OpenFlow Rule Placement In Carrier Networks For Augmented Reality Applications

Rafael George Amado
École de technologie supérieure
University of Quebec
Montreal, Canada
rafael-george.amado.1@ens.etsmtl.ca

Kim-Khoa Nguyen
École de technologie supérieure
University of Quebec
Montreal, Canada
kim-khoa.nguyen@etsmtl.ca

Mohamed Cheriet
École de technologie supérieure
University of Quebec
Montreal, Canada
mohamed.cheriet@etsmtl.ca

ABSTRACT

Today, mobile consumers increasingly use Augmented Reality (AR) devices to stream personal video through carrier networks. Thanks to its flexibility, Software-Defined Networking (SDN) is deployed in many carrier networks to support end-to-end network-slicing, which is substantial for these AR applications. In an OpenFlow-enabled SDN network, a controller must decide the rules to be placed into the switches in the network, subject to multiple constraints such as memory capacity, link bandwidth limitation, and flow continuity. Due to legacy switch models, prior work focuses only on unicast flows which cannot efficiently support AR applications with streaming traffic from one source device (or server) to many destination devices across the network. In this paper, we optimize rule placement in resource-constrained Openflow networks for both unicast and multicast flows. Our approach is to leverage the use of Group Tables, which is recently introduced in the OpenFlow 1.1 specification, to support multicast flows and, at the same time, save switch memory. Traffic to multiple destinations can be aggregated to match a single flow table entry per switch. Therefore, significant link resources can be saved. The experimental results on three different topologies show our solution can support a higher number of flows than the state-of-the-art solutions by reducing both the link usage by up to 30% and the number of flow entries needed to deliver the traffic to destinations by 22%.

CCS CONCEPTS

• **Networks** → **Programmable networks**;

KEYWORDS

augmented reality, software-defined networks, routing policy optimization, multicast routing, rule placement, openflow, group table

ACM Reference Format:
Rafael George Amado, Kim-Khoa Nguyen, and Mohamed Cheriet. 2022. OpenFlow Rule Placement In Carrier Networks For Augmented Reality Applications. In *The 37th ACM/SIGAPP Symposium on Applied Computing (SAC '22), April 25–29, 2022, Virtual Event*. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3477314.3507101

SAC '22, April 25–29, 2022, Virtual Event

ACM ISBN 978-1-4503-8713-2/22/04…$15.00
https://doi.org/10.1145/3477314.3507101

1 INTRODUCTION

Recent AR applications allow mobile devices in the same physical environment to share different points of view on the same event. At the same time, crowdsource applications leverage collaborative tasks such as annotating the physical world. In such applications, users send and receive traffic from both an AR application server and all users in the same session [20]. As an example, is a teacher in a museum streams to students who can remotely watch a live video and, at the same time, receive information about the objects, such as paintings and sculptures. The video is a multicast connection, but the information might be retrieved using unicast connections from an annotation server in the network. Thus, this application requires both unicast and multicast flows flexibly routed in the network according to the mobility of the streamer and the online demand of the receivers.

Nonetheless, due to legacy switch models, most of existing network architectures focus only on IP multicast [9], which is complex by design due to the nature of the protocols involved, such as Internet Group Management Protocol (IGMP) and Protocol Independent Multicast (PIM) [6]. OpenFlow-based multicast has been implemented in state-of-the-art research using Flow Table entries [12], which is expensive since each destination on the multicast group needs one Flow Table entry. The Group Table has been implemented in OpenFlow 1.1, but only recently available hardware has been released with support for newer versions of OpenFlow [5]. Since prior work has not yet taken advantages of new released OpenFlow hardware with Group Table support, they cannot efficiently support AR applications with multicast traffic from one device (or server) to many devices across the network.

