Adaptive Software Defined Multicast

Jeremias Blendin, Julius Rückert, Tobias Volk, and David Hausheer
Peer-to-Peer Systems Engineering Lab, Technische Universität Darmstadt
Email: {jblendin,rueckert,tvolk,hausheer}@ps.tu-darmstadt.de

***Abstract*—Internet Service Providers (ISPs) need to cope with a growing amount of over-the-top (OTT) traffic, often without a share in the high revenues of the content providers. To achieve an efficient global delivery of content, today content providers usually employ content delivery networks (CDNs) located at the edge of ISP networks from where content is delivered to end users via IP unicast. Many OTT services could benefit from a better support within the ISP's network, e.g. by packet duplication to deliver OTT video streams. While traditional solutions like IP multicast did not prevail, SDN-based alternatives have started to gain attention recently. In contrast to traditional approaches, SDN enables ISPs to support network services in a more manageable and flexible manner. However, the approaches proposed so far are quite rigid and keep state at every network device, independent of the multicast group size. To alleviate this problem, this paper proposes a new approach termed ASDM enabling ISPs to dynamically adjust the tradeoff between bandwidth and state for any multicast service. It is shown that, given ISP-defined bandwidth and state cost functions, the optimal parameter for ASDM can be derived and applied for a transparent multicast-to-unicast conversion achieving the desired characteristics. The proposed approach results in up to 30% bandwidth reduction compared to unicast while using only a seventh of the network state compared to traditional multicast.**

I. Introduction

Internet Service Providers (ISPs) are facing increasing customer demands often without being able to benefit from the high profits of over-the-top (OTT) content providers. While the underlying network infrastructure requires large investments to cope with the increasing OTT traffic, ISPs are in a disadvantageous position in the OTT value chain. Many services such as live video streaming, online radio, and multiplayer games could benefit from a more efficient transport of content through the ISPs network by employing network layer multicast, however, traditional approaches like IP multicast did not prevail.

IP multicast has been thoroughly investigated in the past (cf. [1]–[3]). However, existing implementations are complex, do not scale well and do not provide sufficient control for ISPs [4]. Therefore, the deployment of IP multicast is often limited to intra-ISP use cases today, e.g., for distributing IPTV traffic. This situation led to the rise of application layer multicast [5], which improves the delivery efficiency for OTT providers. However, application layer multicast relies on unicast transport, which is less bandwidth efficient than network layer multicast. The result is that today, most multimedia content is transported via unicast and its delivery is organized by OTT providers, which is an unfavorable situation for ISPs.

One approach to improve the situation of ISPs is by offering on-demand network services. This can create new revenue sources and help to increase the share of identifiable and manageable traffic in the ISPs' networks. Offering packet duplication services may further reduce the bandwidth requirements of content providers and can complement application layer multicast as shown by the OpenFlow-based SDM approach [6]. However, SDM and similar approaches are mainly targeted at scenarios with a small number of large multicast groups in order to maximize efficiency gains. While useful for delivering popular content like live streams from major TV stations or sports events, these approaches do not scale well for applications with smaller user groups like web radios or video conferences which, therefore, cannot take advantage of these approaches and need to rely on unicast transmissions instead. This is a problem, since the popularity of Internet content typically follows a Zipf distribution [7] as depicted in Figure 1 and has been confirmed for streaming services [8]. The main reason for the limited scalability of such approaches regarding the number of multicast groups is high amount of required network state per group, regardless of the group size. Each network device on the path from the sender to all receivers must support the multicast mechanism and keep state for every multicast group whose traffic it forwards. While application layer multicast and unicast transmissions help OTT providers and require no extra network state, they do not improve the situation for ISPs. New services such as SDM improve the transmission efficiency, but impose high resource requirements which limit their application to a small number of large multicast groups.

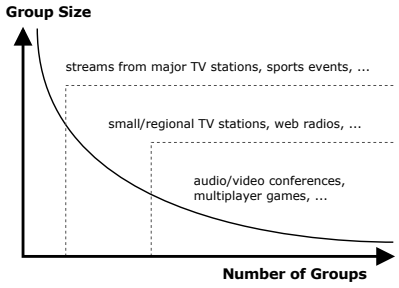

Fig. 1. A Zipf-like multicast group size distribution

To this end, the contribution of this paper is twofold: first it proposes a new packet duplication service that is able to support Zipf distributed group sizes and thereby adapt the system's resource consumption to ISP's network resource requirements. The approach supports the seamless adaptation of different strategies for multicast routing and forwarding, suitable for efficiently accommodating a wide range of multicast group sizes and resource requirements in the network. The approach is based on OpenFlow and enhances the recursive unicast approach to multicast (REUNITE) [3] by a mechanism to dynamically select the depth of the multicast trees. Thereby,

978-1-4799-7899-1/15/$31.00 © 2015 IEEE

multicast trees can be converted into unicast traffic within the ISP network, which reduces the amount of state required, as multicast state is only selectively used. Furthermore, due to the use of unicast forwarding, the approach does not require OpenFlow support of every device in the ISP network and can be used even for inter-ISP traffic. The system behavior is adapted by a single system parameter, termed unicast conversion threshold that controls the size of the multicast trees inside the ISP network. In addition to increasing efficiency and scalability of network layer multicast, the concept enables ISPs to limit either the bandwidth or the network state requirement for the entire multicast forwarding mechanism. Finally, due to the nature of OpenFlow, the multicast concept can be deployed within a single ISP and does not depend on a global rollout.

The rest of this paper is organized as follows: related work is discussed in Section II, followed by the presentation of the system design in Section III. ASDM is evaluated and its results are discussed in Section IV, while finally conclusions are drawn in Section V.

II. Related Work

In classic multicast implementations, intermediate routers have to store each multicast group in their forwarding tables. This limits the amount of groups that can be handled in a network due to the limited maximum size of these tables in currently available hardware. To improve this, research on reducing the state requirements has been conducted. In a MPLS (Multiprotocol Label Switching) [9] based network, the forwarding path for each packet is determined at the edge where a MPLS label is assigned to it. [10] and [11] exploit this approach to optimize multicast forwarding. When the system finds two or more multicast trees which share the same set of recipients, only one label needs to be assigned for this shared tree. This approach saves forwarding table space in scenarios where many multicast groups exist, which share the same or a similar set of recipients. However, given the high number of small groups of Zipf-distributed content, overlapping multicast trees are not expected to occur in large numbers.

