This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2017.2776213, IEEE Communications Surveys & Tutorials

A Survey on Multicasting in Software-Defined Networking

Salekul Islam, *Senior Member, IEEE*, Nasif Muslim, *Student Member, IEEE* , and J. William Atwood, *Life Senior Member, IEEE*

***Abstract*—Existing surveys of research into Software-Defined Networking (SDN) make only minimal mention of multicasting. We present a survey of existing multicast routing protocols in IP multicast, and a survey of existing and proposed multicast routing algorithms in SDN multicast. Each of these protocols/algorithms is designed to optimize a slightly different metric, under specific constraints, so we contrast the approaches based on the optimization goal.**

We then present a survey of existing and proposed approaches to security for IP multicast and SDN multicast. This reveals a striking lack of breadth in the research in the topic of SDN multicast security.

I. Introduction

Traditional network devices that forward traffic (e.g., routers and layer-3 switches) perform forwarding and routing functions at the same time. Routing involves determining the best path for a particular packet based on the destination address and the existing connectivity information. On the other hand, forwarding is transferring an incoming packet to an outgoing interface so that it moves on the best path, as determined in the routing phase. For a given amount of hardware, a router's performance would be improved if it could dedicate its full resources to forwarding. By exploiting this idea, the concept of Software-Defined Networking (SDN) has evolved over the years [1], [2]. Figure 1 shows the main difference between a traditional network and the SDN architecture. In the SDN architecture, a controller has a global view of the network and performs route calculation and other complex tasks, and then sends the forwarding tables to the SDN-enabled switches. The switches are responsible only for forwarding.

At present, various network devices such as routers, switches, virtual switches, and wireless access points are involved in forwarding. In an SDN environment, any network device that is responsible for forwarding is referred to simply as a "switch". The emergence of programmable networks, and more specifically SDN with OpenFlow, has made possible the implementation of many new networking features in a network, such as centralized control of the network, software-based traffic analysis, dynamic updating of forwarding rules, and flow abstractions. It is these new features, rather than any performance improvement, that have driven the adoption of SDN.

S. Islam and N. Muslim are with the Department of Computer Science and Engineering, United International University, Dhaka, Bangladesh (e-mail: salekul@cse.uiu.ac.bd; nasif@cse.uiu.ac.bd).

J.W. Atwood is with the Department of Computer Science and Software Engineering, Concordia University, Montréal, Québec (e-mail: william.atwood@concordia.ca).

Three recent surveys of SDN [1], [2], [3] provide extensive discussion of the properties of networks based on SDN features, and on the features of Openflow. However, multicasting is only briefly mentioned in these surveys.

Note that not all switches are SDN-enabled. An SDN-enabled switch is programmable remotely by the controllers through appropriate configuration protocols. In this article, we are following the SDN terminology and thus switches refer to network devices that are involved in some sort of forwarding.

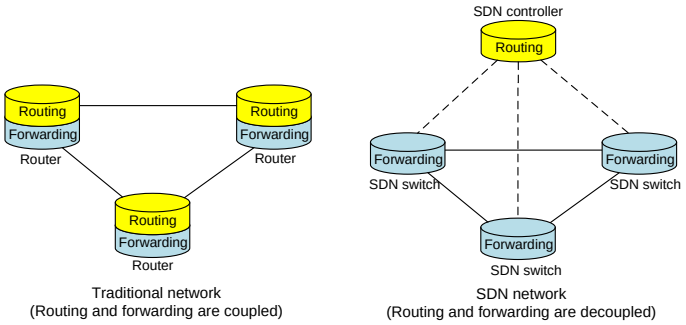


Fig. 1: Basic difference of traditional network and SDN architecture

IP multicast, with all its advantages, is not widely deployed yet, although it was standardized many years ago. The extensions required for a host implementation of the Internet Protocol (IP) to support multicasting were first specified in RFC 1112 [4]. In contrast to the simple point-to-point unicast service, IP multicast provides an m-to-n communication service, from one or more senders to zero or more receivers. Other than this difference, the service provided by IP multicast is the same as the service provide by IP unicast—best-effort delivery of packets, with no guarantees of any kind.

To implement multicasting successfully, we can identify four major steps or processes responsible for the whole complex task. Figure 2 describes the schematic diagram of these four steps.

1) At first, a multicast host group having a suitable multicast address is created by the group owner and this group address is announced to the potential users.
2) A user or host communicates with its local access router to join/leave a multicast group at the network level.
3) The router is responsible for connecting the access router to the multicast data distribution tree (MDDT), by participating in a multicast routing protocol. The details

1553-877X (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2017.2776213, IEEE Communications Surveys & Tutorials

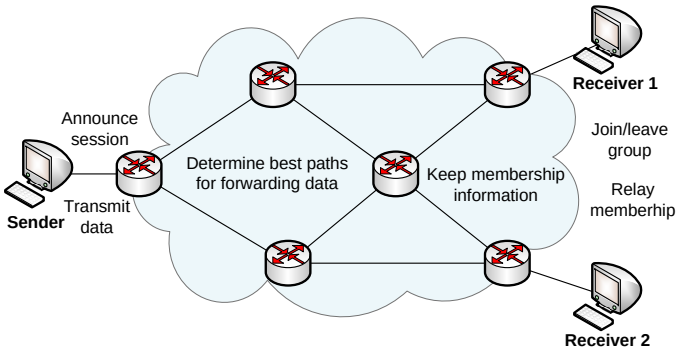


Fig. 2: IP multicast service model

of how this MDDT is constructed will depend on the specific multicast routing protocol used.

4) Finally, there are application-level protocols for creating and managing the multicast data that are distributed through a multicast session.

A wide range of applications involving one-to-many, many-to-many, and many-to-one type communication could benefit from the efficient bandwidth utilization of IP multicast [5]. Some popular multicast applications are audio/video distribution (e.g., IPTV), news or weather updates, file distribution and caching, stock market updates, sensor monitored data distribution, coordination among multiple servers or applications, etc. Among these applications only a few (namely, IPTV and coordination within very large data centers) have successfully leveraged the advantages of IP multicast. The main reason behind the successful deployment of these specific applications is the "walled garden" or restricted approach that has been adopted [6]. Many ISPs have deployed multicast inside their own Autonomous Systems (ASes), but only for applications where they can control all aspects of multicast use. However, large-scale inter-domain deployment of IP multicast has not yet happened.

A number of challenges including scalability, managing dynamic groups, security, etc., that IP multicast needs to address had been identified long before [7]. Note that IP multicast follows a distributed model, which is based on local (adjacent) connectivity information. In this model, each router maintains its connectivity with the routers adjacent to it. Based on this information, the routers collectively build the Shortest Path Tree (SPT). Thus, the way the routing protocol maintains adjacency information and computes the MDDT (i.e., the SPT) is scalable. However, in this paper, the term 'scalability' is used to refer the idea that, with an increase in the number of multicast groups or sources, neither the performance will decline nor the load on the routers will increase. In this context, the MDDT that IP multicast builds is not scalable. In this paper, the scalability aspect is focused on minimizing the number of nodes and edges of the MDDT.

Due to the emergence of programmable networks, and more specifically SDN with OpenFlow, many new features of routing and flexible forwarding semantics (e.g., flooding, multipath, fast reroute, link aggregation and forwarding multiple flows to a common next hop) can be implemented inside the switches. Note that SDN does not provide any out-of-the-box solution for IP multicast. However, by exploiting its group entry (explained in the following section) and complex forwarding semantics, it is possible for the SDN controller to set multicast forwarding rules inside the switches' forwarding tables. Therefore, researchers have concentrated on designing multicast supported controllers and interfaces to program those controllers.

The two largest hurdles that IP multicast needs to overcome are scalability [7] and its `all-or-nothing` deployment nature [6]. For a large dynamic group with many scattered group members, IP multicast does not scale at all and creates many entries in the forwarding tables of the routers. Users will get the benefit of multicasting only if all involved ISPs have deployed it. SDN might be a useful tool to solve many obstacles including these two, which have stopped large scale multicast deployment. In this article, we survey the various efforts that researchers have carried out to study how multicast could be deployed in an SDN-enabled network. Note that "IP multicast in an SDN-enabled network" is also contained in the concept of "IP multicast". Since to use the phrase "IP multicast in an SDN-enabled network" throughout the paper would be extremely cumbersome, we use the short form "SDN multicast" to mean "IP multicast in an SDN-enabled network".

We categorize the existing work on SDN multicasting into two broad categories: multicast routing, and multicast security. The remainder of the paper is organized as follows:

a) Background information: Section II briefly presents the building blocks and protocols of SDN. Section III presents the basics of IP multicasting, and introduces the perspective that multicast routing protocols represent a distributed solution to an optimization problem, subject to certain assumptions and constraints.

b) Multicast routing: Section IV describes examples of IP multicast routing protocols, in recognition of the fact that SDN multicast routing algorithms achieve the same result, but under different constraints. Section V briefly discusses various routing algorithms in SDN multicast and then presents their classification and comparison. Section VI concludes this topic with a comparison of routing protocols in IP multicast and routing algorithms in SDN multicast, and then identifies the open problems that still need to be addressed in SDN multicast.

c) Multicast security: Protecting routing protocol messages, securing data distribution and providing participant access control are three key challenges in IP multicast and in SDN multicast as well. In Section VII we explain different efforts taken by the IETF and individual researchers to address these challenges. Section VIII presents existing work on securing SDN multicast. This topic is concluded in Section IX with a comparison between IP multicast and SDN multicast security services. The open problems that SDN multicast should address are also listed.

d) Conclusion: Section X presents our conclusion.

II. SDN Basics

The development of the concepts of SDN was driven by two complementary approaches: (1) the work of the Forwarding

1553-877X (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2017.2776213, IEEE Communications Surveys & Tutorials

and Control Element Separation (ForCES) working group [8] of the Internet Engineering Task Force (IETF), which grew out of the desire to build hardware forwarding blades out of flexible hardware components; and (2) the work of McKeown [9], which arose from the frustration that network research projects felt at not being able to experiment with new protocols on large-scale networks [10]. Although the basic principle of ForCES and McKeown's work—separating the control and forwarding planes—is the same, there exist some differences in terms of datapath construction, flexibility, and granularity of the protocols being used [10]. The popularity of SDN has grown due to the availability of OpenFlow [11], the protocol that establishes communication between the controller plane and the forwarding plane, which was developed by the Open Networking Foundation (ONF) [12].

In 2012, the Internet Research Task Force (IRTF) has formed the Software-Defined Networking Research Group (SDNRG) with the focus of addressing future challenges in SDN. They have proposed a layered architecture for SDN [13], which is similar to the ONF architecture. Since the work in ForCES is commercially oriented, and the work in SDNRG is looking to future developments, most of the existing research found in the literature is based on the ONF efforts. Therefore, in this paper we will concentrate on the ONF architecture.

A. SDN Architecture

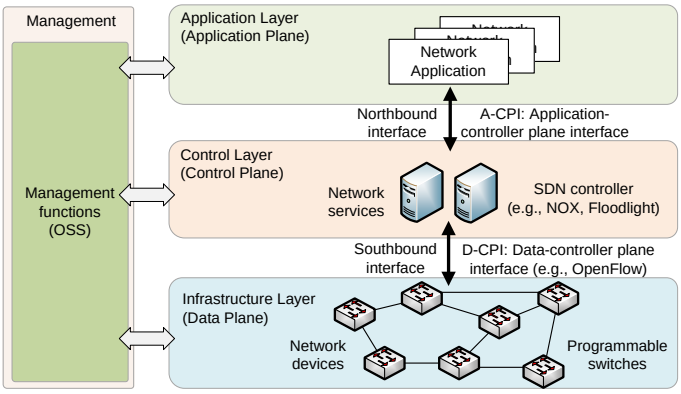


Fig. 3: SDN architecture

The SDN architecture [14] shown in Figure 3 consists of three layers: application, control and infrastructure. This architecture decouples the forwarding plane from the control plane. The forwarding plane is known as the data plane, which is implemented in the infrastructure layer. The network devices of this layer are programmable switches. The main function of these devices is forwarding data packets according to their forwarding tables. They also monitor and gather local information.

The middle layer—the control layer—lies on top of the infrastructure layer. This layer is responsible for control plane services by programming and managing the data plane's network devices. Several software controllers have been designed for this layer. NOX, POX, Beacon and Floodlight are some of the open source controllers. An SDN controller communicates with the data plane through its southbound interface, which is known as the Data-Controller Plane Interface (D-CPI). The ONF has specified the OpenFlow protocol [11] to be used between an SDN controller and the programmable switches. Note that through the OpenFlow protocol, an SDN controller may modify or add entries in the forwarding tables of switches.

The top layer is the application layer, which implements different network applications including security, manageability, load balancing, new features, etc. The controller provides an abstract view to the application layer about the underlying physical network. Using this information the application layer can properly guide the controller. An SDN controller communicates with the application plane through its northbound interface, which is known as the Application-Controller Plane Interface (A-CPI). As of today, the ONF has not specified any protocol for this interface. However, some SDN programming languages have been developed by various researchers, which provide northbound APIs for network applications.

The above-mentioned three layers need various management related configurations. Therefore, in the SDN architecture, a conceptual block called OSS (Operations Support System) has been introduced that hosts all management functions. Although OSS needs interfaces to be connected with all three layers, the ONF has specified the OpenFlow Configuration and Management Protocol (OF-CONFIG) [15] to establish communication between the OSS and an OpenFlow switch. OF-CONFIG makes changes in an OpenFlow switch by considering it as an OpenFlow Logical Switch.

B. Inside an OpenFlow Switch

OpenFlow has emerged as the *de facto* standard for communication between the SDN controller and the SDN switches. Here we briefly explain how an OpenFlow switch takes a decision on packet lookup and performs the actual forwarding. Figure 4 shows different components of an OpenFlow switch, which consists of one or more flow tables and a group table. The switch communicates with one or more controllers using the OpenFlow switch protocol. The flow table and the group table are composed of sets of flow entries and group entries respectively. Figure 5 shows the main components of flow and group entries. The controller using the OpenFlow switch protocol adds, updates, and deletes flow entries in the flow and group tables.

On receiving a packet, a switch extracts the match fields (e.g., ingress port number, source and destination MAC and IP addresses, transport layer port numbers, etc.) from the incoming packet and based on the match fields and priority together determines a unique flow entry. Other than match fields and priority, a flow entry has counters (updated each time a packet is matched), instructions (may update the action list or bypass the packet to some actions), timeouts (life time of this flow), etc. An action might be modifying and forwarding a packet to an output port or forwarding the packet to another flow table or group table.

A group entry has an identifier (a 32-bit unsigned integer), a group type (defines how buckets are implemented), counters, and one or more action buckets. Groups represent complex forwarding rules including flooding, multipath, fast reroute,

1553-877X (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2017.2776213, IEEE Communications Surveys & Tutorials

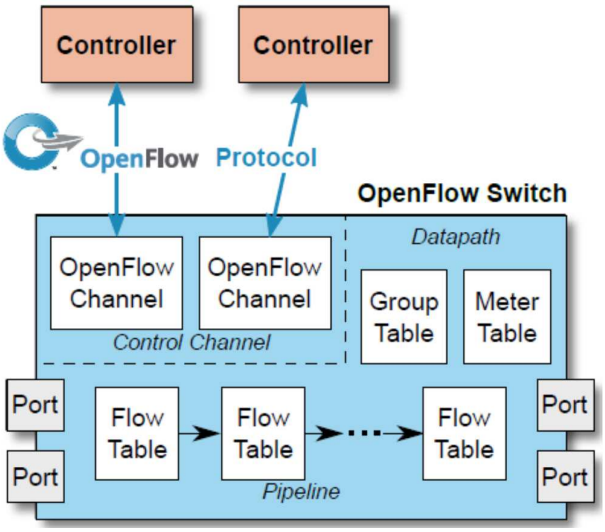


Fig. 4: Components of an OpenFlow switch [11]

link aggregation and forwarding multiple flows to a common next hop. Hence, exploiting the group actions, a multicast routing protocol might be implemented in the OpenFlow switches.

III. Multicasting Basics

A. Architecture

The Internet is a very large distributed system. For ease of management, it is divided into Autonomous Systems (ASes) [16]. Thus, there are three levels of interaction that must be considered when discussing the MDDT: within a local network segment, within an AS (intra-AS), and between ASes (inter-AS).

Within a local network segment, the end user device (host) will indicate its interest in a specific multicast group to the local router. If there is more than one local router on a network segment, one of these is elected as the access router. The multicast routing protocol will typically construct the MDDT only within an AS. If the sender(s) and receiver(s) are in different ASes, it may be necessary (depending on the multicast routing protocol in use) to coordinate the MDDT parts among the involved ASes.

B. Protocols Involved

Successful operation of IP multicast depends on a number of protocols. Since PIM-SM is the most widely accepted multicast routing protocol, in the following we summarize the protocols involved based on PIM-SM:

- **IGMP**: A host uses the Internet Group Management Protocol (IGMP) to join or leave a multicast group. There are three versions of IGMP, of which version 2 (IGMPv2) [17] and version 3 (IGMPv3) [18] are still in use. IGMPv3 supports "source filtering" through which a host can request to receive multicast packets *only* from a specific list of sources, or from *all but* specific sources, sent to a particular multicast group. The scope of IGMP is limited to the one hop between an end host and the directly connected access router. The Multicast Listener Discovery (MLD) protocol [19], [20] provides the same services as IGMP, but for networks using IPv6 addresses.
- **PIM-SM**: Protocol Independent Multicast - Sparse Mode (PIM-SM) [21] efficiently routes multicast data for a group that may span a wide area and be inter-domain in the Internet. PIM is called "protocol independent" because, although it may use the underlying unicast routing to provide reverse-path information for multicast tree building, it is not dependent on any particular unicast routing protocol. PIM-SM is called "sparse mode" for reasons that will be explained in Section III-C. PIM-SM builds the MDDT for a group either as a unidirectional tree, rooted at an administratively-defined point in the network, shared among all the sources, or as a set of shortest-path trees, each rooted at a source. This administratively-defined point is called the Rendezvous Point (RP). See Section III-D for details.
- **MSDP**: To work smoothly in inter-domain environments, PIM-SM depends on Multicast Source Discovery Protocol (MSDP) [22] and Multiprotocol Extensions for BGP-4 (MBGP) [23]. PIM-SM works inside a single domain. It builds the MDDT routed at the RP when the RP and the source reside in the same domain. RPs from different domains are connected to each other through MSDP peering. Through MSDP, an RP learns about the location of a specific source.
- **MBGP**: Regular Border Gateway Protocol (BGP) has been designed to exchange routing information between two ASes. MBGP adds additional features to regular BGP so that it can carry multicast routing information. This routing information is used by PIM-SM.

Note that each of the protocols mentioned in this section have been standardized by the IETF.

C. Multicast Routing Optimizations

Communication from m sources to n destinations is a very general problem. In multicast routing (in general), the objective is to send data from the source(s) to multiple destinations, while minimizing some cost function [24]. When each IP multicast routing protocol was designed, certain assumptions were made about the conditions under which it would operate, and the constraints that would be imposed on its algorithms. The emergence of SDN provides an opportunity to change these assumptions and/or relax the constraints, i.e., to vary the cost function. The *task* of a multicast routing protocol is to construct an appropriate MDDT. Thus, following [24], we regard the design of an IP multicast routing protocol as an optimization problem, subject to certain assumptions and constraints.

- **Assumption: distribution of receivers** The assumption about the expected distribution of receivers is fundamental to the design of a suitable protocol.
  Dense mode protocols are designed under the assumption that the group members are likely to be located close to each other (i.e., their geographical distribution will be "dense"). The available bandwidth tends to be large, and

1553-877X (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2017.2776213, IEEE Communications Surveys & Tutorials

Match Fields	Priority	Counters	Instructions	Timeouts	Cookie	Flags

(a) Flow entry in a flow table

Group Identifier	Group Type	Counters	Timeouts	Action Buckets

(b) Group entry in a group table

Fig. 5: Main components of a (a) Flow entry and (b) Group entry [11]

members are available in most (or all) of the sub-networks involved.

Sparse mode protocols are designed under the assumption that the group members are likely to be located far from each other (i.e., their geographical distribution will be "sparse"). The available bandwidth tends to be small, and members are available only in some of the sub-networks involved.

- **Constraint: local information** For a classical IP multicast routing protocol, the constraint that is imposed on the design of its algorithms is that the individual nodes must act entirely on local information. Each router makes its own, independent decisions, which depend only on information that is available to itself. This information includes the local unicast Forwarding Information Base (sometimes called the routing table), the interface on which the RP can be reached, and the interface(s) on which clients have appeared.
- **Optimization goal** For a classical IP multicast routing protocol, there are two predominant optimization goals:
  1) minimize delay from source to receiver(s);
  2) minimize storage space for router state.

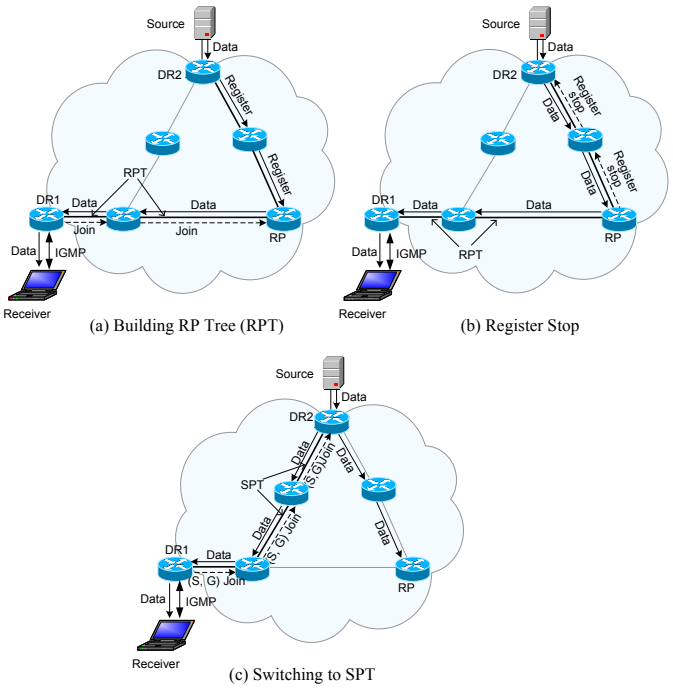
  These goals lead to two design architectures [25], [26]:
  1) source-based tree (SBT);
  2) core-based tree (CBT).

