have to be taken into account to arrive at a full picture. Behringer *et al.* [47] state that TCO is one example for such factors.

There are several research efforts proposing a metric to measure scalability of systems while considering the cost of the systems. [48] proposes a scalability metric, called “P-Scalability”, taking into account the cost of the system in distributed systems. It utilizes a concept called “power” measure and the cost of the system to provide service at a scale factor  $k$ . It is defined as:  $P - Scalability(k_1, k_2) = \frac{P(k_2) \cdot Cost(k_1)}{P(k_1) \cdot Cost(k_2)}$  where  $P(k) = Throughput/ResponseTime$ . This metric combines capacity and response time (both are present in the power  $P$ ) with the cost of the system. [46] defines scalability around “productivity” of the system in distributed systems. Productivity  $F(k)$  is the value delivered per second, divided by the cost per second:  $F(k) = \frac{\lambda(k) \cdot f(k)}{C(k)}$  where  $\lambda(k)$  is throughput in response/sec at scale  $k$ ,  $f(k)$  is the value of QoS at scale  $k$ , and  $C(k)$  is the cost of the system at scale  $k$ .

As proposed in these studies, the cost of providing service while preserving QoS for each service request in a system should be considered as evaluating the scalability of the system. In this context, we define a metric called *Unit Service Cost Scalability* to evaluate unit cost performance of a network for service requests with respect to increasing service request workload. This metric takes into account the network workload and different expenditures incurred to preserve the QoS at the same level for all service requests in the network. We expect that more scalable architectures result in less unit service cost in networks because such networks are able to handle more workload with respect to the same amount of network expenses compared to others.

We characterize the unit cost for a service (request) from a service tier as a function of network CAPEX, OPEX, and Workload over a certain time period. We refer Workload to service requests of all service 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 service tier is shown in Eq. 1. This formula implies that the unit service cost for a request from certain service tier is the ratio of TCO (CAPEX + OPEX) over workload in a given period.

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

where  $bw_j$ ,  $|bw_j|$ ,  $C_{bw_j}$ ,  $C$ ,  $O$  represent the type of (i.e. bandwidth) service with tier  $j$ , the 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  $w_{\delta_j}$  represent the workload and possible additional workload of service  $bw_j$  and  $W = \sum_{j=1} w_j$  and  $W_\delta = \sum_{j=1} w_{\delta_j}$ .  $C_\delta$ ,  $O_\delta$ , and  $W_\delta$  represent possible extra CAPEX, OPEX, and workload, respectively, incurred after introducing different kinds of changes/upgrades (represented as  $\delta$ ) in the network.

## 8.2. Calculation of CAPEX

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

$$C = f(\mathcal{B}, \mathcal{H}, \mathcal{S}, \mathcal{A}, \mathcal{I}) = \mathcal{P}_C(\mathcal{B} + \mathcal{H} + \mathcal{S} + \mathcal{A} + \mathcal{I}) \quad (2)$$

where  $\mathcal{P}_C$  is the CAPEX programmability factor representing the percentage of cost reduction in CAPEX that a network gains by applying a programmable network architecture principles. We use such a factor to accommodate the specificity of programmable networks. We make this factor percentage

based because it is not trivial to quantify and monetize the specificity of programmable networks. However, a percentage based factor makes it easier to comply with the percentage based cost reduction expectations of network decision-makers when simulating CAPEX and OPEX without real data.

### 8.2.1. Initial Physical Infrastructure Expenses ( $\mathcal{H}$ )

The initial physical infrastructure ( $\mathcal{H}$ ) of a network primarily consists of links, network devices, and also extra hardware such as a server that controller(s) installed on (in SDN case). Therefore,  $\mathcal{H}$  can be written as in Eq. 3:

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

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

### 8.2.2. Initial Software Expenses ( $\mathcal{S}$ )

Similarly, the initial software expenses ( $\mathcal{S}$ ) such as the purchase of management systems, licenses for proprietary controllers (in SDN case) can be calculated as in Eq. 4:

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

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

### 8.2.3. Operational Network Upgrade Costs ( $\mathcal{A}$ )

These expenses are incurred from the ongoing network upgrade activities, represented as  $\delta$ , such as adding/deleting/upgrading controller(s) (in SDN case), network device(s), link(s), and so on in the network. These expenses correspond the  $\mathbb{C}_\delta$  and  $\mathbb{O}_\delta$  in the Eq. 1. They are considered because, after foregoing modifications, extra CAPEX and OPEX may be incurred in the network.

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

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

We consider only the initial physical infrastructure ( $\mathcal{H}$ ), the initial software ( $\mathcal{S}$ ) expenses, and the operational network upgrades costs ( $\mathcal{A}$ ) as the main drivers for CAPEX and assume  $\mathcal{B} = 0$  and  $\mathcal{I} = 0$  for both SDN and MPLS cases in this study.

## 8.3. Calculation of OPEX

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 ( $\mathcal{T}$ ), (ii) expenses for network setup ( $\mathcal{N}$ ), and (iii) non-specific expenses ( $\mathcal{U}$ ) 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:

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

where  $\mathcal{P}_{\mathbb{O}}$  represents OPEX programmability factor for the network as explained in Subsection 8.2.

### 8.3.1. OPEX for a Running Network ( $\mathcal{T}$ )

Expenses for a running network ( $\mathcal{T}$ ) are primarily characterized by infrastructure energy expenses ( $\mathcal{K}$ ), maintenance expenses ( $\mathcal{M}$ ), reparation expenses ( $\mathcal{R}$ ), service provisioning/management expenses ( $\mathcal{P}$ ), pricing/billing expenses ( $\mathcal{G}$ ), and marketing expenses ( $\mathcal{W}$ ). Therefore, running network expenses are a function of these foregoing expenses and can be written as in Eq. 7:

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

*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 SDN, we assume that they have a relation with the total messages handled in the network in order to model them without data from a real network. These messages are internal overhead messages ( $O$ ) generated in the network and service requests ( $W$ ) (i.e. workload) entering the network. The idea behind this assumption is that the more messages handled in the network result in the more infrastructure energy 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:

$$\mathcal{K} + \mathcal{M} + \mathcal{R} = C_{pw} \left( \sum_{j=1}^{|d|} pw_{d_j} + \sum_{k=1}^{|c|} pw_{c_k} \right) + \varepsilon \left( \sum_{j=1} w_j + O \right) \quad (8)$$

where  $C_{pw}$ ,  $pw_{d_j}$ ,  $pw_{c_k}$  represent cost of KWh electricity power, energy consumption of network device  $d_j$  and controller  $c_k$  per hour, respectively.