Leaky aggregation is another strategy to further reduce forwarding state. If two multicast trees do not exactly share the same set of recipients, but the sets mostly overlap, the trees can be aggregated anyway. This provides more aggregation opportunities, but also causes packets being sent to recipients, who are not interested in them, therefore reduce efficiency and controllability. This tradeoff between reducing forwarding state and wasting bandwidth is further discussed in [12].

REUNITE [3] implements multicast using recursive unicast trees. Instead of special multicast addresses, REUNITE uses regular unicast IP addresses for the identification of groups and the forwarding. A multicast tree is identified by the IP address of its root node and a destination port number. Each time a multicast packet gets duplicated, its IP address gets replaced with the address of the next node in the tree. If the next node is a leaf node, the address of the final destination is used. This approach saves forwarding table space as existing unicast routing information can be re-used, and only routers involved in packet duplication need to maintain extra forwarding table entries. Routers which are not involved in packet duplication simply can rely on the unicast information they already have, and do not need to keep extra state information at all. However, REUNITE does not differentiate between multicast groups size and does still introduce high state consumption when many small groups are active.

Unlike the approaches discussed so far, the Xcast protocol presented in [2] does not maintain multicast groups in the network but includes a list with the address of every recipient in the packet headers. Intermediate routers parse these lists, split them, and duplicate the packets according to their unicast routing tables. The approach has the advantage that no state is kept in the network. However, the additional per packet overhead limits the maximum group size since all destinations have to be explicitly listed in each packet, limiting the approach to scenarios with very small multicast groups. A further improvement on the concept is the usage of explicit multicast with bit-indexed addressing, as proposed in [13] and currently investigated by the IETF as Bit Indexed Explicit Replication [14]. Using bit indexes reduces the per packet overhead, but does not mitigate the limited group size of Xcast. Therefore, a hierarchical multicasting architecture with per domain bit indexes is proposed. However, the approach is only available as drafts so far and has not been investigated in detail yet. Hence, a comparison of ASDM with bit indexed explicit multicast is left for future work.

III. System Design

On the Internet, the popularity of services follow a Zipf distribution, as shown in [7], and confirmed for streaming services in [8]. There is a small amount of services and websites which are extremely popular, while the majority of websites only attract a few users. This is also assumed to apply for the number of receivers in multicast groups. Therefore, ASDM is designed to handle the amount of multicast groups and the group sizes that appear in such a scenario. The scalability of the used network state, similar to space scalability in [15], is especially important in the context of this work. The amount of the available packet matching memory in switches is limited, which means that the amount of forwarding state that can be stored is limited was well. When multicast is used with OpenFlow, the number of required matcher memory, i.e. flow rules, per switch should at most increase linearly with each additional group Opportunities to further reduce the amount of rules should be used whenever possible, especially in the network core. The amount of resources that the system uses should be predictable as well. This especially applies to flow table space and bandwidth. In order to achieve this, algorithms are chosen that generate a predictable amount of flow rules, preferably with a well-known upper bound.

An extended description of the approach including its application interface can be found in [16]. For a detailed comparison of the general Software-Defined Multicast approach with IP multicast and application layer multicast refer to [6].

A. Multicast Addressing Scheme

The ASDM multicast addressing scheme identifies multicast groups by regular unicast addresses instead of using a separate multicast address space. The approach is derived from the addressing concept introduced by REUNITE [3]. It has the advantage that extra forwarding state for multicasting is

required only at routers that are branching nodes in a multicast tree. This has the benefit that routers which are not multicast enabled can simply forward ASDM multicast packets based on their unicast destination address. The system relies on IPv6 which provides with 2^{128} unique addresses a sufficiently large address space for embedding multicast group identification information inside unicast IPv6 addresses. The system assigns an IPv6 subnet for multicast purposes to each OpenFlow switch. The subnets are carved out of the unicast IPv6 address space that belongs to the ISP network domain. In order to keep the unicast routing tables small, the multicast subnets are assigned in pairs together with the unicast subnets, allowing them to be aggregated into a single routing table entry. To calculate a multicast address, the multicast group identifier of a group is embedded into the host part of a multicast subnet.

Table I shows an example allocation scheme that could be used by an ISP operating a /32 network. In this scheme a multicast group with the ID 0x1234 rooted at the switch responsible for the prefix 2001:db8:abcd:8000::/49 can be reached via the address 2001:db8:abcd:8000::1234. The prefix lengths shown in the table are just examples, they can be chosen differently by individual ISPs depending on their requirements.

TABLE I. EXAMPLE IPV6 ADDRESS ALLOCATION FOR UNICAST AND MULTICAST SUBNETS

Network prefix:	2001:db8::/32
Prefix for switch 0xabcd:	2001:db8:abcd::/48
Unicast subnet:	2001:db8:abcd::/49
Multicast subnet:	2001:db8:abcd:8000::/49
Multicast address for group 0x1234:	2001:db8:abcd:8000::1234

REUNITE, which relies on IPv4, identifies multicast groups by a tuple consisting of a unicast address and a destination port number. Using such tuples for multicast group identification to reduce the number of IP addresses makes sense in an IPv4 environment where addresses are in short supply [17]. However, the approach works only in conjunction with transport layer protocols that employ the concept of port numbers, and also requires routers to inspect transport layer protocol headers in addition to IP headers. For example, sending ICMP Echo Requests to multicast groups for debugging purposes is not possible with this design. As adoption of IPv6 progresses [18], it is a sensible choice to use IPv6 instead of IPv4 in order to take advantage of the larger address space.

B. Multicast Routing Algorithm

Each time a new multicast group is accepted into the system or membership in an existing group changes, suitable OpenFlow rules must be generated and installed into affected switches. This process is split in three steps: First, a multicast tree is generated to determine the switches where new rules must be installed. This information is then used to generate a multicast forwarding table specific for each switch. Finally, the forwarding tables are translated into OpenFlow rules to implement the actual multicast forwarding.