  An SBT-based protocol will construct an MDDT per source, for each active group. A CBT-based protocol will construct a single shared tree per group, rooted on a core router. If the number of sources is greater than one, the CBT design will result in fewer entries in the MRIB. However, as the source(s) must send its/their packets to the core router first, a CBT-based protocol will introduce a larger delay than an SBT-based protocol would.

D. PIM-SM Protocol

To permit a better understanding of multicast routing, the operation of PIM-SM will be described. A PIM-SM router operates entirely on information available to it directly. PIM-SM is a sparse-mode protocol.

Under PIM-SM, multicast data are forwarded from sources to receivers without either the sources or receivers knowing beforehand about the existence of the others. PIM-SM depends on the forwarding table that is stored in the Multicast Routing Information Base (MRIB). The MRIB provides the next hop router along a multicast-capable path to each destination subnet. Thus, the MRIB determines the path along which a Join/Prune PIM message will be forwarded. Data are forwarded in the reverse direction of the Join/Prune message. PIM-SM is done in three phases, which are shown in Figure 6:



(a) Building RP Tree (RPT) (b) Register Stop (c) Switching to SPT

Fig. 6: Three phases of PIM-SM protocol

(a) Join message from a receiver is forwarded towards the RP and the shared tree that converges to the RP is built. This shared tree is called the RP Tree (RPT). The Designated Router (DR) of the sender (directly connected router) encapsulates multicast traffic inside Register messages and sends them to the RP, which decapsulates the multicast data and forwards them through the RPT.

(b) The RP sends a Source-Specific Join message, `(S, G)` to the source, to form a *source-specific* tree between the source and the RP. From this point in time, the RP receives two copies of the data, one is through the source-specific tree and the other is in encapsulated form. The RP then sends a Register-Stop message to the DR connected to the source, which on receiving this message stops sending encapsulated data.

(c) Under certain conditions, the DR of the receiver may send

1553-877X (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2017.2776213, IEEE Communications Surveys & Tutorials

a Source-Specific Join `(S, G)` message towards the source and the Shortest-Path Tree (SPT) converged at the source will be built. From this point, the DR of the receiver receives two copies of data from the SPT and the RPT. The DR of the receiver then sends an `(S, G, rpt)` Prune to the RP to prune the RPT. From this point onward, the DR will receive only one copy of the multicast data.

Thus it can be seen that the initial optimization goal of PIM-SM is to minimize the router state, but that if the conditions are met, it will convert that multicast group to have the goal of minimizing the delay. As a result, the architectural model will switch from CBT to SBT.

E. IP Multicast Challenges

Classical IP multicast has faced a number of challenges from its inception. Several limitations have been identified by researchers from different points of view [7], [6], [27], [28], [29], [30]. Although these limitations have been identified quite a long time ago, most of them are still present and thus relevant to study. In the following we have summarized the findings:

- **Scalability of the MDDT:** The MDDT is maintained by multicast-capable routers to control and deliver traffic to the receivers. An SBT-based protocol, using adjacent local information, constructs the per-source MDDT, based on the Shortest-Path Tree (SPT) algorithm. However, the MDDT that IP multicast builds is not scalable in terms of number of multicast groups, as its construction will cause an additional burden on routers to store the per-group per-source state. To improve the scalability, a CBT-based protocol constructs a single MDDT per group, which is shared by all the sources within that group. However, it is not scalable with the increasing number of nodes and edges of the multicast tree. In graph theory, the optimal shared tree is the Steiner tree [31], which has the minimum number of nodes and edges. The Steiner tree problem is NP-complete, which means that there is no polynomial-time algorithm to solve it. Finding a solution for the Steiner tree problem requires knowledge of the complete network topology, which makes it difficult to deploy as a distributed protocol. A CBT-based protocol may create a suboptimal MDDT. Note that the optimization goal (see Section III-C) of minimizing router state has the side effect of improving scalability.
- **IP Multicast route aggregation:** Frequently, a unicast routing protocol can replace a set of routes by a single route and thus significantly reduce the number of entries in the forwarding table. This is called route aggregation. If IP unicast addresses for an organization are allocated from contiguous ranges, the use of aggregation can result in a substantial reduction in the necessary forwarding state. In the case of multicasting, the multicast group addresses are allocated randomly, and the randomly distributed group members join these groups. In a naive implementation of multicast forwarding, the router must keep the forwarding state for every multicast group passing through it. Under these circumstances, route aggregation for forwarding states in routers is difficult. Different ideas have been explored to aggregate forwarding states. In [27], a per-interface state aggregation technique has been proposed where, depending on the group addresses, adjacency entries for a number of groups at an interface can be aggregated as a single range. State aggregation could also be achieved by forcefully sharing a single distribution tree by multiple multicast groups [32]. However, the amount of state reduction through route aggregation in IP multicast is not as high as it is in unicast.
- **SBT vs. CBT:** The PIM-SM specification states that for low rate sources, the RPT will be used to distribute multicast data. If the data rate from a source crosses a preconfigured threshold, the source-rooted shortest-path tree (i.e., the SPT) will be formed and all receivers will switch from the RPT to the SPT. However, in practice most of the ISPs configure the threshold as 0 kb/s, which results in an immediate switch for an access router from the RPT to the SPT. To be a part of the distribution tree, intermediate routers have to keep path information. In the case of an RPT, a single entry is enough for many sources that share the same tree. However, for each SPT a new entry is required, which increases the memory consumption on the router and the size of the multicast routing table in the multicast design. Moreover, in practice a static RP is used, although many research efforts are found in the literature that establish the benefits of RP relocation by forming an optimal RPT (one such example is found in [33]). Note that choosing a CBT architecture reduces the requirements for router state, and so is the chosen solution when the optimization goal is reduction of router state.
- **Reliability:** Since IP multicast traffic is always sent over the UDP transport layer, reliability is not available in IP multicast. Achieving TCP-like reliability in case of IP multicast is always challenging. Due to the distributed nature of multicast communication, the number of possible failure modes is large. Reliability is essential for many multicast applications; for example, a software developer distributing the latest update to its clients, or a financial institution providing a share-market transaction report to its subscribers, etc. These applications can tolerate delays but cannot function properly with data loss. Supporting reliability thus can be seen as another optimization goal, although satisfying this goal requires knowledge of the application, in addition to knowledge of routing.
- **Quality of Service:** Multimedia applications (e.g., audio streaming, video streaming, online multiplayer games, video conferences) demand stringent Quality of Service (QoS) requirements in terms of end-to-end delay, delay jitter, and packet loss, in order to offer a high quality of service to users. Different multimedia applications require different QoS. Hence, it must be possible to satisfy a set of QoS constraints for a specific multicast stream. As with reliability, this is difficult to do when the QoS capabilities of the different parts of the MDDT are not identical.
- **On-the-fly delivery tree construction**: Due to the dynamic nature of multicast groups, receivers can join/leave any time. Similarly, any sender can start sending any

1553-877X (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2017.2776213, IEEE Communications Surveys & Tutorials

time. Whenever the delivery tree is changed, the related routers have to update their forwarding tables. In the case of an RPT, a new sender does not change the distribution tree. However, a new distribution tree needs to be established if an SBT architecture is being used. If the group has changing membership, this can introduce additional delay, thus compromising a goal of minimum delay.

- **Mobility:** In a mobile environment, both in IPv4 and IPv6, IP multicast faces a number of additional challenges [30] including routing tree construction and maintenance, receiver's join/leave latency, duplicate packets, packet loss, security, QoS, scalability, etc. Although mobility has been studied in IP multicast, we have found no articles that study mobility in SDN multicast. Therefore, this challenge is not further discussed in the rest of the paper.
- **Secured data distribution:** IP multicast was initially specified without any security services. As a result, none of the basic security services—confidentiality, authentication, integrity—were included. Hence, any rogue sender can inject malicious data, and any intermediate node may change or remove data. In a secured multicast group, the sender sends data in encrypted format and the receivers should have the appropriate key to decrypt it. To protect multicast data, different issues such as registration of group members, key generation, key distribution and key management should be considered. The multicast Security (MSEC) Working Group at the IETF has done extensive research in this area. They have already developed a Security Architecture [34] for multicast communication. Moreover, to protect IP multicast packets using IP security (IPsec) services, some extensions to standard IPsec architectures have been specified in [35].
- **Participant access control:** IP multicast without any participant (receiver and sender of multicast group) access control (i.e., authentication and authorization) is vulnerable to many attacks. Without access control any unauthorized host may join a group or change the IGMP reception state at the access router. Any forged sender may generate a replay attack, sender address spoofing or even a DoS attack. The IETF has not proposed any specification for participant access control. A comprehensive participant access control architecture has been designed by Islam and Atwood [36], [37]. It provides receiver and sender access control by incorporating the AAA (i.e., Authentication, Authorization and Accounting) framework into the existing multicast model. An improved proposal for receiver access control is given in [38].
- **Billing:** One of the drawbacks of classical IP multicast is lack of any revenue collection and distribution mechanism. No business model has been developed to generate revenue through content distribution that is being distributed by leveraging IP multicast. For example, an intermediate ISP will deploy a multicast routing protocol to carry multicast data only if it can benefit financially. In [36], a receiver access control mechanism has been presented that provides a business model so that the group owner may collect revenue from an end user before granting access to a secured multicast group.
- **Low motivation for network service provider:** A network service provider (e.g., Internet Service Provider (ISP)) has to deploy a number of protocols to be part of the distribution tree. However, an ISP that has deployed those protocols will not receive any direct benefit through this. For example, an ISP that has no source/receiver of a particular multicast group has very low motivation to deploy multicast specific protocols. Due to such poor business model, most of the ISPs have not deployed multicast routing protocols.

IV. IP Multicast Routing Protocols

Multicast routing is a very broad problem area. Several excellent surveys of IP multicast routing exist, focusing on: protocols, functions, and mechanisms [39], algorithms and protocols [40]; and intra- and inter-domain protocols [41]. This section presents some example protocols, based on the applications for which they are suited, intended to make understanding the SDN multicast approaches easier. The selection is by no means exhaustive. We present three categories: application-independent protocols, application-aware protocols, and the specific problem of video distribution.

A. Application-independent IP Multicast Routing Protocols

Based on the assumed distribution of receivers, intra-domain multicast routing protocols are normally classified into two categories: dense mode and sparse mode. The taxonomy of multicast routing protocols based on the assumed distribution of receivers is shown in Figure 7.

1) Dense Mode Protocols: Dense mode protocols operate by flooding multicast traffic through all sub-networks. If there are no active receivers in a branch, then that branch of the tree is pruned. This strategy is most suitable for areas with a dense concentration of group members.

All dense mode protocols have the implicit goal of minimum delay, because this broadcast-and-prune behavior constructs a shortest-path tree from each source to the associated receivers. The major disadvantage of all dense mode protocols is that they do not scale well, owing precisely to this broadcast-and-prune behavior.

The following are examples of protocols based on dense mode: Multicast Open Shortest Path First (MOSPF) [43] and Distance Vector Multicast Routing Protocol (DVMRP) [44] .

a) Multicast Open Shortest Path First (MOSPF): MOSPF is a link state routing protocol that uses the Bellman-Ford algorithm to find the shortest paths from the source to the destinations. A disadvantage of MOSPF is that the computational complexity of the link-state algorithm increases proportionally with the number of multicast groups and group member sizes.

b) Distance Vector Multicast Routing Protocol (DVMRP): DVMRP is built on the principles of the Routing Information Protocol (RIP), which is a distance vector unicast routing protocol. DVMRP extends the concepts of RIP to multicast. A disadvantage of DVRMP is the large amount of router state that it requires.

1553-877X (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2017.2776213, IEEE Communications Surveys & Tutorials

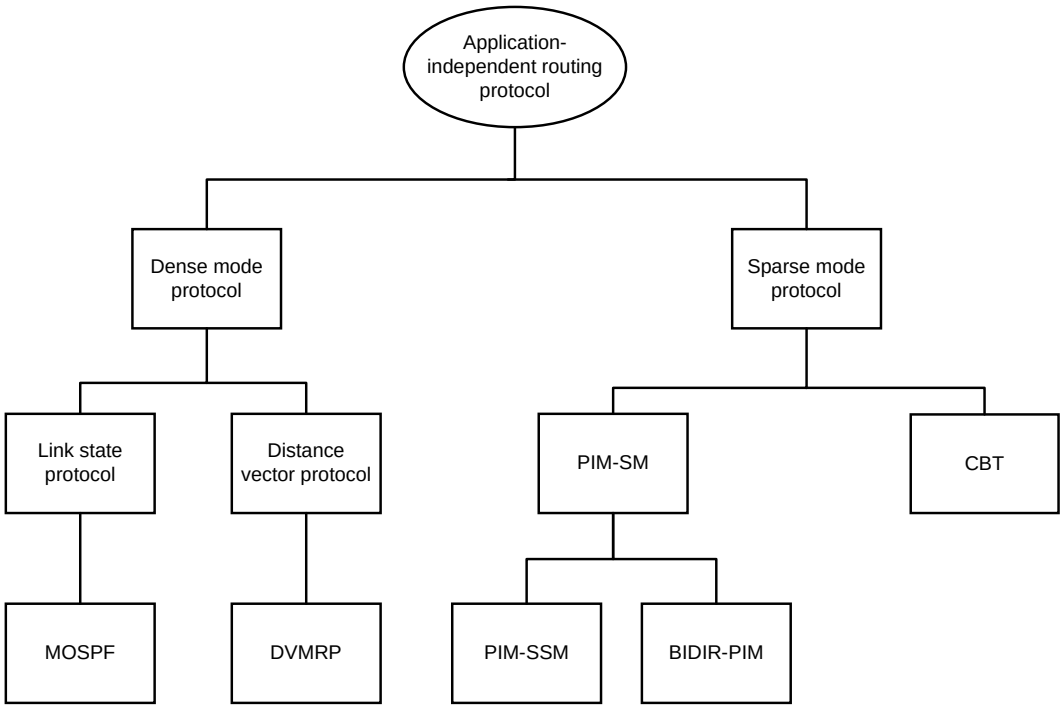


Fig. 7: Taxonomy of IP multicast routing protocols based on the assumed distribution of receivers [42]

2) Sparse Mode Protocols: Sparse mode protocols construct the MDDT in a receiver-driven fashion: a prospective group member must explicitly indicate its desire to join the MDDT, by sending an IGMP join message to its access router. It is then the responsibility of the routing protocol on the access router to find the appropriate place to graft itself onto that tree.

Sparse mode protocols typically have as their goal the minimization of router state. Thus, these types of protocols usually offer better scalability than dense mode protocols, because only the routers on paths from the source to all group members need to keep track of the multicast group. Additionally, sparse mode protocols are more efficient, since they do not flood the network to establish the multicast tree.

The following are examples of protocols based on sparse mode: Protocol Independent Multicast - Sparse Mode (PIM-SM) [21] and Core Based Tree Protocol (CBT) [26].

a) Protocol Independent Multicast - Sparse Mode (PIM-SM): PIM-SM is the most popular CBT protocol. In a CBT protocol, a separate tree is built for each multicast group. The root of the tree is a selected node in the network, which is also known as the Rendezvous Point (RP). The advantage of a CBT architecture is that irrespective to the number of sources, a single tree needs to be constructed for each multicast group. As a result, multicast routers need to keep less state information, which is proportional to the number of multicast groups. Since the single shared tree is used by all sources, the architecture of the tree might not be the optimal tree for all sources. This may result in increased end-to-end delays of multicast traffic. Therefore, the design goal of PIM-SM is to minimize that amount of router state at the expense of low delay.

In PIM-SM, a receiver sends its intention to join a multicast group towards the RP and thus joins the RPT. However, when the data rate of a source exceeds a certain threshold, it is able to switch to an SBT. To construct an optimized path, the intermediate routers will send a join message towards the source and prune the tree towards the RP. This allows operator-controlled change of the design goal to minimize the delay. PIM-SM is also known as Any Source Multicast (ASM), because the receivers receive multicast traffic from any source managed by the RP, and because PIM-SM does not allow any filtering based on the sources.

Source-Specific Multicast (SSM) [45] is a variation of PIM-SM. In SSM, through IGMPv3 join messages, the receivers indicate the source address from which they want to receive multicast traffic. SSM has important benefits over ASM because an SSM channel is defined by both a source and a group address. In SSM, group addresses can be reused by multiple sources while keeping channels unique. It also protects against denial of service attacks. However, SSM exhibits a serious state scalability problem when there is a large number of simultaneous on-going multicast groups in the network [46].

Bidirectional Protocol Independent Multicast (BIDIR-PIM) [47] is another variation of PIM-SM, which creates bidirectional shared trees. In BIDIR-PIM, the source routers can forward traffic downstream toward the RP, while at the same time, the RP can use the same shared tree to send the multicast traffic upstream to a receiver. In this case, minimum delay is ensured. To prevent loops along the path, BIDIR-PIM uses a Designated Forwarder (DF) election mechanism, operating on each segment of a multicast topology. This DF is the only router that is allowed to send multicast traffic towards the RP. Figure 8 shows network traffic flow using BIDIR-PIM protocol.

1553-877X (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2017.2776213, IEEE Communications Surveys & Tutorials

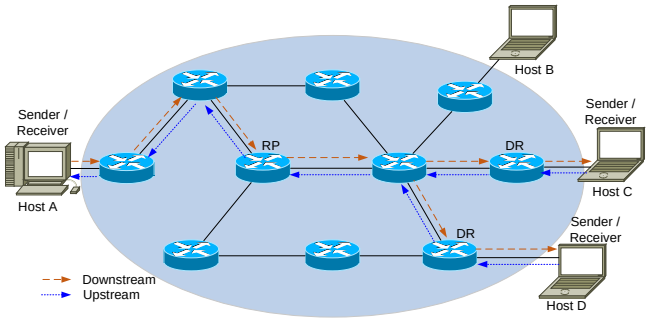


Fig. 8: BIDIR-PIM protocol

b) Core Based Tree Protocol (CBT): CBT is similar to PIM-SM. Nonetheless, CBT has significant differences compared with PIM-SM. Firstly, CBT always uses a shared tree and there is no option to construct the SBT. Secondly, the shared tree is bidirectional, whereas PIM-SM constructs only a unidirectional shared tree. Thus CBT is like BIDIR-PIM, in that the design goal is minimum delay. In PIM-SM, all the packets are sent to the RP, which then sends them down the tree to all the members. However in CBT, intermediate routers directly deliver the packets to the attached members while forwarding the packet towards the RP.

B. Application-aware IP Multicast Routing Protocols

There is often a substantial advantage in achieving a desired operation when specific goals or requirements of an application are combined with the goals of routing. For IP multicast, these application-level goals are normally implemented by a mechanism that is distributed throughout the network, and whose components may or may not be co-located with the routers. These components communicate amongst themselves to achieve the desired overall goal.

1) Reliable IP Multicast Protocols: IP multicast has the same delivery model as IP unicast: best-effort delivery with no guarantees. Many applications require "reliable" multicast, but the range of meanings for "reliable" is quite large, which has resulted in a number of quite different proposals for reliable multicast protocols.

To support reliable transmissions in IP multicast, Floyd et al. presented Scalable Reliable Multicast (SRM) [48], a source-based reliable multicast algorithm to recover the lost packets from the source. However, this approach has poor scalability, because the source needs to provide loss recovery for a potentially large number of destinations. To solve this problem, Speakman et al. proposed Pragmatic General Multicast (PGM) [49] and Paul et al. [50] proposed Reliable Multicast Transport Protocol (RMTP) to place recovery nodes (designated local retransmitters) between the source and the destinations to facilitate data loss recovery. Whetten et al. [51] introduced a new version of the protocol RMTP II.

Atwood [52] provides a taxonomy for reliable IP multicast protocols. He examines the set of requirements that each one meets, the architecture that they use for interaction among the participants, the mechanisms that are used to effect the interaction, and the policies that determine how and when to use the mechanisms.

Each of these reliable IP multicast protocols is clearly an "application aware" multicast routing protocol, which incorporates its definition of "reliable" into the assumptions, goals and constraints imposed on its design.

2) QoS-aware Approaches: If a particular application is aware of its QoS needs, it can use the Resource Reservation Protocol (RSVP) [53] to negotiate its QoS needs with the network, in conjunction with whatever multicast routing protocol is in use.

Two different approaches have been proposed in the literature to accommodate heterogeneous multicast applications and clients. In the first approach, a multicast source may vary the transmission rate by using adaptive coding [54] or combining a layered compression algorithm with a layered transmission scheme [55]. In this case, QoS is not directly supported by the multicast routing protocols, but by the end systems. In the second approach, multicast routing protocols could provide service to the clients with different processing capabilities and QoS requirements. Not all receiving devices have the same capacities, and not all paths between a specific source and its set of receivers have the same capacity. In the case of streamed media, rather than reducing all receivers to the level of the least-capable receiver, it is often acceptable to deliver different QoS to different receivers, in keeping with their capabilities and/or the observed quality of the path. A multicast routing protocol could then take these requirements of the client into consideration during the routing decision. Tyan et al. [56] extended the CBT protocol to support different QoS among the users.

C. Limitation of IP Multicast Routing for Video Distribution

Video streaming over the Internet has become popular due to rapid developments and improvements in network infrastructures, storage capacity, and computing power, which are enabling an increasing number of video applications. Video stream multicast represents a challenging problem, since the goal is to deliver high reliability (but not perfect reliability) as well as low latency. However, due to the discrepancy of network connection qualities and terminal capacities noted above, it is difficult to provide such a facility for video streaming.