A key enabler of the dynamic networking service provisioning required for AR applications is Software Defined Networks (SDN) technology which is essential to accommodate the demands of AR applications [19]. SDN is designed to allow end-to-end support for network slicing. OpenFlow, in particular, can install the rules in the switches so that high-level policies such as the ones required by AR applications can be enforced in a real network, with limited resource capacity. A higher abstraction level hides the complexity of the network devices and exposes a simple interface to the operators. However, this flexibility brings the complex task of allocating the low-level rules in the actual network, which requires handling constraints such as limited available memory on the switches, as well as the capacity of the links [17]. At the same time, the available resources are very scarce, mainly due to memory limitations on OpenFlow switches [14]. Furthermore, since each user can request more than one type of traffic, the flows might have different requirements, augmenting the granularity required and, thus, increasing the number of flow entries to be installed. The flow allocation problem is increasingly complex, with the ever-growing number of users generating streaming traffic to multiple receivers, exponentially expanding the number of rules required in the network.

Contributions

The main contributions of this paper are summarized as follows:

- We leverage OpenFlow Group Table for multicast in the network to reduce the number of flow entries in the switches, increasing the throughput and the number of flows that can be installed.
- We formulate a non-linear integer optimization model for allocating forwarding rules for both unicast and multicast in OpenFlow-based networks, and solve this problem using a traditional optimization method.
- We design an algorithm to solve the optimization problem in a timely manner, then carried out experiments in both simulated networks and a nationwide testbed of Telus-Ciena Lab in Canada.

The remainder of this paper is organized as follows: Section 2 shows some related work, Section 3 describes the system and the scenario proposed. Section 4 presents the problem formulation and solution, Section 5 shows the numerical results, followed by a conclusion.

2 RELATED WORK

While most AR-related works focus on the devices, some propose solutions for the networking issues raised by the requirements of such applications. The authors in [3] investigate multicast AR streams over multiple wireless access points to multiple users. Focusing on tiled panoramic content, [15] proposes a solution to improve users received video quality and reduce battery consumption for small user groups. [1] proposes a probabilistic approach based on the user's channel conditions and applies a rate adaptation algorithm for mobile multicast environments. Focusing on device-to-device (D2D) communications, [16] investigates the appropriate bitrate for the multicast stream from an AR server sending traffic via WiFi to users in the service area. [21] studies Augmented-Reality Multiview-video (ARM) stream scheduling for vehicle-pedestrian situations in small cell networks and proposes a hybrid multicast/unicast D2D architecture that minimizes small cells bandwidth consumption.

However, none of the recent works study the problem from a Software Defined Networks perspective. Our proposed model tackles the AR streams from the network's point of view, formulating a rule placement scheme in OpenFlow switched networks. A rule placement solution defines which rules must be deployed and where to deploy them in the network. Also, each operator can define policies to be enforced in the network. The *endpoint policy* defines where to deliver packets, and the *routing policy* indicates the paths that the flows must follow before being delivered [10]. Many applications can benefit from OpenFlow, especially the Traffic Engineering problem [8]. In this case, the challenge relies on selecting the paths to install the rules that simultaneously satisfy the network policies and constraints.

The problem of flow allocation has been addressed by [17] which relaxes the routing policy and uses a default path towards the controller to allocate the rest of the traffic that cannot be installed. The authors of [4] tackle the congestion problem in flow tables by reducing the number of flow rules for an OpenFlow switch proposing a flow-rule-reduction algorithm. However, they focus on minimizing the memory of the flow table for unicast flows only and do not consider the Group Table in the problem. On the other hand, prior work considers multicast in the context of IP [11]. This approach cannot deliver the dynamic changes required by the AR applications due to the complexity and high-signaling nature of the protocols involved, such as Internet Group Management Protocol (IGMP).

Recently, studies have proposed different approaches for flow rule placement in SDN. [2] presents a mobility-aware flow-rule placement scheme in Software-defined Access Networks (SDAN) to support IoT applications reducing the overall cost in the flow rule placement. The rule placement for traffic engineering was considered by [7]. The authors propose to impose the constraints on the number of routing paths when optimizing flow routing to reduce the number of flow entries in the Ternary Content Addressable Memory (TCAM).

In this paper, we propose a flow allocation model for unicast and multicast flows. It can increase the throughput and number of flows installed using Group Tables to distribute the AR-generated streaming traffic to multiple destinations, reducing overall memory consumption per destination. The objective is to find the optimal placement of Group Table entries for the multicast flows and Flow Table entries (responsible for matching the traffic) for the unicast traffic. The combined placement of both types of rules is essential since the Group Table will receive the packets from the matched Flow Table rule and replicate them to several different ports towards the egress points.

