Journal of Network and Computer Applications 104 (2018) 61–77

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Review

Multicasting in software defined networks: A comprehensive survey

Zainab AlSaeed [a], Imtiaz Ahmad [a,*], Iftekhar Hussain [b]

[a] Computer Engineering Department, Kuwait University, Kuwait
[b] Infinera Corporation, 169 W Java Dr, Sunnyvale, CA, USA

ARTICLE INFO

Keywords:
Software defined network
Multicasing
Multicast group
Spanning tree
Shortest path tree
Steiner tree
Routing
Data center
Traffic engineering

ABSTRACT

The emerging Software Defined Networking (SDN) paradigm is being adopted by the telecom industry since it enables flexible network resource allocation, configuration and management. Multicast is an essential and attractive service for a wide range of todays Internet applications for its bandwidth-preserving efficiency and flexibility. In a multicast, a data stream is delivered from single or multiple sources to a group of destinations simultaneously. SDN has potential to simplify multicast traffic engineering by leveraging the centralized nature of the network control plane, which provides a global view of the network that is built based on the real-time data gathered from network devices. This article aims to provide a comprehensive survey about the recent advances in Software Defined Multicasting. Specifically, it will provide an overview of multicasting in the context of SDNs, discuss tree planning and management, discuss multicast routing and traffic engineering, reliability and scalability in routing and multicast techniques in data centers. It will also summarize key techniques for each important topic related to multicasting that can enable researchers and practitioners to quickly get started. Finally, we identify open challenges for SDN multicasting and outline future research directions.

1. Introduction

According to latest IT industry analysis, the global market for Software Defined Networking (SDN) has recorded a remarkable annual growth rate of 88.1% in five years period only. The market has reached 2.4 $ billion in 2015 and is expected to reach 56.1 $ billion in 2020 (Gijare, 2016). SDN is a new networking paradigm that has been adopted by dominant internet companies like Google and Facebook for their Data Center Networks structure and their connections in Wide Area Networks (De Turck et al., 2016). It was first introduced to overcome the lack of programmability and scalability in network management and configuration that are faced in the traditional IP networks. Although traditional networks are widely adopted, requirements such as: dynamicity, flexibility, and easiness in management and configuration are challenging to achieve in such networks. This makes them unattractive networking solution especially with the evolution of the Internet and the newly evolving technologies that require higher bandwidth, accessibility, increased network programmability and agility. Examples of such technologies are: mobile networking, cloud computing, network function virtualization, social networking, and multimedia applications (Masoudi and Ghaffari, 2016).

The above-mentioned challenges are a result of the vertical integration between the control plane and the data-forwarding plane. Current networks structure consists of different networking devices (routers, switches, middle boxes) where both the control logic and data forwarding functionality are integrated within the same device. Those devices are function-specific and usually designed using chips and Application-Specific Integrated Circuits (ASIC) (Masoudi and Ghaffari, 2016). Therefore, each network device is configured separately using a set of low level pre-defined commands based on their embedded operating system (Kreutz et al., 2015; Masoudi and Ghaffari, 2016). Hence, managing a large number of devices is challenging, time consuming, and error prone. In addition to their complexity, traditional networks lack the mechanisms of auto-reconfiguration in case of dynamic events such as network failures and load changes (Kreutz et al., 2015).

The main concept of SDN is the decoupling between the control plane and the data-forwarding plane. That is, the control logic is be implemented into a centralized entity referred to as the SDN Controller. The controller is responsible for managing the network and setting the forwarding decisions based on the network condition, which is updated periodically by collecting status information from the network devices. The flow-settings decisions are then installed as flow rules into the net-

* Corresponding author.
E-mail addresses: eng.zainab.alsaeed@gmail.com (Z. AlSaeed), imtiaz.ahmad@ku.edu.kw (I. Ahmad), ihussain@infinera.com (I. Hussain).

https://doi.org/10.1016/j.jnca.2017.12.011
Received 20 May 2017; Received in revised form 11 October 2017; Accepted 8 December 2017
Available online 23 December 2017
1084-8045/

work devices that reside in the data plane and which are referred to as the forwarding elements (Gu et al., 2015). The SDN controller has a global view of the network that is updated through real-time communication with network devices. According to network status and users requirements, the controller adjusts routing decisions and control policies and forwards them to the forwarding devices to achieve a well-controlled real time routing. Moreover, the centralized control logic, and the global view of the network at the controller side has other benefits in terms of network management, resource utilization, cost reduction, increasing flexibility, and applying traffic engineering (Prithviraj et al., 2016).

This paper introduces a literature survey on Software Defined Multicasting (SDM), where SDM takes advantage of SDN features to achieve efficient multicasting. Multicasting is a group communication paradigm that aims to transmit data from one or multiple sources to a group of destinations simultaneously through a multicast tree that connects the data sources and the receivers. This saves significant amount of bandwidth when compared to unicast data delivery of multicast traffic where disjoint paths must be reserved for each pairs of communication. Moreover, multicasting eliminates the unnecessary amount of duplicated packets, since duplication will only take place when the stream has to reach all receivers or at the leaf nodes of the tree, unlike unicast delivery where traffic is duplicated at the source and traverses the entire network.

The nature of many applications require applying an efficient and flexible multicast delivery of data. For example, in modern data centers group communication between servers is frequent where same data must be sent from one server to a group of servers. In such scenarios applying multicast instead of unicast delivery is necessary to reduce the number of duplicated packets and save bandwidth significantly. Moreover, the centralized controller in SDM enables flexible deployment of new routing algorithms (Fan et al., 2016).

SDM also supports enforcing QoS constraints which are required in applications like live video streaming, video and audio conferencing, multi-player games, Internet Protocol TV (IPTV). In such applications multicasting is needed as an effective technique for transmitting data to multiple receivers simultaneously (Al Hasrouty et al., 2016). This is difficult to achieve in traditional networks since it requires multicast support, which is not available in most routers and it must maintain certain QoS level.

The authors in Ref. Tang et al. (2014) have introduced a video streaming multicast application that was designed for SDNs. The proposed solution uses a video coding technique named Scalable Video Coding (SVC) where the video is compressed into multiple layers. The layers are non-overlapped with a base layer that represents the original video at a low quality level. The other layers are enhancement layers that gradually emend the quality of reconstructed video. SVC technique implements effective multicasting that can adapt to network status and node capacities to transmit video streams with as high accuracy as possible. SDM^{2}Cast (Jian et al., 2015) applies a layered multicast scheme to deliver SVC videos, which means that each video layer has its own multicast tree.

SVC technique was also studied in Xue et al. (2015) but from a different perspective. Real-time SVC streaming was considered where there are multiple sources of data stream that are geographically distributed among multiple servers and clients can dynamically join and leave the multicasting sessions. This can be defined as a multi-source multi-destination video manycast problem. Several algorithms were designed to solve this problem, starting by formulating an integer linear programming model to solve for small-scale networks. After that two algorithms were designed for practical implementation of the solution. The simulation results proved that the heuristics could provide close-to-optimal solutions (Xue et al., 2015).

Another use case of SDM that is currently being investigated include integrating SDM concepts with cloud-based applications, network function virtualization and geographically distributed data centers (Zeng et al., 2016; Zhang et al., 2015).

The multicast model proposed in Ref. Humernbrum et al. (2016) is an example of a multicast scheme that leverages SDN features to overcome the limitations of traditional IP multicast. Better group membership control is applied using the centralized control structure. Moreover, combining Branch-Aware Modification and Early Branch techniques for tree calculations enhance the scalability since they allow re-using of flow entries in multicast routing tables. A module that combines SDN controller design and tree calculation technique was implemented and evaluated for both simulated networks and real OpenFlow-enabled network, where the results showed a reduction in the required number of flow-table entries.

To the best of our knowledge, in spite of the importance of SDM, there is no comprehensive survey that covers all topics related to this area. An earlier survey was introduced in Ref. Gu et al. (2015), however, it only discussed the basic concepts such as general architecture, multicast in data centers, routing procedure and tree packing. Other fields such as the revolution of multicasting techniques, different multicast mechanisms, tree planning and construction procedures, tree management related issues, new evolving routing concepts, applying Traffic Engineering in multicasting, multicasting deployment in recent data centers, scalable multicasting and suggested research directions were not addressed properly. This paper surveys the state of the art of the multicasting techniques and key challenges in the context of SDN and provides suggestions for future directions. SDN-based multicasting approaches for applications such as mobile networks, vehicular networks, and video-on-demand are not addressed in this survey. A tabular comparison between the two surveys is given in Table 1.

The rest of the paper is organized as follows. Section 2 describes the background of this topic, starting with a brief history of multicasting in general. Then describing the detailed architecture of SDNs with its different layers and the role of each layer in multicasting, and finally presenting different mechanisms of SDM. After that, Section 3 explains the procedure of multicast tree planning and construction, with different construction approaches. Section 4 explains how SDN controller handles tree management including the procedure of maintaining network status, managing dynamics such as group membership changes and network failures, and finally discussing tree packing problem. In Section 5 multicast routing is highlighted from different perspectives. Multicast Traffic Engineering is discussed in Section 6. After that, multicasting in SDN-based data centers is discussed in Section 7. Finally, future research directions are suggested in Section 8. List of all acronyms used in the paper are described in Table 2.

2. Background

2.1. Multicasting history

The multicasting functionality can either be implemented on the network level (IP Multicast) or on the Application Layer Multicast (ALM). Network-layer multicast is built over an IP infrastructure, where the network devices (switches and routers) are responsible for delivering the packets efficiently. That is, the packets are duplicated to reach all receivers while ensuring that they are sent over each network link only once (Gu et al., 2015).

IP multicast was the first communication protocol supporting network-layer multicast. The concept of multicast group was introduced to support multicasting delivery. Each group are assigned a unique address from IP class D address block. Multicast addresses are location independent, and they are in the range from 224.0.0.0 to 239.255.255.255 where each address defines the whole group, not a single host. So when a multicast source wants to send a datagram to a group of receivers, it simply sends it to the assigned group address. Group membership management is done on the network level through routers. The first experimental implementation for IP multi-

Table 1
Comparison between the two surveys.

Scope of comparison	Attributes	Gu et al. (2015)	This work
Background	Multicasting history	Brief definition of multicasting concept was mentioned	The history of multicasting revolution is discussed in Section 2.1
	SDN Architecture	✓	✓
	Multicast Mechanisms	×	Different mechanisms are discussed in Section 2.3
Multicast tree planning and management	Tree planning	×	✓
	Tree Construction	×	✓
	Tree management	×	✓
	Tree packing	✓	✓
Multicast routing	General multicast routing concept	✓	✓
	Reliability in routing	×	✓
	Routing for special conditions	Scalable routing for SDN was discussed	Different routing scenarios were discussed in Section 5
Multicast Traffic Engineering		×	✓
Multicast in Data Centers		✓	✓
Future research directions		×	✓

Table 2
List of acronyms.

ALM	Application Layer Multicast
ASIC	Application-Specific Integrated Circuit
BST	Branch aware Steiner Tree
CDN	Content Delivery Network
DCNs	Data Center Networks
DVMRP	Distance Vector Multicast Routing Protocol
EAM	Elastic Multicast
FRR	Fast Reroute Protocol
IGMP	Internet Group Management Protocol
IPTV	Internet Protocol TV
ISP	Internet Service Provider
LLDP	Link Layer Discovery Protocol
MCF	Minimum Cost Forest
MOSPF	Multicast Open Shortest Path First
MPLS	Multi-Protocol Label Switching
MST	Minimum Spanning Tree
MVN	Multicast Virtual Network
NBI	North Bound Interface
NFV	Network Function Virtualization
PIM	Protocol Independent Multicasting
QoS	Quality of Service
RSVP	Resource Reservation Protocol
RTP	Real-time Transfer Protocol
SBI	South Bound Interface
SBT	Source Based Tree
SDM	Software Defined Multicasting
SDN	Software Defined Networks
SPST	Sequenced Packet Shortest Path Tree
SPT	Shortest Path Tree
SVC	Scalable Video Coding
STP	Steiner Tree Problem
TCAM	Ternary Content Addressable Memory
TE	Traffic Engineering

cast was done on MBone, which is a testbed for deploying IP multicast. Many multicasting protocols where developed and tested on MBone including: DVMRP (Distance Vector Multicast Routing Protocol), PIM (Protocol Independent Multicasting), Multicast Open Shortest Path First (MOSPF), and RTP (Real-time Transfer Protocol) (Yamamoto, 2003).