On the other hand, in MPLS case, we use MTBF (Mean Time Between Failures) and MTTR (Mean Time To Repair) values to calculate Reparation ( $\mathcal{R}$ ) costs of network devices as in Eq. 9:

$$\mathcal{R} = |d| * \frac{\mathbb{T}}{MTBF} * MTTR * p \quad (9)$$

where  $\mathbb{T}$  and  $p$  represent a time period that the OPEX is calculated for and pay rate for the employee who repairs a device, respectively. We assume that each device is repaired by one employee. We use the same formula, as in Eq. 8, for energy expenses ( $\mathcal{K}$ )<sup>3</sup> in MPLS too:

$$\mathcal{K} = C_{pw} \left( \sum_{j=1}^{|d|} pw_{d_j} + \sum_{k=1}^{|c|} pw_{c_k} \right) \quad (10)$$

We also assume that total of Maintenance ( $\mathcal{M}$ ) costs are the half of the Reparation ( $\mathcal{R}$ ) costs,  $\mathcal{R} = \frac{\mathcal{M}}{2}$ .

We should note that service provisioning/management expenses ( $\mathcal{P}$ ), pricing/billing expenses ( $\mathcal{G}$ ), and marketing expenses ( $\mathcal{W}$ ) are not considered in *Running Network OPEX ( $\mathcal{T}$ )* calculation in this study. Therefore, we assume  $\mathcal{P} = 0$ ,  $\mathcal{G} = 0$ , and  $\mathcal{W} = 0$ .

### 8.3.2. OPEX for Network Setup ( $\mathcal{N}$ )

Similarly, network setup expenses ( $\mathcal{N}$ ) mainly come from first time equipment installation expenses ( $\mathcal{F}$ ) and up-front planning expenses ( $\mathcal{X}$ ). Therefore, network setup expenses are a function of these foregoing expenses and can be written as in Eq. 11:

$$\mathcal{N} = f(\mathcal{F}, \mathcal{X}) = \mathcal{F} + \mathcal{X} \quad (11)$$

We do not consider either first time equipment installation expenses ( $\mathcal{F}$ ) or up-front planning expenses ( $\mathcal{X}$ ) in this study. Therefore, we assume  $\mathcal{F} = 0$  and  $\mathcal{X} = 0$ .

<sup>3</sup>The number of controllers ( $|c|$ ) is zero in MPLS case.

### 8.3.3. Non-Specific OPEX ( $\mathcal{U}$ )

Finally, as stated in Section 7, non-specific expenses ( $\mathcal{U}$ ) are specified by non-specific continuous infrastructure expenses ( $\mathcal{Y}$ ) and administration expenses ( $\mathcal{Z}$ ). Therefore, non-specific expenses are a function of these foregoing expenses and can be written as in Eq. 12:

$$\mathcal{U} = f(\mathcal{Y}, \mathcal{Z}) = \mathcal{Y} + \mathcal{Z} \quad (12)$$

We do not consider *Non-Specific OPEX* ( $\mathcal{U}$ ) in this study either. Therefore, we assume  $\mathcal{Y} = 0$  and  $\mathcal{Z} = 0$ .

Regarding expenses that we do not consider, it is possible to include them in OPEX with a correct model as well. However, we leave them as a future work for now.

### 8.4. Cost-to-Service Metric

Programmable networks (e.g. SDN) bring standardized and programmatic interfaces (e.g. Open-Flow) that provide automation in network operations such as configuration across multiple, heterogeneous devices and flow management for efficient resource utilization. They also minimize human intervention in network operations, which helps in reducing network OPEX. This automation increases service velocity, streamlines service creation, and fosters innovative applications and services. On the other hand, in the traditional non-programmable networks (e.g. MPLS), there are a number of devices requiring different skill sets, including different technicians, programmers, and customer care personnel due to lack of standard programmable interfaces. They do not provide the flexibility necessary to make dynamic network changes and create new service offerings. Any changes to these networks are difficult, slow, and risky. Therefore, programmable networking helps reduce the costs of network operations and time required for introduction of a new service.

Network operators may decide to introduce new services for their users for different purposes such as generating new revenue opportunities. This process includes various steps before making the service fully operational. We define these steps as Service Design/Implementation ( $I$ ), Service Testing ( $E$ ), and Service Tuning Up ( $T$ ) in this study.

Service Design/Implementation phase mostly includes planning the configuration specifics for each network entities and sites. This planning involves design details of network elements and sites including port mappings, interface naming, host naming, IP addressing, VLAN addressing and many more. A proper design process is crucial for the continuity and timeliness of the whole service introduction without any errors. In addition, implementing planned design may require necessary coding over various network entities such as network devices, controllers, databases, and management systems depending on the technology, installation, and configuration of equipment and files. In SDN case, these actions/behaviors can be minimized because such actions do not necessarily have to be taken for all network entities. Instead, applying necessary actions once centrally and then distributing them to the relevant network devices saves number of employees and time spent for the service introduction. Service Testing phase aims at detecting network configuration issues causing faulty service functioning and service quality degradation, which may result in revenue losses. This testing process can be automatically conducted by exploiting network programming or manually as the technology allows. In programmable networking, it is possible to program and test every single element in the network remotely and quickly compared to traditional networking, which also saves the number of employees necessary and time to spend for testing purposes for a fast service introduction. Finally, Service Tuning Up phase includes final touches necessary to fix the detected issues from the testing phase and maximize the service quality to generate/retain the revenue from the service. Programmable networks also save costs in this phase due to the automation it provides as in the previous step.

Each step explained above brings its own expenses to the total cost of service introduction process. In this context, we describe a metric called *Cost-to-Service* to economically quantify the cost of introducing a new service in a network technology. To monetize the cost of the service introduction, we define this metric as the total cost of each step described above where the cost of each is the function

of number of employees ( $e$ ), employee pay rate per hour ( $p$ ), and time (in days) spent by employees ( $t$ ) in each step. Therefore, this metric can be written as in Eq. 13:

$$\begin{aligned}\mathfrak{S}_{\mathbf{e}} &= f(e, p, t) = \mathcal{C}^I + \mathcal{C}^E + \mathcal{C}^T \\ &= \sum_{i=1} e_i^I p_i^I t_i^I + \sum_{j=1} e_j^E p_j^E t_j^E + \sum_{k=1} e_k^T p_k^T t_k^T\end{aligned}\quad (13)$$

where subscripts  $i$ ,  $j$ , and  $k$  represent the corresponding employees involved in and  $\mathcal{C}^I$ ,  $\mathcal{C}^E$ , and  $\mathcal{C}^T$  are the cost of each corresponding steps.

### 8.5. Overhead Messages

As we defined in our previous work [49], we refer overhead 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 a rule for the flow in switches over the path calculated by the controller. Therefore, a rule-missing flow results in some other control-messages which are created, processed and sent by a switch and/or 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 devices: *Controller-to-Switch*, *Asynchronous*, and *Symmetric* messages. Each of these message types has its sub-types as well.

We characterize the overhead in a control plane, i.e. a controller, as the function of these three message types of OpenFlow protocol and synchronization messages among controllers. A control plane model generating less of these messages results in less OPEX. As explained in [49], we quantify the number of overhead messages as in Eq. 14:

$$O = f(\alpha, \beta, \gamma, \omega) = \sum_{j=1}^{|d|} (\alpha_j + \beta_j + \gamma_j) + \omega \quad (14)$$

where  $\alpha_j$ ,  $\beta_j$ ,  $\gamma_j$ ,  $\omega$  represent the number of *Controller-to-Switch* messages sent from a controller to 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). A total number of overhead messages in a control plane model such as CCP, DCP, and HCP can be different. 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 [49].