Scalable Video Coding (SVC) [57] is an important solution to the problems posed by the characteristics of video stream multicast. SVC compresses video into multiple non-overlapped layers of streams, with one base layer and multiple enhancement layers. The base layer provides a basic quality for a video stream. A client receiving one or more enhancement layers can increase the video quality of the client. By applying SVC, it is possible to deliver different video qualities to different clients based on their demand. In IP multicast, this can be achieved by offering a different multicast group for each layer, and ensuring that a receiver monitors its received QoS, with appropriate adjustments in its group membership if the observed QoS changes. In this case the multicast routing is determined by the multicast routing protocol being used, and

1553-877X (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2017.2776213, IEEE Communications Surveys & Tutorials

the application-level goals are satisfied using local information at the receiver.

It is difficult to implement SVC because IP multicast utilizes the traditional distributed network in which routers have difficulty to receive global network status and are incapable of calculating an optimal multicast tree to maintain QoS for video streaming. Due to the dynamic nature of the network, the SVC enabled multicast video distribution tree needs to adjust dynamically, which puts an extra burden on the router's processing capability.

V. SDN Multicast Routing Algorithms

The Internet is a decentralized, distributed system, consisting of a number of routers and communication links that provide connectivity to end systems for running network applications. In this distributed system, classical IP multicast routing protocols operate under the constraint of using only localized information about the present connectivity. Furthermore, in response to a change in the group membership, these protocols are slow to recalculate the delivery tree due to the time required to exchange routing information among the routers.

A centralized approach using SDN to perform multicast routing offers several advantages. It has a complete view of the topology. Thus it is possible to remove the constraint of using only local information, to change the assumption about the number of paths between the source and the destinations, and to consider optimization goals other than minimum delay or minimum router state. For example, IP multicast routing protocols typically have "minimize delay" as their objective, and they construct an SPT between the source and the destinations. SDN multicast routing algorithms, in contrast, can be designed to minimize overall resource consumption in the network, because the SDN controller has a global view, and therefore can construct a Minimum Spanning Tree (MST), subject to a variety of constraints. For common data-center network topologies, it is possible to create an optimal delivery tree in polynomial time [58]. It is also possible to respond faster for delivery tree change [59]. As with IP multicast, researchers have developed SDN routing algorithms either for application-independent purposes (i.e., suitable for any multicast dependent application) or targeting specific applications. Based on the applications for which they are suited, SDN multicast routing algorithms can be divided into three categories: application-independent algorithms, application-aware algorithms, and topology-aware algorithms. The details are given in subsequent sections; a summary is shown in Figure 9. When referring to different approaches, if no specific name was given by the author(s), we have introduced names that best describe the routing algorithms. Note that we use the term "routing algorithms" in this section, rather than "routing protocols", because there are, in fact, no protocols. The calculations are performed centrally, so there is no need to communicate from one decision point to another (using a routing *protocol*), as there is in classical IP multicast.

Note also that, in contrast with IP multicast, there are no "standards" for SDN multicast routing. The range of possible optimizations is so large, and the approaches so flexible, that it is unlikely that any one approach is going to be dominant. IP multicast is made operational in a network by enabling the multicast functionality provided by the router manufacturer. SDN multicast is made operational by writing an application for the SDN controller that optimizes the traffic flows to meet the particular needs of the end user. These applications will be specific to the desired optimization goal, and as such, not likely to be candidates for standardization. As a result, discussion of SDN multicast algorithms is of necessity an exploration of "proposals to meet an optimization goal" rather than "proposals for new standards".

The routing algorithms in this section have been analyzed using the following pattern:

- **Motivation:** What problem is being solved, and/or what optimization is driving the work?
- **Gaps covered:** What features of existing protocols/algorithms are missing, and solved by this approach?
- **Operation:** What does this algorithm do, and how does it work?
- **Test environment:** How is it validated (if at all)?
- **Test results:** What new insight is produced?
- **Comparison:** How does this algorithm compare with other possible approaches?
- **Possible extensions:** How could the algorithm be made better?

Not all algorithms have all aspects available in the literature, so certain aspects will be missing from some descriptions. A summary of some of the important aspects may be found in Table I.

A. Application-independent SDN Multicast Routing Algorithms

An application-independent routing algorithm is not used for any particular type of specialization. Its goals are strictly related to routing. It can be used in a range of circumstances such as data, audio or video streaming.

1) Castflow: Marcondes et al. [60] proposed Castflow, which provides a faster response time during multicast group events (i.e., joining or leaving a multicast group by a host).

Existing IP multicast protocols use a distributed approach and rely on local information to develop the MDDT. During multicast group events, this process is slow to respond because routers need to exchange information among themselves and update their routing tables. Additionally, it generates a flood of messages, which reduces the bandwidth of the network. Castflow uses a centralized approach to create an optimal multicast distribution tree and thus provides faster response without flooding of messages.

Castflow applies Prim's algorithm to calculate all possible routes between the sources and each host of the multicast group in advance by utilizing the global view of the SDN controller. Prim's algorithm [61] generates the MST as a function of the weights associated with the edges of the graph. Figure 10 shows how Castflow builds the data distribution tree using Prim's algorithm, where `Client1` receives data via the path, N1-N3-N2-N4-N9-N13-N12.

1553-877X (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2017.2776213, IEEE Communications Surveys & Tutorials

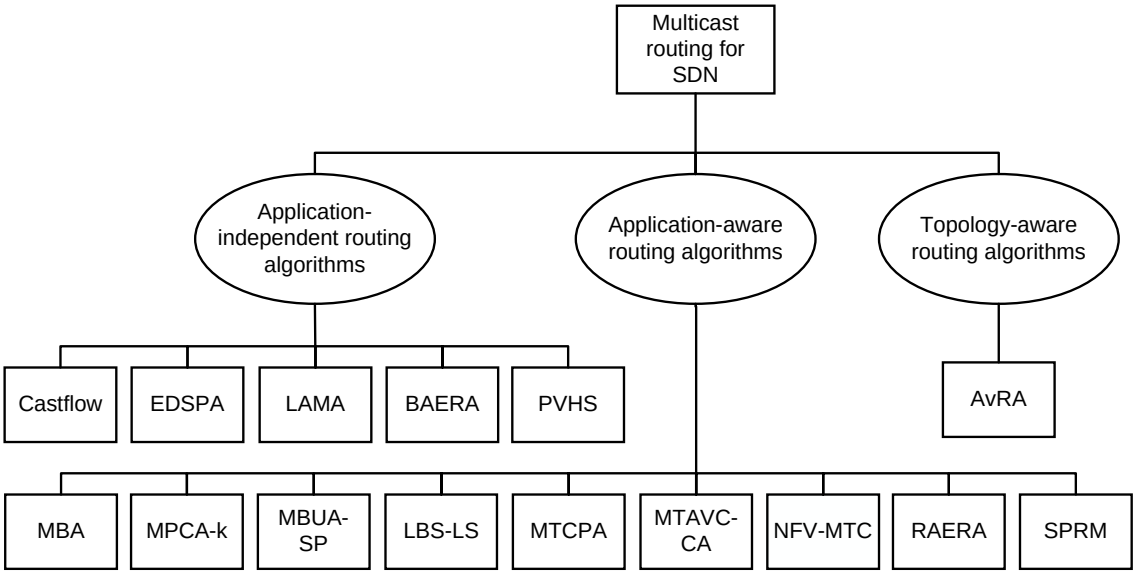


Fig. 9: Taxonomy of SDN multicast based on algorithms based on applications

To evaluate the performance, Castflow is implemented and tested in the topology emulation platform Mininet [62] with 10 to 50 nodes. NOX [63] is used for the SDN controller. BRITE [64] is used to generate network topologies and a custom program is used to emulate topology events. To analyze the efficiency of the algorithm, the tree construction time is measured by varying network sizes. Although the multicast distribution tree construction time shows linear growth, it is not compared with other IP multicast algorithms. The performance of Castflow needs to be measured in an SDN testbed with a large number of nodes.

2) Extended Dijkstra's shortest path algorithm (EDSPA): Ananta et al. [65] proposed to apply Extended Dijkstra's Shortest Path Algorithm (EDSPA) [66] to calculate the shortest path from source to destination to minimize delay.

Dijkstra's algorithm computes the shortest path from source to destination considering only the edge weights. If the average processing delay at a node is significant, it is possible to include this average processing delay while determining the edge weights, and obtain a better result. In Extended Dijkstra's algorithm, the node weight of the destination node is *not* added to the edge weight of the arrival edge, thus giving an even more correct result.

Figure 11 shows the data distribution tree using EDSPA [65] where `Client1` receives data via the path, N1-N2-N4-N9-N12.

EDSPA is implemented in the emulation platform Mininet [62] with 11 switches. The simulation experiment compares the performance of EDSPA with Dijkstra's Shortest Path (DSPT) algorithm and the Best-First Search Tree (BFST) algorithm in terms of throughput, jitter, and packet loss in the Abilene network topology [67]. The results show that EDSPA outperforms the two other algorithms. The performance of EDSPA needs to be evaluated in an SDN testbed with a large number of nodes.

3) Locality-Aware Multicast Approach (LAMA): Lin et al. [68] proposed a Locality-Aware Multicast Approach (LAMA) in SDN to minimize the computation time of the multicast distribution tree and the number of flow entries in the switches to enhance the scalability of the controller and the switches.

Castflow [60], EDSPA [65] and MTCPA [69] algorithms are based on per-source trees. Note that the SMVCMTC algorithm is further discussed in Section V-B5. These algorithms increase the computation time of the controller and the number of flow entries installed in the switches for video streaming. In order to solve this scalability problem, LAMA constructs a shared multicast tree to maintain a desirable video quality.

LAMA adaptively clusters several multicast groups into a single multicast-cluster and thus, a shortest-path multi-group shared tree is built for each multicast-cluster. The algorithm consists of three steps. In the first step, a distance-based clustering algorithm, named Complete linkage clustering [70], is used to cluster the multicast sources located in the vicinity. In the second step, a locality-aware RP selection algorithm [71] is applied to determine the proper RP that has the minimum distance to all multicast sources within a multicast cluster. The final step constructs the shortest paths from each multicast source to the selected RP in the same cluster and also constructs the shortest paths from RP to the hosts for each multicast cluster.

LAMA is implemented and tested on the topology emulation platform Mininet [62] with 1000 nodes, 100 multicast sources and 500 clients. Ryu [72] is used as the SDN controller and BRITE [64] is used to generate network topologies. The simulation results show that the time required to generate the distribution tree is 70 ms, which is significantly faster compared to per-source tree. Additionally, the total number of installed flow entries is about 94% lower compared to the per-source tree approach. However, the simulation results do not investigate the bandwidth utilization overhead incurred by this approach [73]. LAMA is yet to be implement in an SDN testbed to measure the bandwidth utilization overhead.

4) Branch Aware Edge Reduction Algorithm (BAERA): Huang et al. [74] proposed the Branch-aware Steiner tree

1553-877X (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2017.2776213, IEEE Communications Surveys & Tutorials

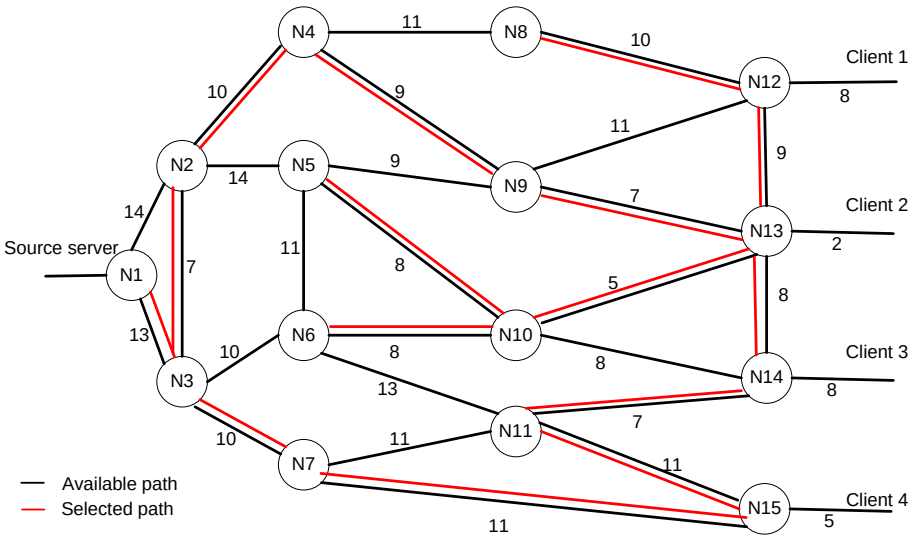


Fig. 10: Castflow [60] distribution spanning tree using Prim's algorithm

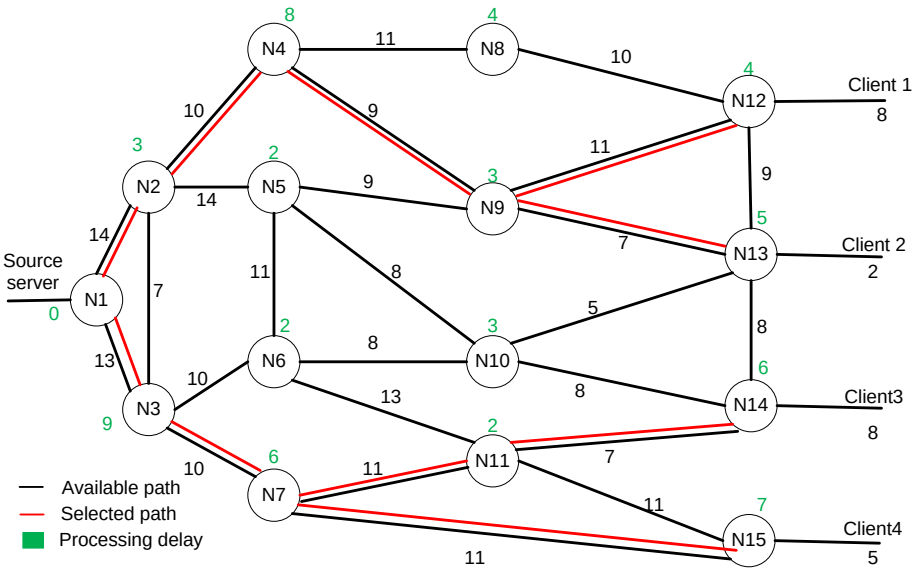


Fig. 11: Data distribution tree using EDSPA [65]

(BST) to improve the scalability of multicast routing in SDN by minimizing the total number of edges and branch nodes of the multicast tree.

Both Castflow [60] and EDSPA [65] work well for a limited number of nodes, but show poor scalability performance. In SDN, the forwarding rules are implemented inside networking equipment, such as high-performance routers and switches, using expensive but high-speed memory (e.g., Ternary Content Aware Memory (TCAM)). As a result, the number of forwarding rules or the size of the forwarding tables are limited by the available memory capacity. Clearly, the goals of these two algorithms (minimum response time and minimum delay) are achieved at the expense of poor scalability. To solve this scalability problem in multicasting, Tian et al. [75] proposed the branch forwarding technique where the forwarding states need to be stored only in the branch nodes rather than in every node of a multicast tree. A branch node has at least three incident edges. To improve the scalability properties further, higher weight can be assigned to each branch node in BST.

The BST problem is NP-hard. To solve this problem, an approximation algorithm called Branch Aware Edge Reduction Algorithm (BAERA) was designed. BAERA includes two phases: edge optimization and branch optimization. In the edge optimization phase, at each iteration, BAERA iteratively chooses and adds a terminal node to the solution tree $T(V^T, E^T)$ for constructing a basic BST, where V^T and E^T denote the nodes and edges currently in T. Initially, a random root node is added to the V^T. Then, the BAERA algorithm finds the minimum distance $d_{v,T}$ between the root node and terminal node v where $v \varepsilon k$ and $k \subset V$ given $G(V, E)$. To reduce the number of edges, two paths are aggregated that share common edges by substituting the shortest path $d_{v,T}$ with a longer path $d^1_{v,T}$. However, more branch nodes will be generated when more paths are aggregated in the solution tree. To solve

1553-877X (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2017.2776213, IEEE Communications Surveys & Tutorials

this problem, the branch optimization phase re-routes the tree T, to reduce the number of branch nodes. Figure 12a shows the data distribution tree using BAERA (edge optimization phase), where N1 is the root node. Node N7 is connected with node N1 via node N2. Node N5 is connected with node N2 via node N3 and node N4. Node N30 is connected with the solution tree via N26. Node N29 is connected with the solution tree via created branch node N26. Figure 12b shows the data distribution tree using BAERA (branch optimization phase), where branch node N2 is removed. Then, node N1 and N7 are re-routed to the solution tree T via node N15.

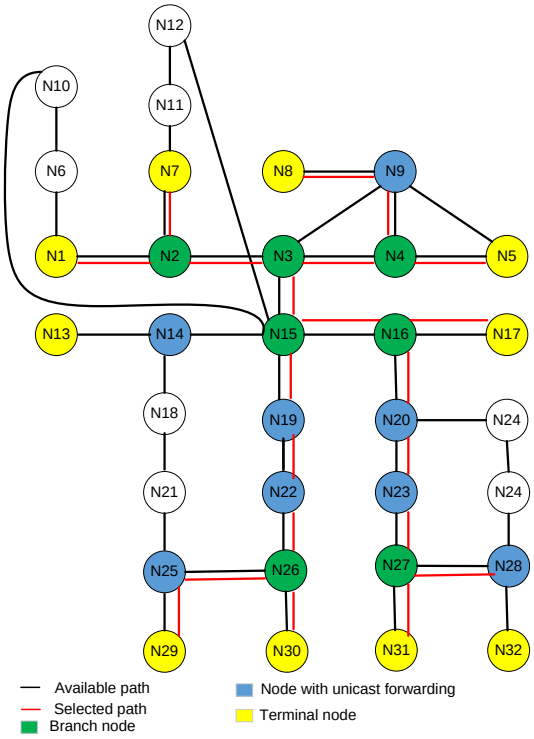
BAERA is tested in real network topologies [76], the Uunet network with 49 nodes and 84 links and the Deltacom network with 113 nodes and 183 links. Additionally, it is tested in a large synthetic network topology with 10,000 nodes [77]. The simulation results show that the multicast tree obtained by BAERA includes fewer edges and branch nodes compared to the SPT and Steiner trees. Moreover, it can generate the multicast tree structure faster. The performance of BAERA needs to be tested in an SDN testbed with a large number of nodes.

5) Path-Vector based Harmony Search (PVHS): Zhou et al. [78] proposed the Degree-dependent Branch-node Weighted Steiner Tree (DBWST) problem to minimize the resource consumption, including the cost of links as well as the number of branch nodes and the number of forwarding entries in the group tables of multicast tree.

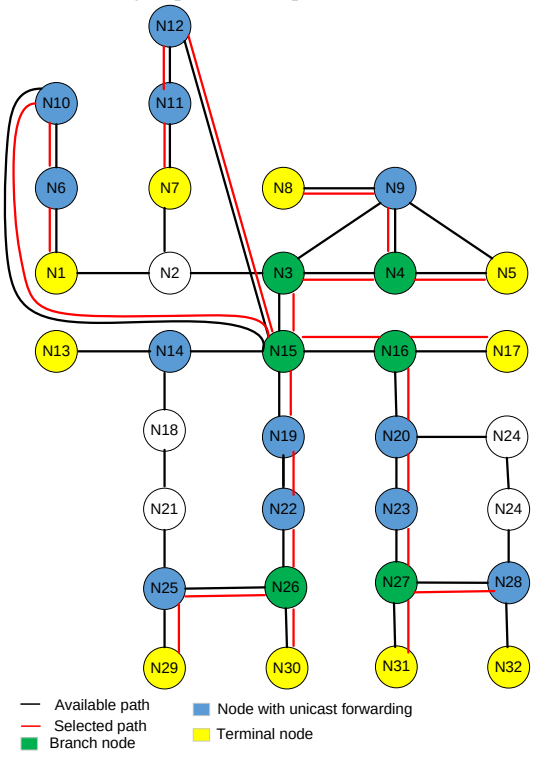
Although BAERA [74] is scalable and computationally efficient, it overlooked the impact of branch degree on the cost of branch nodes. DBWST problem uses the branch forwarding technique and considers the edge cost as well as the degree-dependent branch node cost.

The DBWST problem is NP-hard, because it is equivalent to the Steiner tree problem [31]. To solve this problem, an approximation algorithm, named Path-Vector based Harmony Search (PVHS) algorithm has been designed. The PVHS algorithm consists of three phases: initialization, improvisation and updating. In the initialization phase, a random Depth First Search (DFS) algorithm is used to generate random spanning trees. Figure 13 shows the data distribution spanning tree using PVHS [78]. Then, each tree is converted to its associated path vector representation by traversing the path reversely and storing it in harmony memory. In the improvisation phase, a candidate path vector is created by generating a random sequence of candidate nodes repeatedly until a path is created between the source and the destination. Figure 13 shows the candidate path from the source N1 and the destination N12 via N2-N4-N9-N12. In the updating phase, the total cost of the new improvised tree is calculated. If the cost is lower, then the new tree structure is recorded in the harmony memory.

PVHS is tested in both small topologies (with 40 and 100 nodes) and large topologies (with 500 and 1000 nodes) [79]. The simulation results show that for small networks, PVHS has the advantage of getting more cost efficient and scalable solutions. However, for large topologies, BAERA performs better than PVHS. The performance of the PVHS needs to be improved for a large number of nodes. Additionally, it is required to be tested in an SDN testbed.