In spite of its high forwarding efficiency, IP multicast is challenging to scale up to the current Internet structure where this solution was not capable to scan more than an individual network island. This is due to many challenges like: the dependence on internet infrastructure and the involved stakeholders, rapid resource-consumption of network devices, and technical challenges of the solution (Rückert et al., 2015; Gu et al., 2015). An example of the technical challenges is that all routers must be replaced with multicast-enabled routers, which is difficult to achieve. Application layer multicast overcomes these limitations by transferring the multicast functionality to the application layer instead of network layer. This is done using Content Delivery Networks (CDNs), where the multicasting is achieved on the application layer level without involving the network layer. That is, the data streams from content providers is spread among CDN nodes, which in turn act as server nodes and deliver the data-streams to the clients in Client-Server communication model. IP datagrams are sent to multicast groups using virtual addressing space that is determined by application level routing (Yamamoto, 2003).

Although CDNs is more applicable to current Internet structure, Internet Service Providers (ISPs) consider CDN traffic as unpredictable traffic inside their well-controlled networks. Moreover, it is less efficient than IP multicast from bandwidth perspective since it depends on unicast delivery of data from CDN nodes to the clients (Client-Server model) (Rückert et al., 2015).

Software Defined Multicast (SDM) was introduced to overcome the difficulties in the previous mentioned solutions. This is achieved by applying SDN techniques to provide efficient and well-managed multicast delivery services on network level within the ISPs networks while being transparent to the clients. Transparency here applies that clients at the end systems receive the multicast stream in the form of normal IP packets that are not distinguishable from unicast packets that are received directly from the content provider.

2.2. General architecture

For a better understanding of Software Defined Multicast, the SDN archi-tecture must be understood clearly. The architecture consists of three layers: the application plane, control plane, and data plane as illustrated in Fig. 1.

- Data plane: represents the network infrastructure where the different networking devices reside. Those devices are simple forwarding elements without embedded control logic or network intelligence. The forwarding decisions and control policies are determined by the controller and installed at the forwarding devices as flow rules that are held in a flow table. The communication between the control plane and data plane is enabled through South Bound Interfaces (SBI). OpenFlow is one of the well-known interfaces used in SDN (Masoudi and Ghaffari, 2016).
- Control plane: is the middle layer where the control logic is implemented. The controller plays the main role in managing the network

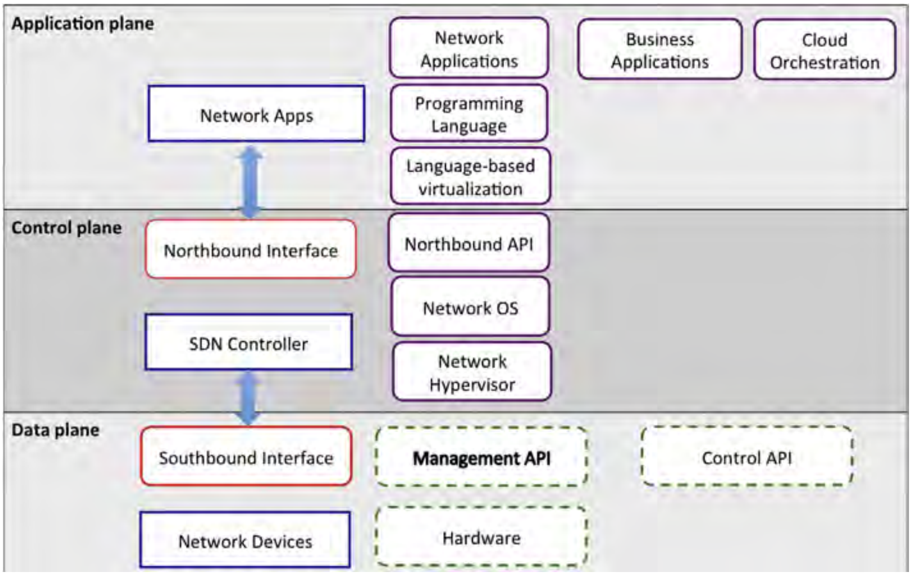

Fig. 1. SDN architecture.

and implementing the network functionalities including Multicasting in SDM. It communicates with the applications layer through North Bound Interface (NBI) to receive SDM clients high-level requests such as: creating new multicast groups or managing the membership of the existing groups. The forwarding and control decisions are then translated to flow rules that are passed to the devices in the data plane through SBI.

- Application layer: is the layer at the top of the SDN architecture. This is where all the business applications are located. Those applications represent SDM clients who need to send multicast streams where they communicate with the controller through NBI.

2.3. SDM different mechanisms

Logically, a multicast group refers to the group of receivers that should receive the same data content from a common sender. This group can be represented as a tree data structure where the sender or the source is usually connected to the root of the tree and the receivers are attached to the leaf nodes. This tree is either established as a Source Based Tree (SBT) or a shared tree. In SBT a tree is formulated for each source by computing the shortest path from that source to all destinations. While in the shared tree approach, one tree is computed and shared among all sources and destinations within the group. This approach can be classified to core-based tree algorithms and Steiner Tree (ST) based algorithms (Sheu et al., 2015). Special types of Steiner tree may be constructed due to specific network requirements. Branch-Aware Steiner Tree (BST) is an example of Steiner trees that aim to increase the scalability, that is, it finds the minimum summation of the number of edges and branch nodes in a tree. Reducing the number of branch nodes allows more support for BSTs in SDN when compared to ST. To solve the BST problem, a k-approximation algorithm was introduced in Ref. Huang et al. (2014), which is known as Branch Aware Edge Reduction Algorithm (BAERA) and will be deployed at the controller side. The algorithm works in two phases, Edge Optimization phase and Branch Optimization Phase. According to simulation results, BAERA was proven to produce trees that contain fewer edges and branch nodes compared to shortest path tree and Steiner Tree. More details about tree constructing and management algorithms are discussed in the later sections.

On the network level, when a new multicast group is registered the ISP will assign a group socket to the group source, where the socket consists of an IP address and a port allocated by the ISP. This socket is used by the group source to send multicast data stream, where this stream is delivered by ISP in form of unicast packets that follows normal IP routing procedures. The multicast delivery steps take place as illustrated in Fig. 2. After reaching the ingress switch of the ISP, a unicast to multicast translation takes place by matching the packets using OpenFlow flow entry that was installed by SDM controller upon group registration. The packets of an individual group are either identified using the group socket information or using an internal group identifier that is presented as a packet header field marked by the ingress switch. In addition to uniquely identifying the group, the group identifier is also used to install forwarding and duplication rules at all switches involved in a group multicast tree.

After traversing the SDM domain, packets arrive at the egress switches where the multicast-to-unicast translation takes place before being delivered to the individual clients. That is, the packet header is re-written where the group identifier is removed and replaced by the SDM client IP address and port. Moreover, packet duplication may take place at this point according to the duplication rules that are defined by the multicast tree. This is the traditional SDM model, however, different models have been proposed to improve the limitations of traditional SDM, which are classified below.

2.3.1. Adaptive SDM

SDM was introduced as a multicast technique that reduces the bandwidth requirements of content providers when compared to Application Layer Multicast (ALM) solutions. However, SDM was designed for scenarios with a small number of large multicast groups. Therefore, it does not scale well for applications that have a large number of smaller groups like video/audio conferences and web radio. This is due to the high amount of network state that must be obtained for each group regardless of its size.

The authors in Ref. Prithviraj et al. (2016) proposed the Adaptive SDM model, which improves the original SDM concept by allowing the ISPs to dynamically select the depth of multicast trees. That is, the ISPs can control where in the network the multicast-to-unicast translation takes place. Two translation strategies were introduced in Adaptive SDM, where the selection of one strategy depends on the tradeoff between bandwidth consumption and amount of network state. “Late Duplication” is the strategy that results in low bandwidth consumption and higher number of flow rules at each switch and its similar to the original SDM where translation takes place at the egress switch. This is suitable for large multicast groups where the traffic reduction is more important than the space at each switch. In the other strategy “Early Duplication” translation is performed at the ingress switch. This effectively reduces the required network state per client to a single

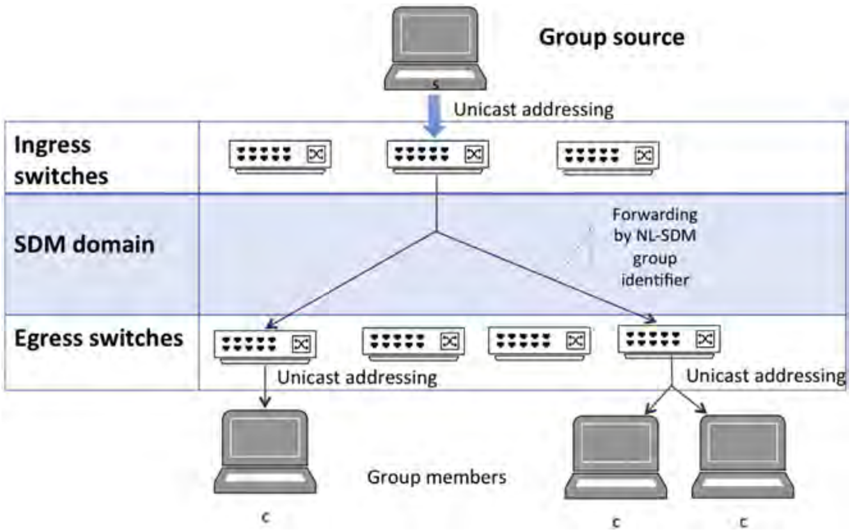

Fig. 2. Multicast delivery steps in SDM.

switch; however, the amount of traffic inside the network is noticeably increased (Rückert et al., 2016; Prithviraj et al., 2016).

2.3.2. Dynamic SDM

One of the limitations in SDM model is that it does not allow ISPs to manage the multicast streams delivery along with the rest of their unicast IP traffic. In Ref. Rückert et al. (2015) Dynamic SDM (Dyn-SDM) was introduced as a complete practical solution to be adopted by ISPs. Dyn-SDM provides traffic-engineering support, that is, planning and constructing the multicast trees will be achieved according to QoS parameters that are defined by ISPs. Moreover, Dyn-SDM provides mechanisms for dynamically connecting and dis-connecting clients to delivery trees. This will enhance the ability of handling dynamics such as changing client population and link failures. In addition, Dyn-SDM provides a multi-tree approach that will result in better traffic distribution over the links. Also, an SDN-based service discovery feature is presented in Dyn-SDM where it allows discovering the Dyn-SDM services along the routing paths (Rückert et al., 2015; Ruckert et al., 2015).

3. Multicast tree planning and construction

3.1. Tree planning

A well-studied multicast tree planning is an essential requirement that leads to successful multicast delivery of data within the ISP network. The steps starting from tree planning until flow rules installation are summarized as below and illustrated in Fig. 3.

- Obtaining Graph model of the ISP topology.
- Calculating edge weights.
- Tree construction based on ISP preferences.
- Network Layer path setup.

At the first step, a weighted graph representing the topology of ISP network is constructed according to real-time information that are being provided and updated by the ISPs monitoring system. The Graph G = (V, E) represents each openflow switch/router as vertex v and each physical link between two openflow devices as an edge e between two vertices. The second step is to calculate and assign weight to each edge of the graph according to number of parameters. The ISP defines those parameters and they may reflect current traffic condition, QoS metrics, internal ISP policies, and network resources consumption. An example of weights calculation equation that was introduced in Ref. Ruckert et al. (2015):

$$weight_e := (K_1 \times bandwidth_e) + (K_2 \times utilization_e) + (K_3 \times delay_e) + (K_4 \times lossrate_e) + (K_5 \times failurerate_e)$$

The coefficients are used to define the importance of each parameter and its share of the final edge weight. The ISP has the full control to prioritize different QoS parameters according to the nature of applications and client requirements.

3.2. Tree construction: single tree approach

After defining the weighted graph model, the actual tree construction takes place using a tree construction algorithm. For this purpose, two requirements must be defined: the entry point and the list of group members. The entry point is actually represented by the group socket that was assigned upon group registration as mentioned in Section 2 of this paper.

After defining the required information, a graph algorithm is used to construct a tree that connects the entry point with all the clients of the group. The constructed tree could be a Minimum Spanning Tree (MST) or Shortest Path Tree (SPT), where the choice depends on ISP needs. A Minimum Spanning Tree is a one that results in minimum sum of weights of all the used edges. It s appropriate to use for network bandwidth optimization. A Shortest Path Tree algorithm finds the shortest path between the entry point and each member of the multicast group. This makes it ideal to apply on delay-sensitive applications since it will reduce the delay of multicast data delivery (Ruckert et al., 2015).