## 9. Evaluation

In this section, we present numerical results to provide some insights in order to understand the economic impact of SDN architecture with different control plane models (CCP, DCP, and HCP) and MPLS architecture on unit service cost. In this context, we analyze unit service cost and service introduction cost for the video service by total number of satisfied requests, CAPEX and OPEX in different SDN models and MPLS architecture.

### 9.1. Experimental Setup

SDN Setup. We have used Mininet emulator [50] with POX controller [51] in SDN case. While there is one controller in CCP model, we have divided the whole network into 4 fully-connected subnetworks with a controller for each in DCP and HCP models with a varying number of switches in different

simulation cases shown in the figures. There is also a master controller on top of local domain controllers in HCP model.

*MPLS Setup.* Regarding MPLS setting, we have used ns3 [52] network simulator. We need to use a signaling protocol such as RSVP-TE or CR-LDP to support constraint-based routing in MPLS. Since none of them has been implemented in ns3 at the time of this writing and it is time-consuming and effort-greedy to implement them in ns3, we have generated extra packets between network elements to mimic link state advertisements and state refresh messages for LSPs from aforementioned signaling protocols in MPLS.

*Shared Setup.* In the experiments, we have used 3 Mbps flow sending rate for all service requests. Therefore, there is only one service tier and  $|bw| = 3$  for all requests. Other numerical calculations have been done using MATLAB platform. Also, we have used a modified version of Waxman [53] random topology generator 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 [54], to find a feasible path through the network because constraint-based routing with two or more constraints has been shown to be an NP-hard [55]. A\*Prune algorithm combines A\*-search with a correct pruning technique. A\*Prune algorithm can be used to solve finding the K shortest paths subject to multiple constraints (KM CSP). Finally, all experiments were performed on Ubuntu 14.04 in Oracle VirtualBox using an Intel Core i7-5500 system with 12GB RAM.

Parameter	Value			
	CCP	DCP	HCP	MPLS
$ d $	4, 8, 12, 16, 20, 24, 28, 32	4, 8, 12, 16, 20, 24, 28, 32	4, 8, 12, 16, 20, 24, 28, 32	4, 8, 12, 16, 20, 24, 28, 32
$C_d$	\$1000	\$1000	\$1000	\$2000
$ l $	6, 12, 18, 24, 30, 36, 42, 48	6, 12, 18, 24, 30, 36, 42, 48	6, 12, 18, 24, 30, 36, 42, 48	6, 12, 18, 24, 30, 36, 42, 48
$C_l^4$	\$500	\$500	\$500	\$500
$ c $	1	4	4 + 1	N/A
$C_c$	$\frac{\$500 *  d }{r}$	$\frac{\$500 *  d }{r}$	$\frac{\$500 *  d }{r}$	N/A
$ s $	1	4	4 + 1	N/A
$C_s$	$\frac{\$500 *  d }{r}$	$\frac{\$500 *  d }{r}$	$\frac{\$500 *  d }{r}$	N/A
$\varepsilon$	$\$10^{-8}$	$\$10^{-8}$	$\$10^{-8}$	N/A
$\mathcal{P}_C$	40%	40%	40%	0%
$\mathcal{P}_O$	50%	50%	50%	0%
$C_{pw}$	\$0.116 KWh	\$0.116 KWh	\$0.116 KWh	\$0.116 KWh
$pw_d$	48 W	48 W	48 W	60 W
$pw_c$	400 W	400 W	400 W	N/A
$T$	1 year	1 year	1 year	1 year
$MTBF$	N/A	N/A	N/A	2000 hours
$MTTR$	N/A	N/A	N/A	10 hours
$e^I, p^I, t^I$	$2, \$35, \frac{ d  * r}{25}$	$2, \$35, \frac{ d  * r}{25}$	$2, \$35, \frac{ d  * r}{20}$	$2, \$35, \frac{ d }{4}$
$e^E, p^E, t^E$	$2, \$35, \frac{ d  * r}{35}$	$2, \$35, \frac{ d  * r}{35}$	$2, \$35, \frac{ d  * r}{30}$	$2, \$35, \frac{ d }{4}$
$e^T, p^T, t^T$	$2, \$35, \frac{ d  * r}{35}$	$2, \$35, \frac{ d  * r}{35}$	$2, \$35, \frac{ d  * r}{30}$	$2, \$35, \frac{ d }{10}$

Table 1: List of parameters and their values used in calculation of CAPEX and OPEX.

an average of each parameter based on Internet research, literature review ([3, 6, 12, 15], [56]-[65]), and our discussions. Although we should note that these input numbers may not be reflecting precise and/or realistic values, they should not impact the nature of the calculation framework since these values are very relative for every network company. We have used the same cost for all links ( $C_l$ ) in both SDN models and MPLS case. We have assumed that device cost is double ( $C_d$ ) in MPLS case since traditional network equipment is expected to be more expensive than SDN equipment due

We have used the scheme explained in our previous work [49] to get overhead messages with respect to number of requests in SDN models. In the experiments, we have used three 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). In each pattern, the source and destination switches are chosen randomly for service request while preserving the traffic pattern condition. 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.

Table 1 lists the parameters and their values used in calculation of CAPEX and OPEX in this study. It is difficult to gather precise input values for some of these parameters because they are proprietary and companies are not willing to publicly share them. These value assumptions constitute

<sup>4</sup>We have used \$100 for the link cost in 1 Gbps link bandwidth case.

to integrated control plane (i.e. proprietary software implementation). We have also assumed the same cost, which is proportional to the number of network devices ( $|d|$ ) and controller ( $r$ ) in different device number cases, for a controller hardware ( $C_c$ ) and software ( $C_s$ ) in all models. [57, 58] reveal that CAPEX reductions are expected (40%) in SDN case, but not as large as OPEX (50%) because it requires deploying new equipment (e.g. controller(s)), developing new operational processes, training employees, and so on. Therefore, we have used 40% and 50% CAPEX and OPEX programmability factor values, respectively, in SDN case while 0% programmability factor for CAPEX and OPEX due to non-programmability in MPLS case. Regarding device energy consumption, these values depend on many factors such as number of ports, capacity of port, memory type used, number of coming flows/packets and so on. Therefore, we have used an average value from these studies ([56], [60]-[63], [65]) for both SDN and MPLS devices. Furthermore, we have used the same number of employees and pay rate in service introduction steps in both SDN and MPLS cases for all switch cases. We have made the time spent by an employee proportional to the number of network devices and controllers in service introduction steps of SDN models. These time ratios are based on customer feedback from SDN use cases explained in [59]. In MPLS, this time is much more compared to SDN case due to mostly manual configurations over multiple heterogeneous devices. Finally, these values are the same for the different traffic patterns.