Before generation of the multicast tree starts, a suitable root node is selected. In cases where the multicast group has a single source, the nearest OpenFlow switch connected to the host is chosen. Then, a tree that reaches all receivers in the multicast group is calculated. The corresponding Steiner tree problem is NP-complete [19]; therefore, an approximation is used instead. Since the multicast forwarding of the system partly depends on its unicast forwarding, existing unicast routes are used. Therefore, the path from the root node to a certain receiver can be constructed in a similar fashion as a packet travels from source to destination. Starting at the root node, a routing table lookup for the IP address of the receiver yields the switch port number of the link to the next switch. The switch and the link are added to the multicast tree. This process continues until the receiver node is reached, and is repeated for all other receivers in the multicast group. The end result is a multicast tree where each link is annotated with a set of IP addresses, as shown in Figure 2. Switches which have only one outgoing link in the multicast tree are non-branching nodes or unicast nodes, while switches with at least two outgoing links are branching or multicast nodes.

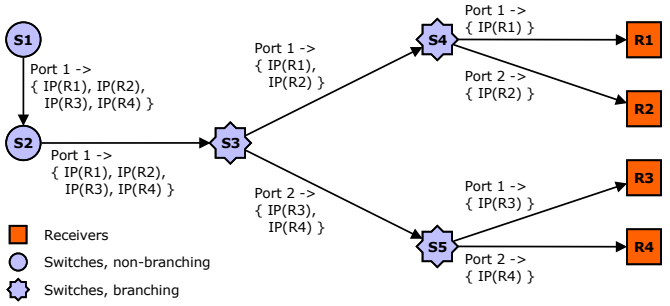

Fig. 2. Annotated multicast tree, with IP address information added to each link

After the generation of the multicast tree, post-processing is used to match constraints set by the network administrator. For example, an ISP might want to disable multicast branching in some switches. In case a branching node in the multicast tree is actually a switch where multicast branching is not allowed, a backwards search for an alternative node is performed in the multicast tree. The search starts at the disabled branching node and moves closer to the root node by one hop in each iteration. The search terminates when a suitable intermediate node or the root node has been found, which is then declared a branching node and takes over the packet duplication that would have been done at the original node.

The post-processed multicast tree is used as an input to the multicast forwarding table generator, which is executed for the root node and each branching node in the multicast tree. Starting at the root node, a multicast address for the group is generated by combining the node IPv6 prefix and the multicast group identifier according to the addressing scheme. For each outgoing link, it is checked whether at least two receivers are reachable. If this is the case, the multicast address that is valid at the next branching node is calculated and inserted into the forwarding table as the next destination address, together with the corresponding outgoing switch port number. The process is then recursively started again at the next branching node. If only one receiver is left, its unicast address is inserted into the forwarding table and the process terminates.

In the final step, the multicast forwarding tables are translated into OpenFlow rules. For each multicast group, a flow rule is generated on the root switch and each branching switch. Each rule matches for the IPv6 Ethertype and the multicast group address valid on the corresponding switch. The action

list contains a pair of header rewrite and output actions for each packet copy that is intended to leave the switch. In the header rewrite action, the IPv6 destination address header field of the packet is rewritten with the multicast group address valid at the next branching node. After the rewrite, the packet is sent out through the appropriate switch port.

C. Duplication Strategies

The algorithm described in III-B implements a late duplication strategy, where multicast packets are duplicated at the last possible branching points in a multicast tree. This minimizes bandwidth consumption in the network and data is sent only once over any given link in the network. However, this also maximizes the forwarding state that needs to be maintained in the network, as additional flow rules need to be installed on all branching nodes in the multicast tree. When considering the cost of this additional forwarding state, this strategy can be inefficient in scenarios with a large number of small multicast groups. Figure 3 a) shows a multicast group implemented using a late duplication strategy.

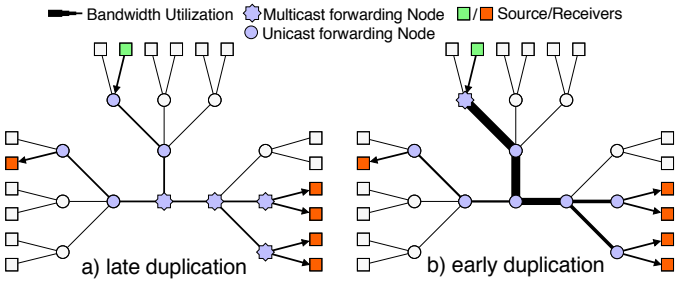

Fig. 3. Late and early duplication multicast forwarding strategies

When using an early duplication strategy, multicast traffic is converted into unicast packets at earlier points in the multicast tree. This reduces forwarding state in the network, as flow rules only need to be installed in a subset of the branching nodes that are required for late duplication. The downside is higher bandwidth consumption, as the same data needs to be transferred multiple times over the same links. However, this strategy can be more efficient when a large number of low bandwidth multicast groups has to be accommodated, especially when the cost of keeping forwarding state is high and the cost of additional bandwidth consumption is low. Figure 3 b) shows a multicast group implemented using an early duplication strategy.

An early duplication strategy can be especially useful when bandwidth is cheap in the core of an ISP network, but expensive at the connections to the customers. For example, a customer operating a small web radio from his home using a slow ADSL link wants to use multicast because his Internet connection does not have enough bandwidth to serve all the listeners via unicast. The radio stream itself is low bandwidth and attracts just a small amount of listeners. From the point of view of the ISP, using a classic late duplication strategy for this multicast stream would be inefficient since the additional forwarding state cost would not justify the bandwidth savings. Instead, the ISP can decide to convert the multicast stream to unicast packets at the first hop where the customer access link terminates. Using this strategy, additional forwarding state is only required at the switch where the unicast conversion happens, and the stream can be handled as regular unicast traffic in the rest of the network.

The choice between late duplication and early duplication strategies is a tradeoff between bandwidth and memory consumption. Due to the expected group size distribution in ISP network environments, both strategies have to be combined in order to make efficient use of network resources. Late duplication strategies are preferable for a small amount of large groups, while early duplication strategies are better suited for a large amount of small groups.

D. Unicast Conversion Threshold

Early duplication can be implemented by making a few small modifications to the algorithm presented in Section III-B. A simple method is to introduce a branch limit by limiting the recursion depth of the algorithm. When the branch limit is reached, traversal of the tree is aborted and flow rules that convert packets to unicast are installed. Setting a high branch limit equals using a late duplication strategy, while setting a low branch limit equals using an early duplication strategy. However, the parameter has to be chosen differently depending on group size for optimal results. Furthermore, the branch limit parameter is topology-dependent: A network consisting of many aggregation levels will need to use higher branch limits than a network with a flatter structure. Therefore, this approach is complex to apply.