The problem of finding a MST or SPT is similar to the concept of the Steiner Tree Problem (STP) which is a fundamental design problem in networking area. Many polynomial time algorithms were developed to find near optimal solution for that problem (Gijare, 2016). Detailed explanation of STP along with different algorithms that can be applied to solve it can be found in Ref. Bezenšek and Robič (2014). However, many of the polynomial-time heuristic Steiner tree are not sufficient for large networks as they sometimes produce output values that are far from optimal. Therefore, population-based intelligence algorithms were proposed to overcome the limitations in polynomial-time algorithms. Examples of such algorithms are: the Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Artificial Fish Swarm Algorithm (AFSA). Although these algorithms are able to reach almost-optimal solutions, they are considered to be time consuming. Multi Agent branch based multicast (BBMC) algorithm was introduced in Ref. Matsuura (2016), where it can match the optimality of intelligence algorithms while keeping the fast speed of polynomial-time algorithms (Matsuura, 2016). A Path based Harmony Search Algorithm (PVHS) was proposed in Ref. Zhou et al. (2015) to solve degree-dependent branch-node weighted Steiner Tree problem. This problem aims to minimize

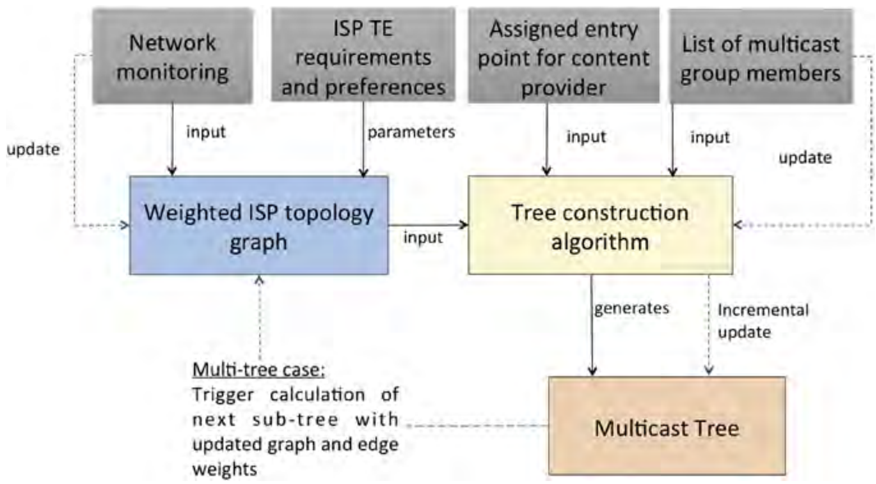

Fig. 3. Multicast tree planning and construction steps.

the total cost of edges and branch nodes.

After constructing the tree, the actual network topology is updated, and flow rules are programmed in all the involved network devices.

3.3. Tree construction: multi-tree approach

Some of the limitations of using a single multicast tree per group are the un-even distribution of load, and the lack of dynamicity in response to failures and change in client population (Ruckert et al., 2015). To overcome those limitations, a multi-tree approach was introduced, where the data stream is divided into sub-streams each carried by an independent tree. That is, for each multicast group there are multiple independently built sub-trees instead of a single multicast tree that connects the group sources to the clients. This overcomes the above-mentioned limitations and makes it possible to apply traffic engineering methods on specific part of the stream since it is delivered independently.

The initial tree planning and construction procedure remain the same as dis-cussed before. After defining the group entry point and list of clients, the first tree is constructed. Then, more trees connecting that point to the group members are generated iteratively. Whenever a new tree is added, previously constructed ones are updated incrementally by changing edge weights or even removing some of them if necessary. The whole procedure is illustrated in Fig. 3. A weighted ISP network topology graph is constructed based on the input received from the network monitoring tools, and the parameters defining ISP Traffic Engineering requirements and preferences. The graph is used as in input for the tree construction algorithm along with other inputs: entry point for content provider, and list of multicast group members. The algorithm generates the multicast tree based on the received input and incrementally updates it whenever group membership updates occur.

The literature works (Sun et al., 2016; Jiang and Chen, 2016) discussed detailed approaches for constructing multiple multicast trees. After constructing the multicast tree, an adjustment to the paths may be required due to changes in network conditions which may degrade the whole multicasting performance. This may be difficult to achieve in case of unchanged group source and destinations. The authors in Ref. Ge et al. (2013) introduced an openflow-based mechanism that dynamically adjusts the paths of a multicast tree to achieve better multicasting performance. Moreover, a distributed tree construction strategy was introduced where its inspired by the dynamic path adjustment mechanism. More tree management concepts and approaches are discussed in the next section.

4. Tree management

After constructing the multicast tree, an efficient management method must be defined to control the procedure of initiating and updating flow rules. It must be taken into consideration that the process of modifying those rules and computing the multicast tree architecture requires a significant amount of time that may affect the performance of the multicast application. Therefore, its important to define an efficient management procedure that is able to adapt to the frequent changes in the network environment such as network failures and group membership modifications. In SDN-based environment, all management functionalities are implemented in the logically centralized controller. The controller obtains the state of network links and nodes in addition to group membership status by communicating with open-flow switches and applies the management procedure accordingly. Fig. 4 shows the communication scenario between the controller and the switches.

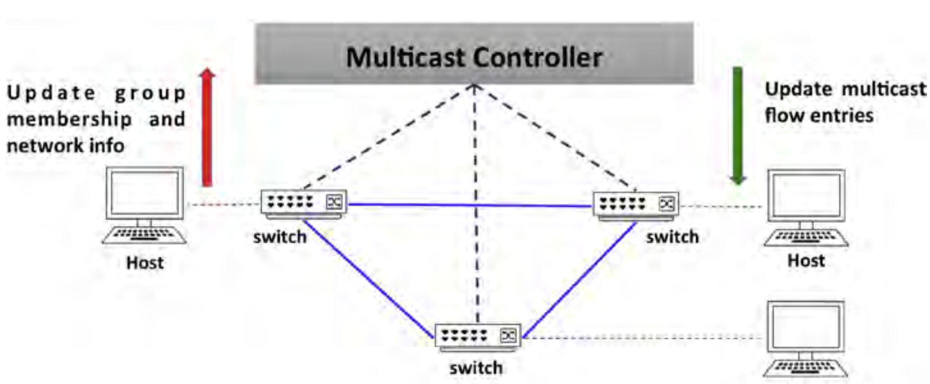

Fig. 4. Relationship between switches and multicast controller.

4.1. Obtaining network status

All the needed information to setup the trees are stored into the controllers database. This database can be divided into tree database and group membership database. The tree database is responsible for holding the multicast trees with the status of each tree (active or inactive). Its organized as follows: at the top-level record the group entry is stored, where group entries are defined by (Source IP address, multicast address) pair. Each group entry contains the list of trees within that group where a unique ID defines a tree. The tree entry contains a list of tree components (nodes, edges) in addition to the state of each one of them. In addition to that it maintains pointers that map the nodes and edges to the physical network components such as switch ID and port number.

On the other hand, the group membership database could be divided into senders database and receivers database. The senders database has a record for each sender that consists of the switch ID and the port number of the root switch where that sender is attached. The controller either lookups this database for a specific sender location using the IP address of the sender and the multicast address of the group, or it retrieves a list of all senders within a specific group using the multicast address of the group. The receivers database stores a list of receivers where receiver record contains the switch ID and the port number of the leaf node where the receiver is attached. Similar to the above, the receivers database is either used to fetch a list of all receivers within a group or to update the status of one receiver where its location is retrieved by sender IP address, multicast address, switch ID and port number.

4.2. Managing group membership

Any change in group membership will result in new calculations at the controller side to modify the trees where the change is reflected, in addition to updating the flow entries at the involved switches. Therefore, frequent membership changes will cause a heavy load on both the controller and the network switches.

The authors in Ref. Kotani et al. (2016) proposed a pre-planned management approach that helps to reduce the load and shorten the needed time to handle group membership changes. That is, when a new sender and at least one receiver appears in the network, the controller calculates and stores at least two trees that cover all the leaf switches where the receivers are attached. Those multiple trees per group are stored in the tree database in the controller and are removed when none of the sender or receiver exists anymore.

After that, whenever a membership change (join or leave) happens at one of the leaf switches, the below scenario will take place:

- The group membership database will be modified including all switches and ports that are affected by the change.
- Compute the updated multicast trees.
- Update the flow entries in the involved switches accordingly.

The most time-consuming tasks that this approach aims to minimize are calculating the trees and updating the switches flow entries. This is achieved by the database hierarchal design that enables quick look up for the leaf node where the receiver is attached or removed. After locating that node, the controller tracks each stored tree and then modifies all flow entries of the involved switches within that tree.

4.3. Handling failures

In addition to group population changes, the multicast mechanism must be able to recover from failures that may exist within the network. Those failures could happen at the network nodes (switches, routers) or the edges (links) that connect those nodes. Multi-Protocol Label Switching (MPLS) fast re-route is a well-known mechanism that sets back-up routing paths in advance and re-route the packets on those paths in case of a failure. However, in this approach the parent router of the failure source (router/switch or link) will be responsible for re-routing the traffic, which limits the flexibility of back-up trees. In the pre-planned approach introduced in Ref. Kotani et al. (2016) for each multicast group there is an (active) tree that is currently being used for data delivery and a number of back-up (inactive) trees. A unique ID identifies each tree where the root switch will embed the ID of the active tree in the packet header to be used for packet delivery. That is, the next switches forward packets based on the tree ID, source address, and multicast address. In case of a failure, the network devices at the data plane send a notification to the controller. It in turn checks whether the currently active tree has been disconnected. In case of tree disconnection, the tree recovery procedure is applied as follows:

- The controller chooses one back-up tree that is not affected by the failure, selecting a certain back-up tree depends on pre-defined criteria based on network preferences.
- The flow entry in the root switch is updated, such that the ID of the newly selected tree is embedded in packet headers instead of the earlier faulty one.

After doing so, any algorithm could be applied to calculate back-up trees for the new active tree. Moreover, the controller only needs to update flow entries at the root switch and then start multicast delivery without updating entries in other related switches.

4.4. Tree packing

Tree packing problem can be defined as systematically scheduling a series of multicast sessions to locate available routing for each multicast session (Ren et al., 2017). Traditional multicast tree packing solution depends on reducing the total link cost based on Minimum Steiner Tree. That is, a limited amount of reserved bandwidth is used to concurrently serve as many multicast sessions as possible. However, the limited amount of reserved bandwidth reduces the possible utilization of network resources (Gu et al., 2015).

SDN approach has introduced the possibility of full packing for multicast trees. That is, all network resources could be utilized to accommodate the needs of multicast groups without specifying a limited amount of bandwidth. This is because of SDN features that enable monitoring real-time links traffic and global adjustment on the routing algorithm to host new multicast groups and manage group membership at running time (Gu et al., 2015).

However, many applications apply content-replica designed to achieve better robustness and efficiency. Examples are content distribution networks, IP television networks, and data center networks (Ren et al., 2017). In such designs, each multicast session may have multiple potential sources; therefore each destination node has the chance to select any node in the replica as its source. This kind of multicasting is referred to as uncertain multicast. The routing structure in such multicast is a Minimum Cost Forest (MCF), which consists of multiple disjoint trees where each tree is rooted at different sources (Ren et al., 2017).

The input to the uncertain multicast problem is the network topology along with the set of destinations. The output would be a constructed forest with minimum summation of costs for all edges. The authors in Ref. Hu et al. (2016) proposed an uncertain multicast routing scheme that is designed for SDNs. Their approach is to divide the uncertain multicast problem into a set of smaller deterministic multicast problems, where in each one the minimum Steiner tree is found for a specific source. This is a challenging to solve since its equivalent to solving a set of NP-Hard problems (Hu et al., 2016). Later on, the authors in Ref. Ren et al. (2017) discussed the packing problem for uncertain multicast. In such multicasting, it is impossible to satisfy all the existing uncertain sessions with their optimal MCFs. One of the limitations is the link capacity constraint. That is, in practice, the available capacity of each link from source to destination may not be enough to accommodate all the simultaneous multicast sessions. In case one of the links was blocked while constructing the optimal MCF, the building algorithm must find an alternative routing path to accommodate the traffic (Ren et al., 2017).

The paper (Ren et al., 2017) presented the packing problem of uncertain Multicast (MPU) to minimize the overall cost of all multicast sessions while taking the link capacity constraint into consideration. That is, given a network that has a capacity constraint on each of its links, a multicast forest is constructed for each ongoing multicast session while optimizing the total cost of all forests instead of reducing the cost for each individual forest. An earlier work has introduced an efficient method E-MCF for a single uncertain multicast; however, the MPU studies the packing problem for multiple multicast sessions.