## 9.2. Experimental Results

Fig. 4 shows the relation between the total number of satisfied QoS-based requests regarding the different switch numbers in SDN models and MPLS under the different traffic patterns. This experiment consists of two parts: 100 Gbps and 1 Gbps link bandwidth parts. In the first part, (Fig. 4a, 4b, and 4c), we have provided enough bandwidth (100 Gbps) in links so that there is no service request rejection due to network resource limitations, while we have reduced the link bandwidth to 1 Gbps to see their performances under network resource limitations in the second part of the experiment (Fig. 4d, 4e, and 4f). In SDN models, the satisfied requests numbers represent the total number of requests that have been serviced in the corresponding models in a second by all controller(s) before rejecting a request in each switch number case. This rejection happens due to controller message handling capacity. In each traffic pattern, the total number of satisfied requests in all SDN 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 addition, as traffic becomes more inter-domains, total satisfied request numbers also reduce in all SDN models because paths become longer, which also results in more overhead in control planes that controllers need to deal with. In MPLS case, the network is able to serve requests at least HCP model in 100 Gbps case, which gives the best result in SDN case. Although MPLS could serve more due to no controller capacity constraint and enough bandwidth on links, we have left it at the same number as HCP model. We should note that these numbers also depend on SDN controller performance. However, comparing controllers performance is out of this paper's scope. In 1 Gbps link bandwidth case, the link bandwidth resource is exhausted before controllers reach their message handling capacity in SDN models. In this part of the experiment, the total number of satisfied requests show the tendency of increase in SDN and MPLS cases while switch number increases because adding more switches in the topology results in more link connectivities. This increases the number of possible end-to-end paths from a source to destination that can be used for a request. However, the number of satisfied requests is less than CCP because the flooding of available link bandwidth information (i.e. link state advertisements) and tunnel refresh messages (i.e. overhead) also consumes usable link bandwidth in MPLS case. The difference between CCP and MPLS regarding the total number of satisfied requests increases as the switch and link numbers increase because the aforementioned advertisement and refresh messages increase as well.

Fig. 5 shows the relation between the TCO with respect to the different switch numbers in SDN models and MPLS under the different traffic patterns. This experiment also consists of two parts: 100 Gbps (Fig. 5a, 5b, 5c) and 1 Gbps (Fig. 5d, 5e, 5f) link bandwidth parts. Main drivers of TCO are CAPEX and OPEX. Therefore, we consider that TCO is the sum of CAPEX and OPEX.



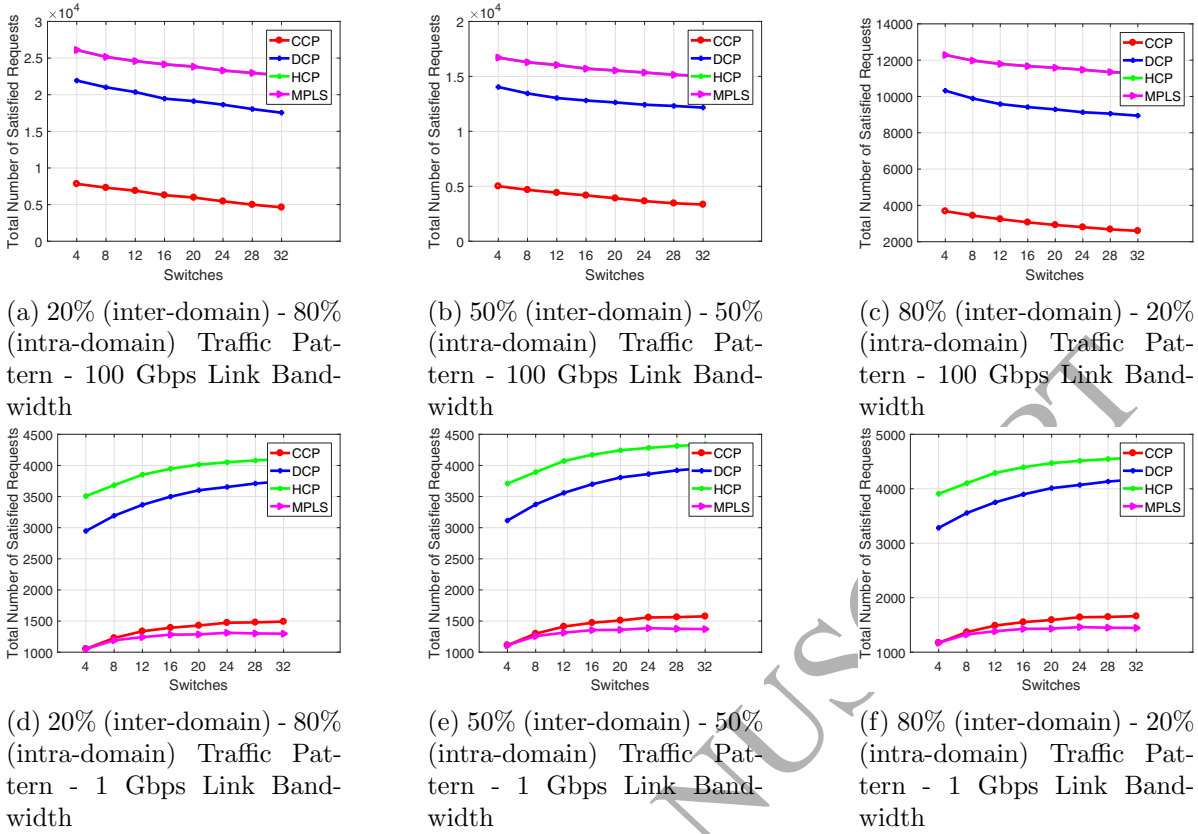


Figure 4: Total number of satisfied (controller(s)) requests (i.e. Workload) with QoS with respect to the different switch numbers in SDN models and MPLS under the different traffic patterns with 100 Gbps and 1 Gbps link bandwidth.