(a) Edge optimization phase



(b) Branch optimization phase

Fig. 12: Data distribution tree using BAERA [74]

1553-877X (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2017.2776213, IEEE Communications Surveys & Tutorials

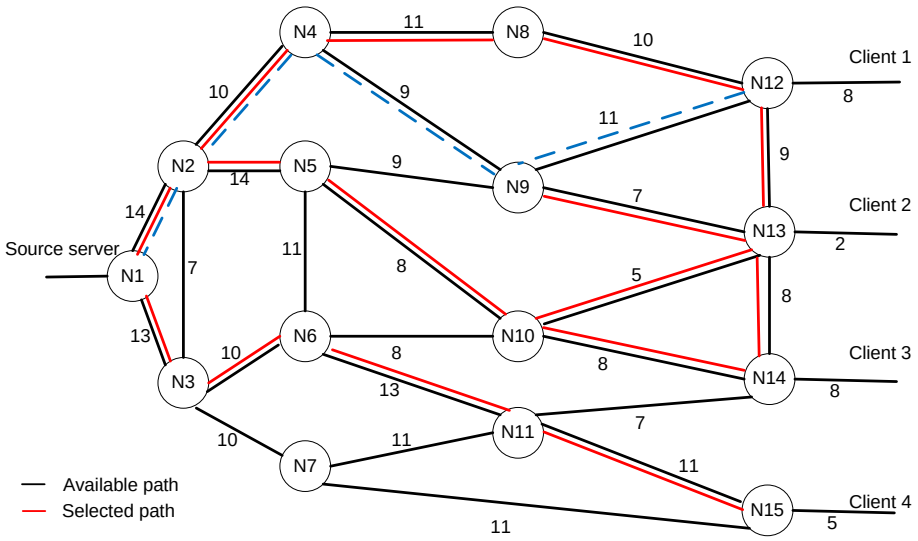


Fig. 13: Data distribution spanning tree using PVHS [80]

B. Application-aware SDN Multicast Routing Algorithms

As mentioned in Section IV-B, in IP multicast, specific non-routing goals can be met using a mechanism that is distributed throughout the region that is being managed. For SDN multicasting, this distributed mechanism can be concentrated in the SDN controller, leading to a much simpler implementation.

For example, in case of SVC enabled video streaming, the controller computed the multi-rate multicast tree for the multicast group and assigned the SVC encoded video according to network conditions and router capability. The global view of the entire network by the controller resulted in efficient and dynamic management of the SVC streaming.

1) Maximum Bandwidth Algorithm (MBA): Tang et al. [81] proposed to use the Maximum Bandwidth Algorithm (MBA) [82] to build the optimal multicast tree for an SDN multicast group, which can be used to deliver as many SVC video layers as possible to all group members according to their individual bandwidth constraint.

Traditional IP multicast routing algorithms rely on local information, which makes it difficult to achieve both efficiency and flexibility for video streaming [83]. MBA uses a centralized approach and SVC to provide a video streaming service that can adapt to the video quality demands of the individual client.

When a new node joins the group, MBA finds a set of candidate paths from a newly joined node to the multicast tree. An optimal path is the path with highest concatenation bandwidth, selected from these candidate paths, by calculating concatenation bandwidth of each candidate path.

MBA is implemented and tested in the topology emulation platform Mininet [62] with 10 switches and 20 hosts. However, the observed performance (e.g., number of packets received) is not compared with any other algorithm. The algorithm assumes that all the video layers utilize the same path to reach different destinations from the source. Allowing multiple paths to carry different video layers could yield more bandwidth for high quality video streaming. However, it would increase jitter at the receiver and require complex session management. Therefore, it is essential to develop a solution that can balance between high quality video and delay.

2) Minimum Path Cost Algorithm with Extra Delay Time k (MPCA-k): Sheu et al. [84] proposed two separate algorithms to build the optimal multicast tree for SVC. The first algorithm is Minimum Path Cost Algorithm with extra delay time k (MPCA-k), which is used to minimize the communication cost from source to clients under the delay-time constraint.

MBA [81] assumes that each client demands homogeneous video streaming quality, which is limited by the residual bandwidth of the edges in the path. However, each client has its own demand of video quality depending on types of video applications. A real-time video application requires bounded end-to-end delay. The video packet that exceeds the time constraint is useless and can be considered lost. MPCA-k minimizes the delay for real-time SVC video streaming in SDN.

MPCA-k sorts the clients by their bandwidth requests in decreasing order. Next, the edges that have residual bandwidths smaller than the first client's bandwidth requirement are removed. Then, the shortest path (based on hop count) is calculated from the source to the first client using the Breadth First Search (BFS) algorithm [61]. If there is more than one shortest path, then the path with Maximum Bottleneck Bandwidth (MBB) is selected. Similarly, the shortest path is calculated from the source to the second client. Then, combine the paths calculated for the first and the second clients to get a new multicast tree. Figure 14 shows the multicast data distribution tree using the MPCA-k [84] algorithm, where `Client1` receives data via path, the N1-N2-N4-N8-N12.

MPCA-k is implemented and tested in a simulator with 200 switches and 10 hosts. The simulation results show that the average hop counts, link utilization and multicast success rate increase when the extra delay time, k increases. Note that this can be adjusted according to the requirement for the video streaming and network condition. The average hop counts from the source to each client are approximately the same

1553-877X (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2017.2776213, IEEE Communications Surveys & Tutorials

in both MPCA-0 and SPT. However, the performance of link utilization and multicast success rate of MPCA-0 is better than SPT. The weakness of the MPCA-k algorithm is the delay time k needs to be adjusted manually. Some arrangements need to be developed to adjust the value of k dynamically.

3) Maximum Bandwidth Utilization Algorithm with Shortest Path (MBUA-SP): The second algorithm proposed by Sheu et al. is Maximum Bandwidth Utilization Algorithm with Shortest Path (MBUA-SP) [84], which is used to maximize the link utilization.

By applying SVC, it is possible to deliver different video qualities to different clients based on their demands. MBUA-SP can be used to find the maximum bottleneck bandwidth or the maximum transmission rate that could be achieved from the source to each client with the minimal delay path.

MBUA-SP first applies the relax operation of Dijkstra's algorithm to find the MBB from the source to all switches in the network. Then, the clients are sorted by MBB of their connected switches in decreasing order. Next, the switch links that have residual bandwidths smaller than the MBB of the first client are removed. Now, the BFS algorithm is applied to find the shortest path from the source to the first client. Similarly, the shortest path is calculated from the source to the second client. Afterword, the paths calculated for the first and the second clients are combined to get a new multicast tree. After establishing a multicast tree, the residual bandwidth in all edges of the network is updated. Figure 15 shows the multicast data distribution tree using MBUA-SP [84], where `Client1` receives data via the path, N1-N2-N4-N8-N12.

The simulation results show that the communication delay of MBUA-SP is slightly higher than the SPT under the same simulation environment mentioned in [84]. However, the multicast success rate and the bandwidth utilization are much higher than the SPT. A drawback of the MBUA-SP algorithm is that it treats the unused bandwidth as an available resource. As a result, a situation may occur where some clients do not receive the basic layers of their requested contents, while the clients that were served earlier receive not only the basic layer but also the extended layers.

4) Layer-based Serving with Layer Switching (LBS-LS): Xue et al. [85] proposed to solve multi-source multi-destination manycast problem [90] to ensure that all the clients get basic video layer in an environment where video sources reside in geographically distributed servers and clients can join and leave the streaming services dynamically.

Traditional IP multicast algorithms do not have the capability of reconfiguring the routing paths dynamically and adaptively. The controller of the SDN architecture can properly select the source video server for each client and adjust the manycast forest (e.g., a group of multicast trees) dynamically when the network status changes.

To solve multi-source multi-destination SVC video manycast problem, an integer linear programming (ILP) model is formulated to perform the theoretical analysis. To reduce computational complexity, two time efficient heuristics: LBS and LBS-LS are proposed. In order to avoid the service unfairness, LBS first attempts to serve the base layers to the group of clients, and then moves to the enhancement layers in sequence. The streaming path is selected in the ascending order of their delays. If multiple paths with the same minimum delay exist, then the path with maximum available bandwidth is selected to reduce the network congestion. LBS-LS is applied when no feasible path for the base layer for a client is available. In this case, the service streaming for the extended layer is switched off to make room for the base layer towards the client.

To measure the performance of LBS and LBS-LS, an experimental testbed with 16 OpenFlow switches is developed. In this testbed, the network links have heterogeneous bandwidth. POX [91] is used for the SDN controller, which is running on a Linux server (Lenovo ThinkServer RD540). The performance of LBS and LBS-LS are compared with traditional IP multicast based on streaming bandwidth, packet-loss-rate (PLR) and delay jitter. The experimental results show that the proposed scheme could allocate bandwidth dynamically according to the network status and hence, improves the quality of the SVC video streaming.

5) Multicast Tree Construction and Packing Algorithm (MTCPA): Zhao et al. [69] proposed an architecture for multi-party video conferencing by utilizing SDN multicast. Additionally, a time-efficient Multicast Tree Construction and Packing Algorithm (MTCPA) is designed to maximize the link utilization under a guaranteed end-to-end delay.

Currently available commercial, multi-party conferencing systems are based on a multipoint control unit (MCU). Each participant sends her video to the MCU, which sends the video to the receivers along separate connections. However, this process may cause large delays due to the detouring to the MCU. Moreover, a single point of failure may occur and the entire communication session might be disrupted. SMVCS proposed to utilize SDN-enabled multicasting where SDN controller helps the MCU to build up multicast trees for video streaming.

First, the link capacity constraint is relaxed to find a delay-bounded multicast tree for each source, which supports the highest multicast success rate. Nonetheless, there may exist some congested links shared by multiple delay-bounded multicast trees with the aggregate link rate violating the link capacity constraint. Next, the congestion problem is solved iteratively by re-routing some multicast paths from the congested link or adjusting the video rates. The SDN controller obtains whole network information: network topology, link bandwidth and link delay. The controller sends this information to the MCU, which calculates the flow route and the transmission rate for each source using the proposed algorithm. The MCU then notifies each source concerning its video rate and informs the SDN controller to set up the flow routes inside each switch.

In this proposed video conference architecture, a single-layer video multicast is deployed, which lacks the support of in-network bit-rate adaptation. To use a layered multicast approach, a mechanism of supporting heterogeneous terminal devices and in-network transmission adaptation is needed. Moreover, the simulation results presented in the paper do not provide any information about the global bandwidth consumption and only focus on small-scale networks (under 150 nodes). Detailed simulation environment related information

1553-877X (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2017.2776213, IEEE Communications Surveys & Tutorials

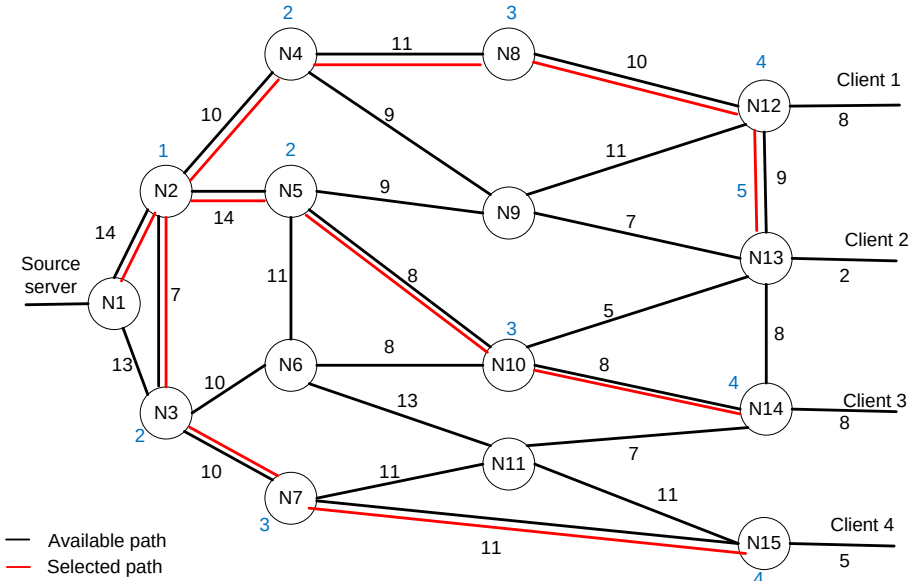


Fig. 14: A multicast data distribution tree using MPCA-k [84]

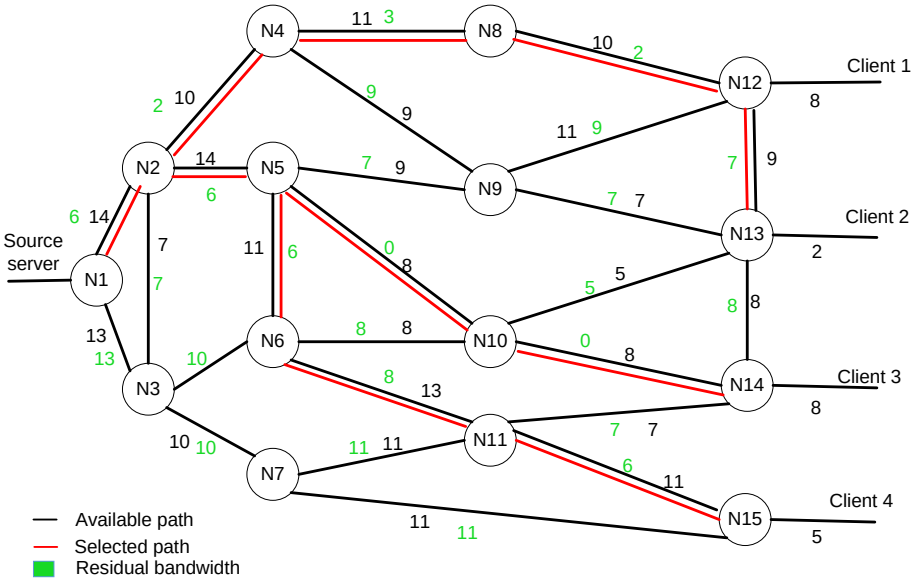


Fig. 15: A multicast data distribution tree using MBUA-SP [84]

is missing.

6) SVC Video Conferencing System (SVCS): Yang et al. [92] proposed SVC enabled Video Conferencing System (SVCS) for multicasting to save the network bandwidth and reduce the delay. Instead of single-layer video multicasting (e.g., Zhao et al. [69]), the SSVCMVCS architecture could provide the base layer and additional enhancement layers dynamically to the participants for a better video quality using SVC.

In SVCS, the conference manager manages the multicasting group and sends the relevant information to the controller for constructing multicast trees for each video layer to implement SVC multicast streaming. Since SVCS does not propose any routing algorithm, it is not included in Tables I and III.

An experimental testbed to measure the performance of SVCS is built consisting of 16 OpenFlow switches (Netgear WNDR3800 routers with OpenWrt [93]), which are connected by network links with homogeneous bandwidth. POX [91] is used as the SDN controller, which is running on a Linux server. The experimental results show that the flexible and controllable SVC multicast stream service of the video conferencing system can reduce the packet-loss ratio and provide better network traffic management. SVCS allows the participants to receive different video layers according to the device capability rather than the network access link properties. Additionally, it allows some degradation of the video stream dynamically in case of network congestion. The evaluation has only been performed on a small-scale grid topology.

7) Multicast Tree Algorithm for Video Conference with Call Adaptation (MTAVC-CA)): Hasrouty et al. [86] proposed SVC enabled video conferencing system using the SDN ar-

1553-877X (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2017.2776213, IEEE Communications Surveys & Tutorials

TABLE I: An overview of SDN multicast routing algorithms

Routing algorithm	Specialization	Integer linear programming (ILP) Model	ILP solution approach	Emulator	OpenFlow controller	Testbed
Castflow [60]	Application-independent routing	No	-	Mininet	NOX	No
Extended Dijkstra's shortest path algorithm (EDSPA) [65]	Application-independent routing	No	-	-	POX	No
Locality-Aware Multicast Approach (LAMA) [68]	Application-independent routing	No	-	Mininet	Ryu	No
Branch aware edge reduction algorithm (BAERA) [74]	Application-independent routing	Yes	Approximation	-	-	Yes
Path-vector based harmony search (PVHS) [80]	Application-independent routing	No	Meta-heuristics	-	-	No
Maximum bandwidth algorithm (MBA) [81]	Video streaming application	No	-	Mininet	NOX	No
Minimum path cost algorithm-k (MPCA-k) [84]	Video streaming application	No	-	-	-	No
Maximum bottleneck bandwidth utilization algorithm with shortst path (MBUA-SP) [84]	Video streaming application	No	-	-	-	No
Layer-based Serving with Layer Switching (LBS-LS) [85]	Video streaming application	Yes	Heuristics	-	POX	Yes
Multicast tree construction and packing algorithm (MTCPA) [69]	Video streaming application	No	-	-	-	No
Multicast tree algorithm for video conference with call adaptation (MTAVC-CA) [86]	Video streaming application	No	-	-	-	No
NFV-enabled multicast topology construction (NFV-MTC) [87]	Network Function Virtualization application	Yes	Approximation & Heuristics	-	-	Yes
Recover aware edge reduction algorithm (RAERA) [88]	Reliability application	Yes	Approximation	-	Floodlight	Yes
Subtree protection and recovery mechanism (SPRM) [89]	Reliability application	No	-	Mininet	POX	No
Avalanche routing protocol (AvRA) [58]	Topology-aware application	No	-	Mininet	OpenDaylight	No

chitecture. This architecture has no conference manager or MCU. Instead the SDN controller is responsible for managing groups and setting up the video conference call. Additionally, this paper presents the Multicast Tree Algorithm for Video Conference with Call Adaptation (MTAVC-CA) to set up a video conference call. MTAVC-CA minimizes the total bandwidth consumption of the network link, but maximizes the bit rates received by each participant.

SVCS [92] solution primarily focuses on reducing the bandwidth consumption on core network links. To solve this problem, MTAVC-CA defines at which node some video layers should be dropped in order to adapt the video streams to the bandwidth capacities of the receiving devices.

In this proposed video conference architecture, the controller collects the downlink bandwidth and computes the receiving and sending bit rates for each participant. If the network bandwidth capacity permits then a multicast tree from each participant to all the others is built. The latencies over all paths between the participants is checked. If it is below the threshold then conference call is dropped. This multicast tree building algorithm is also used in VOIP Conference system [94].

An evaluation of the performance of the MTAVC-CA is implemented using a static simulator in Python. The simulation result compares the performance between MTAVC-CA and a unicast-based video conference system. The results shows that it uses only a fraction of the bandwidth of a unicast-based video conference system (70% at 8 participants and 40% at

1553-877X (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2017.2776213, IEEE Communications Surveys & Tutorials

12 on a 2000 node network) while the average path latency increased by 25% at 12 participants, still remaining under 150 ms. It also increased the call capacity of the network 35% to 49% (depending on the network size, at a given core link bandwidth). The weakness of the video call set up algorithm is that it assumes the downlink bandwidth of the participant is not varied over a session.

8) NFV-enabled Multicast Topology Construction (NFV-MTC): Zhang et al. [87] proposed the NFV-enabled Multicast Topology Construction (NFV-MTC) routing algorithm, which guarantees that each multicast flow traverses through the NFV node before reaching the destination, while minimizing the total topology construction cost.

Network Function Virtualization (NFV) allows efficient implementation of multicast services that involve intermediary processing steps [95]. For example, a video streaming session may require transcoding before it reaches the clients. In this case, transcoding can be implemented as a NFV component placed in the servers or virtual machines (VM). The main challenge for building an NFV-enabled multicast mechanism on SDN is to determine the placement of the NFV node and the path that connects the source and the destination through the NFV node.

The NFV-MTC problem is NP-hard. To find a solution in polynomial time, three approximation algorithms have been designed for three different cases. The first case is the Unchanged Multicast Flow (UMF) bandwidth problem, where the bandwidth usage does not change after traversing through the NFV node. In this case, the approximation algorithm selects a single NFV node, which is used by all the destination clients. Then, the SDN controller, using the complete view of the network topology, constructs an MST among the end users and the selected NFV node. The traffic flow first traverses through the selected NFV node and then it is multicast to each end user using the MST.

The second case is the Reduced Multicast Flow (RMF) bandwidth problem, where the bandwidth of multicast flow decreases after traversing through NFV. This problem can be solved by almost the same approximation algorithm as the UMF. The only difference is that the bandwidth requirement of the flow reduced after traversing through the NFV, resulting in two link weight metrics, θ_{before}, the distance from the source to the NFV node and θ_{after}, the distance from the NFV node to the client.

The third case is the Increased Multicast Flow (IMF) bandwidth problem, where the bandwidth consumption increases after passing through the NFV nodes, which causes higher link cost. This approximation algorithm attempts to find the shortest path between source to NFV node and NFV node to client node for each source and NFV pair.

In real life, end users may dynamically join and leave the existing multicast topology, which causes reconstruction of the MST, which in turn increases the computation cost. To deal with this problem, the authors proposed a dynamic heuristic algorithm. When a new user joins the system, the algorithm tries to find an existing user of the MST, who is closest to the new user. Then, a new path is created from the user in the multicast tree to the new user.

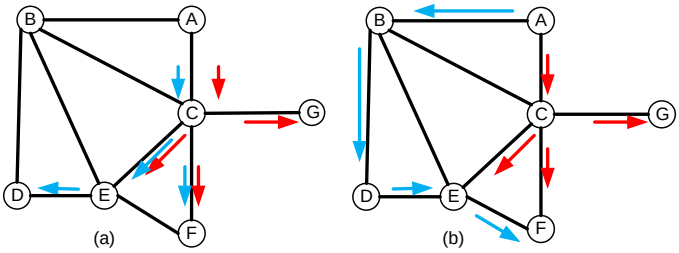


Fig. 16: (a): Two multicast trees (shown in red and blue arrow) carry traffic from source A to their own clients, resulting in double traffic load on path (A, C), (C, F), (C, E). (b) After load balancing, traffic flows through underutilized links.