This MPU problem is proven to be NP-hard, therefore there is no exact optimal solution for it. However, approximation methods may be used to derive an approximate solution. The authors in Ref. Ren et al. (2017) derived two approximation methods: Based on priority, and by adjusting congested links. The priority based method assigns a priority for each multicast session and serve the multicast requests according to that priority. That is, when packing a set of uncertain multicasts, the sessions with higher priorities are served first. The higher priority is assigned for multicast trees with more destination nodes and less source nodes. This is because the sessions with more sources have better performance in finding new route and utilizing network resources. The second method initially calculates the MCF for each multicast session. After constructing the forests, an approximation method is used to adjust the overloaded links. That is, the congested links are deleted from the MCF and then the end points are reconnected with an alternative shortest path.

5. Multicast routing

Routing in SDN is more challenging than in traditional networks due to the difficulty of network traffic aggregation. Moreover, the majority of multicast applications require high bandwidth and lower delay. Therefore, choosing an optimal multicast routing approach in SDN is critical to fulfill the bandwidth and delay requirements of multicast applications as well as applying QoS parameters based on the current availability of network resources. That is, multicast streams should not cause blockage of other traffic in the network (Huang et al., 2016a; Gu et al., 2015).

As discussed in the previous section, a minimum Steiner tree is constructed to connect the data source to all the members of the multicast group. After that, two control components play major role in routing: Internet Group Management Protocol (IGMP) and multicast routing protocol, PIM is an example of commonly used routing protocol in the Internet (Fan et al., 2016). These two components reside in the control plane of SDN and are considered as part of the SDN controller.

IGMP is a communication protocol that is responsible for managing the relationship between routers and hosts. That is, the requests for joining and leaving multicast groups are sent as IGMP join and IGMP leave messages, respectively. If a join message reaches an edge switch and the multicast group was already registered (if flow rules of the group address exist in the switch flow table), it simply adds the input port address which has first received the join request to the group of output ports of the targeted group. Otherwise, the request are forwarded to the controller, the controller in turn runs the Steiner tree algorithm to add the new edge switch and update the flow rules in the affected switches. After that, routing is done on the forwarding devices by simply checking the installed flow rules and forwarding the packets accordingly. The same concept is applied in case of IGMP leave request (Huang et al., 2016a; Xu et al., 2015). In SDN networks, the routing rules along with group membership information are exchanged between the switches and the controller in form of OpenFlow Protocol messages (Xu et al., 2015).

A number of constraints must be taken into consideration when determining the routing rules. For example, the authors in Ref. Huang et al. (2016a) introduced an SDN-based multicast routing algorithm that aims to optimize bandwidth utilization. That is, multicast traffic will be routed based on the current available network bandwidth. To do so, sFlow (Phaal, 2004), which is a network monitoring technology that provides detailed network traffic information, is used to calculate bandwidth accurately.

Fig. 5 shows the proposed routing architecture in Ref. Huang et al. (2016a). The Topology manager at the control plane is where the net-

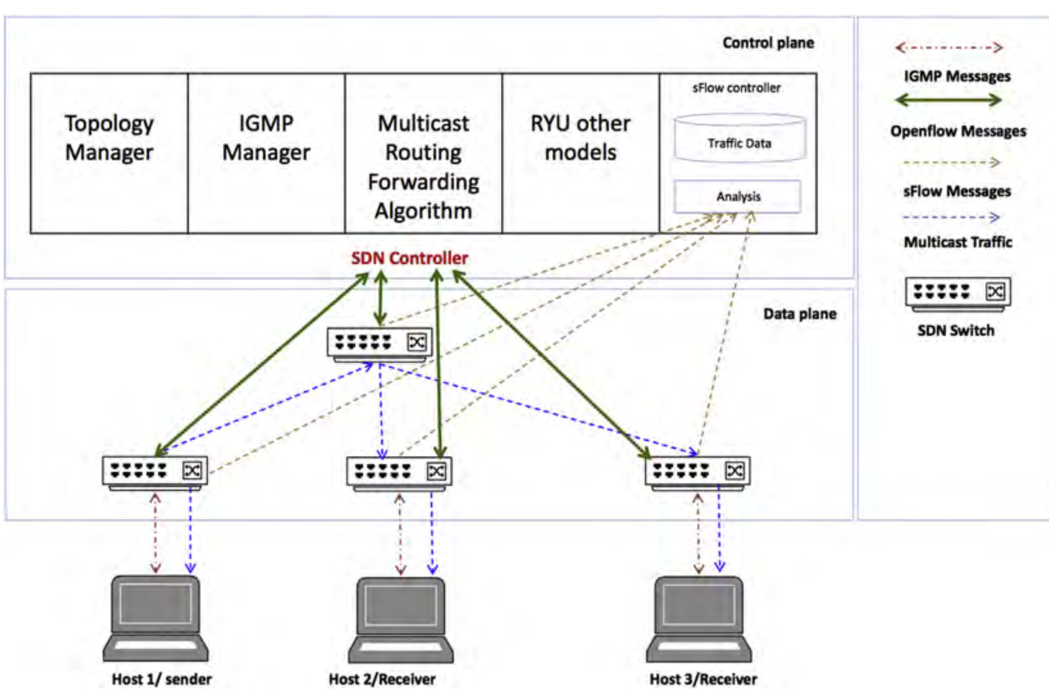

Fig. 5. Routing system design.

work topology is maintained where as Link Layer Discovery Protocol (LLDP) is responsible for discovering new devices in the network. The other important control component: IGMP manager, will control the procedures for joining and leaving multicast groups as discussed earlier.

The proposed algorithm takes links bandwidth constraint into consideration, however it does not consider the limited forwarding table capacity. This limitation is due to using Ternary Content Addressable Memory (TCAM) memory for storing the forwarding rules at the SDN switches. This memory is expensive and power hungry therefore it imposes limited size of forwarding tables. The authors in Ref. Huang et al. (2016b) proposed a dynamic multicast routing scheme for network throughput maximization in SDNs taking both link bandwidth and forwarding table size along with user bandwidth requirements into consideration.

The basic idea of the algorithm is to apply an admission control policy that either admits or rejects each incoming multicast request based on a threshold value configured on the node and link resource consumption. Whenever a new multicast request k is initiated, the summation of all weights for node derived and link derived edges will be calculated for the corresponding multicast tree s_k. The summation is then compared against the threshold value to determine whether to admit or reject the request (Huang et al., 2016b).

Delay optimization is another parameter that is more important for some types of traffic such as video data streams. A delay-optimized routing system was introduced in Ref. Liu et al. (2016), where its dedicated for software-defined inter-Data Center networks.

5.1. Network function virtualization enabled routing

Many multicast applications involve intermediary processing of the traffic before its being delivered to the end users such as video transcoding and packet inspection. In such application Network Function Virtualization (NFV) may be applied to implement network functions using virtualization techniques. That is, the required network functions are implemented as software that runs on the top of network component referred to as (NFV) node. In such applications, a multicast mechanism is required to determine the placement of NFV nodes and routing path. That is, a multicast tree must connect the source to every client through an NFV node, which is challenging to achieve.

The authors in Ref. Zhang et al. (2015) proposed an NFV multicast routing system for SDN net-works, where the controller will be responsible for locating the NFV nodes and choosing the optimal routing paths. The proposed algorithm is a single layer mechanism on SDN that could be applied for both static and dynamic multicast.

5.2. Routing with sequenced packet transmission

A well-constructed multicast routing tree is the one that is able to reach all possible destinations with minimum delay while utilizing network resources efficiently (e.g., minimum bandwidth consumption). Most routing algorithms therefore aims to find minimum-cost tree: Minimum Spanning Tree, Shortest-path tree, or Steiner tree. However, in a sequenced packet transmission scenario, the least cost path is not necessarily the least time cost. Sequenced packet transmission routing is the case when a network node cannot send a packet until the node on the other side of the link has received the previously sent one. This scenario is found on ns-3 open source network simulator but its not limited to this environment (Yu et al., 2016). The authors in Ref. Yu et al. (2016) have studied the problem of finding Sequenced Packet Shortest Path Tree (SPST). Assuming that all network nodes send packets one by one, and there is at most a single packet on any link between two nodes at a given time. The proposed algorithm aims to construct a multicast routing tree such that the amount of time by which all the receivers have received all data is minimized. In such situation, the total time delay cannot be simply estimated by summing all the involved links delays. The authors have therefore design a shortest path algorithm that extends Dijkstra's algorithm while introducing new cost models for link costs and routing paths. The proposed algorithm was then applied to SDN environment and simulation results has proven that the amount of enhancement in data-transfer time has exceeded 10% (Yu et al., 2016). The authors in Ref. Reed et al. (2016) proposed a SDN-based multicasting technique based on Bloom filter in order to reduce the TCAM state size. Table 3 shows a comparison of the previously mentioned routing techniques.

5.3. Reliability in routing

Failures can greatly affect the quality of real-time multicasting services. Node or link failures in an ongoing multicast session cause delays or even packet loss. Therefore, a multicast protection technique is one of the key requirements to achieve reliable multicasting.

Generally, multicast protection is either done using proactive or reactive techniques. When applying reactive techniques, the backup paths are calculated only upon failure detection. This results in a long recovery time, which is undesirable for real-time streaming applications. On the other hand, proactive methods calculate and configure the back up paths or trees in advance before the occurrence of a failure (Raja et al., 2016). However, re-calculating those trees each time a new receiver joins or leaves a multicast tree results in limitations in terms of the scalability and flexibility of the approach. The work (Wei et al., 2010) applied the same backup tree concept but for a sub-tree instead of the entire tree which lowers the complexity and increases the flexi-

Table 3
Routing techniques comparison.

Reference	Technology Used	Objective	Environment/Applications
Huang et al. (2016a)	sFlow, LLDP, IGMP manager	Optimize bandwidth utilization	SDN-based multicast
Huang et al. (2016b)	Admission control policy	Network throughput maximization taking both link capacity and table size into consideration	SDN-based multicast
Liu et al. (2016)	Differentiate path selection for different sessions based on delay sensitivities	Delay optimization	Transmitting video data streams in inter-data center networks
Zhang et al. (2015)	Optimal placement of NFV nodes along the routing path	Optimizing multicast routing paths while handling NFV nodes location at the controller side.	NFV multicast routing for SDN
Ren et al. (2017)	Divide the uncertain multicast problem into smaller deterministic ones.	Finding near-optimal Minimum Cost Forest for uncertain multicast	Multi-media distribution systems with content replica designs
Yu et al. (2016)	Extended Dijkstra's algorithm	Finding Sequenced Packet Shortest Path Tree (SPST)	Sequenced packet multicast in SDN
Reed et al. (2016)	Bloom filter	Minimizing the state size in TCAM	SDN-based multicast

Table 4
Reliability Techniques summary.

Reference	Technique	Proactive	Reactive	Methodology
Shen et al. (2015)	Packet re-transmission		✓	Acknowledgement messages
Shen et al. (2015)	Recovery nodes	✓		Temporal cache/proxy server
Shen et al. (2015)	Recover-Aware Steiner Tree (RST)	✓		Find minimum cost Steiner tree that spans maximum amount of recovery nodes
Zhang et al. (2016)	Elastic-Loss recovery solution (Ecast)	✓		Elastic local multicast
Li et al. (2014)	Reliable multicast protocol for data center networks (RDCM)	✓		Construct backup overlay to send repair traffic in case of loss

bility of the approach. The authors in Ref. Raja et al. (2016) extended (Wei et al., 2010) sub-tree approach to be applied for SDNs while the earlier work (Wei et al., 2010) was designed for traditional networks. The backup paths are calculated by the controller, which is much more efficient than the Resource Reservation Protocol (RSVP) that is used in traditional IP networks. Moreover, the high complexity and control overhead of RSVP and Fast Reroute (FRR) protocols are eliminated. The experimental results showed that the sub-tree approach achieved a good restoration time from failure detection point of view.

Packet re-transmission is a well-known reactive solution for packet loss that occurs during data delivery. The source re-transmits the data if it did not receive an acknowledgement message indicating that the packets has not reached the destination. This solution is simple to implement and can reduce the amount of packet loss, however, it’s not scalable for a multicast group with large number of destinations. That is, the source could be overwhelmed by the huge amount of acknowledgement messages and the required re-transmitted traffic (Shen et al., 2015). Moreover, this scheme may result in recovery redundancy problem. To reduce the burden on the source node, the idea of recovery nodes was introduced. A recovery node can be defined as a cache/proxy server that stores small amount of packets temporarily, where they are placed between the source and destinations to support local loss recovery. The distance between the destination and the recovery node defines the recovery cost, where closer nodes imply less cost. However, reducing the distances means including more recovery nodes within the multicast tree, which results in increasing the tree size and the caching overhead. Therefore, its necessary to find a solution that is able to find minimum cost routing tree while selecting optimal amount of recovery nodes that can support reliable delivery of data with least recovery cost.