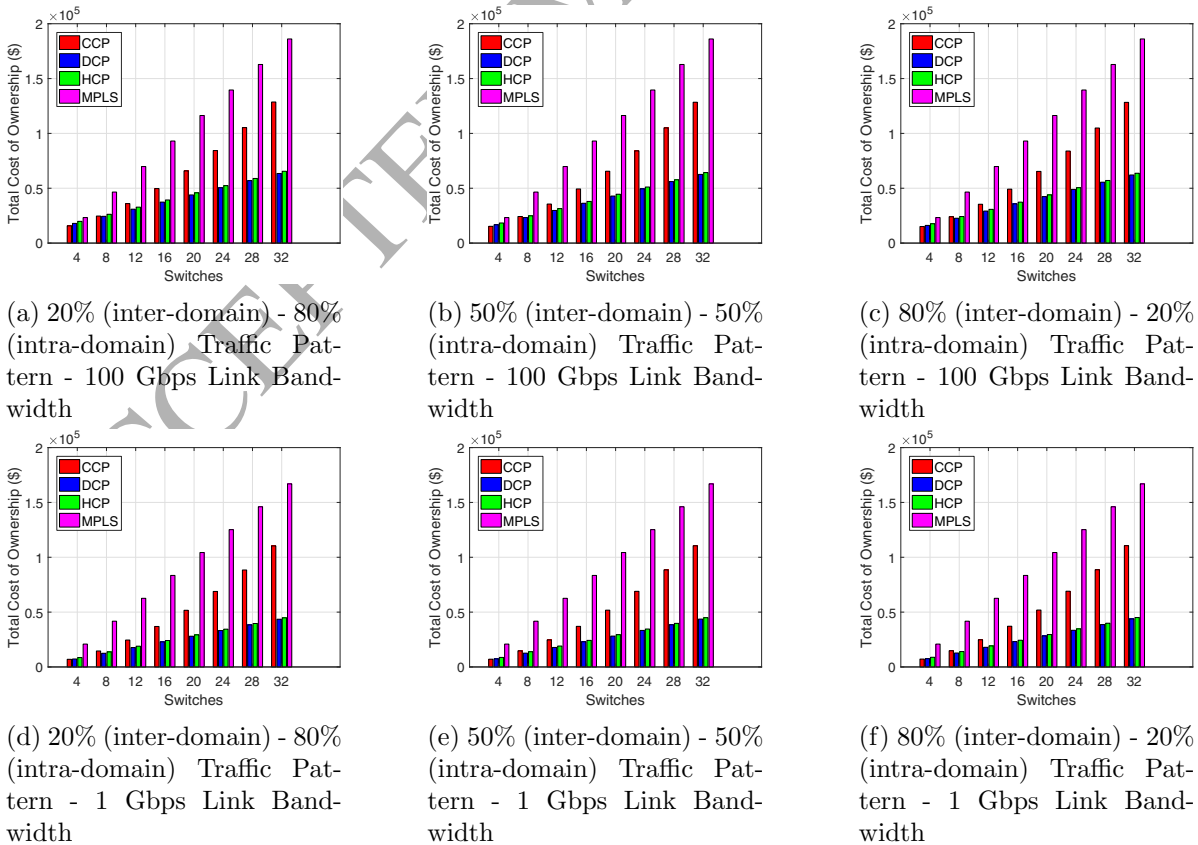


Figure 5: Total Cost of Ownership (TCO) with respect to the different switch numbers in SDN models and MPLS under the different traffic patterns with 100 Gbps and 1 Gbps link bandwidth.

OPEX cost is mostly dominated by the total number of messages (overhead + workload) handled by the controller(s) in SDN case as discussed in Subsection 8.3.1. In this experiment, when controllers reach their maximum throughput point, which happens in the first part of the experiment, we have added more switches to the network and started 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 exponential CAPEX addition since the previous controller is not used anymore. On the other hand, in DCP 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 patterns and link bandwidth types, which is an example for  $\mathbb{C}_{\Delta_s}$  as well discussed in Subsection 8.2.3. Furthermore, TCO of HCP model is more than DCP model because HCP model handles more workload ( $W$ ) than DCP model, see Eq. 8, and there is extra master controller cost in HCP model. On the other hand, MPLS shows more TCO compared to all SDN models because of its OPEX, which is not programmable and does not bring any cost reduction. In the second part of the experiment, TCO shows reduction in all SDN models and MPLS under all traffic patterns because CAPEX reduces due to link cost and OPEX reduces due to the number of workload and overhead.

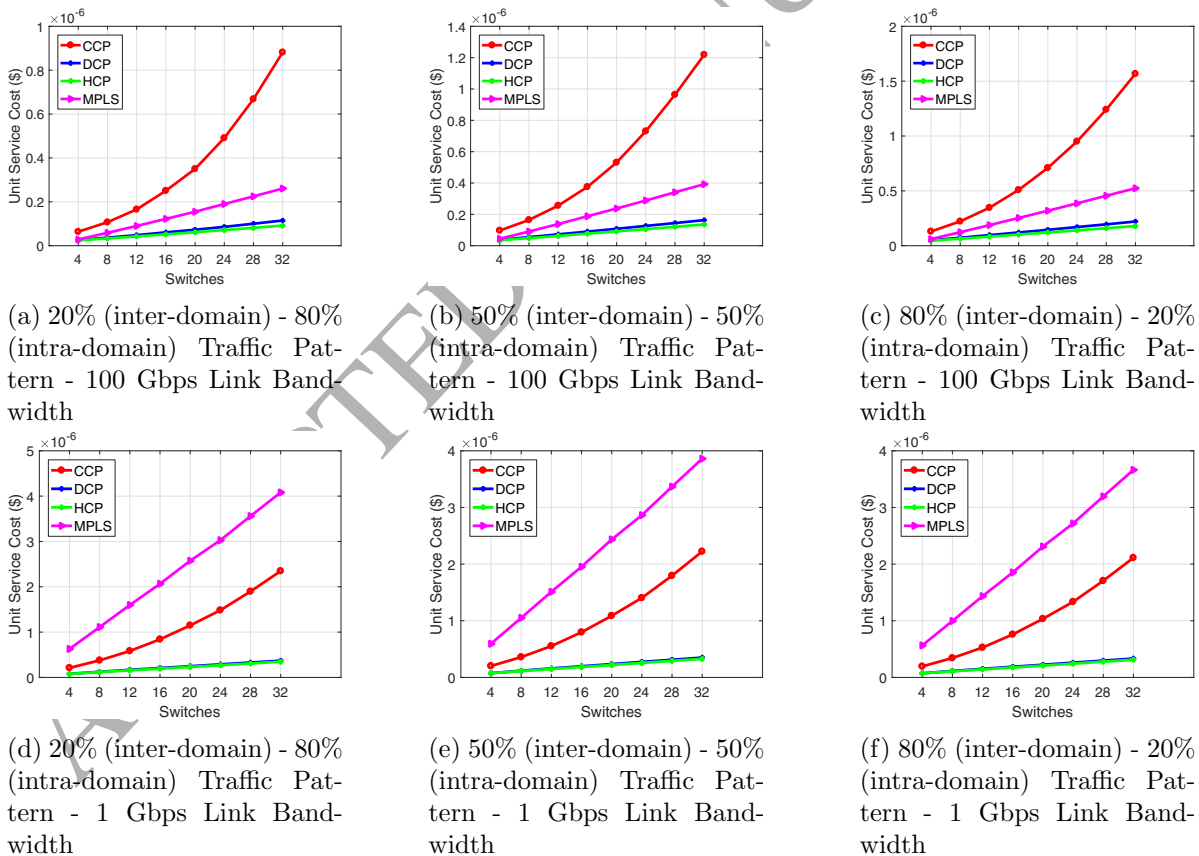


Figure 6: Unit service cost with respect to the different switch numbers in SDN models and MPLS under the different traffic patterns with 100 Gbps and 1 Gbps link bandwidth.