Instead of directly setting the branch limit, a unicast conversion threshold is used that is based on the residual tree size, which denotes the number of receiver nodes in a multicast sub tree. When traversing a multicast tree, the late duplication algorithm checks the residual tree size for each link. Packets get converted to unicast if only one receiver is left on a given link. To implement earlier duplication, this threshold can be changed to a higher value. This approach has the advantage that a single threshold value is used for all group sizes, as unicast conversion is performed automatically when the number of receivers behind a link falls below the threshold. This allows a network administrator to adjust the bandwidth-memory tradeoff by setting a single parameter, without having to deal with different group sizes and stream bandwidths.

Further refinement of this approach is possible by including bandwidth data into the decision process. These can be obtained dynamically by measuring the bandwidth consumption of each multicast stream, for example by using the per-flow metering functionality introduced in OpenFlow 1.3. It is then possible to set the unicast conversion threshold based on the product of stream bandwidth and the residual tree size.

E. Incremental OpenFlow Deployment & Inter-Domain Multicast

Deploying ASDM in partially OpenFlow enabled network domains is possible due to its ability to restrict multicast state and forwarding to selected areas of the network domain. The interoperability with unicast forwarding is advantageous for inter-domain multicasting as well; once the traffic is converted to unicast, locally multicasted traffic can be forwarded to other networks without requiring any cooperation. It is also possible to receive a multicast stream from an outside sender and

deliver it via multicast inside the ISP network. To achieve this, a multicast tree is generated which is rooted at the appropriate peering switch. The resulting multicast address is then published to the outside sender, where the stream is sent out via regular unicast packets. As soon as these packets are delivered from the foreign network to the peering switch, the packets are forwarded and duplicated according to the system, just like any other intra-domain multicast group.

While this approach works without cooperation from other networks, it causes high bandwidth consumption at the peering points when a large number of receivers in a multicast group are located within foreign networks. Depending on the peering contracts that an ISP has with other networks, this may be prohibitively expensive. Therefore, ISPs may decide to cooperate with each other in order to enable inter-domain multicasting while keeping bandwidth consumption low. To achieve this, cooperating ISPs can grant each other access to their system API or join an SDX [20] that offers corresponding services. This approach can also be used between ISPs which are not connected directly to each other, but via an intermediate network which may not be interested in such cooperation.

IV. Evaluation

In this section, the effect of the unicast conversion threshold parameter on the system behavior is investigated. The bandwidth and flow table space consumption is measured on different emulated network topologies.

A. Goals & Metrics

The evaluation focusses on the bandwidth and flow table space consumption of the system, as well as the effect of the different multicast forwarding strategies on these values. Given a fixed network topology and workload, the input variable for the measurements is the unicast conversion threshold, which is abbreviated as the parameter t in the remaining sections of this chapter. A value of $t = 1$ represents the classic late duplication strategy, which causes multicast trees to be constructed all the way from source to destinations. This saves the most bandwidth compared to pure unicast, but also produces the highest number of flow rules. Higher values of t produce shorter multicast trees since the conversion to unicast streams happens earlier, resulting in reduced flow rule numbers at the cost of higher bandwidth consumption. Finally, a value of $t = \infty$ converts all multicast traffic to unicast at the first hop in the network, which requires a single OpenFlow rule for every group only. This parameter still implements multicast behavior of the network from the perspective of the client as the packet duplication takes place inside the network.

During evaluation runs the bandwidth is measured by monitoring the byte counters of each network interface in the testbed at fixed intervals. These raw values are then normalized to bytes per second for further processing. The bandwidth consumption for the whole network is obtained by summing up the outgoing bandwidth values at each interface in the network. Bandwidth numbers obtained in this way tend to increase with the average hop count in the network. Therefore, the bandwidth numbers are normalized against a comparison measurement which uses pure unicast forwarding for the given workload. This yields bandwidth numbers in a range between 0 and 1, which are comparable across different topologies. For easier reading, these numbers are always given as percentages (0% - 100%).

The number of flow rules is measured by querying the flow tables of each emulated switch in the testbed, and counting the multicast related flow rules. For per-switch analysis, the resulting numbers are used directly without further processing. For network-wide analysis, the flow rule numbers are summed up across all switches, and then divided by the number of multicast stream receivers in the network. This results in a flow rules per user metric, which is comparable across different topologies, similar to the bandwidth percentages. The numbers range between 0 and 1, as the multicast routing algorithm of the system is guaranteed to never produce more than one flow rule per multicast receiver.

B. Scenario & Workload

The structure of a typical ISP network is hierarchical, and can be roughly divided in access, edge, and core sections. Aggregation networks carry traffic from the individual ISP customers to a nearby PoP (Point of Presence) located at the edge of the ISP network. The PoPs in the edge section connect the attached aggregation networks to the network core, which usually employs multiple redundant links between its routers in order to increase availability. Also located at the edge are the peering points with other ISPs, which connect the network to the rest of the Internet.

For evaluation two distinct topologies mimicking ISP network topology characteristics are used. The first topology is a simple tree-shaped topology as shown in Figure 4 a), representing networks without redundant links and a single central core switch which are typical for aggregation networks. The second topology is modeled after the Germany network of Deutsche Telekom [21]. It consists of a triangle-shaped inner core, which is extended by outer core nodes, each maintaining two links to the inner core and therefore forming further triangles. A tree-shaped aggregation section is attached to each outer core node, forming the network as depicted in Figure 4 b). Table II shows the parameters used to construct the two topologies and their characteristics. The topology sizes are chosen to be as large as possible with respect to the resource limitations imposed by the testbed.