The authors in Ref. Shen et al. (2015) proposed the Recover-aware Steiner Tree (RST) model, which aims to solve the Reliable Steiner Tree problem. The inputs to this problem are: source and destinations of a specific multicast group, number of nominations for recovery node, a pre-defined non-negative integer r. The output is a multicast tree that connects the source to all destinations and spans maximum number of r recovery nodes within the tree.

The main contribution was to reduce both the tree cost and the recovery cost of the tree. The tree cost is the summation of all edges in the tree, while the recovery cost is the cost of the path from a destination v to its local cache u. The parameter r controls the tradeoff between the recovery cost and the caching overhead. The SDN controller tunes the value of r according to the application requirements (Shen et al., 2015). The authors in Ref. Popovic et al. (2017) compared the performance of three different algorithms for constructing node-redundant multicast trees in SDN-based networks in order to evaluate the number of forwarding rules and the effects of nodes failures.

The authors in Ref. Zhang et al. (2016) have introduced an Elastic loss recovery solution (ECast) that was designed to minimize the recovery redundancy while achieving good multicasting scalability. The re-transmitting method is done using a communication model known as Elastic Multicast (EAM). Elastic Multicast utilizes the flow-matching feature of OpenFlow switches to mitigate the redundancy that may occur during the re-transmission of data. That is, in case of packet loss, EAM performs elastic local multicast using any sub-tree of the multicast tree. The re-transmitted packets can be sent to a sub-set of the multicast group receivers where this subset covers only the receivers that were affected by the packet loss. This is not the case in traditional packet re-transmission schemes where recovery packets are either delivered to receivers one by one or distributed as a multicast to the whole group of available destinations. Compared to these schemes, EAM minimizes the recovery cost and therefore redundancy. Moreover, ECast supports using recovery nodes, which improves the scalability.

Most of the existing reliable multicast solutions are not applicable for data center networks environments. Such networks have high link density therefore traffic rate are typically high. This results in higher packet loss rate, which in turn causes degradation in the multicast session throughput. The authors in Ref. Li et al. (2014) introduced a reliable multicast protocol for data center networks (RDCM), which aims to limit the amount of throughput degradation that is caused by packet loss.

RDCM takes advantage of the high link density by constructing a backup overlay that is used to send repair traffic in case of packet loss. This traffic is sent on peer-to-peer mechanism and mostly avoids the congested/failed link that caused packet loss within the multicast tree. In addition to throughput enhancement, RDCM also achieves congestion control, tree adjustment, and handling failures with low individual overhead. Table 4 shows the summary of the above discussed reliability mechanisms.

6. Multicast traffic engineering

Traffic Engineering (TE) is one of the most challenging topics in communication networks. The term refers to applying scientific principles and strategies on operational networks to achieve optimal performance. That is, traffic is routed throughout the network such that traffic demands are met while certain performance objectives are optimized. Those objective are as in the below list, and they are selected for each network based on the nature and applications of the network since some of them may be contradicting:

- Congestion minimization
- End-to-end delay minimization
- Packet loss minimization
- Energy consumption minimization
- Resource utilization minimization

The ease in network management, high network programmability, and centralized control logic in SDNs are all factors that support applying powerful TE strategies. Out of this point, most ISPs, Research Education Networks, and Telecom Companies are currently heading to adopt SDN techniques for their networks. Examples include Energy Sciences Network (ESnet) in the United States, AT& T, and pan European research and education center (Mendiola et al., 2017).

The importance of Traffic Engineering is that it plays a key role in providing the required network services with Quality of Service (QoS). For example, energy consumption minimization is one of the key TE objectives, which is widely used in the scope of green computing in order to reduce the impact of ICT on the environment. Generally, the routing protocols plays an important role in this TE objective for the energy management in the network (Baker et al., 2013; Baker et al., 2015). When it comes to multicasting applications, achieving QoS add more complications to the multicasting problem as it requires scalable and efficient network support and some applications may require more specific requirements related to end-to-end delay, delay jitter, and packet loss. During a multicast session lifecycle, three major events that affect QoS may take place: group dynamics, network dynamics, and traffic dynamics (Manimaran and Striegel, 2002).

The group dynamics management has a number of related issues such as: QoS-aware routing, tree re-arrangement, and core/tree migration. QoS aware routing can be defined as: given a new member M joining the group, the multicast routing protocol must find a path from M to any tree node while satisfying the QoS requirements of M. Those requirements may be link constraints (bandwidth consumption) or path constraint (end-to-end delay). In addition to these basic requirements, a QoS-aware routing protocol must be able to:

- Improve successful join probability.
- Reduce the cost of joining path and time.
- Has the ability to scale up to large networks.

For the tree re-arrangement, it is important to ensure that members join or leave do not disturb the current multicast session and that the constructed multi-cast tree still satisfies the QoS requirements for all receivers after a successful join/leave. Another important issue related to group dynamics is the core and tree migration. Since the selection of the core affects tree cost and delay and therefore the quality of the tree, it is important to ensure that this quality is maintained after dynamic changes related to group membership or failures.

Moreover, handling failures is a fundamental QoS issue. That is, multicast routing protocols must be able to discover and recover from link/node failures to keep an adequate QoS level. However, this is more challenging in multicasting than unicasting. This is due to the share resource reservation and group dynamics that result in network re-configuration.

Load balancing is another important traffic engineering objective for multicast traffic that was studied by many literature works. The paper (Fabregat et al., 2005) presented a novel taxonomy for traffic engineering load balancing where thirty-five publications were classified according to their objective functions, constraints, and proposed heuristic. Based on the classification results, a Generalized Multitree model (GMM-model) was presented, where this model can handle any type of flow and any number of flows in a general multi-objective context (Fabregat et al., 2005).

As explained earlier, the selected performance optimization objectives differ from one network to another depending on the nature/applications of the network. According to Mendiola et al. (2017), in SDNs its required to apply traffic engineering strategies to optimize the network performance in four different aspects: Scalability and Availability, Reliability, Consistency, and Accuracy.

- Scalability and Availability: Flow management

As mentioned earlier, each new flow results in generating a new flow entry at the flow table of the involved switches where installing these entries yield in some delay. Moreover, a high amount of newly generated flows cause a significant overhead at both the controller and the data forwarding devices. Therefore, a TE solution for SDNs must consider the tradeoff between latency and load balancing by prioritizing the application needs. Different load balancing techniques for SDNs along with the research challenges related to each one were discussed in Ref. Mendiola et al. (2017).

- Reliability: Fault Tolerance

Providing a failure recovery technique is essential to ensure the reliability of the network. In SDNs, since the control logic is separated from the data forwarding devices (switches). A switch is not be able to recover from a failure without receiving the updates from the centralized controller. The controller is also responsible for re-constructing the optimal routes and network topology for the current traffic. Different fault tolerance techniques that could be applied for data plane and/or control plane were introduced in Ref. Mendiola et al. (2017).

- Consistency: Topology update

Consistency in SDNs could be categorized into: per-packet consistency and per-flow consistency. Per-packet consistency ensures that each ongoing packet through the network must be processed according to unified network configuration. Per-flow consistency means that all packets within the same flow follow the same version of network policy.

- Accuracy: Traffic Analysis

Traffic analysis techniques in SDN include monitoring network traffic using a monitoring framework, checking network errors, and debugging program-ming errors. All these techniques along with associated research challenges were discussed in Ref. Mendiola et al. (2017).

The features of SDN have facilitated applying traffic engineering techniques. Since traffic monitoring, group management, and multicast routing functionalities are all implemented within the centralized controller, this enables flexible resources allocation according to network conditions, which in turn accomplishes traffic-engineering goals. However, when it comes to multicasting, the current Internet multicast standard adopts a shortest path tree approach to transfer data from the source to the destination, but the design of such a tree does not support traffic engineering (Craig et al., 2015). Steiner Tree is an alternative solution that is conducive to traffic engineering standards, however, its not adopted yet in current Internet standard (Huang et al., 2015). The authors in Ref. Craig et al. (2015) proposed an SDN controller architecture that supports multicast traffic engineering by applying real-time link cost adjustment to achieve better distribution of the traffic load among the links. Those link costs are then used by Dijkstraś shortest path algorithm for multicast tree calculations. Fig. 6 shows the proposed controller architecture. The component that is responsible for real-time bandwidth consumption is the FlowTracker module. The measurement is done periodically by polling all switches to query all existing flows on all switches.

The FlowTracker module captures a map of link utilization keyed by switches IDs and port numbers. These measurements are used by the GroupFlow model to adjust link costs in a way that directs the traffic away from congested links. Although this approach is effective in achieving better load balancing, the periodic polling causes undesired overhead in the control plane. A more scalable traffic engineering solution was proposed in Ref. Huang et al. (2015). Their approach has focused on optimizing the bandwidth consumption of all multicast groups in the network while taking the scalability limitations of SDM into consideration. According to Huang et al. (2015), the total possible number of multicast groups in a network with n nodes is $O(2^n)$. To overcome this limitation, the branch forwarding technique was introduced where the multicast flow entries are installed in branch nodes only instead of all nodes in the network. A branch node is the node that has a minimum of three incident edges and it can act as a branch state node that has an installed multicasting forwarding rule or branch stateless node that acts as any other forwarding element. Packets are forwarded from one branch state node to the next one in unicast tunneling fashion. That is the intermediate forwarding elements no longer needs to maintain forwarding entries in their flow tables.

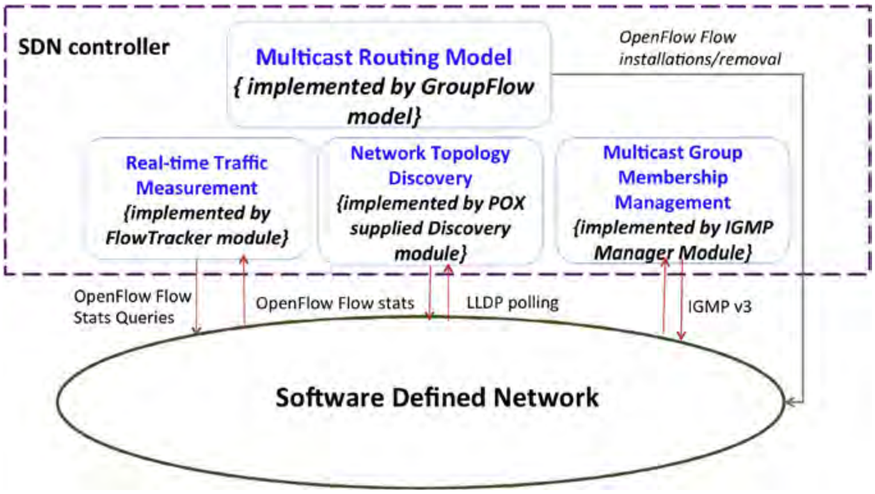

Fig. 6. Deployment diagram of the proposed SDN controller.

Two scalability constraints must be highlighted here: link capacity and node capacity. The link capacity constraint states that the total consumption rate of a link bandwidth by all multicast trees should not be higher than the assigned link capacity. The node capacity constraint specifies the size of group table in a branch state node such that its sufficiently large to accommodate all multicast trees.

The Scalable Multicast Traffic Engineering (SMTE) was introduced in Huang et al. (2015) where its contribution is to reduce the total bandwidth cost of all multicast trees. This is achieved by finding an optimum multicast tree for each group and allocating branch state nodes within that tree such that both link capacity and node capacity constraints are not violated.

Another scalable multicasting solution was introduced in Li and Freedman (2013). The approach was designed for scaling up the possible number of supported multicast groups within a large data center network. The main idea is to partition the multicast address space where switches are divided to cooperating sets instead of treating each switch as a stand-alone entity. Actually switch sets are either a group of core switches or upper layer switches, which is known as pod. Applying this partitioning technique allows a fat-tree network that is composed of 27 K servers with switches that holds a maximum of 1000 group addresses to support 4 K–30 K multicast groups.

This group capacity is further increased through local multicast address aggregation. Indirection and re-writing mechanisms were introduced to aggregate local groups into virtual meta-groups. Finally, link/switch failures mechanisms were introduced and implemented (Li and Freedman, 2013). A comparison between the discussed Traffic Engineering techniques is represented in Table 5.