Fig 6 shows the relation between the unit service cost with respect to the different switch numbers in SDN models and MPLS under the different traffic patterns. This experiment has two parts as well: 100 Gbps and 1 Gbps link bandwidth parts. In the 100 Gbps link bandwidth part (Fig. 6a, 6b, and 6c), while CCP shows the highest unit service cost among all models, DCP gives higher unit service

cost than HCP based on one-to-one point comparison of curves. Although MPLS and HCP have the same and the highest number of satisfied requests, MPLS gives higher unit service cost than DCP and HCP because its CAPEX and OPEX is higher than that of DCP and HCP under all traffic patterns. Furthermore, both SDN models and MPLS result in lower unit service costs as the traffic becomes more local (i.e. intra-domain) because more requests are satisfied as explained previously. The unit service cost increases while the number of switches increase because CAPEX and OPEX increase and total satisfied number of requests decreases in both SDN models and MPLS as implied by the Eq. 1. In 1 Gbps link bandwidth part (Fig. 6d, 6e, and 6f), MPLS gives higher unit service cost than SDN models because it shows the lowest number of satisfied requests. CCP gives the highest unit service cost among SDN models while HCP gives slightly lower cost than DCP under all traffic patterns. Both SDN models and MPLS unit service cost results are similar (due to a similar number of satisfied requests) under all traffic patterns because link bandwidth is exhausted before controller capacity. Therefore, traffic pattern has little to no effect on unit service cost in 1 Gbps link bandwidth case. The CAPEX and OPEX increase ratio is faster compared to total number of satisfied requests in both SDN models and MPLS as the number of switches increase. Therefore, the unit service cost also increases while the number of switches increase. Finally, the unit service cost is higher for both SDN models and MPLS in 1 Gbps link bandwidth case compared to 100 Gbps link bandwidth case in each switch case due to the total number of satisfied requests, CAPEX, and OPEX results.

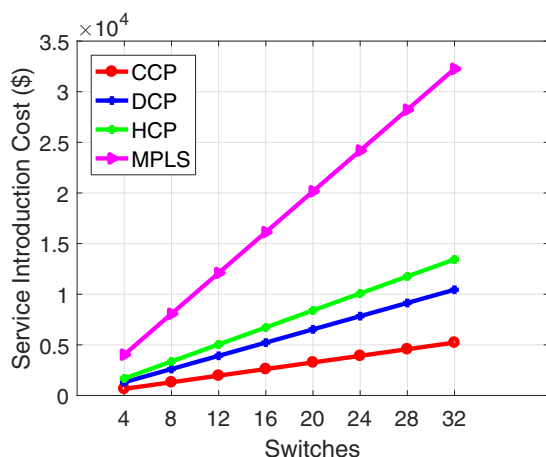


Figure 7: Service introduction cost with respect to the different switch numbers in SDN models and MPLS. The results are the same for the different traffic patterns and link bandwidth cases as well.

ability, which is reflected in time. HCP shows higher service introduction cost than CCP and DCP in each switch cases because the time spent in each service introduction step is more due to extra controller in the top layer. Similarly, DCP gives more service introduction cost compared to CCP because of the number of controllers, which impacts the time spent in each service introduction step too.

## 10. Conclusions

SDN paradigm has several key attributes that have an impact on the CAPEX and OPEX equations of a network. It has got the attention of researchers from both academia and industry as a means to be leveraged in order to decrease network costs and generate revenue for service providers due to features it promises in networking. In this article, we have investigated how programmable network architectures, i.e. SDN technology, affects the network economics compared to traditional network architectures, i.e. MPLS technology. To this end, we have defined two metrics, *Unit Service Cost Scalability* and *Cost-to-Service*, to evaluate how SDN architecture performs compared to MPLS architecture. We have

also presented mathematical models to calculate certain cost parts of a network. In addition, we have compared different popular SDN control plane models, Centralized Control Plane (CCP), Distributed Control Plane (DCP), and Hierarchical Control Plane (HCP), to understand the economic impact of them with regards to the defined metrics. Our simulation results have revealed that MPLS shows more TCO compared to all SDN models because of its OPEX, which is not programmable and does not bring any cost reduction. TCO of HCP model is more than DCP model because HCP model handles more workload than DCP model and there is extra master controller cost in HCP model. In addition, CCP shows the highest unit service cost because it results in more CAPEX and least workload among all models. On the other hand, HCP results in the lowest unit service cost because it handles the most number of workload. The results have also demonstrated that as the number of switches increases the unit service cost increases as well because the total number of satisfied requests are decreasing due to longer paths in both SDN models and MPLS. Furthermore, it has been shown that MPLS gives the higher service introduction costs compared to SDN models owing to lack of automation and programmability, which is reflected in time. In SDN case, on the other hand, the number of network elements, such as controllers, impact the total cost of service introduction because of the time spent in each service introduction step. These results have pointed out that programmability has great impact on network economics.

## References

- [1] I. F. Akyildiz, A. Lee, P. Wang, M. Luo, W. Chou, A Roadmap for Traffic Engineering in SDN-OpenFlow Networks, *Computer Networks* 71 (2014) 1 – 30. doi:<http://dx.doi.org/10.1016/j.comnet.2014.06.002>. URL <http://www.sciencedirect.com/science/article/pii/S1389128614002254>
- [2] H. Farhady, H. Lee, A. Nakao, Software-Defined Networking: A survey, *Computer Networks* 81 (2015) 79 – 95. doi:<http://dx.doi.org/10.1016/j.comnet.2015.02.014>. URL <http://www.sciencedirect.com/science/article/pii/S1389128615000614>
- [3] S. Verbrugge, D. Colle, M. Pickavet, P. Demeester, S. Pasqualini, A. Iselt, A. Kirstädter, R. Hülsermann, F.-J. Westphal, M. Jäger, Methodology and input availability parameters for calculating opexand capex costs for realistic network scenarios, *J. Opt. Netw.* 5 (6) (2006) 509–520.
- [4] S. Verbrugge, S. Pasqualini, F. J. Westphal, M. Jager, A. Iselt, A. Kirstadter, R. Chahine, D. Colle, M. Pickavet, P. Demeester, Modeling Operational Expenditures for Telecom Operators, in: *Conference on Optical Network Design and Modeling*, 2005., 2005.
- [5] A. Swisscom, T. Danmark, S. T. SA, Extended investment analysis of telecommunication operator strategies, *Euroscm* 2.
- [6] S. Verbrugge, D. Colle, M. JÄGER, R. HUELSESMANN, F. WESTPHAL, M. Pickavet, P. Demeester, Impact of resilience strategies on capital and operational expenditures, in: *Proceedings of ITG-Fachtagung Photonical Networks 2005*, 2005, pp. 109–116. URL <http://dx.doi.org/10.1109/MCOM.2005.1470804>
- [7] S. Pasqualini, S. Verbrugge, A. Kirstädter, A. Iselt, D. Colle, M. Pickavet, P. Demeester, Influence of a control plane on network expenditures, in: *Asia-Pacific Optical Communications, International Society for Optics and Photonics*, 2005, pp. 60220I–60220I.
- [8] S. Pasqualini, A. Kirstadter, A. Iselt, R. Chahine, S. Verbrugge, D. Colle, M. Pickavet, P. Demeester, Influence of gmpls on network providers' operational expenditures: a quantitative study, *IEEE Communications Magazine* 43 (7) (2005) 28–38. doi:10.1109/MCOM.2005.1470804.
- [9] B. Naudts, M. Kind, F. J. Westphal, S. Verbrugge, D. Colle, 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.
- [10] R. Sherwood, G. Gibb, K.-K. Yap, G. Appenzeller, M. Casado, N. McKeown, G. Parulkar, Flowvisor: A network virtualization layer, *OpenFlow Switch Consortium*, Tech. Rep.
- [11] E. Hernandez-Valencia, S. Izzo, B. Polonsky, How will NFV/SDN Transform Service Provider Opex?, *IEEE Network*.
- [12] N. Zhang, H. Hmminen, Cost efficiency of sdn in lte-based mobile networks: Case finland, in: *2015 International Conference and Workshops on Networked Systems (NetSys)*, 2015, pp. 1–5. doi:10.1109/NetSys.2015.7089056.
- [13] T. M. Knoll, A Combined CAPEX and OPEX Cost Model for LTE Networks, in: *Telecommunications Network Strategy and Planning Symposium (Networks)*, 2014 16th International, 2014, pp. 1–6. doi:10.1109/NETWKS.2014.6958531.