It is possible that multiple multicast streams may coexist to deliver their own information to different end user groups using the same path. As a result, it may cause network links and routers to be overloaded, and consequently reduce network throughput. Thus, it is important to find a solution to accommodate changing traffic. Figure 16 illustrates the load balancing problem. A heuristic algorithm is proposed to solve the load balancing problem. The algorithm assumes that each multicast stream has a priority level. Now, first, schedule the multicasting stream with highest priority. Then, update the link costs based on their remaining resources. Similarly, schedule the rest of the multicasting streams.

To evaluate the performance of the proposed algorithm, the NFV-enabled multicast mechanism is implemented on ViNO [96] running on the SAVI [97] testbed. The network topology consists of 4 nodes, 12 OpenFlow switches with a set of predefined network link weights. One of the nodes is the NFV node. Experimental results show that the proposed routing algorithm produces near optimal multicast topology. However, the proposed algorithm only considers issues of load balancing specific to multicast. It is essential to consider other network performance parameters such as delay and characteristics of NFV for load balancing.

9) Software-defined Scalable Multimedia Multicast Streaming (SDM^{2}Cast): In SDM^{2}Cast [98], an SDN multicast framework has been proposed, where SVC streaming could be identified and processed by the in-network nodes.

One of the salient features of SVC is in-network bit rate adaptation with respect to the available bandwidth. Since in the current Internet model, routers or switches are oblivious to the SVC streaming application, it is not possible for the network to distinguish SVC streaming data from other IP packets. As a result only end-to-end bit rate adaptation is possible now. To solve this problem SDM^{2}Cast implements network-layer multicast, which is similar to IP multicast. However, this paper does not propose any routing algorithm for SVC streaming. Therefore, it is not included in the Tables I and III.

To measure the support on heterogeneous devices, an experimental testbed has been developed, which consists of OpenFlow switches (Netgear WNDR3800 routers) and a management server has been deployed inside a Linux server with 60 heterogeneous terminals. Joint Scalable Video Model (JSVM) [99] software is used to implement SVC and POX [91] is used as the SDN controller. Experimental results show

1553-877X (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2017.2776213, IEEE Communications Surveys & Tutorials

that SDM^{2}Cast has a lower bandwidth cost compared to IP multicast-based single-layer video distribution. This is due to the customized layered multicast trees of an SVC video, which allows each terminal to receive its desired video layers. A similar experiment shows that the in-network adaptation strategy of SDM^{2}Cast improves the QoS of the video streaming by adjusting the number of layers according to the network condition. SDM^{2}Cast is further discussed in Section VIII-A, where we present different models for participant access control in SDN multicast.

10) Transformation Algorithm (Tran-Alg): Rückert et al. [100] presented, a cross-layer approach where an ISP with more than one client requesting the same real-time video streaming data, builds a data distribution tree through network-layer multicasting. However, instead of IP multicast, Software-Defined Multicast (SDM) is deployed, which assumes that the routers inside the ISP are SDN enabled. Thus, using OpenFlow, multicast group management and distribution tree construction are achieved.

At the border of an ISP, a transformation algorithm (we name it as Tran-Alg) transforms incoming unicast streams to multicast streams, and thus reduces traffic inside the ISP. Such a Tran-Alg has been developed for over-the-top (OTT) services. Audio, video and other multimedia contents that are delivered through the Internet as an alternate to standard telecom services are known as OTT services. Examples include text messaging through Facebook and Whatsapp, video streaming through Netflix, Sling TV and Hulu, etc. While delivering content of OTT services, ISPs work as a dumb pipe that only transports packets. At present the OTT contents are delivered through application layer messaging and through unicast at the network layer. To solve this problem, SDM is proposed, enabling the ISPs to offer network layer multicast support for OTT and overlay-based live streaming as a service.

In this model, an ingress switch receives a single copy of unicast data, which is distributed to the egress switches through the multicast distribution tree. The clients are directly connected to the egress switches, which forward data to the clients using unicast. The ingress and egress switches convert unicast data to multicast data and again change it back to unicast data. This whole process is accomplished through SDN switches. In this paper, the routing is implemented with static routes and no new routing algorithm has been proposed. Thus, it is not included in the Tables I and III.

To evaluate the performance, the SDM model is implemented and tested in the network topology emulation platform Mininet [62]. Ryu [72] is used as the SDN controller. The host system is a XEN virtual machine (VM) [101]. Experiment results show that overlay network with SDM integration reduces the network traffic by 30% compared to the network without SDM. This SDM model has been further extended in Dynamic Software-Defined Multicast (DYNSDM) [102], which introduces the network-layer multiple distribution trees and dynamic multicast groups with fast response in link failure.

11) Recover Aware Edge Reduction Algorithm (RAERA): Shen et al. [88] proposed a new reliable multicast tree algorithm for SDN, named Recover-aware Steiner tree (RST). The goal of RST is to minimize the total cost of all edges in the tree and to facilitate the local loss recovery by selecting the recovery nodes that are close to the destinations.

Nowadays, many multimedia applications rely on reliable transmissions such as YouTube, Microsoft Smooth Streaming, and MPEG DASH [103]. Floyd et al. [48] proposed to recover the lost packets from the source. However, the source suffers from the scalability problem due to the extra burden to provide loss recovery for a large number of destinations. Whetten et al. [51] proposed to place the recovery nodes between the source and the destinations to facilitate local loss recovery. RST proposed to span suitable recovery nodes for local data loss recovery and minimize the recovery load and the corresponding traffic for a multicast tree in an SDN network.

Note that the RST problem is NP-hard. To reduce computational complexity, an approximate algorithm called Recover Aware Edge Reduction Algorithm (RAERA) is proposed in [88]. RAERA includes two phases: tree routing and recovery selection. In the tree routing phase, the path from the source to destination is reduced iteratively by re-routing the path on the solution tree $T(V^T, E^T)$. The re-routing path needs to include at least one recovery node to facilitate local loss recovery. The recovery selection phase selects an optimal recovery node for minimizing the recovery cost, using a dynamic programming algorithm.

The performance of RAERA is evaluated by simulation in a real network topology, Biznet [104] with 29 nodes, 33 network links using Estinet network simulator [105] and in a synthetic network with 4000 to 10000 nodes generated by Inet [106]. The evaluation results show that RAERA provides lower overall cost, less re-transmission and lower latency. Additionally, the performance of RAERA is tested in a real world environment using an experimental testbed with 14 nodes and 20 network links with heterogeneous bandwidth. Floodlight [107] is used as the SDN controller. The recovery nodes are implemented inside HP ProLiant servers on the top of Click Software Router [108]. The experimental results show that bandwidth consumption of the Steiner tree and the SPT are 24% and 35% more than RAERA. During video playback, RAERA requires less than 1 second for video re-buffering while Steiner tree and SPT require 33.5 and 7.8 seconds, respectively. Therefore, RAERA shows better network performance and can improve user experience during recovery. For further evaluation, RAERA needs to be deployed in a network where the links' bandwidths change dynamically. It also needs to be tested in a testbed with a large number of nodes.

12) Subtree Protection and Recovery Mechanism (SPRM): Based on the subtree protection and recovery mechanism, Raja et al. [89] proposed an algorithm to increase the resiliency of multicast in SDN [109]. In this paper, we name this algorithm as Subtree Protection and Recovery Mechanism (SPRM).

In traditional networks, using only the routing protocols, it is difficult to distinguish between a link and a node failure [110]. If a node failure occurs, routers have to identify if all links of a particular node have also failed. It is a time consuming process, which results in high delay, packet losses and degradation of the QoS of delay-sensitive applications

1553-877X (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2017.2776213, IEEE Communications Surveys & Tutorials

such as video streaming and video conferencing. The SPRM algorithm proposed to combine programmable control plane and the global view of the network topology in SDN. These two features are used to detect link or node failures of a multicast tree and to determine which part of the multicast tree must be updated in the flow table to recover from this failure.

The proposed algorithm divides the multicast tree into subtrees and installs backup paths for these subtrees instead of an entire redundant backup tree. It computes multiple disjoint paths for each leaf of the subtree. In case of a link or a node failure, the controller searches for the subtree it needs to recover and responds to the failure by installing or modifying flow entries for fast recovery.

The proposed algorithm is implemented and tested on the topology emulation platform Mininet [62] with 9 nodes. The simulation results show that the average failure localization time is mostly less than 40 μs. The total recovery time, which consists of the failure localization time and the time taken by the controller to respond to the failure, is on average 0.5 ms. However, the evaluation results ignored the failure detection time and assumed zero latency between the controller and the switches. The POX [91] SDN controller, used for simulation, is slow to respond to failure because it periodically checks OpenFlow switches. The Bidirectional Forwarding Detection (BFD) protocol [111], which detects failures by detecting packet loss in frequent streams of control messages, could be used so that an event can be triggered to reduce failure detection time.

C. Topology-aware SDN Multicast Routing Algorithms

The interest of IP multicat has renewed due to the increased deployment of data centers around the world. Modern data centers may exploit multicast group communication to provide different services: publish-subscribe services for data dissemination, web cache updates, system monitoring, etc. [112]. The importance of data centers is increasing rapidly due to the supporting and sustaining Internet-based applications including search (e.g., Google, Yahoo, Bing), video content hosting and distribution (e.g., YouTube, NetFlix, Amazon), social networking (e.g., Facebook, Twitter, Tumblr), and large-scale computations (e.g., data mining).

Data center networks deploy *complete graph* like connectivity between nodes, which creates multiple equal-cost paths between two hosts. Traditional IP multicast could not utilize this network topology property to optimize the performance. Instead it suffers from poor scalability, which is due to forwarding state explosion in switches/routers. It is assumed that multicast state capacity of a commodity switch is 1000. Another limitation of IP multicast is formation of the MDDT centered at a single root (i.e., either at the RP or at the source). A single-rooted tree fails to leverage the unique topological property of data center architecture, which is more symmetric and have many redundant paths between two hosts. Due to the centralized architecture, data center networks are suitable to exploit centralized traffic engineering using SDN. The advantage of SDN in a data center is that it can dynamically deliver services based on demand and availability of resources within the data center.

1) Avalanche Routing Algorithm (AvRA): Iyer et al. [58] developed the Avalanche Routing Algorithm (AvRA) for data center networks. AvRA has been designed to utilize the topological properties of data center networks.

The typical data center network architecture usually consists of switches and routers in a two or three level hierarchy, with high link density. There are many equal cost paths between a pair of servers or switches. In this scenario, multicast trees formed by traditional multicast routing can result in severe link waste. AvRA attempts to minimize the size of the routing tree created for each multicast group. It creates a spanning tree by utilizing the rich path diversity that is commonly available in data center topologies, such as Tree and FatTree [113], which ensures higher bandwidth utilization.

Instead of trying to find the shortest path from a group member to the source node of the multicast group, AvRA attempts to find the shortest path to the existing multicast tree. Figure 17 shows that, initially, the shortest path exists between the source server and the client, `R1`. Next, another client, `R2` joins the multicast group with node `I` at level 1. However, node `I` is not part of the multicast tree. The AvRA algorithm searches in all adjacent nodes and neighbour nodes of node `I` to check if any of them is on the multicast tree. Node `I` has an adjacent node `E` at level 2, which has neighboring node `D`. Node `D` is adjacent to node `A` at level 3, which is on the multicast tree. Therefore, the algorithm attaches the client `R2` to the existing multicast tree at node `A`.

AvRA is implemented and tested on the FatTree topology [113] using the topology emulation platform Mininet [62] with 24 hosts, 6 Top-of-Rack (ToR) switches, 6 aggregation switches, and 2 core switches. The simulation results show that compared to IP multicast, AvRA improves application data rate by up to 12%, and lowers packet loss by 51% on an average. Moreover, this algorithm reduces the number of links that are less than 5% utilized, by at least 35%, compared to IP multicast.

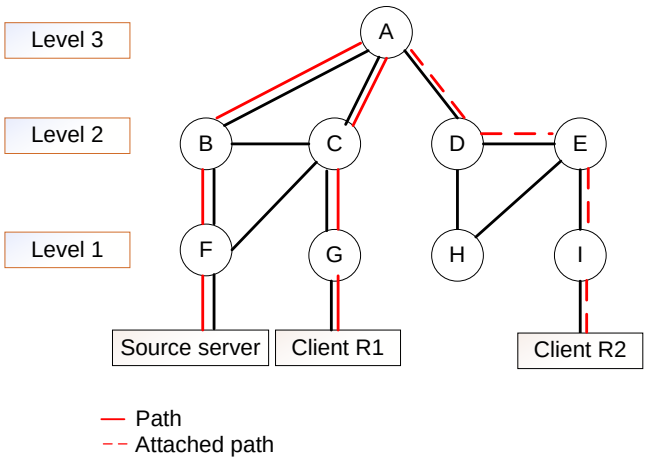


Fig. 17: A multicast data distribution tree using AvRA [58]

2) Scale-out IP Multicast (SoIM) : Li et al. [112] presented an SDN multicast architecture that solves the scalability issue

1553-877X (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2017.2776213, IEEE Communications Surveys & Tutorials

by developing multi-rooted data distribution trees. In this paper we refer to this model as the Scale-out IP multicast (SoIM) model. This model proposed to partition and distribute the multicast address space across cooperating switches in a data center network.

In IP multicast addresses are randomly assigned to groups and thus address aggregation is not possible at the switches. However, SoIM introduced a local multicast address aggregation mechanism in switches to aggregate local groups for forwarding packets and increasing the number of supported groups. Since SoIM does not propose any routing algorithm, it is not included in Tables I and III.

SoIM is implemented and tested on the 3-tiered AB FatTree topology [114] with 48-port switches, which connects 27,648 physical end hosts. The simulation results show that, without local aggregation, the datacenter can support up to about 30K multicast groups. Applying local aggregation can support 100K multicast groups. The traffic overhead due to local aggregation is only about 0.2%. If there are 1000 dynamic events (join and leave) per second in the system then on average in every second, each edge, aggregation and core switches have 42, 72 and 78 updates, respectively where tenant's VM are distributed across the network randomly. Reconstructing the multicast shared tree can be an expensive process, which may significantly impact network performance in case of network failures in datacenters. However, generally the failures in datacenter networks are short term, so immediately reconstructing the multicast shared tree can be inefficient and unnecessary. This paper proposes a fault-tolerant routing solution that consists of two parts: (1) short-term local rerouting for fast failover where switches locally reroute the packets immediately after a failure and (2) long-term multicast tree reconstruction.

3) Scalar-pair and Vectors Routing and Forwarding (SVRF): To minimize the multicast forwarding space and reduce the workload of the routers, Scalar-pair and Vectors Routing and Forwarding (SVRF) technique has been proposed in [115]. SVRF does not propose any routing algorithm but designs the digital logic and hardware of an SDN switch to reduce forwarding state space and forwarding decision time. Hence, it is not mentioned in Tables I and III.

The disadvantage of traditional IP multicast protocols is that they require excessive system resources, such as memory space and processor speed. SVRF proposed to solve the scalability problem in terms of multicast forwarding states using a novel unified unicast and multicast forwarding algorithm based on the prime number and Chinese remainder theorem.

SVRF is implemented in SDN switches and controlled by the SDN controller. The SDN controller updates an SDN switch about its scalar-pair, which is the encoded version of all unicast and multicast group addresses allowed to traverse through this switch. Hence, scalar-pair is the entire forwarding information (also named as vector) for an SDN switch. On receiving a unicast/multicast packet, the necessary forwarding information is extracted from the packet and is mapped to a unique input key. The switch takes decision on forwarding (i.e., sent to a specific port or dropped) through arithmetic calculation that uses the input key and the previously received scalar-vector.

Bloom Filter [116] is another efficient algorithm to compress the multicast forwarding states. However, significant traffic leakage may occur when group membership querying is false positive. Compared to Bloom filters, the simulation results shows that SVRF offers better performance in scalability of number of flows. The total hardware costs of BF are much greater than proposed scheme's in any conditions, especially for achieving low false positive probabilities. The only drawback of the proposed scheme is that it becomes less time efficient in ultra-scalable routing/forwarding engines because of the characteristics of prime numbers [117].

4) Dual-structure Multicast (DuSM): It is observed that multicast traffic in the data center networks shows strong group heterogeneity. Based on this observation, Cui et al. [118] presented Dual-structure Multicast (DuSM) system to balance the network traffic among the shared multicast trees.

Traditional IP multicast protocols could not utilize the topology of the data center network. For example, Protocol Independent Multicast Sparse Mode (PIM-SM) [21] chooses a single core switch on a Fat Tree topology [113] as the rendezvous point (RP). When all the multicast groups in the data center network are using the same core switch, network congestion could occur due to the limited state capacity of the switch. DuSM proposed to balance the traffic to reduce congestion.

The DuSM categorized the multicast traffic into two groups: *elephant groups* and *mice groups*. Elephant groups consist of that small fraction of the groups that are contributing the majority of the group traffic. Mice groups have a very low traffic volume. The system installs multicast rules on the switches to balance the traffic among multicast trees to avoid congestion for the elephant groups. For mice groups, the system applies state-free multicast such as multicast-to-unicast translation to save state space in exchange for bandwidth capacity. All new groups are treated as *mice*. When the traffic of a group exceeds a predefined threshold, it is promoted to *elephant group* and new rules will be installed on corresponding switches. DuSM does not propose any routing algorithm, so it is not included in Tables I and III.

The DuSM system is implemented on the FatTree topology [113] using ns-2 simulator [119]. In order to evaluate the performance of the DuSM and PIM-SM, three performance metrics are considered: multicast state capacity, network traffic distribution and number of switch updates. For the multicast state capacity, PIM-SM can support around 15K group whereas using the 10 KB and 100 KB thresholds, DuSM can support around 32K (100% more) and 64K (300% more) groups respectively. Compared to PIM, DuSM can achieve better traffic balance by assigning packets to multiple shared trees and reduce the occurrence of network congestion. The number of switch updates depends on join and leave events. DuSM can save around 90% of switch updates when the threshold is set to 10K. Simulation results show that DuSM can support many more groups using commodity switches and achieves better load balance, compared to IP multicast. Nonetheless, dynamic group classification between elephant group and mice group is not explored in this paper.

1553-877X (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2017.2776213, IEEE Communications Surveys & Tutorials

TABLE II: Multicast Routing Issues

Multicast Routing Issues	IP Multicast	SDN Multicast
Scalability	Thaler and Handley [27], Fei et al. [32]	LAMA [68], BAERA [74], PVHS [80]
Reliability	Floyd et al. [48], Speakman et al. [49]	RAERA [88], SPRM [89]
Quality of service	Taubman et al. [54], Bolot et al. [55], Tyan et al. [56], Schwarz et al. [57]	MBA [81], MPCA-k [84], MBUA-SP [84], LBS-LS) [85], MTCPA [69], MTAVC-CA [86]

VI. Discussion and Open Problems in Multicast Routing

Although the rise of multicasting looks promising, its widespread use is still limited due to the traditional architecture of the Internet, which is designed to provide best-effort delivery for one-to-one communication. Multicasting involves one-to-many communication and may require minimal delay, low resource consumption, high delivery reliability, or high availability of resources in the path, depending on the application.

Section VI-A provides a brief discussion about the SDN multicast routing algorithms that overcome the limitations of IP multicast routing protocols. Section VI-B outlines the open problems and challenges of SDN multicast routing algorithms.

A. Discussion

The multicast routing issues that hinder large-scale deployment are: balancing minimum delay against other issues, scalability, reliability, and quality of service. Traditional IP multicast protocols are optimized primarily for minimum delay. The alternate routing "issues" represent different optimizations (see Section III-C) that can be important to the application. Table II shows different approaches to address these three issues in cases of IP multicast and SDN multicast. Multicast routing protocols and algorithms can also be differentiated from the perspectives of the optimization goal and the imposed constraints. Table III shows the various characteristics of the protocols and algorithms that have been discussed in this paper.

As noted in Section III-C, given the constraint on traditional IP multicast routing (i.e., that a router use only locally-available information), the proposed solutions for achieving these optimizations are implemented using a mechanism that is distributed throughout the network, and whose components may or may not be co-located with the routers. For SDN multicasting, the constraint can be removed; the SDN controller can act on information from the entire controlled region, which may result in superior solutions. Table I shows an overview of the different SDN multicast routing algorithms. Figure 9 shows the taxonomy of SDN multicast routing algorithms based on the applications.

1) Scalability of MDDT: As noted in Section III-E, traditional IP multicast protocols, especially those that are designed to produce minimum delay, suffer from scalability issues. To solve the scalability problem of multicast, it is crucial to reduce the number of nodes and edges of the shared MDDT. The original approaches attempted to balance between the SBT and CBT architectures. Thaler and Handley [27] and Fei et al. [32] have proposed gentle and aggressive approaches to achieving scalability, using only local adjacency information.

For a graph $G(V, E, B, \Delta)$, V represents different types of nodes such as the source server, the destination server, the forwarding server, E represents the edges, B represents the bandwidth capacity, Δ represents both transmission and processing delay. In the literature, the problem of finding the minimum weighted tree that spans the set V is called the Steiner tree problem. Finding a solution to the Steiner tree problem [31] is a promising way to solve the scalability problem. SDN multicast routing algorithms could solve Steiner tree problem faster and efficiently by utilizing the centralized approach and global view of the network topology.

Lin et al. [68] proposed Locality-Aware Multicast Approach (LAMA) to minimize the computation time of the multicast distribution tree and the number of flow entries in the switches to enhance the scalability of the controller and the switches. Huang et al. [74] proposed Branch-aware Steiner tree (BST) to solve the scalability problem. Zhou et al. [80] proposed the Degree-dependent Branch-node Weighted Steiner tree (DB-WST) to solve the scalability problem by minimizing the resource consumption, including the cost of links as well as the number of branch nodes and the number of forwarding entries in the group tables on the multicast tree.