7. Multicast in data centers

7.1. General multicasting solutions for data centers

Applying multicast in modern data centers is necessary due to the frequent group communication that takes place between servers and for cloud-based services (Baker et al., 2017). HDFS in Hadoop (2009) for distributed file storage, MapReduce in Hadoop (2009) for distributed data execution and Nova batch VM provisioning in OpenStack (Open, 2009) are some examples of applications that have inherent group communication patterns in servers. However, the structure of data centers must be taken into consideration when designing a multicast model. That is, modern data center topologies tend to have high link density due to the use of large number of low-end switches rather than using fewer high-end ones. This makes it inefficient to design a multicast scheme that applies common Internet multicast protocols such as IGMP and PIM. This is because such protocols do not consider the network topology and therefore cannot utilize multiple equal-cost links when constructing multicast trees. Moreover, applying traditional multicast protocols result in a scalability problem due to the limited space in the forwarding tables of the multicast switches (Cui and Qian, 2014). In general, a well-designed Multicast solution for Data Center Networks (DCNs) must fulfill the special requirements imposed by the features of group communications in such networks. Those features are as below:

- Small groups structure
- Reliable delivery of data
- Sender-initiated communication
- Efficiency
- Robustness

However, most of traditional multicast solutions fail to achieve all the above communication goals. That is, most solutions are receiver-initiated protocols, which is not compatible with the sender-initiated mode. Moreover, the solutions mainly focus on finding shortest path trees without considering links congestion, which may significantly affect the performance of multicast delivery. In addition to that, most of these solutions do not have a mechanism for failure discovery (Zhu et al., 2016; Zhu et al., 2017).

The evolution of SDN-based multicast has introduced new scope of multicasting solutions that are able to fulfill the special requirements of multicasting communication in data centers. Avalanche and OFM are well-known SDN-based solutions that utilize the SDN controller abilities to improve the management. Avalanche, which was presented by the work in (Iyer et al., 2014) enables multicast routing in the commodity switches through a new routing Algorithm called Avalanche Routing Algorithm AvRa. This algorithm aims to provide a near optimal solution for the Steiner tree multicast problem (Gu et al., 2015; Fan et al., 2016). However, the centralized nature of the algorithm has some limitations such as the single point of failure and scalability problems.

As mentioned above, the communication requirements in DCNs led to developing specialized multicasting schemes based on the needs of these networks. MCDC is one solution introduced in Ref. Shukla et al. (2016) that utilizes the multiple equal-cost paths feature to reduce the congestion on links. That is, the load on the link will be taken into consideration when choosing the routing path.

Another congestion control was proposed in (Akamatsu et al., 2016) where the transmission rate at the sender side is adjusted according to the available throughput at the most congested receiver. The throughput at the most congested receiver is estimated according to pre-defined equations that were introduced in this literature work. The simulation results showed that the proposed solution achieved good link utilization.

Other approaches aim to enhance the multicasting delay in DCNs. The literature works (Marcondes et al., 2012), and

Table 5
Traffic Engineering Techniques summary.

Reference	Technique	Congestion minimization	Bandwidth consumption optimization	Scalability enhancement
Craig et al. (2015)	Real-time link cost adjustment	✓		
Huang et al. (2015)	Branch forwarding technique		✓	✓
Li and Freedman (2013)	Partitioning multicast address space			✓

(Bondan et al., 2013) discuss the clean-slate multicast approach which depends on pre-calculation of all available routes from multicast group source to all destinations. This in turn speeds up the multicasting process.

Dynamic multicast routing algorithm was proposed by Ref. Ge et al. (2013), which enables adjustable routing while the source and destination of data is unchanged. Recover-Aware Edge Reduction Algorithm was introduced to construct reliable multicast routing tree in SDN environments.

However, all these solutions have only focused on routing related issues and not the multicast protocol design.

The authors in Refs. Zhu et al.(2016) and Zhu et al. (2017) introduced MCTCP, which is an SDN-based multicasting solution, designed for small-multicast groups. The solution is sender-initiated, congestion-aware and applies reliable routing concept. All these features make it ideal to apply in data center networks. MCTCP utilizes the centralized control concept in SDNs and assign the multicast flows according to real-time link status, where the flows are assigned to the active and less-utilized links.

MCTCP mainly consists of two modules, HSP (Host-Side Protocol) and MGM (Multicast Group Manager). HSP is a sender-initiated transport layer protocol where the multicasting transmission is started at the sender side and the receivers are not aware of the address and do not need to register in advance. HSP notifies MGM each time a new transmission session is initiated or closed. MGM dynamically schedules the multicast flows according to network status, where it ignores any congested or failed link. The experimental results showed a good enhancement of multicasting performance when applying MCTCP (Zhu et al., 2016; Zhu et al., 2017).

Recently, the authors in Mahajan et al. (2017) proposed a multicast scheme called ATHENA for group communication in SDN-based data centers. ATHENA multicast scheme, which is based on earlier work by the authors Avalanche (Cui and Qian, 2014), guarantees reliability, statelessness and TCP-friendliness. Table 6 summarizes the solutions discussed in this section so far. Scalability is an important requirement for multicasting in general and for data centers more specifically. Scalable multicasting solutions are discussed in the next subsection.

7.2. Scalable multicasting in data centers

Multicast trees can be categorized into two types: per-source tree and shared tree. In the first category, a multicast tree is assigned to each multicast source while in the shared tree; the same multicast tree will be shared among different groups. In SDN there are three alternatives for implementing the shared tree: per-group shared tree, multi-group shared tree, and single shared tree. In the per-group shared tree, the multicast sources within the same group share the same tree. In the multi-group shared tree, the multicast tree is shared among several groups. Lastly, in the single shared tree all the possible multicast sources will share the same multicast tree (Lin et al., 2017). For a network that has N multicast groups with M sources in each group, per-source tree and per-group shared tree will result in NxM and N total trees, respectively, which is a heavy burden on the controller and the multicast switches. On the other hand, a single shared tree only needs one tree, however, the end-to-end delay can be significantly high. Therefore, the multi-group shared tree is considered to be the most appropriate approach for SDN (Lin et al., 2017).

The authors in Lin et al. (2017) presented a Locality-Aware Multicast Approach (LAMA) that aims to enhance the scalability in multicasting for video streaming services. LAMA depends on clustering several multicast groups, and then generating a shortest-path multi-group shared tree for each multicast cluster. Since this problem is an NP complete problem, it was divided to three sub-problems that are independent and solved separately: multicast group clustering, Rendezvous Point (RP) selection, and tree construction. A distance-based clustering algorithm was proposed to solve the multicast group-clustering problem, where all the nearby-multicast sources are clustered in the same multicast cluster. For the RP selection, a locality-aware selection algorithm was introduced to determine the appropriate RP within a cluster, such that it has the minimum distance to all the multicast sources. Finally, a shortest-path tree is generated from the selected RP to each multicast cluster host.

In addition to constructing multi-group shared trees for multicasting in SDN, LAMA aims to improve the scalability of both SDN controller and forwarding switches. The improvements are in terms of both the computation time and the number of entries in the flow tables while maintaining the required quality of service. Most data center networks currently use a hierarchical multi-rooted tree topologies such as fat-tree and Clos topologies. In a standard fat-tree architecture, the network consists of three layers: core, aggregate, and edge switches that are connected to the end-hosts. A pod is a set of connected hosts along with the set of aggregate and edge switches that connects those hosts. The fat-tree network can be divided to a number of k pods that are connected by (k/2) edge switches and (k/2) aggregate switches (Cui and Qian, 2014).

The literature work (Fan et al., 2016) has introduced a distributed multicast system that was designed for data centers with fat-tree topology. The proposed algorithm implements a distributed management system that eliminates the need for a centralized controller. To do so,

Table 6
Multicasting in data centers.

Reference	Technique	Congestion minimization	Delay minimization	Reliability enhancement
Shukla et al. (2016)	Utilizing multiple equal-cost paths	✓		
Akamatsu et al. (2016)	Adjusting transmission rate at sender end	✓		
Marcondes et al. (2012); Bondan et al. (2013)	Clean-slate multicast		✓	
Ge et al. (2013)	Recover-aware Edge Reduction Algorithm			✓
Zhu et al. (2016)	Real-time adjustment for routing	✓		✓
Mahajan et al. (2017)	ATHENA	✓	✓	✓

first the multicast group concept is re-defined. That is, a group G is defined as a set of hosts who are permitted to receive the same content of data regardless of the identity of the sender. This significantly minimizes the number of multicast groups which in turn reduces the address space and helps to improve scalability. After defining the groups, any host can join or leave a specific group just by sending a request to the targeted group address. Two addresses are defined for each group, a static one and a dynamic one. The static address is referred to as marking address and its used by the application layer to define a specific group and the hosts belonging to that group. The dynamic address or routing address is the actual address that defines the source and destination of data at the network level, however, it is hidden from the multicast application. An address dispatcher is used to map each marking address to the corresponding routing address.

After obtaining the routing address, a host can join or quit the group at any time and dynamic adjustment of the group will take place without the need to re-create the group. When compared with centralized multicast methods, the controller must re-define the group each time a member joins or leaves which is a heavy load especially in data centers with the large size of groups (Fan et al., 2016). Another scalable multicast approach was introduced in Cui and Qian (2014), where experimental results showed that the amount of groups it can support in data centers is 300% compared to traditional IP multicast. In addition to the scalability improvement, the proposed solution aims to enhance load balancing and therefore helps to reduce congestion within the data center network. The solution focuses on the wide heterogeneity of multicast traffic in data center networks. The multicast groups are divided into two categories based on their amount of traffic: elephant groups and mice groups. Mice groups form the majority of groups and they have low traffic volume, while elephant groups are small fraction of the total multicast groups and they have higher volume of traffic that exceeds a pre-defined threshold value. This value is defined by the group classification module which is a part of the SDN controller. That is, the controller classifies each group based on the amount of traffic that is monitored and updated periodically. The flow rules are then set and installed on the switches of each group according to the group classification.

The solution aims to find a balance on the tradeoff between bandwidth capacity and state capacity. That is, since Mice groups are more frequent but they have smaller sizes, state-free multicast are applied where state capacity has the priority over bandwidth capacity. In Elephant groups, the bandwidth capacity have the higher priority due to their large amount of traffic. Therefore, multiple shared trees are applied among groups. That is, if all group members are located in the same pod of the fat-tree network, a single shared tree would be used. Otherwise, multiple shared trees are constructed by selecting a number of cores randomly and then running a Steiner Tree algorithm to construct a tree for each core. Multi-cast flow rules are then installed on the switches of each tree. Incoming packets can then be matched to one of the trees and delivered to receivers through that tree.

Code-Oriented explicit multicast (COXcast) is another scalable multicasing solution that was introduced in (Jia, 2014). COXcast was developed based on explicit multicast Xcast concept which is a new multicasting scheme that aims to overcome the scalability challenges in multicasting for a large number of variant multicast groups. Instead of using a multicast group address Xcast encodes the list of group destinations explicitly in the packet header. Therefore, inter-mediate routers will perform stateless forwarding of packets. Similar to the situation in Xcast, the entire packet forwarding information are held in the multicast packet header using a common identifier and a node-specific key. Therefore, the packets are self-routed to all group destinations. Eliminating the multicast forwarding rules in inter-mediate network devices result in significant enhancements in terms of reducing the overhead and latency. Moreover, it offers better scalability support of large scale multicast applications that has few-to-few groups structure.

ESM that was introduced in Li et al. (2012) constructs efficient multicast routing trees using a source-to-receiver expansion mechanism that eliminates unneeded inter-mediate network switches that are used in traditional receiver-driven routing protocols. For the scalability enhancement ESM makes a tradeoff between the amount of supported multicast groups and the bandwidth overhead by using both in-packet Bloom Filters and in-switch entries. Experimental results recorded an improvement in multicast throughput, computation complexity, and traffic leakage.

Scalar-pair Vectors Routing and Forwarding (SVRF) was introduced in Jia and Wang (2013) to mathematically solve the multiple membership problem by introducing a multiple membership query algorithm that calculates the output ports of each multicast group by dividing a common scalar-pair over a group specific key. This division method is done within pseudo-polynomial time. The simulation results prove that the proposed algorithm recorded a significant enhancement in reducing memory space, processing time, and hardware cost. Table 7 shows a summary of all scalability-related schemes that were discussed in this survey.

8. Future directions

As mentioned earlier, dominant Internet companies are now adopting SDN techniques for their Data Center Networks structure. Therefore, multicasting in SDN-based data centers is an area with rich content to be explored. Earlier literature works have introduced models for applying multicasting in such environments as discussed in Section 5 of this paper. However, several challenges still remain that opens interesting research opportunities for future work.

8.1. Bit Indexed Explicit Replication (BIER) architecture

IETF is defining a new scalable architecture for optimal forwarding of multicast data packets called Bit Index Explicit Replication (BIER) (Tony et al., 2017). Unlike the traditional IP/MPLS multicast approaches, BIER architecture neither requires a protocol for explicitly building multicast distribution trees (e.g., PIM, mLDP, RSVP-TE) nor needs to maintain per-flow state in the intermediate nodes which enables a flexible and scalable multicast solution. Although BIER allows optimal multicast forwarding, however, it does not support traffic engineering capabilities and lacks fast network resiliency mechanism to protect against link and node failures. The Traffic Engineering for Bit Index Explicit Replication (TE-BIER) architecture addresses these deficiencies (Eckert et al., 2016b; Eckert et al., 2016a; Braun et al., 2017). TE-BIER is a SDN controller based approach to support traffic engineering and network resiliency through the Fast ReRoute (FRR) mechanism.