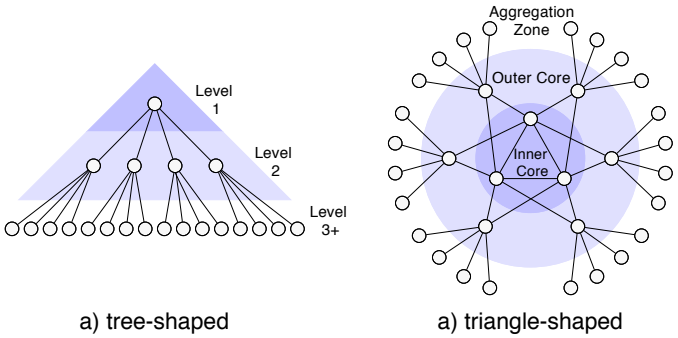

Fig. 4. Network topologies used in the evaluation

Link speeds and traffic volumes are comparatively low in the aggregation networks, but get orders of magnitude higher close to the network core. The consequence of this is that

TABLE II. CHARACTERISTICS OF THE NETWORK TOPOLOGIES USED FOR INDIVIDUAL EVALUATION RUNS

	Tree topology		Triangle topology	
Zone	**# Nodes**	**Fanout**	**# Nodes**	**Fanout**
Level 1 / Inner Core	1	4	3	2
Level 2 / Outer Core	4	4	6	4
Level 3 / Aggregation Level 1	16	4	24	4
Level 4 / Aggregation Level 2	64	16	96	11
Hosts	1024	-	1056	-

specialized equipment is needed at the core, which is able to make packet forwarding decisions very quickly in order to support such high data rates. However, the memory required for this purpose is still very limited in the hardware available today [22]. Therefore, the MPLS protocol is often utilized in ISP networks to reduce complexity in the core as much as possible. In such a setup, the path of each packet through the network is determined at the edge; the switches in the network core then make forwarding decisions based on a restricted set of MPLS labels instead of lookups in the global IP routing table, which is very large [23]. To analyze the systems applicability in such networks, the triangle-shaped network topology is evaluated twice: one time in the previously described configuration, and one time with a unicast core, where packet duplication is restricted to the network edge and aggregation sections.

The workload in the evaluation is modeled after the expected Zipf multicast group distribution. The evaluated scenario is home users during the daily peak traffic in the evening [24]. A large portion of users is assumed to be at home, using the fixed Internet access. Given the wide range of applications that support multicast, a participation ratio of 50% of the active fixed Internet access users is used. Each fixed Internet access consumes at maximum one multicast service at a time.

A total of 252 multicast groups and 3072 multicast receivers are used for analysis. Therefore, the group numbers and sizes are selected as follows: 4 groups with 128 members, 8 groups with 64 members, 16 groups with 32 members, 32 groups with 16 members, 64 groups with 8 members, and 128 groups with 4 members. The different group sizes are measured separately and the measurements are combined later during analysis, to reduce the load on the testbed. This is possible because the used routing algorithm implements each multicast group independently from other groups. Therefore, the maximum number of active hosts in the network during individual runs is 512 of 3072 total hosts.

The measurements with multiple multicast groups being active at the same time are performed separately for each group size and are repeated 30 times. A randomly generated set of multicast groups is used for each measurement. The measurements are repeated with varying unicast conversion thresholds, including a pure unicast measurement. The individual measurement sets for the different group sizes are combined by summing up the bandwidth and flow rule numbers of each measurement set.

For each multicast group the workload is an audio stream using a 128 kilobit per second MP3 file. This is a popular bitrate configuration used by web radio streams, as it provides decent audio quality with moderate bandwidth consumption. Emulating high bitrate streams like HDTV is unfeasible due to the performance limitations of the testbed. The workload is generated by a custom application, which creates a packet stream that emulates the output of VLC media player in UDP streaming mode. To detect potential packet loss, another custom application is employed to count the number of bytes received at each multicast group member.

The prototype is implemented using the Ryu OpenFlow controller, with the OpenFlow version set to 1.3. The evaluation is conducted using the Mininet 2.0 [25] network emulator with Open vSwitch 2.0.1 on Ubuntu Server 14.04. The custom network topology relies solely on IPv6. The evaluation is conducted on a VM with 4 CPU cores and 6GB memory on a server with an Intel Xeon E5-1410 Processor and 48GB memory.

C. Results & Discussion

By adjusting the unicast conversion threshold, the system can be optimized for either minimal bandwidth consumption, minimal flow table space consumption, or a balanced use of these two resources according to the requirements. The discussion focusses on the values $t \in \{1..16\}$ that have a relevant impact on either the bandwidth or flow rule consumption. Figure 5 depicts this bandwidth-state tradeoff for all measured scenarios, with the x-axis starting at 35%. Each point in the plot represents the bandwidth consumption and the number of flow rules produced by a given unicast conversion threshold configuration. Additionally, a bandwidth-state profile is generated for each topology by fitting a third degree polynomial to the points in the plot using a least squares method to illustrate the potential adaption space. For the evaluated scenarios the bandwidth savings range from nearly $60\%, t = 1$ to slightly less than $30\%, t = 16$ compared to unicast traffic. The required flow rules per user vary between $0.1, t = 1$ and $0.7, t = 16$. Note that both the standard deviation as well as the 95% confidence interval are small and are omitted for that reason.

The bandwidth-state profiles can be used to configure limits for either bandwidth consumption or the number of flow rules. An ISP that wants to limit the number of flow rules used by the system enters the desired limit into the system configuration, and the system then selects the lowest unicast conversion threshold value which stays below the flow rule limit. Alternatively, the administrator can limit the bandwidth consumption instead, and the system selects the highest threshold which stays below the bandwidth limit. An example configuration for the tree topology is shown in Figure 6. The number of flow rules is limited to 0.3 per user, and the system sets the threshold to $t = 4$ in order to stay below the limit. Alternatively, the example can also be interpreted as a bandwidth limit of 46%, with the system setting the threshold to $t = 3$. The profiles vary depending on network topology and workload. It is possible to obtain the profile data for any network by repeating the evaluation steps as described before and choosing the emulated topology and workloads accordingly.