2) Reliability: In IP multicast, Floyd et al. have proposed Scalable Reliable Multicast (SRM) [48], which uses a centralized approach to achieve greater reliability. However, it has limited scalability. Speakman et al. have proposed Pragmatic General Multicast (PGM) [49], which uses a distributed recovery mechanism, and shows superior scalability. In SDN multicast, Shen et al. [88] proposed Recover-aware Steiner tree (RST) to minimize total cost of all edges in the tree and to facilitate local loss recovery by selecting the recovery nodes close to the destinations, along with an approximation algorithm (RAERA) to calculate the routes efficiently.

Raja et al. [89] proposed SPRM to increase the resiliency of SDN multicast based on the subtree protection and recovery mechanism.

3) Quality of service: In IP multicast, methods have been proposed where the source varies the effective rate [54], [55], or the routing protocol takes certain characteristics of the receivers into account [56]. In addition, the source can send multiple non-overlapping streams using SVC, and an individual receiver can subscribe to the base stream and zero or more enhancement streams [57]. In SDN multicasting, Tang et al. [81] have proposed to build the optimal multicast tree, which can deliver SVC streaming layers all multicast group members according to their individual bandwidth constraint. Sheu et al. [84] have proposed two separate algorithms, Minimum Path Cost Algorithm with extra delay time k (MPCA-k) and Maximum Bandwidth Utilization Algorithm with Shortest Path (MBUA-SP) to build the optimal multicast tree for SVC.

Zhao et al. [69] proposed a time-efficient multicast tree construction and packing algorithm (MTCPA) for SDN-enabled Multi-party Video Conferencing to maximize the link utiliza-

1553-877X (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2017.2776213, IEEE Communications Surveys & Tutorials

tion under a guaranteed end-to-end delay. Hasrouty et al. [86] proposed Multicast tree algorithm for video conference with call adaptation (MTAVC-CA) to construct multicast tree for SVC Multicast Video Conference system, which minimizes the total bandwidth consumption of the network link, while maximizing the bit rates received by each participant.

B. Open Problems in SDN Multicast Routing

After reviewing the current proposals of SDN multicast routing, we list the following open problems:

1) Multicast tree construction: An open challenge of SDN multicast routing is to balance two conflicting multicast tree construction mechanism between Shortest-Path Tree (SPT) and Minimum Spanning Tree (MST). SPT is used to minimize the length of the paths (usually in terms of hops) to the individual destinations. It minimizes end-to-end delay but it causes high consumption of network resources. It is suitable for applications with interactive data dissemination where end-to-end delay minimization is crucial. On the other hand, MST is used to minimize the sum of link cost to all the destinations. As a result, the MST optimizes resource consumption of the network but the pair-wise paths may not be optimal and can cause large end-to-end delays. It is suitable for applications with non-interactive data dissemination where end-to-end delays are not an issue.

2) Multicast tree size: Another open challenge of SDN multicast routing is to minimize the size of the multicast tree to enhance the network performance. After multicast tree construction, refinement is done to improve the quality of the tree structure. However, it is an expensive operation. Thus it is essential to investigate an efficient method to perform the tree refining procedure.

3) Quality of service (QoS): It is extremely challenging to maintain QoS and optimize resource utilization using SDN multicast routing simultaneously. A trade-off can be made between QoS and resource utilization based on monetary cost, since guaranteeing a higher level of QoS will cause a large consumption of resources, which has to be paid for.

VII. Secured Communication in IP Multicast

Three areas have been identified that need to be addressed to secure IP multicast communications [121]. These three areas are (i) Protecting routing protocol messages, (ii) Protecting multicast data and (iii) Participant access control. In the following sections we discuss the existing solutions in IP multicast of these three areas including the security services provided by these solutions.

A. Secured Routing Protocol

As we have mentioned earlier, IP multicast basically depends on two routing protocols: hosts use IGMP to join or leave a multicast group while the routers use one of the multicast routing protocols to maintain or update their own forwarding tables. Since PIM-SM is the only multicast routing protocol that is widely deployed and supported by the Internet Service Providers (ISP), the discussion on routing protocol security is limited to the security of the PIM-SM protocol.

1) Securing PIM-SM Messages: There are two types of PIM-SM control messages: unicast messages (i.e., `Register` and `Register-Stop`) and link-local multicast messages (i.e., `Hello`, `Join/Prune`, `Assert`).

To provide data origin authentication and data integrity protection for the unicast messages, an IP security (IPsec) [122] Security Association (SA) needs to be established between the RP and the DRs of a PIM domain [21]. An SA provides security services to the traffic at the IP layer. IPsec must implement one of the protocols—Authentication Header (AH) or Encapsulating Security Payload (ESP)—to offer security services. In order to simplify the management problem, the PIM-SM specification suggests the use of the same authentication algorithm and authentication parameters for Register messages regardless of the sending DR or the destination RP. Although a unique SA (which should be identified by a unique Security Parameter Index (SPI) [122]) is needed for each DR, the same authentication algorithm and authentication parameters (secret key) can be shared by all DRs and by all RPs. Similarly, a unique SA per DR with the same authentication algorithm and authentication parameters is recommended for Register-Stop messages.

Authentication and confidentiality in PIM-SM link-local messages are addressed separately in [123]. To authenticate the link-local messages, IPsec with ESP has been specified. An optional ESP-based mechanism is also suggested for providing confidentiality. Manual keying is specified as the mandatory and default group key management solution, while an optional support for automated group key management mechanism is provided.

2) Securing IGMP Messages: To protect IGMP messages, the IGMPv3 specification [18] suggests the use of an IPsec SA with the AH protocol. Use of an SA with AH provides data origin authentication and integrity. In the absence of automated key management, a single symmetric key should be used for the whole LAN. This will only ensure that an IGMPv3 message has arrived from a system on the LAN with the proper secret key. The use of an IPse SA with automated key management and an asymmetric signature algorithm would provide better security.

Li and Atwood [124] have proposed a solution for securing IGMP messages, which is called Secure IGMP (SIGMP). In contrast to the solution suggested in the IGMPv3 specification, the proposed solution differentiates the security parameters (keys, SAs) on a per-group basis, and is independent of the version of IGMP (or MLD) in use. The security parameters used by SIGMP are negotiated among the group members on the network segment and the access router(s) on that segment, using a new automated key-management protocol called GSAM.

B. Protecting Multicast Data

For secured data distribution over multicast groups, the IETF has designed a Group Security Architecture, which is independent of any multicast routing or admission control protocols [34]. The reference framework shown in Figure 18 organizes the elements of the architecture into three functional areas pertaining to security. Here, the sender and the

1553-877X (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2017.2776213, IEEE Communications Surveys & Tutorials

TABLE III: Multicast Routing Goals and Constraints

Name	Membership	Upward Degree	Category	Optimization	Constraint	
					Local Info	Other
MOSPF [43]	Dense	1	IP multicast routing	Minimum delay	Yes	
DVMRP [44]	Dense	1	IP multicast routing	Minimum delay	Yes	
PIM-DM [120]	Dense	1	IP multicast routing	Minimum delay	Yes	
PIM-SM [21]	Sparse	1	IP multicast routing	Routing table size(1)	Yes	
PIM-SSM [45]	Sparse	1	IP multicast routing	Minimum delay	Yes	
BIDIR-PIM [47]	Sparse	1	IP multicast routing	Minimum delay	Yes	
CBT [26]	Sparse	1	IP multicast routing	Minimum delay	Yes	
SRM [48]	Sparse	1	IP reliable multicast	Recovery of lost packets	Yes(2)	Recovery from source node
RMTP II [51]	Sparse	1	IP reliable multicast	Recovery of lost packets	Yes(2)	Recovery from intermediate nodes
PGM [49]	Sparse	1	IP reliable multicast	Recovery of lost packets	Yes(2)	Recovery from intermediate nodes
CastFlow [60]	Sparse	1	SDN multicast routing	Minimum response time for join and leave event	No	
Extended Dijkstra [65]	Sparse	1	SDN multicast routing	Minimum delay	No	
LAMA [68]	Sparse	1	SDN multicast routing	Minimum distribution tree computation time, minimum # flow entries	No	
BAERA [74]	Sparse	1	SDN multicast routing	Minimum # edges and # branch nodes	No	
PVHS [80]	Sparse	1	SDN multicast routing	Minimum resource consumption	No	
MBA [81]	Sparse	1	SDN video distribution	Deliver as many layers as possible	No	Individual bandwidth available
MPCA-k [84]	Sparse	1	SDN video distribution	Minimum communication cost	No	Delay-time limit
MBUA-SP [84]	Sparse	1	SDN video distribution	Maximize link utilization	No	
LBS-LS [85]	Sparse	1	SDN video distribution	Ensure all clients get basic layer	No	
MTCPA [69]	Sparse	1	SDN video distribution	Maximize the link utilization	No	
MTAVC-CA [86]	Sparse	1	SDN video distribution	Minimize bandwidth consumption	No	
NFV-MTC [87]	Sparse	1	SDN NFV multicast	Minimum topological cost	No	Guarantee flow through NFV node
Zhang et al. [87]	Sparse	1	SDN load balancing	Load balancing (avoidance of overload)	No	
RAERA [88]	Sparse	1	SDN reliable multicast	Fast local recovery of lost packets	No	
SPRM [89]	Sparse	1	SDN reliable multicast	Subtree protection and recovery mechanism	No	
AvRA [58]	Sparse	> 1	SDN data centers	Minimum size of the routing tree	No	

(1) can switch to minimum delay
(2) Uses only local information for the routing, but needs more global information for the proposed optimization

receivers will interact with the Group Controller and Key Server (GCKS) for authentication, authorization, and key management purposes. Key management includes obtaining the new keys and the key updating materials.

The reference framework shown in Figure 18 makes use of an IPsec Group Security Association (GSA), an SA established among the members of a multicast group. IPsec was primarily designed to provide security services at the IP layer for unicast packets. The necessary extensions for applying IPsec security services to IP multicast packets are specified in Weis et al. [35]. Through the multicast extension to IPsec, several security services become available, such as confidentiality, group source authentication, group integrity protection, sender data origin authentication, anti-replay protection, etc.

The reference framework and GSA described above do not assume any particular group key management protocol. The basic idea of group key management is that whenever a single user joins or leaves, the entire set of existing receivers must be re-keyed. The challenge here is to reduce the number of receivers that have to be re-keyed. In [125], using the concept of proxy encryption, an efficient group key management protocol has been presented.

C. Participant Access Control

Participant access control is defined as authentication, authorization and accounting (AAA) functionalities for both

1553-877X (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2017.2776213, IEEE Communications Surveys & Tutorials

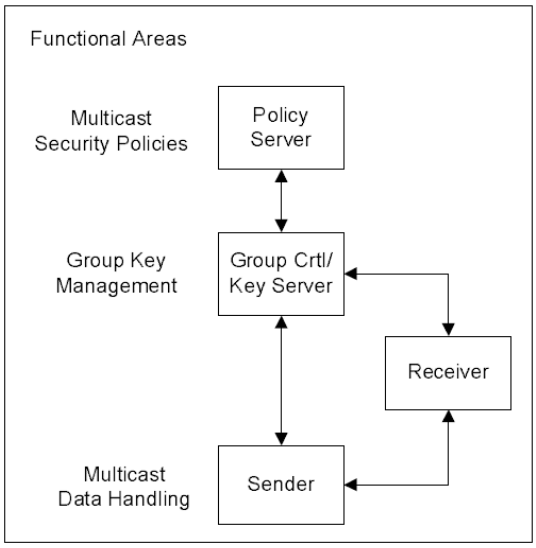


Fig. 18: Centralized Multicast Security Reference Framework [34]

sender(s) and receivers of a multicast group [126]. Figure 19 shows the extensions needed to provide sender and receiver access control by incorporating the AAA framework into the existing multicast model.

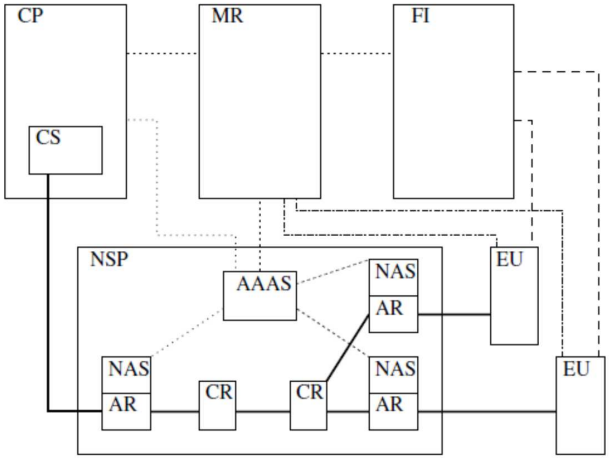


Fig. 19: Participant Access Control Architecture [127]

The framework shown in Figure 19 has different entities: End User (EU), Content Provider (CP), Merchant (MR), Network Service Provider (NSP), Access Router (AR), Core Router (CR), Network Access Server (NAS) and AAA Server (AAAS). The MR and the AAAS are allowed to update the participants' database. The NAS or the AAA client will reside inside the ARs. The CP or the sender will deliver service or data packets through the Content Server (CS). The one-hop AR of the sender will authenticate and authorize the sender by communicating with the AAAS. Similarly, the one-hop AR of the EUs will authenticate and authorize the end users. The ARs keep accounting information for a session. At the end of a session this information will be forwarded to the AAAS.

Access control is performed in two layers: the application layer provides the admission decision while the network layer enforces that decision. Hence, there must be correlation so that the decision taken at the application layer is properly mapped to the enforcement taken at the network layer. There are two possible approaches to map the relationship between the two layers:

- Both application and network layer credentials are coupled and thus are carried together.
- Credentials of these two layers are carried separately and are correlated.

1) IGMP-AC: Islam and Atwood [36] have proposed a participant access control solution called IGMP-AC, where the credentials are coupled. IGMP-AC is an extension to IGMP, designed to incorporate the IETF AAA functionalities into the IGMP exchanges. It enforces authentication and authorization of an end user or receiver before joining or leaving a secured multicast group. IGMP-AC provides a mechanism within the IGMP exchange to carry a generic client-server authentication protocol. Any suitable authentication protocol (e.g., Extensible Authentication Protocol(EAP) [128]) having client-server entities can be encapsulated within the IGMP-AC packets. IGMP-AC will not disrupt the usual function of IGMPv3 (to be used for classical multicast groups), and the access control mechanism of IGMP-AC will be used to join/leave a secured or restricted multicast group only.

This approach has the advantage that only one interaction is necessary. Its disadvantage is that the IGMP packet format is different from the IGMP packet format, so that, although (secure) IGMP-AC interactions will not interfere with (insecure) IGMP interactions, all hosts on the network segment must understand IGMP-AC.

2) PANA and SIGMP: Islam and Atwood [38] have proposed an alternate participant access control solution, where the credentials are separated. The application-layer credentials are carried by Protocol for Carrying Authentication for Network Access (PANA). The network-layer request is carried by Secure IGMP (see Section VII-A2). The derived key from the application-layer interaction is used as a credential when GSAM negotiates the security parameters for SIGMP.

This approach has two advantages:

1) The operation of SIGMP and GASM are completely independent of the method used to create the derived key, and therefore of the business model used at the application layer.
2) The packet formats of IGMP and SIGMP are identical. The only change is in the semantics of the interactions when the associated group is a secure group.

3) Sender Access Control (SAC): In [37], a Sender Access Control (SAC) architecture has been presented that enforces that only an authenticated and authorized sender will be permitted to send data to a secured group. Depending on the underlying business model, accounting may also be required. In the SAC model, authentication of a sender is optional, so that the architecture also supports open groups that operate according to the classical IP multicast model. The SAC model incorporates AAA protocols and depends on EAP (carried by PANA) to authenticate a sender.

1553-877X (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2017.2776213, IEEE Communications Surveys & Tutorials

VIII. Secured Communication in SDN Multicast

Although we identified three different areas of secure multicasting, researchers so far have mostly focused on participant access control in SDN multicast models. This section explains briefly those efforts and also presents a comparison among those methods.

A. Participant Access Control

1) Kiciński Model: A three layer SDN multicast architecture has been proposed by Kiciński et al. [129], where, in addition to the control and forwarding layer, an application layer has been added. The application layer lies on top of the control layer and performs a number of multicast group control functions including AAA, signalling, group management, access control, etc. Figure 20 shows the architecture with message flows for the Kiciński model. In this architecture, a client sends its request to join a group to the Upper Layer Multicast Application (ULMA). On successful authentication and authorization, the ULMA notifies the controller that a new node has to be added (in case of a new client join) or the topology has to be readjusted (in case of the mobility of an existing client). The controller is responsible for implementing topology discovery, maintaining multicast tree, mobility, etc. In both cases the multicast tree needs to be readjusted. The controller updates the forwarding states of the routers and switches using the OpenFlow [11] protocol. In the Kiciński model, the ULMA maintains a list of authorized senders, which is passed to the controllers. When an unauthorized sender tries to transmit to the group the controller prevents it from doing so.

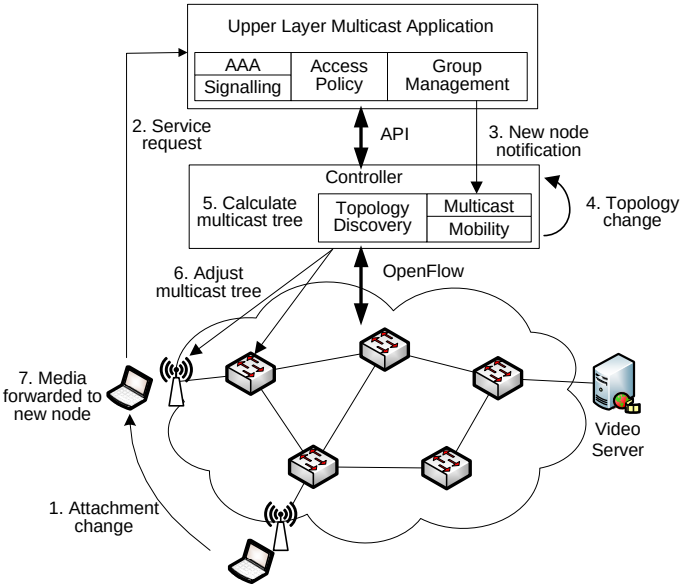


Fig. 20: Multicast communication in Kiciński model [129]

2) Leu Model: Leu et al. [130] has proposed a receiver access control scheme for SDN multicast. Figure 21 shows the joining of a receiver in this model, where a receiver is authenticated by the source and then supplied an authentication token. If the receiver would like to receive group data the receiver provides this token inside the IGMP membership report (through the auxiliary data field of the IGMP membership report), which is supplied to the controller through an OpenFlow PacketIn message. Note that the insertion of an "authentication token" inside the IGMP membership report is the same idea as was mentioned in Section VII-C for IGMP-AC. The controller forwards this authentication token to the source. The source authenticates and authorizes the receiver, and then informs the controller of its approval or rejection. Since the controller has a centralized view of the network, in case of successful authentication, the controller updates the forwarding states of the routers and switches accordingly. Note that in this model only the receivers have been authenticated by the source. The identity of the source has not been authenticated.

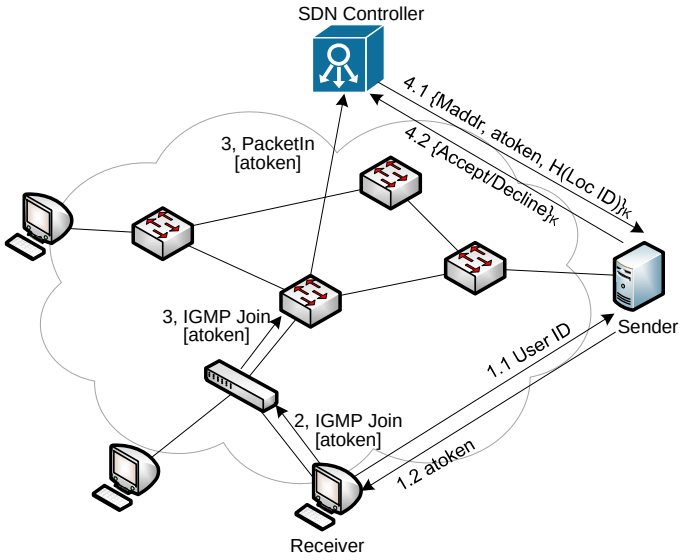


Fig. 21: Multicast communication in Leu model [130]

3) Zou Model: Unlike the Kiciński model, a two layer (i.e., controller and forwarding) SDN multicast model has been presented by Zou et al. [131]. The detailed architecture of this model is shown in Figure 22. In this model, the central controller is responsible for all sorts of user authentication, group management, topology maintenance, route configuration, etc. At first, a user sends an IGMP membership report to the OpenFlow switch, which forwards this to the controller. This proposed model does not support any receiver/user authentication mechanism and thus no authentication token is being sent inside the IGMP membership report message. The controller inspecting only the user's identity/address authorizes the user. As we have described in the earlier models, if the user is authorized to join the multicast group, the controller configures the forwarding entries of the switches/routers so that the distribution tree is extended up to the newly joined user.