Table 7
Scalability techniques summary.

Reference	Technology Used	Scalability applications
Lin et al. (2017)	Distance-based clustering	Multicasting for video streaming services
Fan et al. (2016)	Distributed management system	Data centers with fat-tree topology
Cui and Qian (2014)	Divide the multicast groups into two categories based on their amount of traffic	Widely heterogenic traffic in data center networks
Jia (2014)	Code-Oriented explicit multicast	Multicasting for a large number of variant multicast groups
Li et al. (2012)	Source-to-receiver expansion mechanism	Data centers networks
Jia and Wang (2013)	Scalar-pair Vectors routing and forwarding	Data centers networks

An interesting area of future direction could be to study performance of the TE-BIER for (a) traffic engineering with respect to path lengths, traffic loads, and required network capacity (b) network resiliency with respect to protection coverage (c) scalability with respect to state required in large networks (d) network convergence with respect to amount of traffic loss following topology changes and link/node failures. It could also be interesting to explore architectural extensions of TE-BIER to allow computation of end-to- end multicast TE-BIER trees across multi-domain (i.e., multiple IGP areas, multiple autonomous systems) in conjunction with BGP-LS and hierarchical PCE (King and Farrel, 2012).

8.2. Network reliability

As mentioned earlier, network reliability is a fundamental requirement in realtime applications where time-sensitive content is delivered to multiple receivers simultaneously using network multicast functionalities. The reliable delivery of this content depends on the existence of a fault resilient network that is able to restore network services in case of a failure. Failure protection can be implemented using either local protection schemes such as fast reroute or end-to- end schemes such as redundant trees. The link-coloring method presented in Bejerano and Koppol (2013) is an example of redundant trees schemes. The idea is to color all the links for a given source node in either blue or red, such that for any given destination the red and blue paths are disjoint. Therefore, redundant trees are formed regardless of the used tree selection method. Redundant trees connect the source node to all its destinations such that in case of a failure every destination still have a connected path to the source node in at least one of the trees. The authors in Ref. Bejerano and Koppol (2014) conducted a comparison to study the trade-offs between the fast reroute and redundant tree approaches. The comparison considered different aspects such as recovery time, protection availability, resource reservation, and management complexity. Fast reroute schemes have faster recovery time, however, it results in high resource reservation and complex network management. On the other hand, redundant trees schemes are more efficient in resource protection and offer simpler to manage. However, fast recovery time cannot be achieved without requiring a hot standby. According to these conflicting objectives, one of the future directions could be to explore possibility of designing a hybrid scheme that is able to utilize the advantages of the two approaches while eliminating their drawbacks. Another interesting area of future research could be to study the computed multicast scheme applied to MPLS based Segment Routing and evaluate its performance in terms of reliability and the amount of multicast state required in the network (Allan, 2017).

8.3. Security in Software Defined Multicasting

The security of Software Defined Multicasting is another direction that requires more investigation since the centralized control is vulnerable to attacks which may disrupt the entire network (Karam et al., 2012). In particular, dynamic nature of multicast protocols (e.g., joining and leaving of multicast groups) creates opportunity for unauthorized users to join multicast sessions and receive packets easily. Moreover, the safety of data source is not guaranteed in multicast applications. That is, any client could act as the multicast packets source without any control from the application (Zou et al., 2013). An earlier work in Li et al. (2016) classified security challenges to SDN network components including switch, controller, and communication channel. Due to the above-mentioned challenges, a multicast model is required to provide secure multicast delivery of data in SDN environment such that the relationship between applied security measures and the performance of multicast delivery is well analyzed. The literature work Zou et al. (2013) introduced a secure multicast scheme that is designed for SDNs. User authentication and multicast group management mechanisms are implemented at the SDN controller. User authentication technique prevents unauthorized groups joining. Moreover, forwarding devices send packets according to flow rules that are generated by the controller. This ensures that no client can turn into data source without the permission of the controller. However, the proposed security scheme was designed for traditional IP multicast. A future direction of research could be to enhance this security framework with respect to holistic real-time system monitoring, detection, containment, and correlation of all potential threats, and automated activation of controls for multicast multi-tenant access in virtualized cloud environments.

8.4. Power consumption

Power consumption is another network-related concern that requires more attention from economic and environmental perspectives. According to Dharmaweera et al. (2015), it's estimated that the power consumption by the Internet is 1% of the total electricity consumption in broad-band enabled countries. The three Internet domains consume this power in access networks, metro networks and backbone or core networks. The power consumption in the access networks increases as the number of subscribers increase. Whereas in backbone network, the consumption is proportional to network traffic volume which is expected to exceed the zettabyte threshold in the near future (Dharmaweera et al., 2015). The authors in Ref. Dharmaweera et al. (2015) introduced a survey for the power reduction techniques that could be applied in backbone networks. When it comes to multicasting, few literature works have studied the relationship between power consumption and multicast routing. The literature work in Wang et al. (2015) has introduced a power-efficient routing scheme for many-to-many multicasting that was designed for green multi-granularity transport networks. An interesting future areas of research could be to expand SDN functionality to collect power consumption information across multi-domain core network elements (Alferness et al., 2013). This information could enable computation of end-to- end power efficiency metrics therefore allowing evaluation of new energy efficient multicast approaches such as TE-BIER.

9. Conclusion

In this paper, a systematic survey of multicasting in software-defined net-working was presented. The goal of this survey was to study different aspects of multicasting in SDN and try to cover all the related challenges to each aspect and the existing solutions to overcome those challenges. First, the importance of software defined networking was introduced while highlighting the features of SDN that can be utilized to improve the quality of multicasting. Then multicasting techniques were surveyed starting from the history and background of multicasting. After that the existing solutions for multicasting main fields of interest were presented such as: tree planning, tree management, multicast routing, multicast TE and multicast in data centers. Comparisons were conducted between the different existing solutions/techniques that were introduced in the paper. Moreover, future research directions were suggested to improve the existing solutions or to highlight new problems that were not studied earlier.

Acknowledgement

The authors would like to thank the anonymous reviewers for their valuable and thoughtful comments which definitely improved overall quality of the manuscript.

References

Akamatsu, J., Matsushima, K., Yamamoto, M., 2016. Equation-based multicast congestion control in data center networks. In: Network Operations and Management Symposium (APNOMS), 2016 18th Asia-Pacific. IEEE, pp. 1–6.

Al Hasrouty, C., Autefage, V., Olariu, C., Magoni, D., Murphy, J., 2016. Sdn-driven multicast streams with adaptive bitrates for voip conferences. In: IEEE International Conference on Communications, pp. 1–7.

Alferness, R., et al., 2013. Ict Core Networks: towards a Scalable, Energy-efficient Future. http://www.infonetics.com/cgp/downloads/UCSB-REPORT-Scalable-Energy-Efficient-ICT-Core-Networks.pdf.

Allan, D., 2017. A Framework for Computed Multicast Applied to Mpls Based Segment Routing. https://www.tools.ietf.org/html/draft-allan-spring-mpls-multicast-framework-03.

Baker, T., Al-Dawsari, B., Tawfik, H., Reid, D., Ngoko, Y., 2015. Greedi: an energy efficient routing algorithm for big data on cloud. Ad Hoc Netw. 35, 83–96.

Baker, T., Asim, M., Tawfik, H., Aldawsari, B., Buyya, R., 2017. An energy-aware service composition algorithm for multiple cloud-based iot applications. J. Netw. Comput. Appl. 89, 96–108.

Baker, T., Ngoko, Y., Tolosana-Calasanz, R., Rana, O.F., Randles, M., 2013. Energy efficient cloud computing environment via autonomic meta-director framework. In: Developments in eSystems Engineering (DeSE), 2013 Sixth International Conference on. IEEE, pp. 198–203.

Bejerano, Y., Koppol, P.V., 2013. Link-coloring based scheme for multicast and unicast protection. In: High Performance Switching and Routing (HPSR), 2013 IEEE 14th International Conference on. IEEE, pp. 21–28.

Bejerano, Y., Koppol, P.V., 2014. Resource reservation comparison of fault resilient routing schemes. In: Local Computer Networks (LCN), 2014 IEEE 39th Conference on. IEEE, pp. 81–89.

Bezenšek, M., Robič, B., 2014. A survey of parallel and distributed algorithms for the steiner tree problem. Int. J. Parallel Program. 42 (2), 287–319.

Bondan, L., Müller, L.F., Kist, M., 2013. Multiflow: multicast clean-slate with anticipated route calculation on openflow programmable networks. J. Appl. Commun. Res. 2 (2), 68–74.

Braun, W., Hartmann, J., Menth, M., 2017. Scalable and reliable software-defined multicast with bier and p4. In: Integrated Network and Service Management (IM), 2017 IFIP/IEEE Symposium on. IEEE, pp. 905–906.

Craig, A., Nandy, B., Lambadaris, I., Ashwood-Smith, P., 2015. Load balancing for multicast traffic in sdn using real-time link cost modification. In: 2015 IEEE International Conference on Communications (ICC). IEEE, pp. 5789–5795.

Cui, W., Qian, C., 2014. Dual-structure Data Center Multicast Using Software Defined Networking. arXiv preprint arXiv:1403.8065.

De Turck, F., Chemouil, P., Boutaba, R., Yu, M., Rothenberg, C.E., Shiomoto, K., 2016. Guest editors introduction: special issue on management of softwarized networks. IEEE Trans. Netw. Service Manag. 13 (3), 362–365.

Dharmaweera, M.N., Parthiban, R., Şekercioğlu, Y.A., 2015. Toward a power-efficient backbone network: the state of research. IEEE Commun. Surveys Tutor. 17 (1), 198–227.

Eckert, T., Cauchie, G., Braun, W., Menth, M., 2016a. Fast Reroute (frr) Extensions for Bier-te. Internet Engineering Task Force. Internet-Draft draft-eckert-bier-te-frr-00.

Eckert, T., Cauchie, G., Braun, W., Menth, M., 2016b. Traffic Engineering for Bit Index Explicit Replication bier-te. Tech. rep. Internet Engineering Task Force, Internet-Draft draft-eckert-bier-te-arch-04. https://tools.ietf.org/html/draft-eckert-bier-te-arch-04.

Fabregat, R., Donoso, Y., Baran, B., Solano, F., Marzo, J.L., 2005. Multi-objective optimization scheme for multicast flows: a survey, a model and a moea solution. In: Proceedings of the 3rd International IFIP/ACM Latin American Conference on Networking. ACM, pp. 73–86.

Fan, F., Hu, B., Yeung, K.L., 2016. Distributed and dynamic multicast scheduling in fat-tree data center networks. In: Communications (ICC), 2016 IEEE International Conference on. IEEE, pp. 1–6.

Ge, J., Shen, H., Yuepeng, E., Wu, Y., You, J., 2013. An openflow-based dynamic path adjustment algorithm for multicast spanning trees. In: Trust, Security and Privacy in Computing and Communications (TrustCom), 2013 12th IEEE International Conference on. IEEE, pp. 1478–1483.

Gijare, A., 2016. Software-defined Networking: Technologies and Global Markets. Tech. rep., BCC Research.

Gu, W., Zhang, X., Gong, B., Wang, L., 2015. A survey of multicast in software-defined networking. In: n Proceedings of the fth International Conference on Information Engineering for Mechanics and Materials(ICIMM), pp. 1096–1100.

Hadoop, A., 2009. Hadoop. URL 2009-03-06]. http://hadoop. apache. org.

Hu, Z., Guo, D., Xie, J., Ren, B., 2016. Multicast routing with uncertain sources in software-defined network. In: Quality of Service (IWQoS), 2016 IEEE/ACM 24th International Symposium on. IEEE, pp. 1–6.

Huang, L., Zhi, X., Gao, Q., Kausar, S., Zheng, S., 2016a. Design and implementation of multicast routing system over sdn and sflow. In: Communication Software and Networks (ICCSN), 2016 8th IEEE International Conference on. IEEE, pp. 524–529.

Huang, L.-H., Hsu, H.-C., Shen, S.-H., Yang, D.-N., Chen, W.-T., 2015. Multicast Traffic Engineering for Software-defined Networks. arXiv preprint arXiv:1507.08728.

Huang, L.-H., Hung, H.-J., Lin, C.-C., Yang, D.-N., 2014. Scalable Steiner Tree for Multicast Communications in Software-defined Networking. arXiv preprint arXiv:1404.3454.

Huang, M., Liang, W., Xu, Z., Xu, W., Guo, S., Xu, Y., 2016b. Dynamic routing for network throughput maximization in software-defined networks. In: Computer Communications, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on. IEEE, pp. 1–9.