Given the high impact on flow rules compared to the impact on the bandwidth consumption, setting a fixed optimal value for a given network could be desirable. For example, by assuming that the costs of bandwidth and flow table space consumption grow linearly, the maximum yield for the utilized flow rules is identified by analyzing the saved

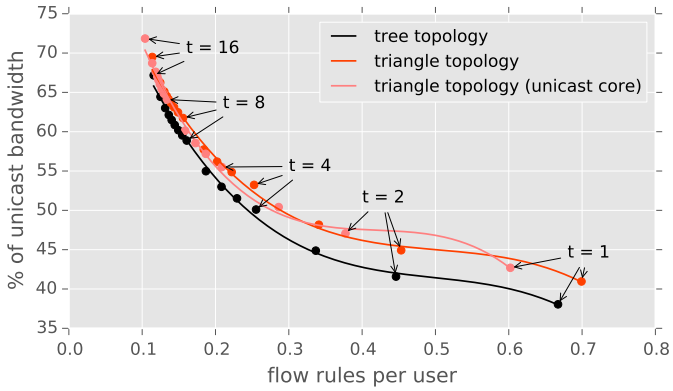

Fig. 5. The bandwidth-state profiles of each network topology

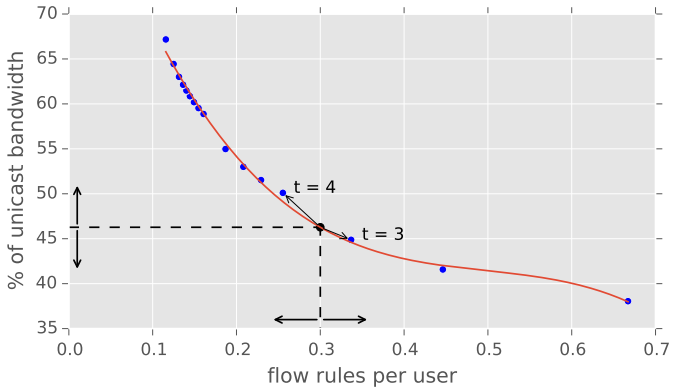

Fig. 6. Configuration of a flow rule or bandwidth limit using a bandwidth-state profile

bandwidth per flow rule. For the evaluated topologies and workload, the optimal configuration is $t = 15$, as shown in Figure 7. For $t = 15$ the bandwidth usage is 30% reduced compared to unicast distribution while using only a seventh of the network state compared to classic multicast with late duplication. However, in real networks, the associated costs of bandwidth and flow table space consumption may follow non-linear growth patterns. For example, additional bandwidth consumption may be free for the ISP up to a certain point until one or more links in the network need to be upgraded. Likewise, additional flow rules may be tolerable until the flow

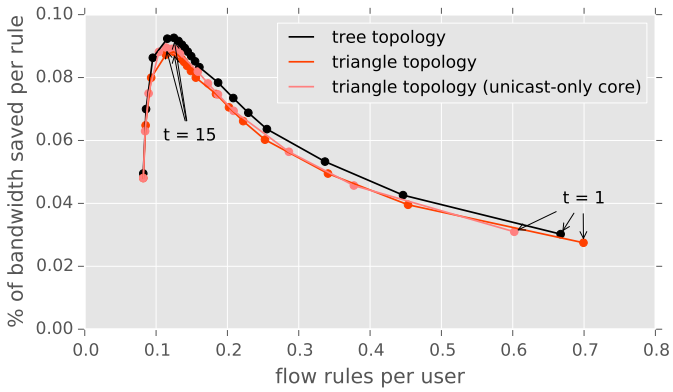

Fig. 7. Bandwidth savings per flow rule

table at one switch in the network reaches capacity, requiring the switch to be replaced with another model supporting more flow rules. These non-linear costs need to be taken into account by an ISP in order to find the most cost effective configuration for a given network.

The system enables ISPs to configure the bandwidth and network state consumption of multicast traffic between 70% and 40% of the unicast bandwidth and between 0.1 and 0.7 flow rules per multicast user for the evaluated networks. This allows the ISP to both to adapt the system to dynamically varying network conditions as well as to fixed network design requirements. The first is achieved by generating a bandwidth-state tradeoff profile and dynamically selecting appropriate resource limits. For the latter use case, depending on individual costs for bandwidth and flow rules, a fixed optimal value can be determined. Both require the selection of a single parameter which ensures easy applicability.

D. System Analysis & Costs

In this section, the characteristics and resource requirements of the system are investigated. Figure 8 depicts the average bandwidth consumption and Figure 9 the number of flow rules for each topology. The bars are divided in five sections, each representing the different zones where a network node can be located. Bandwidth measurements are assigned to the zone where the traffic originates. For example, upstream traffic from a host node to the next aggregation switch is counted as host zone traffic, while downstream traffic on the same link is counted as aggregation zone traffic. Flow rule numbers of a given switch are assigned to the zone where the switch is located. Confidence intervals are not shown since they are very narrow.

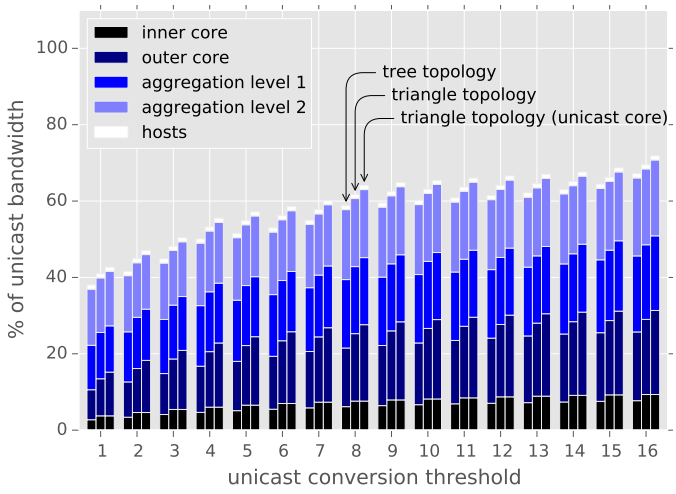

Fig. 8. Used bandwidth, in % of unicast bandwidth

The host zone bandwidth numbers stay the same throughout all topologies and unicast conversion threshold settings. This is expected since traffic originating from hosts is unaffected by both parameters. Exceptions to this are the unicast measurements, where all streams are duplicated directly on the host nodes, which obviously increases the outgoing bandwidth. The number of flow rules in the host zone is always zero since the hosts are not part of the OpenFlow domain.