- [14] T. M. Knoll, Life-cycle cost modelling for nfv/sdn based mobile networks, in: 2015 Conference of Telecommunication, Media and Internet Techno-Economics (CTTE), 2015, pp. 1–8. doi:10.1109/CTTE.2015.7347225.
- [15] C. Bouras, P. Ntarzanos, 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), 2016, pp. 56–61. doi:10.1109/ICUMT.2016.7765232.
- [16] M. Karakus, A. Durrezi, Service cost in software defined networking (sdn), in: 2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA), 2017, pp. 468–475. doi:10.1109/AINA.2017.111.
- [17] E. j. Kwak, G. e. Kim, J. h. Yoo, Network operation cost model to achieve efficient operation and improving cost competitiveness, in: ICACT, 2011, 2011, pp. 1107–1112.
- [18] R. G. Addie, Y. Peng, M. Albdair, C. Xing, D. Fatseas, M. Zukerman, Cost modelling and validation in network optimization, in: ITNAC, 2015 International, 2015, pp. 11–16. doi:10.1109/ATNAC.2015.7366782.
- [19] L. R. de Lope, K. Hackbarth, A. E. Garca, T. Plueckebaum, D. Ilic, Cost models for next generation networks with quality of service parameters, in: Networks 2008 - The 13th International Telecommunications Network Strategy and Planning Symposium, Vol. Supplement, 2008, pp. 1–9. doi:10.1109/NETWKS.2008.6231323.
- [20] L. Popa, S. Ratnasamy, G. Iannaccone, A. Krishnamurthy, I. Stoica, A Cost Comparison of Datacenter Network Architectures, in: Proceedings of the 6th International Conference, Co-NEXT '10, 2010, pp. 16:1–16:12.
- [21] K. Casier, S. Verbrugge, R. Meersman, J. Van Ooteghem, D. Colle, M. Pickavet, P. Demeester, A fair cost allocation scheme for capex and opex for a network service provider, Proceedings of CTTE2006, the 5th Conference on Telecommunication Techno-Economics.
- [22] R. Bailey, Elements of techno-economic modelling planning, provisioning and operation of virtualised networks, in: Telecommunications Network Strategy and Planning Symposium (Networks), 2016 17th International, IEEE, 2016, pp. 25–30.
- [23] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, J. Turner, Openflow: Enabling innovation in campus networks, SIGCOMM Comput. Commun. Rev. 38 (2) (2008) 69–74.
- [24] SDN architecture, Tech. rep., Open Networking Foundation (ONF) (June 2014).  
URL [https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/TR\\_SDN\\_ARCH\\_1.0\\_06062014.pdf](https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/TR_SDN_ARCH_1.0_06062014.pdf)
- [25] P. Lin, J. Bi, S. Wolff, Y. Wang, A. Xu, Z. Chen, H. Hu, Y. Lin, A west-east bridge based sdn inter-domain testbed, Communications Magazine, IEEE 53 (2) (2015) 190–197. doi:10.1109/MCOM.2015.7045408.
- [26] M. Karakus, A. Durrezi, A Survey: Control Plane Scalability Issues and Approaches in Software-Defined Networking (SDN), Computer Networks 112 (2017) 279–293. doi:http://dx.doi.org/10.1016/j.comnet.2016.11.017.  
URL <http://www.sciencedirect.com/science/article/pii/S138912861630411X>
- [27] A. S. da Silva, P. Smith, A. Mauthe, A. Schaeffer-Filho, Resilience Support in Software-Defined Networking: A Survey, Computer Networks 92, Part 1 (2015) 189–207. doi:http://dx.doi.org/10.1016/j.comnet.2015.09.012.  
URL <http://www.sciencedirect.com/science/article/pii/S1389128615003229>
- [28] M. Casado, M. J. Freedman, J. Pettit, J. Luo, N. McKeown, S. Shenker, Ethane: taking control of the enterprise, SIGCOMM Comput. Commun. Rev. 37 (4) (2007) 1–12.
- [29] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, S. Shenker, Nox: Towards an operating system for networks, SIGCOMM Comput. Commun. Rev. 38 (3) (2008) 105–110. doi:10.1145/1384609.1384625.  
URL <http://doi.acm.org/10.1145/1384609.1384625>
- [30] A. Tootoonchian, Y. Ganjali, Hyperflow: A distributed control plane for openflow, in: Proceedings of the 2010 Internet Network Management Conference on Research on Enterprise Networking, INM/WREN'10, 2010, pp. 3–3.
- [31] K. Phemius, M. Bouet, J. Leguay, Disco: Distributed multi-domain sdn controllers., CoRR abs/1308.6138.  
URL <http://dblp.uni-trier.de/db/journals/corr/corr1308.html#PhemiusBL13>
- [32] Y. Fu, J. Bi, K. Gao, Z. Chen, J. Wu, B. Hao, Orion: A hybrid hierarchical control plane of software-defined networking for large-scale networks, in: Network Protocols (ICNP), 2014 IEEE 22nd International Conference on, 2014, pp. 569–576. doi:10.1109/ICNP.2014.91.
- [33] M. Karakus, A. Durrezi, A Scalable Inter-AS QoS Routing Architecture in Software Defined Network (SDN), in: Advanced Information Networking and Applications (AINA), 2015 IEEE 29th International Conference on, 2015, pp. 148–154. doi:10.1109/AINA.2015.179.
- [34] M. Walker, A Growth Opportunity for Vendors: Telco Opex (OVUM, Oct 31 2012).
- [35] H. G. Perros, Connection-Oriented Networks: SONET/SDH, ATM, MPLS and Optical Networks, Wiley, 2005.