4) SDM^{2}Cast: Software-defined scalable multimedia multicast streaming (SDM^{2}Cast) has been designed to provide optimized multimedia flow management, scalability and flexibility [98]. The main focus of this model is to provide Scalabale Video Coding (SVC) multicast in an SDN-enabled

1553-877X (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2017.2776213, IEEE Communications Surveys & Tutorials

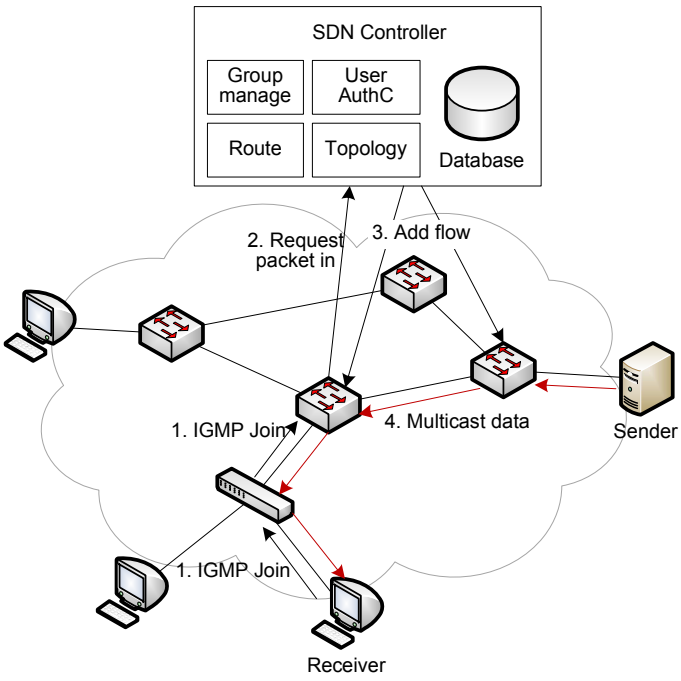


Fig. 22: Multicast communication in Zou model [131]

network through in-network adaptive transmission and access control. Hence, user authentication and authorization is not the primary focus of this model. Figure 23 shows the architecture of SDM^{2}Cast, which consists of controller, media server and management server. In this architecture the media server stores the contents and is also responsible for SVC encoding and streaming to different SVC layers. The SDN controller performs multicast group management and distribution tree formation functions. On the other hand the management server performs user manager, video manager, AAA and policy related functions. Note that the management server communicates with the SDN controller and the media server over secured HTTP protocols.

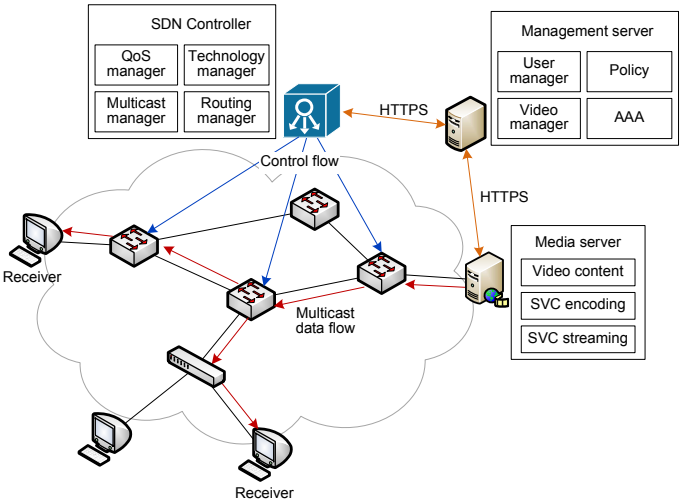


Fig. 23: Multicast architecture of SDM^{2}Cast [98]

5) Summary of Different Efforts: A comparative summary of four different approaches for participant access control in SDN multicast is presented in Table IV. In this summary we have compared these methods based on participant access control and other related issues including the controller, distribution tree, prototype implemented and other supported services.

Among the four methods, Leu [130] and Zou [131] models depend on IGMP for receivers' joining to multicast group. The other two methods have not mentioned any specific protocol for the purpose of joining. Although Kiciński model [129] and SDM^{2}Cast [98] have mentioned sender access control, details of the procedures have not been explained. All four methods develop an MST for data distribution. Kiciński model [129] has developed the most comprehensive prototype where an IPTV application has been integrated with mininet based forwarding switches and thus data transmission has been achieved through SDN multicasting. In addition to SDN multicast, Kiciński model [129] has explained how it could be used in case of client mobility. As we have mentioned earlier, SDM^{2}Cast [98] has been developed for SVC supported multimedia streaming applications.

B. Secured Routing Protocol

Unlike IP multicast, SDN multicast does not depend on any specialized routing protocols. SDN controllers communicate with switches and routers using OpenFlow or similar protocols. Therefore, protecting OpenFlow messages provides security to unicast and multicast routing protocol messages as well.

1) Protecting OpenFlow Messages: According to the initial specification of OpenFlow (v1.0.0) [132], the Transport Layer Security (TLS) [133] protocol must be deployed to protect the communication channels between the controller and switches. However, TLS implementation needs many steps to be configured correctly, and thus emerges as a barrier for overall SDN deployment. Compared to TLS, plaintext communication requires only the configuration of the controller address. To speed up SDN deployment, the later version of the specification including the existing version [11] removed the requirement and made TLS optional to implement. It has been observed that among the OpenFlow switch vendors and the popular OpenFlow controllers only a few are supporting TLS [134]. In the absence of TLS support, the communication channel has become vulnerable to several threats, including man-in-the-middle attack, flow enforcement, Denial-of-Service (DoS), etc.

2) Validating Forwarding Rules: To mitigate insider attacks and ensure consistency checks, SDN needs to implement cryptographic primitives [135]. These cryptographic primitives also protect control planes from outside attackers. Since SDN architecture is highly modularized and distributed, the consistency checking needs to be implemented not only in the SDN controller but also in the switches. A symmetric key based extension to SDN, named as SDNsec [136] has been proposed, where the controller shares separate symmetric keys with all switches or forwarding devices. In this model, only

1553-877X (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2017.2776213, IEEE Communications Surveys & Tutorials

TABLE IV: Summary of different proposals on secured multicast in SDN

Proposal	Summary	Receiver authentication	Sender authentication	Authorization and accounting	Controller	Distribution tree	Prototype	Other services
Kiciński model [129]	A complete architecture with three layers: application, control and forwarding	No details on authentication and group key management. Receivers are authenticated by the Upper Layer Multicast Application. Does not use IGMP.	Provides sender control	AAA and access policy	Floodlight	A simple MST, not Steiner tree but could be supported	Gstreamer-based IPTV application, which has been integrated with mininet-based forwarding. Floodlight controller in between.	Supports mobility
Leu model [130]	A receiver access control architecture where the source authenticates the receiver.	Receivers first receive authentication token from the sender and then send IGMP join messages where this token is included. The controller forwards this join message to the sender and the sender authenticates the receivers.	Nothing has been mentioned.	Source can authorize receivers	POX-based GroupFlow	GroupFlow is based on CastFlow, where MST-based routing paths are computed in advance for known sources to all other switches in the network.	A simple prototype has been implemented based on mininet where the GroupFlow has been implemented.	None
Zou model [131]	An architecture with multicast controller. No application interface to controller.	IGMP messages are forwarded to controller that authenticates receivers. Authentication is based on MAC address.	Discussed about sender control.	None	NOX	Kruskal algorithm to calculate a MST centralized on source.	A simple prototype with user authority databases, which keeps the MAC addresses of the source and the receiver.	None
SDM^{2}Cast [98]	SDN-based, scalable multimedia multicast streaming framework, which optimizes multimedia flow management and provides scalability and flexibility.	Provision of AAA services but no specific or detailed receiver authentication was proposed. Does not use IGMP.	Nothing has been mentioned.	AAA module.	SDM^{2}Cast controller in POX.	A simple MST.	A simple prototype with heterogeneous devices and experiments with in-network adaption. Nothing has been mentioned about receiver authentication or other AAA services.	SVC

the edge switches (i.e., ingress and egress switches) store the forwarding tables. The controller computes the whole path and updates the forwarding tables of the edge switches. The ingress switch receives a packet from the source and inscribes the forwarding information in the packet. The core switches validate this information before forwarding the packet and finally, the egress switch removes this information. SDNsec provides a number of security services including integrity of forwarding rule, validation and enforcement of forwarding rules inside the switches, consistency during configuration, mitigating state exhaustion and path validation.

C. Protecting Multicast Data

In IP multicast, some group key management protocol needs to be deployed to distribute necessary keys to protect multicast data. To our knowledge no such group key management has been studied for SDN-enabled networks.

IX. Discussion and Open Problems in Multicast Security

So far, we have presented existing solutions to protect multicast communication in cases of IP multicast and SDN multicast. In Table V these two service models are compared with respect to three areas of secure multicast communications. In the following we have identified the existing limitations and the open problems that exist in SDN multicast.

A. Access Control

So far four different efforts have been carried out to introduce access control in SDN multicast. However, if we focus on security services, all these efforts seem incomplete and in the primary stage. Note that SDN multicast itself is not an established service model and many challenges exist how to deliver multicast data in SDN networks. Therefore, researchers are concentrating in how to design and implement the controller, how to develop interfaces to communicate with the controller, etc. As a result, access control has been addressed as a subproblem or by-product of the actual solution. For example,

1553-877X (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2017.2776213, IEEE Communications Surveys & Tutorials

although AAA functionality has been mentioned in SDM^{2}Cast [98], detailed architecture and protocols are missing.

B. Routing Protocol

Since TLS is the only suggested method to secure the channels between the controller and the switches, researchers do not have many options to look around. On the other hand, we have already mentioned that making TLS mandatory was slowing down the deployment procedure. Therefore, later versions has opted it out and made TLS implementation optional. Clearly this opens up the threats of DDoS and other attacks. The OpenFlow community should look for some middle way. Some light-weight application layer security protocols should be mandated. For example, in network layer IPsec with AH is preferable over IPsec with ESP, if secrecy is not required. Similarly, in addition to TLS other security protocols might be explored.

C. Secured Data Transmission

To the best of our knowledge, secured data transmission to a group of receivers has not yet been explored in SDN networks. Group key management has no direct relation with the underlying routing and forwarding and thus it might be concluded that no extra benefits or obstacles would be experienced in group key management in SDN network. However, in case of SDN multicast, we may exploit the centralized architecture of the SDN network. Figure 18 shows that GCKS is a central entity that establishes SAs with all receivers. If we compare the logical location and connectivity of the SDN controller and the GCKS, we find these two identical. Thus, future research might be focused on developing a scalable group key management protocol to secure data transmission in SDN multicast.

X. Conclusion

IP multicast routing protocols are, in fact, a distributed algorithm for optimizing the delivery of information to multiple recipients, subject to the constraint of using only local information.

Proposals for SDN multicast routing algorithms reflect the fact that the centralized nature of the control actions in SDN provides an opportunity to (a) remove the "local information" constraint, and (b) broaden the scope of feasible optimizations.

A. Multicast Routing

IP multicast routing approaches have been proposed that optimize:

1) delivery to dense receiver populations
2) delivery to sparse receiver populations
3) reliable delivery
4) time-constrained delivery
5) video delivery

SDN multicast routing approaches have been proposed that optimize:

1) overall resource usage in the controlled network
2) reliable delivery
3) NFV environments
4) video delivery
5) use of parallel paths in the network

In general, the use of SDN permits a much wider set of possible optimizations. However, most of the optimizations proposed for SDN environments are quite specific to a particular application, or reflect a particular constraint on acceptable solutions. Therefore, use of SDN will neither encourage nor discourage widespread deployment of "general" (application-independent) multicasting. However, the proposed approaches do have the potential to provide relatively convenient mechanisms for deploying multicasting in support of specific applications.

B. Multicast Security

IP multicast security approaches have been proposed that provide security for:

1) protection of user data
2) protection of routing protocol messages
3) participant access control

SDN security approaches have concentrated on participant access control, because protection of user data is outside the scope of SDN, and procedures for protection of SDN control messages are defined in the OpenFlow standards.

We believe that security for SDN is extremely important, especially given that if an SDN controller is compromised, this will open the entire controlled network area to attack. However, due to the difficulty of using the security features of OpenFlow, there has been an elimination of the requirement to protect OpenFlow messages. This leads us to suggest that a useful area of study would be to find easier-to-use security approaches.

Acknowledgments

Support for Salekul Islam was provided in part by United International University (UIU), through its UIU Research Grant program.

Support for J. William Atwood was provided in part by the Natural Sciences and Engineering Research Council of Canada, through its Discovery Grants program.

References

[1] B. A. A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turletti, "A survey of software-defined networking: Past, present, and future of programmable networks," *IEEE Communications Surveys & Tutorials*, vol. 16, pp. 1617–1634, 2014.

[2] D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015.

[3] A. Lara, A. Kolasani, and B. Ramamurthy, "Network innovation using openflow: A survey," *Communications Surveys Tutorials, IEEE*, vol. 16, no. 1, pp. 493–512, First Quarter 2014.

[4] S. Deering, "Host extensions for IP multicasting," RFC 1112, Tech. Rep. 1112, Aug. 1989, updated by RFC 2236. [Online]. Available: https://rfc-editor.org/rfc/rfc1112.txt

[5] B. Quinn and K. C. Almeroth, "IP Multicast Applications: Challenges and Solutions," RFC 3170, Tech. Rep. 3170, Sep. 2001. [Online]. Available: https://rfc-editor.org/rfc/rfc3170.txt

1553-877X (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2017.2776213, IEEE Communications Surveys & Tutorials

TABLE V: Comparison of security services in IP multicast and SDN multicast

Area of consideration	Basis of comparison	IP multicast	SDN multicast
Participant access control	Dependency on IGMP	Both IGMP-AC [36] and SIGMP [124] depend on IGMP for determining users' interest to join multicast group.	In SDN multicast, Leu [130] and Zou [131] models use IGMP for joining multicast groups. The other two models have not mentioned any specific protocol for joining multicast group.
	Implementation layer	End user authentication is performed in the application layer while access control is enforced at the network layer, although IGMP-AC and SIGMP couple these two differently.	In Leu and Zou models, IGMP transports end users' credentials to OpenFlow switches, which then forward this information to the controller using OpenFlow switch protocol. This protocol runs at the application layer over TLS/TCP.
	Receiver access control	IGMP-AC provides EAP-based receiver authentication and incorporates AAA protocol (i.e., Diameter). SIGMP provides a shared key based authentication and authorization, which is derived during the GSAM establishment phase.	In Leu model, an authentication token-based (received from the sender) authentication is proposed. Zou model does not propose any authentication model rather end user is authorized based on user's identity and address.
	Sender access control	SAC model provides EAP-based sender access control and incorporates AAA protocols.	Only Kiciński model provides sender access control by maintaining a list of authorized sources. Details of authentication and authorization procedures are not mentioned.
	Completeness	Both IGMP-AC and and SIGMP could be considered as a focused and complete work with rigorous security analysis.	All four SDN multicast efforts are at the initial stage and many designs issues are yet to be addressed.
Secured data transmission	Security architecture	The IETF has developed GSA that provides secured data transmission through establishing IPsec SAs.	No significant study has been carried out in SDN multicast.
	Group key management	The IETF has designed the necessary architecture and a number of protocols have been developed by independent researchers outside the IETF community.	No significant study has been carried out in SDN multicast.
	Security services	GSA provides confidentiality, source authentication and integrity protection, source data origin authentication and anti-replay protection.	Since no proposal exists multicast data could not be distributed securely in SDN multicast.
Secured routing protocol	Implementation layer	At the network layer since IGMP and PIM-SM and other routing protocols are layer-3 protocols.	OpenFlow is an application layer protocol. However, SDN does not mandate nor preclude the use of IGMP. Security of IGMP needs to be assured at the network layer.
	Security mechanism	PIM-SM unicast messages [21] and IGMP messages [18] are suggested to be protected through IPsec with AH, which ensures data origin authentication and data integrity protection. PIM-SM link-local messages are mandated [123] to be protected using IPsec with ESP, which provides confidentiality, data origin authentication and data integrity protection.	OpenFlow messages exchanged by the controller and the switches are suggested to be protected through TLS, which ensures confidentiality and data integrity [11]. However, TLS is not being used in most of the cases. If IGMP is also used the IGMP messages could be protected as described in regular IP multicast case.

[6] M. Eubanks, "The Basics of Multicast," South Asian Network Operators Group (SANOG 7), Mumbai, India, Jan. 2006. [Online]. Available: http://www.sanog.org/resources/sanog7/eubanks-multicast-tutorial.pdf

[7] C. Diot, B. N. Levine, B. Lyles, H. Kassem, and D. Balensiefen, "Deployment issues for the ip multicast service and architecture," *Network, IEEE*, vol. 14, no. 1, pp. 78–88, 2000.

[8] [Online]. Available: https://datatracker.ietf.org/wg/forces/charter/

[9] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, and J. Rexford, "Openflow: Enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, pp. 69–74, Apr. 2008.

[10] S. Hares, "Analysis of Comparisons between OpenFlow and ForCES," Internet Engineering Task Force, Internet-Draft draft-hares-forces-vs-openflow-00, Jan. 2013, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-hares-forces-vs-openflow-00

[11] "OpenFlow Switch Specification, Version 1.5.0, ONF TS-020," 12 2014. [Online]. Available: https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-switch-v1.5.0.noipr.pdf

[12] "Open Networking Foundation," http://www.opennetworking.org.

[13] E. Haleplidis, K. Pentikousis, S. Denazis, J. H. Salim, D. Meyer, and O. Koufopavlou, "Software-Defined Networking (SDN): Layers and Architecture Terminology," RFC 7426, Tech. Rep. 7426, Jan. 2015. [Online]. Available: https://rfc-editor.org/rfc/rfc7426.txt

[14] "SDN Architecture Overview, Version 1.1, ONF TR-504," 11 2014. [Online]. Available: https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/TR_SDN_ARCH_1.0_06062014.pdf

[15] "OpenFlow Management and Configuration Protocol, OpenFlow Config 1.2, ONF TS-016," 2014. [Online]. Available: https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow-config/of-config-1.2.pdf

[16] J. A. Hawkinson and T. J. Bates, "Guidelines for creation, selection, and registration of an Autonomous System (AS)," RFC 1930, Tech. Rep. 1930, Mar. 1996. [Online]. Available: https://rfc-editor.org/rfc/rfc1930.txt

[17] B. Fenner, "Internet Group Management Protocol, Version 2," RFC 2236, Tech. Rep. 2236, Nov. 1997. [Online]. Available: https://rfc-editor.org/rfc/rfc2236.txt

[18] B. Cain, S. E. Deering, B. Fenner, I. Kouvelas, and A. Thyagarajan, "Internet Group Management Protocol, Version 3," RFC 3376, Tech. Rep. 3376, Oct. 2002. [Online]. Available: https://rfc-editor.org/rfc/rfc3376.txt

[19] S. E. Deering, B. Fenner, and B. Haberman, "Multicast Listener Discovery (MLD) for IPv6," RFC 2710, Tech. Rep. 2710, Oct. 1999. [Online]. Available: https://rfc-editor.org/rfc/rfc2710.txt

[20] L. Costa and R. Vida, "Multicast Listener Discovery Version 2 (MLDv2) for IPv6," RFC 3810, Tech. Rep. 3810, Jun. 2004. [Online]. Available: https://rfc-editor.org/rfc/rfc3810.txt

[21] H. Holbrook, I. Kouvelas, M. J. Handley, R. Parekh, Z. J. Zhang, B. Fenner, and L. Zheng, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)," RFC 7761, Tech. Rep. 7761, Mar. 2016. [Online]. Available: https://rfc-editor.org/rfc/rfc7761.txt

[22] D. Meyer and B. Fenner, "Multicast Source Discovery Protocol (MSDP)," RFC 3618, Tech. Rep. 3618, Oct. 2003. [Online]. Available: https://rfc-editor.org/rfc/rfc3618.txt

[23] T. Bates, R. Chandra, D. Katz, and Y. Rekhter, "Multiprotocol

1553-877X (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2017.2776213, IEEE Communications Surveys & Tutorials

Extensions for BGP-4," RFC 4760, Tech. Rep. 4760, Jan. 2007, updated by RFC 7606. [Online]. Available: https://rfc-editor.org/rfc/rfc4760.txt

[24] C. A. S. Oliveira and P. M. Pardalos, "A survey of combinatorial optimization problems in multicast routing," *Computers & Operations Research*, vol. 32, no. 8, pp. 1953–1981, 2005.

[25] A. Ballardie, P. Francis, and J. Crowcroft, "Core based trees (cbt) an architecture for scalable inter-domain multicast routing," in *SIGCOMM 93*, Ithaca, NY, Sep 1993, pp. 85–95.

[26] A. Ballardie, "Core Based Trees (CBT) Multicast Routing Architecture," RFC 2201, Tech. Rep. 2201, Sep. 1997. [Online]. Available: https://rfc-editor.org/rfc/rfc2201.txt

[27] D. Thaler and M. Handley, "On the aggregatability of multicast forwarding state," in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3. IEEE, 2000, pp. 1654–1663.

[28] M. Yamamoto, "Multicast communications–present and future," *IEICE transactions on communications*, vol. 86, no. 6, pp. 1754–1767, 2003.

[29] A. Ganjam and H. Zhang, "Internet multicast video delivery," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 159–170, Jan 2005.

[30] I. Romdhani, M. Kellil, H.-Y. Lach, A. Bouabdallah, and H. Bettahar, "IP Mobile Multicast: Challenges and Solutions," *IEEE Comm. Surveys & Tutorials*, vol. 6, no. 1, pp. 18–41, 2004.

[31] F. K. Hwang, D. S. Richards, and P. Winter, *The Steiner tree problem*. Elsevier, 1992, vol. 53.

[32] A. Fei, J. Cui, M. Gerla, and M. Faloutsos, "Aggregated multicast: an approach to reduce multicast state," in *Global Telecommunications Conference, 2001. GLOBECOM'01. IEEE*, vol. 3. IEEE, 2001, pp. 1595–1599.

[33] R. Mukherjee and J. Atwood, "Rendezvous point relocation in protocol independent multicast-sparse mode," *Telecommunication Systems*, vol. 24, no. 2-4, pp. 207–220, 2003.

[34] T. Hardjono and B. Weis, "The Multicast Group Security Architecture," RFC 3740, Tech. Rep. 3740, Mar. 2004. [Online]. Available: https://rfc-editor.org/rfc/rfc3740.txt

[35] G. Gross, B. Weis, and D. Ignjatic, "Multicast Extensions to the Security Architecture for the Internet Protocol," RFC 5374, Tech. Rep. 5374, Nov. 2008. [Online]. Available: https://rfc-editor.org/rfc/rfc5374.txt

[36] S. Islam and J. W. Atwood, "Multicast receiver access control by igmp-ac," *Comput. Netw.*, vol. 53, no. 7, pp. 989–1013, May 2009.