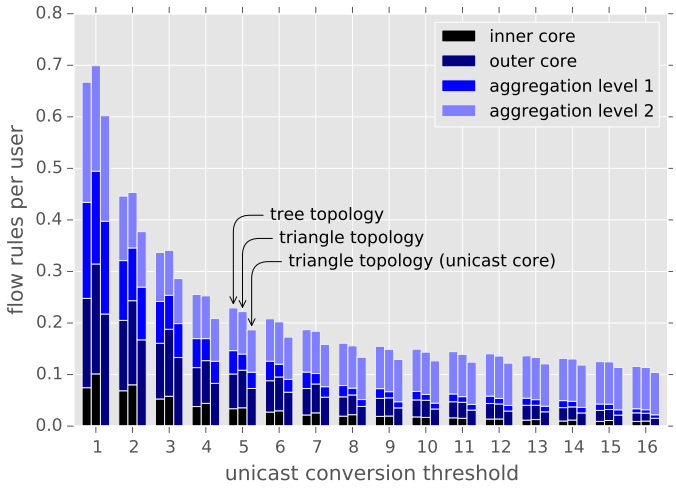

Fig. 9. Flow rules consumption per user

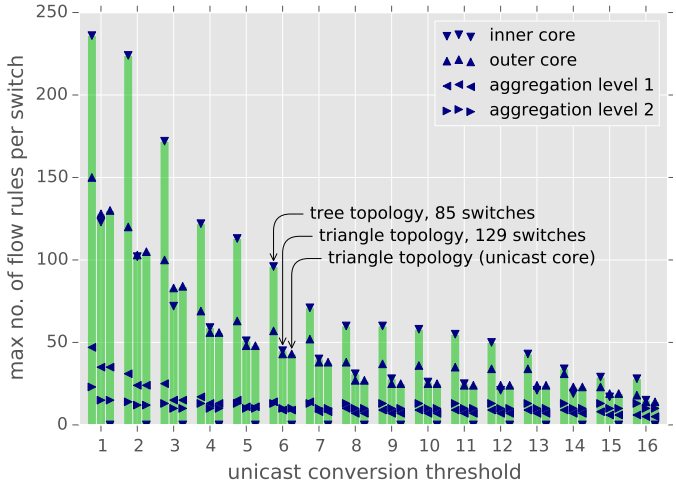

Fig. 10. Peak number of flow rules per switch

Modifying the unicast conversion threshold does not only change the number of flow rules installed into the network, but also their distribution. When the threshold is set to $t = 1$, about a third of the flow rules in the network are installed on switches located in the second aggregation level, for $t = 16$ this number increases to 75%. This effect is a result of the early duplication strategy, as it incrementally shrinks the multicast trees in the direction of the entry nodes. While the number of flow rules is reduced in other zones, the switches in the second aggregation level are entry nodes for most multicast groups, and therefore must maintain at least one flow rule per group they are responsible for. This distribution is advantageous since the flow rules are moved away from the high bandwidth network core to sections with lower traffic density. Across topologies, the bandwidth and flow rule numbers vary slightly due to the different network structures. When multicast forwarding is disabled at the inner core of the triangle topology, the flow rules that would be installed in the core are traded for additional bandwidth consumption, with minor flow rule increases in the outer core zone.

Since the maximum size of OpenFlow tables is fixed in hardware switches, it is also necessary to analyze the peak number of flow rules installed in switches for the purpose of network capacity planning. Figure 10 depicts the maximum number of flow rules installed on at least one switch in the different network topologies. Additional markers on each bar show the maximum values separately for each network zone. The per-switch peak flow rule numbers vary strongly with changes in network topology. At $t = 1$, the peak number of flow rules in the tree topology is about 84% higher than in the triangle topology, despite having the same workload. This is caused by the lower number of switches in the tree topology (85 switches) compared to the triangle topology (129 switches). Since the number of multicast groups is the same in both topologies, the flow rules implementing the multicast groups are distributed on fewer switches, which lead to higher per-switch numbers. The most flow rules accumulate at the single root switch of the tree topology, where the majority of multicast groups pass through. A similar effect is seen in the triangle topology, where core switches have higher flow rule numbers than aggregation switches. When multicast forwarding is switched off in the inner core of the triangle topology, the number of flow rules in the inner core drops to zero as expected. The numbers for the other zones do not change significantly, as the removed flow rules are traded for additional bandwidth consumption instead. The peak flow rule numbers decrease with increasing values of the unicast conversion threshold. At $t = 16$, the required flow rule capacity is reduced to 12% of the capacity that would be needed for the late duplication strategy, across all measured topologies. This is highly advantageous considering the limited flow rule capacity of OpenFlow hardware switches.

To illustrate the effect of ASDM on a Zipf-like multicast group size distribution, the number of flow rules per group member for varying group sizes is depicted in Figure 11. When compared to classic approaches that solely rely on late duplication strategies with $t = 1$, ASDM can significantly reduce flow table space requirements, even with moderate increases of t. This is especially true for small group sizes which are dominant in Zipf-like group size distributions. For example, with $t >= 8$, flow table space consumption for groups with 8 members is reduced by 80% when compared to the implementation with $t = 1$.

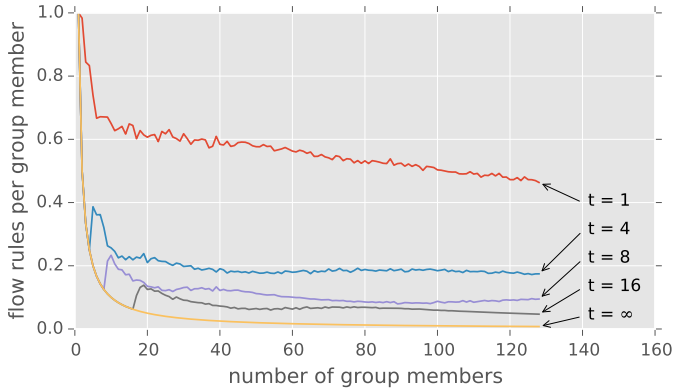

Fig. 11. Flow rules per group member

V. Conclusion

This paper proposes a new approach called ASDM that enables ISPs to offer controllable and efficient on-demand network layer multicast services for typical multicast applications with Zipf-distributed group sizes. In contrast to existing network layer multicast approaches, ASDM enables the ISP to dynamically adjust the bandwidth-state tradeoff of the multicast system, depending on its requirements.