Humernbrum, T., Hagedorn, B., Gorlatch, S., 2016. Towards efficient multicast communication in software-defined networks. In: Distributed Computing Systems Workshops (ICDCSW), 2016 IEEE 36th International Conference on. IEEE, pp. 106–113.

Iyer, A., Kumar, P., Mann, V., 2014. Avalanche: data center multicast using software defined networking. In: Communication Systems and Networks (COMSNETS), 2014 Sixth International Conference on. IEEE, pp. 1–8.

Jia, W.-K., 2014. A scalable multicast source routing architecture for data center networks. IEEE J. Sel. Area. Commun. 32 (1), 116–123.

Jia, W.-K., Wang, L.-C., 2013. A unified unicast and multicast routing and forwarding algorithm for software-defined datacenter networks. IEEE J. Sel. Area. Commun. 31 (12), 2646–2657.

Jian, Y., Enzhong, Y., Yongyi, R., Shuangwu, C., 2015. Sdm^2 cast an openflow-based, software-defined scalable multimedia multicast streaming framework. IEEE Internet Computing 19 (4), 36–44.

Jiang, J.-R., Chen, S.-Y., 2016. Constructing multiple steiner trees for software-defined networking multicast. In: Proceedings of the 11th International Conference on Future Internet Technologies. ACM, pp. 1–6.

Karam, Y., Baker, T., Taleb-Bendiab, A., 2012. Security support for intention driven elastic cloud computing. In: Computer Modeling and Simulation (EMS), 2012 Sixth UKSim/AMSS European Symposium on. IEEE, pp. 67–73.

King, D., Farrel, A., 2012. The Application of the Path Computation Element Architecture to the Determination of a Sequence of Domains in Mpls and Gmpls. IETF RFC 6805.

Kotani, D., Suzuki, K., Shimonishi, H., 2016. A multicast tree management method supporting fast failure recovery and dynamic group membership changes in openflow networks. J. Inf. Process. 24 (2), 395–406.

Kreutz, D., Ramos, F.M., Verissimo, P.E., Rothenberg, C.E., Azodolmolky, S., Uhlig, S., 2015. Software-defined networking: a comprehensive survey. Proc. IEEE 103 (1), 14–76.

Li, D., Li, Y., Wu, J., Su, S., Yu, J., 2012. Esm: efficient and scalable data center multicast routing. IEEE/ACM Trans. Netw. (TON) 20 (3), 944–955.

Li, D., Xu, M., Liu, Y., Xie, X., Cui, Y., Wang, J., Chen, G., 2014. Reliable multicast in data center networks. IEEE Trans. Comput. 63 (8), 2011–2024.

Li, W., Meng, W., Kwok, L.F., 2016. A survey on openflow-based software defined networks: security challenges and countermeasures. J. Netw. Comput. Appl. 68, 126–139.

Li, X., Freedman, M.J., 2013. Scaling ip multicast on datacenter topologies. In: Proceedings of the Ninth ACM Conference on Emerging Networking Experiments and Technologies. ACM, pp. 61–72.

Lin, Y.-D., Lai, Y.-C., Teng, H.-Y., Liao, C.-C., Kao, Y.-C., 2017. Scalable multicasting with multiple shared trees in software defined networking. J. Netw. Comput. Appl. 78, 125–133.

Liu, Y., Niu, D., Li, B., 2016. Delay-optimized video traffic routing in software-defined interdatacenter networks. IEEE Trans. Multimed. 18 (5), 865–878.

Mahajan, K., Sharma, D., Mann, V., 01 2017. Athena: reliable multicast for group communication in sdn-based data centers. In: 2017 9th International Conference on Communication Systems and Networks (COMSNETS), pp. 174–181.

Manimaran, G., Striegel, A., 2002. A survey of qos multicasting issues. IEEE Commun. Mag. 40 (6), 82–87.

Marcondes, C.A., Santos, T.P., Godoy, A.P., Viel, C.C., Teixeira, C.A., 2012. Castflow: clean-slate multicast approach using in-advance path processing in programmable networks. In: Computers and Communications (ISCC), 2012 IEEE Symposium on. IEEE, pp. 94–101.

Masoudi, R., Ghaffari, A., 2016. Software defined networks: a survey. J. Netw. Comput. Appl. 67, 1–25.

Matsuura, H., 2016. Multi-agent steiner tree algorithm based on branch-based multicast. IEICE Trans. Info Syst. 99 (11), 2745–2758.

Mendiola, A., Astorga, J., Jacob, E., Higuero, M., 05, 2017. A survey on the contributions of software-defined networking to traffic engineering. IEEE Commun. Surveys Tutor. 19, 918–953.

Open, s., 2009. Openstack. http://openstack.org.

Phaal, P., 2004. sflow Specification Version 5. http://www.sFlow.org/.

Popovic, M., Khalili, R., Le Boudec, J.-Y., 2017. Performance comparison of node-redundant multicast distribution trees in sdn networks. In: Networked Systems (NetSys), 2017 International Conference on. IEEE, pp. 1–8.

Prithviraj, P., Akram, H., Aniruddha, G., 2016. Adaptive and Flexible sdn-based Multicast for Efficient Data Center Networking.

Raja, V.R., Lung, C.-H., Pandey, A., Wei, G.-m., Srinivasan, A., 2016. A subtree-based approach to failure detection and protection for multicast in sdn. Front. Inform. Technol. Electron. Eng. 17 (7), 682–700.

Reed, M.J., Al-Naday, M., Thomos, N., Trossen, D., Petropoulos, G., Spirou, S., 2016. Stateless multicast switching in software defined networks. In: Communications (ICC), 2016 IEEE International Conference on. IEEE, pp. 1–7.

Ren, B., Guo, D., Xie, J., Li, W., Yuan, B., Liu, Y., 2017. The packing problem of uncertain multicasts. Concurrency Comput. Pract. Exp. 29 (16). https://doi.org/10.1002/cpe.3985, e3985–n/a, e3985 cpe. 3985.

Ruckert, J., Blendin, J., Hark, R., Hausheer, D., 2015. Dynsdm: dynamic and flexible software-defined multicast for isp environments. In: Network and Service Management (CNSM), 2015 11th International Conference on. IEEE, pp. 117–125.

Rückert, J., Blendin, J., Hark, R., Hausheer, D., 2016. Flexible, efficient, and scalable software-defined over-the-top multicast for isp environments with dynsdm. IEEE Trans. Netw. Service Manag. 13 (4), 754–767.

Rückert, J., Blendin, J., Hark, R., Wächter, T., Hausheer, D., 2015. An Extended Study of Dynsdm: Software-defined Multicast Using Multi-trees. Tech. Rep. Peer-to-Peer Systems Engineering Lab, TU Darmstadt, Germany.

Shen, S.-H., Huang, L.-H., Yang, D.-N., Chen, W.-T., 2015. Reliable multicast routing for software-defined networks. In: 2015 IEEE Conference on Computer Communications (INFOCOM). IEEE, pp. 181–189.

Sheu, J.-P., Chang, C.-W., Chang, Y.-C., 2015. Efficient multicast algorithms for scalable video coding in software-defined networking. In: Personal, Indoor, and Mobile Radio Communications (PIMRC), 2015 IEEE 26th Annual International Symposium on. IEEE, pp. 2089–2093.

Shukla, S., Ranjan, P., Singh, K., 2016. Mcdc: multicast routing leveraging sdn for data center networks. In: Cloud System and Big Data Engineering (Confluence), 2016 6th International Conference. IEEE, pp. 585–590.

Sun, M., Zhang, X., Wang, L., Shi, H., Zhang, W., 2016. A multiple multicast tree optimization solution based on software defined network. In: 2016 7th International Conference on Information and Communication Systems (ICICS). IEEE, pp. 168–173.

Tang, S., Hua, B., Wang, D., 2014. Realizing video streaming multicast over sdn networks. In: Communications and Networking in China (CHINACOM), 2014 9th International Conference on. IEEE, pp. 90–95.

Tony, P., Sam, A., Wijnands, I., Eric, R., Andrew, D. a., sep 2017. Multicast Using Bit Index Explicit Replication Draft-ietf-bier-architecture-06. Internet-Draft draft-ietf-bier-architecture-06, Internet Engineering Task Force, work in Progress. https://tools.ietf.org/html/draft-ietf-bier-architecture-06.

Wang, X., Qu, D., Huang, M., Li, K., Das, S.K., Zhang, J., Yu, R., 2015. Multiple many-to-many multicast routing scheme in green multi-granularity transport networks. Comput. Network. 93, 225–242.

Wei, G., Lung, C.-H., Srinivasan, A., 2010. Protecting a mpls multicast session tree with bounded switchover time. In: Performance Evaluation of Computer and Telecommunication Systems (SPECTS), 2010 International Symposium on. IEEE, pp. 236–243.

Xu, S., Wu, C., Li, Z., 2015. Software defined mobile multicast. In: Mobile Ad Hoc and Sensor Systems (MASS), 2015 IEEE 12th International Conference on. IEEE, pp. 208–216.

Xue, N., Chen, X., Gong, L., Li, S., Hu, D., Zhu, Z., 2015. Demonstration of openflow-controlled network orchestration for adaptive svc video manycast. IEEE Trans. Multimed. 17 (9), 1617–1629.

Yamamoto, M., 2003. Multicast communications–present and future. IEICE Trans. Commun. 86 (6), 1754–1767.

Yu, P., Wu, R., Zhou, H., Yu, H., Chen, Y., Zhong, H., 2016. Multicast routing tree for sequenced packet transmission in software-defined networks. In: Proceedings of the 8th Asia-Pacific Symposium on Internetware. ACM, pp. 27–35.

Zeng, M., Fang, W., Rodrigues, J.J., Zhu, Z., 2016. Orchestrating multicast-oriented nfv trees in inter-dc elastic optical networks. In: IEEE International Conference on Communications 2016. IEEE.

Zhang, S.Q., Zhang, Q., Bannazadeh, H., Leon-Garcia, A., 2015. Routing algorithms for network function virtualization enabled multicast topology on sdn. IEEE Trans. Netw. Service Manag. 12 (4), 580–594.

Zhang, X., Yang, M., Wang, L., Sun, M., 2016. An openflow-enabled elastic loss recovery solution for reliable multicast. IEEE Systems J. PP 99, 1–12.

Zhou, S., Wang, H., Yi, S., Zhu, F., 2015. Cost-efficient and scalable multicast tree in software defined networking. In: International Conference on Algorithms and Architectures for Parallel Processing. Springer, pp. 592–605.

Zhu, T., Feng, D., Wang, F., Hua, Y., Shi, Q., Xie, Y., Wan, Y., 2017. A congestion-aware and robust multicast protocol in sdn-based data center networks. J. Netw. Comput. Appl. 95, 105–117.

Zhu, T., Wang, F., Hua, Y., Feng, D., Wan, Y., Shi, Q., Xie, Y., 2016. Mctcp: congestion-aware and robust multicast tcp in software-defined networks. In: Quality of Service (IWQoS), 2016 IEEE/ACM 24th International Symposium on. IEEE, pp. 1–10.

Zou, J., Shou, G., Guo, Z., Hu, Y., 2013. Design and implementation of secure multicast based on sdn. In: Broadband Network & Multimedia Technology (IC-BNMT), 2013 5th IEEE International Conference on. IEEE, pp. 124–128.

Zainab AlSaeed received her B.Sc. degree in Computer Engineering from Kuwait University in 2013. Currently, she is pursuing her master degree in Computer Engineering at Kuwait University. Since 2014, she has been working at Kuwait National Petroleum Company. Her research interests include computer networks and software defined networks.

Imtiaz Ahmad received his B.Sc. in Electrical Engineering from University of Engineering and Technology, Lahore, Pakistan, a M.Sc. in Electrical Engineering from King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia, and a Ph.D. in Computer Engineering from Syracuse University, Syracuse, New York, in 1984, 1988 and 1992, respectively. Since September 1992, he has been with the Department of Computer Engineering at Kuwait University, Kuwait, where he is currently a professor. His research interests include design automation of digital systems, high-level synthesis, distributed computing, and software defined networks.

Iftekhar Hussain received his B.Sc. in Electrical Engineering from University of Engineering and Technology, Lahore, Pakistan, M.S. in Electrical and Computer Engineering from The University of Tennessee, Knoxville, Tennessee, USA, and a Ph.D. in Electrical and Computer Engineering from the University of California, Davis, California, USA. Iftekhar is currently a senior principal software architect at Infinera Corporation, where he is involved in design of IP/MPLS control and data plane system architecture of high-capacity packet switching systems for data center and core network applications. His research interests include routing, switching, computer vision, and machine learning.