- [36] J. Kharel, D. Adhikari, Performance evaluation of voice traffic over mpls network with te and qos implementation, Master's thesis, Blekinge Institute of Technology, Sweden (November 2011).
- [37] B. Jamoussi, L. Andersson, R. Callon, R. Dantu, L. Wu, P. Doolan, T. Worster, N. Feldman, A. Fredette, M. Girish, et al., RFC 3212: Constraint-based LSP setup using LDP (2002).  
URL <https://tools.ietf.org/html/rfc3212>
- [38] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, G. Swallow, RFC 3209: RSVP-TE: extensions to RSVP for LSP tunnels (2001).  
URL <https://tools.ietf.org/html/rfc3209>
- [39] J. W. Evans, C. Filsfils, Deploying IP and MPLS QoS for Multiservice Networks: Theory & Practice, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2007.
- [40] K. Hwang, Z. Xu, Scalable Parallel Computing: Technology, Architecture, Programming, McGraw-Hill, Inc., New York, NY, USA, 1998.
- [41] X. H. Sun, D. T. Rover, Scalability of parallel algorithm-machine combinations, IEEE Trans. Parallel Distrib. Syst. 5 (6) (1994) 599–613. doi:10.1109/71.285606.  
URL <http://dx.doi.org/10.1109/71.285606>
- [42] V. Kumar, A. Grama, A. Gupta, G. Karypis, Introduction to Parallel Computing: Design and Analysis of Algorithms, Benjamin-Cummings Publishing Co., Inc., Redwood City, CA, USA, 1994.
- [43] A. Y. Grama, A. Gupta, V. Kumar, Isoefficiency: Measuring the scalability of parallel algorithms and architectures, IEEE concurrency (3) (1993) 12–21.
- [44] L. Pastor, J. Bosquwe Orero, An efficiency and scalability model for heterogeneous clusters, in: Cluster Computing, 2001. Proceedings. 2001 IEEE International Conference on, 2001, pp. 427–434. doi:10.1109/CLUSTR.2001.960009.
- [45] X. H. Sun, Y. Chen, M. Wu, Scalability of heterogeneous computing, in: Proceedings of the 2005 International Conference on Parallel Processing, ICPP '05, IEEE Computer Society, Washington, DC, USA, 2005, pp. 557–564. doi:10.1109/ICPP.2005.69.  
URL <http://dx.doi.org/10.1109/ICPP.2005.69>
- [46] P. Jogalekar, M. Woodside, Evaluating the scalability of distributed systems, IEEE Trans. Parallel Distrib. Syst. 11 (6).
- [47] M. Behringer, A. Retana, R. White, G. Huston, RFC 7980: A Framework for Defining Network Complexity.
- [48] P. Jogalekar, C. Woodside, A scalability metric for distributed computing applications in telecommunications, Teletraffic Science and Engineering 2 (1997) 101–110.
- [49] M. Karakus, A. Dursesi, A Scalability Metric for Control Planes in Software Defined Networks (SDNs), in: 2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA), 2016, pp. 282–289. doi:10.1109/AINA.2016.112.
- [50] Mininet.  
URL <http://mininet.org/>
- [51] Pox.  
URL <http://www.noxrepo.org/pox/about-pox/>
- [52] The ns-3 Network Simulator.  
URL <https://www.nsnam.org/>
- [53] B. M. Waxman, Routing of multipoint connections, IEEE Journal on Selected Areas in Communications 6 (9) (1988) 1617–1622. doi:10.1109/49.12889.
- [54] G. Liu, K. G. Ramakrishnan, A\*prune: an algorithm for finding k shortest paths subject to multiple constraints, in: Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No.01CH37213), Vol. 2, 2001, pp. 743–749 vol.2. doi:10.1109/INFCOM.2001.916263.
- [55] O. Younis, S. Fahmy, Constraint-based routing in the internet: Basic principles and recent research, Commun. Surveys Tuts. 5 (1) (2003) 2–13. doi:10.1109/COMST.2003.5342226.  
URL <http://dx.doi.org/10.1109/COMST.2003.5342226>
- [56] J. Lahteenmaki2016Cost, H. Hammainen, N. Zhang, M. Swan, Cost modeling of a network service provider cloud platform, in: 2016 IEEE International Conference on Cloud Engineering Workshop (IC2EW), 2016, pp. 148–153. doi:10.1109/IC2EW.2016.40.
- [57] Data Center SDN Strategies Global Service Provider Survey, Tech. rep., IHS Technology, Infonetics (October 2015).
- [58] Data Center SDN Strategies North American Enterprise Survey, Tech. rep., IHS Technology, Infonetics (February 2015).
- [59] NEC SDN Customer Cases.  
URL <http://www.nec.com/en/global/solutions/sdn/case/index.html>

- [60] H. Hlavacs, G. D. Costa, J. M. Pierson, Energy consumption of residential and professional switches, in: 2009 International Conference on Computational Science and Engineering, Vol. 1, 2009, pp. 240–246. doi:10.1109/CSE.2009.244.
- [61] P. Mahadevan, P. Sharma, S. Banerjee, P. Ranganathan, A power benchmarking framework for network devices, in: Proceedings of the 8th International IFIP-TC 6 Networking Conference, NETWORKING '09, Springer-Verlag, Berlin, Heidelberg, 2009, pp. 795–808. doi:10.1007/978-3-642-01399-7\_62.  
URL [http://dx.doi.org/10.1007/978-3-642-01399-7\\_62](http://dx.doi.org/10.1007/978-3-642-01399-7_62)
- [62] K. Xie, X. Huang, S. Hao, M. Ma, P. Zhang, D. Hu,  $e^3$ mc: Improving energy efficiency via elastic multi-controller sdn in data center networks, IEEE Access 4 (2016) 6780–6791. doi:10.1109/ACCESS.2016.2617871.
- [63] D. B. Rawat, S. R. Reddy, Software defined networking architecture, security and energy efficiency: A survey, IEEE Communications Surveys Tutorials 19 (1) (2017) 325–346. doi:10.1109/COMST.2016.2618874.
- [64] Cost of KWh Electricity Power.  
URL [https://www.eia.gov/electricity/monthly/epm\\_table\\_grapher.php?t=epmt\\_5\\_6\\_a](https://www.eia.gov/electricity/monthly/epm_table_grapher.php?t=epmt_5_6_a)
- [65] F. Kaup, S. Melnikowitsch, D. Hausheer, Measuring and modeling the power consumption of openflow switches, in: Network and Service Management (CNSM), 2014 10th International Conference on, IEEE, 2014, pp. 181–186.

## Author Biography



**Murat Karakus (mkarakus@iupui.edu)**

Murat Karakus received the BS Degree in Mathematics from Suleyman Demirel University, Turkey, in 2009, and the MS Degree in Computer Science and Information Systems from the University of Michigan-Flint, in 2013. He is currently working through his PhD in the Department of Computer and Information Science at Indiana University Purdue University - Indianapolis. He is the recipient of the Best Paper Award at ACM SIGITE 2011 conference. His current research interests include new network architectures (particularly Software-Defined Networking (SDN)), scalability, Quality of Service (QoS), routing, and introducing programming to non-CS majors.



**Arjan Durresi (adurresi@iupui.edu)**

Arjan Durresi is a Professor of Computer Science at Indiana University Purdue University in Indianapolis, Indiana. In the past, he held positions at LSU, and The Ohio State University. His research interest include network architectures, security and trust management. He has published over eighty papers in journals and over 200 papers in conference proceedings, and seven book chapters. He also has over thirty contributions to standardization organizations such as IETF, ATM Forum, ITU, ANSI and TIA. His research has been funded by NSF, the States of Ohio and Louisiana, as well as university and industry sources.