The evaluation results show that the resource consumption required by multicast services can be optimized by choosing an appropriate forwarding strategy. The system can be adjusted for minimal bandwidth consumption, minimal flow table space consumption, or a balance between the two by setting a single parameter value. This bandwidth-state tradeoff can be adjusted by ISPs depending on the costs associated with the different resources, allowing them to choose the most cost-effective configuration for a given multicast workload and network topology. For the evaluated scenarios, the bandwidth reduction can be selected in the range from nearly $60\%, t = 1$ to slightly less than $30\%, t = 16$ compared to unicast traffic, with the flow rule consumption as the dependent variable. The flow rules consumption per user can be set to values be between $0.1, t = 1$ and $0.7, t = 16$ with the bandwidth usage as the dependent variable. It is shown that for a given ISP-defined bandwidth-state cost function, the optimal parameter setting for ASDM can be derived and directly applied. For scenarios investigated in this work, the selection of the most efficient usage of network state capacity leads to a 30% bandwidth reduction compared to unicast transport while using only a seventh of the network state compared to classic multicast with late duplication. Furthermore, the results also provide information about the peak flow rule load in the different zones of a network. Together with the control facilities provided by the system, this can help ISPs to plan ahead for the resource usage caused by multicast services. The use of unicast network forwarding allows using ASDM in networks with partial OpenFlow support. The results also show that with ASDM selected areas can be designated unicast-only as required.

Research opportunities for future work include an extension of ASDM that enables the system to configure the unicast conversion threshold based on the product of stream bandwidth and the residual tree size. Furthermore, the system behavior under dynamically changing workloads should be investigated with respect to the systems' reaction latency and the number of flow modifications needed to reconfigure the network. Finally, ASDM should be compared to the developing approach of bit index explicit multicast as discussed in Section II.

Acknowledgments

This work has been funded in parts by the European Union (FP7/#317846, SmartenIT and FP7/#318398, eCOUSIN) and the German Research Foundation (DFG) as part of project C03 within the Collaborative Research Center (CRC) 1053 – MAKI.

References

[1] J.-H. Cui, J. Kim, D. Maggiorini, K. Boussetta, and M. Gerla, "Aggregated Multicast – A Comparative Study," *Cluster Computing*, vol. 8, no. 1, pp. 15–26, 2005.

[2] R. Boivie, N. Feldman, Y. Imai, W. Livens, and D. Ooms, "Explicit Multicast (Xcast) Concepts and Options," IETF, RFC 5058, 2007.

[3] I. Stoica, T. Ng, and H. Zhang, "REUNITE: a Recursive Unicast Approach to Multicast," in *IEEE INFOCOM*, 2000.

[4] C. Diot, B. Levine, B. Lyles, H. Kassem, and D. Balensiefen, "Deployment Issues for the IP Multicast Service and Architecture," *IEEE Network*, vol. 14, no. 1, pp. 78–88, 2000.

[5] M. Hosseini, D. T. Ahmed, S. Shirmohammadi, and N. D. Georganas, "A survey of application-layer multicast protocols." *IEEE Communications Surveys and Tutorials*, vol. 9, no. 1-4, pp. 58–74, 2007.

[6] 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.

[7] L. A. Adamic and B. A. Huberman, "Glottometrics," *Glottometrics*, vol. 3, no. 1, pp. 143–150, 2002.

[8] K. Sripanidkulchai, B. Maggs, and H. Zhang, "An Analysis of Live Streaming Workloads on the Internet," in *ACM IMC*, 2004.

[9] E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol Label Switching Architecture," IETF, RFC 3031, 2001.

[10] A. Boudani and B. Cousin, "A new Approach to Construct Multicast Trees in MPLS Networks," in *IEEE ISCC*, 2002.

[11] G. Apostolopoulos and I. Ciurea, "Reducing the Forwarding State Requirements of Point-to-Multipoint Trees Using MPLS Multicast," in *IEEE ISCC*, 2005.

[12] P. I. Radoslavov, D. Estrin, and R. Govindan, "Exploiting the Bandwidth-Memory Tradeoff in Multicast State Aggregation," University of Southern California, Tech. Rep. TR99-697, 1999.

[13] J. Blendin, "Cross-layer Optimization of Peer-to-Peer Video Streaming in OpenFlow-based ISP Networks," Diploma Thesis, Technische Universität Darmstadt, 2013.

[14] I. Wijnands, E. Rosen, D. A., T. Przygienda, and S. Aldrin, "Multicast using Bit Index Explicit Replication," IETF, Internet-Draft, 2015. [Online]. Available: https://tools.ietf.org/id/draft-wijnands-bier-architecture-04.txt

[15] A. B. Bondi, "Characteristics of Scalability and Their Impact on Performance," in *Workshop on Software and Performance*, 2000.

[16] T. Volk, "Supporting Multicast in Application-controlled Software Defined Networks," Masters' Thesis, Technische Universität Darmstadt, 2014.

[17] V. G. Cerf, "2012 Isn't the End of the World," *IEEE Internet Computing*, vol. 14, no. 6, pp. 95–96, 2010.

[18] J. Kim, N. Sarrar, and A. Feldmann, "Watching the IPv6 Takeoff from an IXP's Viewpoint," Technische Universität Berlin, Tech. Rep. 2014-01, 2014.

[19] M. R. Garey, R. L. Graham, and D. S. Johnson, "The Complexity of Computing Steiner Minimal Trees," *SIAM Journal on Applied Mathematics*, vol. 32, no. 4, pp. 835–859, 1977.

[20] A. Gupta, L. Vanbever, M. Shahbaz, S. P. Donovan, B. Schlinker, N. Feamster, J. Rexford, S. Shenker, R. Clark, E. Katz-bassett, G. Tech, and U. C. Berkeley, "SDX : A Software Defined Internet Exchange," in *ACM SIGCOMM*, 2014.

[21] M. Düser and A. Gladisch, "Evaluation of Next Generation Network Architectures and Further Steps for a Clean Slate Networking Approach," in *ITG EuroView*, 2006, Presentation.

[22] D. Y. Huang, K. Yocum, and A. C. Snoeren, "High-fidelity Switch Models for Software-defined Network Emulation," in *ACM HotSDN*, 2013.

[23] X. Meng, Z. Xu, B. Zhang, G. Huston, S. Lu, and L. Zhang, "IPv4 Address Allocation and the BGP Routing Table Evolution," *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 1, pp. 71–80, 2005.

[24] G. Maier, A. Feldmann, V. Paxson, and M. Allman, "On Dominant Characteristics of Residential Broadband Internet Traffic," in *ACM IMC*, 2009.

[25] N. Handigol, B. Heller, V. Jeyakumar, B. Lantz, and N. McKeown, "Reproducible network experiments using container-based emulation," in *ACM CoNEXT*, 2012.