[37] ——, "Sender access and data distribution control for inter-domain multicast groups," *Comput. Netw.*, vol. 54, no. 10, pp. 1646–1671, Jul. 2010.

[38] ——, "Multicast receiver access control using pana," in *Proceedings of the 1st Taibah University International Conference on Computing and Information Technology (ICCIT 2012)*, 2012, pp. 816–821.

[39] C. Diot, W. Dabbous, and J. Crowcroft, "Multipoint communication: A survey of protocols, functions, and mechanisms," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 3, pp. 277–290, Apr. 1997.

[40] L. H. Sahasrabuddhe and B. Mukherjee, "Multicast routing algorithms and protocols: a tutorial," *IEEE Network*, vol. 14, no. 1, pp. 90–101, Jan/Feb 2000.

[41] M. Ramalho, "Intra- and inter-domain multicast routing protocols: A survey and taxonomy," *IEEE Communications Surveys & Tutorials*, vol. 3, no. 1, pp. 2–25, First Quarter 2000.

[42] R. Rummler, A. D. Gluhak, and H. Aghvami, *Multicast in Third-Generation Mobile Networks: Services, Mechanisms and Performance*. Wiley Publishing, 2009.

[43] J. Moy, "Multicast Extensions to OSPF," RFC 1584, Tech. Rep. 1584, Mar. 1994. [Online]. Available: https://rfc-editor.org/rfc/rfc1584.txt

[44] D. Waitzman, C. Partridge, and S. Deering, "Distance Vector Multicast Routing Protocol," RFC Editor, RFC 1075, Nov. 1988. [Online]. Available: https://rfc-editor.org/rfc/rfc1075.txt

[45] H. Holbrook and B. Cain, "Source-Specific Multicast for IP," RFC 4607, Tech. Rep. 4607, Aug. 2006. [Online]. Available: https://rfc-editor.org/rfc/rfc4607.txt

[46] J.-H. Cui, D. Maggiorini, J. Kim, K. Boussetta, and M. Gerla, "A protocol to improve the state scalability of source specific multicast," in *Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE*, vol. 2, Nov 2002, pp. 1899–1904 vol.2.

[47] T. Speakman, L. Vicisano, M. J. Handley, and I. Kouvelas, "Bidirectional Protocol Independent Multicast (BIDIR-PIM)," RFC 5015, Tech. Rep. 5015, Oct. 2007. [Online]. Available: https://rfc-editor.org/rfc/rfc5015.txt

[48] S. Floyd, V. Jacobson, C.-G. Liu, S. McCanne, and L. Zhang, "A reliable multicast framework for light-weight sessions and application level framing," *IEEE/ACM Trans. Netw.*, vol. 5, no. 6, pp. 784–803, Dec. 1997. [Online]. Available: http://dx.doi.org/10.1109/90.650139

[49] T. Speakman, J. Crowcroft, J. Gemmell, D. Farinacci, S. D. Lin, D. Leshchiner, M. Luby, T. L. Montgomery, L. Rizzo, A. Tweedly, N. Bhaskar, R. Edmonstone, R. Sumanasekera, and L. Vicisano, "PGM Reliable Transport Protocol Specification," RFC 3208, Tech. Rep. 3208, Dec. 2001. [Online]. Available: https://rfc-editor.org/rfc/rfc3208.txt

[50] S. Paul, K. K. Sabnani, J.-H. Lin, and S. Bhattacharyya, "Reliable multicast transport protocol (rmtp)," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 3, pp. 407–421, 1997.

[51] B. Whetten and G. Taskale, "An overview of reliable multicast transport protocol ii," *Network, IEEE*, vol. 14, no. 1, pp. 37–47, 2000.

[52] J. W. Atwood, "A classification of reliable multicast protocols," *IEEE Network*, vol. 18, pp. 24–34, 2004.

[53] L. Zhang, S. Berson, S. Herzog, and S. Jamin, "Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification," RFC 2205, Tech. Rep. 2205, Sep. 1997. [Online]. Available: https://rfc-editor.org/rfc/rfc2205.txt

[54] D. Taubman and A. Zakhor, "Multirate 3-d subband coding of video," *Trans. Img. Proc.*, vol. 3, no. 5, pp. 572–588, Sep. 1994. [Online]. Available: http://dx.doi.org/10.1109/83.334984

[55] J.-C. Bolot, T. Turletti, and I. Wakeman, "Scalable feedback control for multicast video distribution in the internet," *SIGCOMM Comput. Commun. Rev.*, vol. 24, no. 4, pp. 58–67, Oct. 1994. [Online]. Available: http://doi.acm.org/10.1145/190809.190320

[56] H.-Y. Tyan, J. Hou, B. Wang, and Y.-M. Chen, "Qos extension to the core based tree protocol," in *Proc. NOSSDAV99*, 1999.

[57] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the h.264/avc standard," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 17, no. 9, pp. 1103–1120, 2007.

[58] A. Iyer, P. Kumar, and V. Mann, "Avalanche: Data center multicast using software defined networking," in *Communication Systems and Networks (COMSNETS), 2014 Sixth International Conference on*, Jan 2014, pp. 1–8.

[59] D. Kotani, K. Suzuki, and H. Shimonishi, "A design and implementation of openflow controller handling ip multicast with fast tree switching," in *Applications and the Internet (SAINT), 2012 IEEE/IPSJ 12th International Symposium on*. IEEE, 2012, pp. 60–67.

[60] C. Marcondes, T. Santos, A. Godoy, C. Viel, and C. Teixeira, "Castflow: Clean-slate multicast approach using in-advance path processing in programmable networks," in *Computers and Communications (ISCC), 2012 IEEE Symposium on*, July 2012, pp. 000 094–000 101.

[61] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson, *Introduction to Algorithms*, 2nd ed. McGraw-Hill Higher Education, 2001.

[62] M. J. Miic and S. R. Gajin, "Simulation of software defined networks in mininet environment," in *Telecommunications Forum Telfor (TELFOR), 2014 22nd*, Nov 2014, pp. 1055–1058.

[63] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker, "Nox: towards an operating system for networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 3, pp. 105–110, 2008.

[64] A. Medina, A. Lakhina, I. Matta, and J. Byers, "Brite: An approach to universal topology generation," in *Modeling, Analysis and Simulation of Computer and Telecommunication Systems, 2001. Proceedings. Ninth International Symposium on*. IEEE, 2001, pp. 346–353.

[65] M. T. Ananta, J.-R. Jiang, and M. A. Muslim, "Multicasting with the extended dijkstras shortest path algorithm for software defined networking," *International Journal of Applied Engineering Research*, vol. 9, no. 23, pp. 21 017–21 030, 2014.

[66] E. W. Dijkstra, "A note on two problems in connexion with graphs," *NUMERISCHE MATHEMATIK*, vol. 1, no. 1, pp. 269–271, 1959.

[67] A. B. Network, "Advanced networking for research and education," *nd)[Online]. http://abilene. internet2. edu*, 2003.

[68] Y.-D. Lin, Y.-C. Lai, H.-Y. Teng, C.-C. Liao, and Y.-C. Kao, "Scalable multicasting with multiple shared trees in software defined networking," *Journal of Network and Computer Applications*, vol. 78, pp. 125–133, 2017.

[69] M. Zhao, B. Jia, M. Wu, H. Yu, and Y. Xu, "Software defined network-enabled multicast for multi-party video conferencing systems," in *Communications (ICC), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1729–1735.

[70] S. C. Johnson, "Hierarchical clustering schemes," *Psychometrika*, vol. 32, no. 3, pp. 241–254, 1967.

1553-877X (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2017.2776213, IEEE Communications Surveys & Tutorials

[71] D. Zappala and A. Fabbri, "An evaluation of shared multicast trees with multiple active cores," *NetworkingICN 2001*, pp. 620–629, 2001.

[72] P. RYU, "Ryu sdn framework using openflow 1.3. website," 2014.

[73] A. Craig, B. Nandy, I. Lambadaris, and P. Koutsakis, "Bloomflow: Openflow extensions for memory efficient, scalable multicast with multi-stage bloom filters," *Computer Communications*, 2017.

[74] L.-H. Huang, H.-J. Hung, C.-C. Lin, and D.-N. Yang, "Scalable and bandwidth-efficient multicast for software-defined networks," in *Global Communications Conference (GLOBECOM), 2014 IEEE*, Dec 2014, pp. 1890–1896.

[75] J. Tian and G. Neufeld, "Forwarding state reduction for sparse mode multicast communication," in *INFOCOM'98. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2. IEEE, 1998, pp. 711–719.

[76] S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, and M. Roughan, "The internet topology zoo," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 9, pp. 1765–1775, 2011.

[77] C. Jin, Q. Chen, and S. Jamin, "Inet: Internet topology generator," 2000.

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

[79] B. M. Waxman, "Routing of multipoint connections," *IEEE journal on selected areas in communications*, vol. 6, no. 9, pp. 1617–1622, 1988.

[80] S. Zhou, H. Wang, S. Yi, and F. Zhu, "Cost-efficient and scalable multicast tree in software defined networking," in *Algorithms and Architectures for Parallel Processing*. Springer, 2015, pp. 592–605.

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

[82] N. Shacham, "Multipoint communication by hierarchically encoded data," in *INFOCOM'92. Eleventh Annual Joint Conference of the IEEE Computer and Communications Societies, IEEE*. IEEE, 1992, pp. 2107–2114.

[83] D. Wu, Y. T. Hou, W. Zhu, Y.-Q. Zhang, and J. M. Peha, "Streaming video over the internet: approaches and directions," *IEEE Transactions on circuits and systems for video technology*, vol. 11, no. 3, pp. 282–300, 2001.

[84] J.-P. Sheu, C.-W. Chang, and Y.-C. Chang, "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, 2015, pp. 2089–2093.

[85] N. Xue, X. Chen, L. Gong, S. Li, D. Hu, and Z. Zhu, "Demonstration of openflow-controlled network orchestration for adaptive svc video manycast," *Multimedia, IEEE Transactions on*, vol. 17, no. 9, pp. 1617–1629, Sept 2015.

[86] C. Al Hasrouty, C. Olariu, V. Autefage, D. Magoni, and J. Murphy, "Svc videoconferencing call adaptation and bandwidth usage in sdn networks," 2017.

[87] S. Zhang, Q. Zhang, H. Bannazadeh, and A. Leon-Garcia, "Routing algorithms for network function virtualization enabled multicast topology on sdn," *Network and Service Management, IEEE Transactions on*, vol. 12, no. 4, pp. 580–594, Dec 2015.

[88] S.-H. Shen, L.-H. Huang, D.-N. Yang, and W.-T. Chen, "Reliable multicast routing for software-defined networks," in *Computer Communications (INFOCOM), 2015 IEEE Conference on*. IEEE, 2015, pp. 181–189.

[89] V. R. Raja, L. Chung-Horng, A. Pandey, G.-m. Wei, and A. Srinivasan, "A subtree-based approach to failure detection and protection for multicast in sdn a preliminary version was presented at the 13th wireless and wired international conference, malaga, spain, may 25–27, 2015," *Frontiers of Information Technology & Electronic Engineering*, vol. 17, no. 7, pp. 682–700, 2016.

[90] Q. She and J. P. Jue, "Min-cost tree for multi-resource manycast in mesh networks," in *Advanced Networks and Telecommunication Systems, 2007 First International Symposium on*. IEEE, 2007, pp. 1–2.

[91] S. Kaur, J. Singh, and N. S. Ghumman, "Network programmability using pox controller," in *ICCCS International Conference on Communication, Computing & Systems, IEEE*, no. s 134, 2014, p. 138.

[92] E.-z. Yang, L.-k. Zhang, Z. Yao, and J. Yang, "A video conferencing system based on sdn-enabled svc multicast," *Frontiers of Information Technology & Electronic Engineering*, vol. 17, no. 7, pp. 672–681, 2016.

[93] "Openwrt," https://openwrt.org/, (Accessed on 06/24/2017).

[94] C. Al Hasrouty, V. Autefage, C. Olariu, D. Magoni, and J. Murphy, "Sdn-driven multicast streams with adaptive bitrates for voip conferences," in *Communications (ICC), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1–7.

[95] J. Matias, J. Garay, N. Toledo, J. Unzilla, and E. Jacob, "Toward an sdn-enabled nfv architecture," *IEEE Communications Magazine*, vol. 53, no. 4, pp. 187–193, April 2015.

[96] S. Bemby, H. Lu, K. H. Zadeh, H. Bannazadeh, and A. Leon-Garcia, "Vino: Sdn overlay to allow seamless migration across heterogeneous infrastructure," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, May 2015, pp. 782–785.

[97] J.-M. Kang, H. Bannazadeh, and A. Leon-Garcia, "Savi testbed: Control and management of converged virtual ict resources," in *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on*. IEEE, 2013, pp. 664–667.

[98] J. Yang, E. Yang, Y. Ran, and S. Chen, "SDM^{2}Cast an openflow-based, software-defined scalable multimedia multicast streaming framework," *Internet Computing, IEEE*, vol. 19, no. 4, pp. 36–44, 2015.

[99] J. Reichel, H. Schwarz, and M. Wien, "Joint scalable video model jsvm-8," 2006.

[100] J. Rückert, J. Blendin, and D. Hausheer, "Software-defined multicast for over-the-top and overlay-based live streaming in isp networks," *Journal of Network and Systems Management*, vol. 23, no. 2, pp. 280–308, 2015.

[101] J. N. Matthews, E. M. Dow, T. Deshane, W. Hu, J. Bongio, P. F. Wilbur, and B. Johnson, *Running Xen: a hands-on guide to the art of virtualization*. Prentice Hall PTR, 2008.

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

[103] I. MPEG, "Information technology-dynamic adaptive streaming over http (dash)-part 1: Media presentation description and segment formats," *ISO/IEC MPEG, Tech. Rep*, 2012.

[104] "The internet topology zoo," http://www.topology-zoo.org/dataset.html, (Accessed on 06/26/2017).

[105] "Estinet 9.0 — simulator," http://www.estinet.com/ns/?page_id=21140, (Accessed on 06/26/2017).

[106] H. Tangmunarunkit, R. Govindan, S. Jamin, S. Shenker, and W. Willinger, "Network topology generators: Degree-based vs. structural," in *ACM SIGCOMM Computer Communication Review*, vol. 32, no. 4. ACM, 2002, pp. 147–159.

[107] "Floodlight openflow controller -project floodlight," http://www.projectfloodlight.org/floodlight/, (Accessed on 06/26/2017).

[108] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek, "The click modular router," *ACM Transactions on Computer Systems (TOCS)*, vol. 18, no. 3, pp. 263–297, 2000.

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

[110] E. D. Osborne and A. Simha, *Traffic engineering with MPLS*. Cisco Press, 2002.

[111] D. Katz and D. Ward, "Bidirectional Forwarding Detection (BFD)," RFC 5880, Tech. Rep. 5880, Jun. 2010. [Online]. Available: https://rfc-editor.org/rfc/rfc5880.txt

[112] X. Li and M. J. Freedman, "Scaling ip multicast on datacenter topologies," in *Proceedings of the ninth ACM conference on Emerging networking experiments and technologies*. ACM, 2013, pp. 61–72.

[113] C. E. Leiserson, "Fat-trees: universal networks for hardware-efficient supercomputing," *IEEE transactions on Computers*, vol. 100, no. 10, pp. 892–901, 1985.

[114] V. Liu, D. Halperin, A. Krishnamurthy, and T. E. Anderson, "F10: A fault-tolerant engineered network." in *NSDI*, 2013, pp. 399–412.

[115] W.-K. Jia and L.-C. Wang, "A unified unicast and multicast routing and forwarding algorithm for software-defined datacenter networks," *Selected Areas in Communications, IEEE Journal on*, vol. 31, no. 12, pp. 2646–2657, 2013.

[116] D. Li, H. Cui, Y. Hu, Y. Xia, and X. Wang, "Scalable data center multicast using multi-class bloom filter," in *Network Protocols (ICNP), 2011 19th IEEE International Conference on*. IEEE, 2011, pp. 266–275.

[117] V. Shoup, *A computational introduction to number theory and algebra*. Cambridge university press, 2009.

1553-877X (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/COMST.2017.2776213, IEEE Communications Surveys & Tutorials

[118] W. Cui and C. Qian, "Scalable and load-balanced data center multicast," in *Global Communications Conference (GLOBECOM), 2015 IEEE*. IEEE, 2015, pp. 1–6.

[119] T. Issariyakul and E. Hossain, *Introduction to network simulator NS2*. Springer Science & Business Media, 2011.

[120] J. Nicholas, A. Adams, and W. Siadak, "Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification (Revised)," RFC 3973, Tech. Rep. 3973, Jan. 2005. [Online]. Available: https://rfc-editor.org/rfc/rfc3973.txt

[121] S. Islam and J. W. Atwood, *Multicast Security*. Horizons in Computer Science Research, Volume 2. Nova Science Publishers, 2011, pp. 127–149.

[122] S. Kent and K. Seo, "Security Architecture for the Internet Protocol," RFC 4301, Tech. Rep. 4301, Dec. 2005. [Online]. Available: https://rfc-editor.org/rfc/rfc4301.txt

[123] W. Atwood, S. Islam, and M. Siami, "Authentication and Confidentiality in Protocol Independent Multicast Sparse Mode (PIM-SM) Link-Local Messages," RFC 5796 (Proposed Standard), Tech. Rep. 5796, Mar. 2010. [Online]. Available: https://rfc-editor.org/rfc/rfc5796.txt

[124] B. Li and J. W. Atwood, "Secure receiver access control for ip multicast at the network level: Design and validation," *Computer Networks*, vol. 102, pp. 109–128, 2016.

[125] R. Mukherjee and J. W. Atwood, "Scalable solutions for secure group communications," *Computer Networks*, vol. 51, no. 12, pp. 3525–3548, 2007.

[126] S. Islam, *Participant Access Control in IP Multicasting*. PhD Thesis, Concordia University, Montréal, Canada, 2008.

[127] J. W. Atwood, "An architecture for secure and accountable multicasting," in *Proceedings of the 38th IEEE Conference on Local Computer Networks*. IEEE, 2007, pp. 73–78.

[128] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, and H. Levkowetz, "Extensible Authentication Protocol (EAP)," RFC 3748, Tech. Rep. 3748, Jun. 2004. [Online]. Available: https://rfc-editor.org/rfc/rfc3748.txt

[129] J. Kiciski and M. Hoeft, "A novel multicast architecture of programmable networks," in *Computer Networks*, ser. Communications in Computer and Information Science, P. Gaj, A. Kwiecie, and P. Stera, Eds. Springer International Publishing, 2015, vol. 522, pp. 280–292. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-19419-6_27

[130] P. Leu, "SDN-assisted IP Multicast," Master's thesis, Institut für Technische Informatik und Kommunikationsnetze, Switzerland, December 2014.

[131] J. Zou, G. Shou, Z. Guo, and Y. Hu, "Design and implementation of secure multicast based on sdn," in *Broadband Network Multimedia Technology (IC-BNMT), 2013 5th IEEE International Conference on*, Nov 2013, pp. 124–128.

[132] "OpenFlow Switch Specification, Version 1.0.0," 12 2009. [Online]. Available: http://www.openflow.org/documents/openflow-spec-v1.0.0.pdf

[133] T. Dierks, "The Transport Layer Security (TLS) Protocol Version 1.2," RFC 5246, Tech. Rep. 5246, Aug. 2008. [Online]. Available: https://rfc-editor.org/rfc/rfc5246.txt

[134] K. Benton, L. J. Camp, and C. Small, "Openflow vulnerability assessment," in *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*. ACM, 2013, pp. 151–152.

[135] T. Wan, A. Abdou, and P. C. van Oorschot, "A framework and comparative analysis of control plane security of SDN and conventional networks," *CoRR*, vol. abs/1703.06992, 2017. [Online]. Available: http://arxiv.org/abs/1703.06992

[136] T. Sasaki, C. Pappas, T. Lee, T. Hoefler, and A. Perrig, "Sdnsec: Forwarding accountability for the sdn data plane," in *2016 25th International Conference on Computer Communication and Networks (ICCCN)*, Aug 2016, pp. 1–10.

Salekul Islam (IEEE M'08-SM'17) is an Associate Professor and Head of the Computer Science and Engineering (CSE) Department of United International University (UIU), Bangladesh. Earlier, from 2008 to 2011, he worked as an FRQNT Postdoctoral Fellow at the Énergie, Matériaux et Télécommunications (EMT) center of Institut national de la recherche scientifique (INRS), Montréal. He completed his PhD in 2008 from Computer Science and Software Engineering Department of Concordia University, Canada. His present research interests are in software-defined networks, Edge Cloud computing, future Internet and analysis of security protocols. He also carried out research in IP multicast specially in the area of security issues of multicasting.

Nasif Muslim (IEEE S'17) received the B.Sc. degree in Computer Engineering from American International University-Bangladesh (AIUB), Dhaka, Bangladesh, in 2007, and the M.Sc. degree in Communication Technology from University of ULM, ULM, Baden Württemberg, Germany, in 2012. He is currently pursuing another M.Sc. degree in Computer Science and Engineering at United International University (UIU), Dhaka, Bangladesh. His current research interest includes Edge cloud computing, software-defined networks, and future Internet architecture.

J. William Atwood graduated from McGill University in 1963 with a Bachelor of Engineering, from the University of Toronto in 1965 with a Master of Applied Science, and from the University of Illinois at Urbana-Champaign in 1970 with a Doctor of Philosophy, all in Electrical Engineering.

He is a Distinguished Professor Emeritus in the Department of Computer Science and Software Engineering at Concordia University, Montreal. His research interests include the design, analysis, validation and deployment of protocols for secure multicasting, secure routing, software-defined networks, and autonomic networks.

1553-877X (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.