3 SYSTEM DESCRIPTION

Control Plane

We illustrate our system in Figure 1, where a centralized controller calculates where to install entries into Flow and Group Tables of all switches in a backhaul. The controller exposes a Northbound Interface (NBI) where high-level policies and applications can be implemented, and via the Southbound Interface (SBI), it manages the switches through OpenFlow messages. The Topology Manager is an application that uses LLDP packets to discover the topology. With this information, the controller can create an abstracted topology graph where the policy enforcement decisions will be made.

In the example shown in Figure 1, an endpoint policy is enforced by the controller to guarantee the end-to-end bandwidth of a collaborative AR application. By the use of AR technology, it allows several users to collaborate over the same video-stream traffic generated by the server. When the users want to use the AR application, the controller installs flow entries into the switches to deliver the video-stream traffic from the server across the network. Each user can interact within the same scene since several users can participate in the same session and insert overlay information over the video in real-time. All users in the session receive the original video from the server (clean Eiffel Tower on the left-hand side)

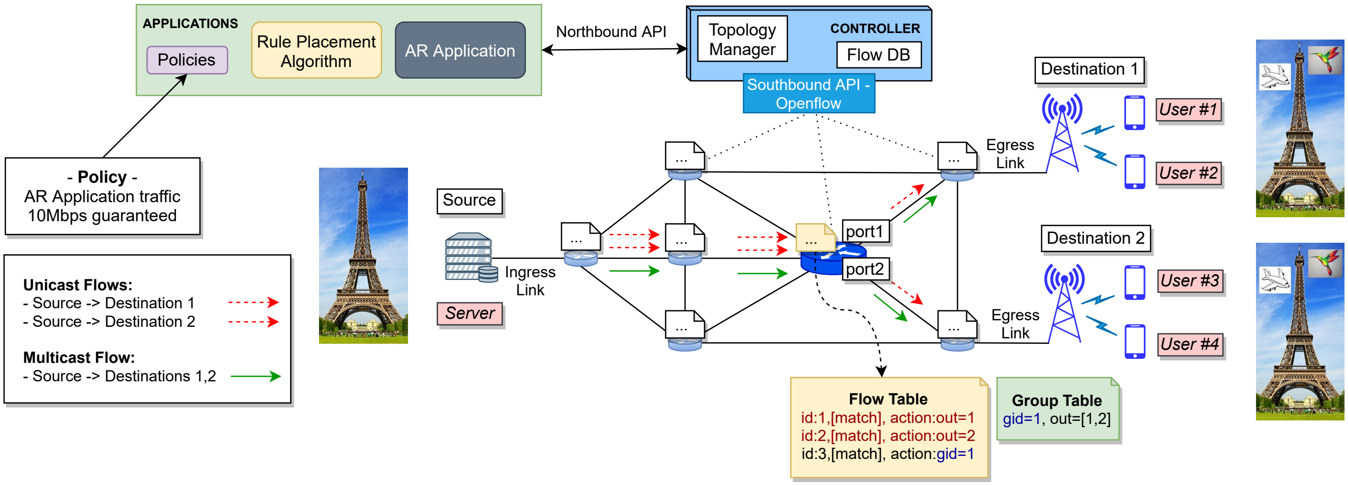

Figure 1: Interactive AR application streaming video to multiple users

and the interactions generated by each user. The user interactions are replicated by the server to the other participants, as illustrated by the inserted elements in the right-hand side of Figure 1. The traffic source is a server connected to one aggregation switch in the network via an ingress link. The egress links connect the aggregation switches to the destination Base Station (BS) to which up to 10 users are simultaneously attached. Since the Policy requires a guaranteed bandwidth for each user, we assume the egress links are the central limit for the number of users attached to the BS. The controller receives a request from the AR application and installs the flow entries to the switches within the network, i.e., from one ingress link to one or more egress links. If only unicast is supported by the AR application, each pair (source, destination) requires one entry in the Flow Table of every switch along the path, as shown by the red dashed arrows. Fortunately, Openflow Group Table allows the controller to make a more scalable decision by replicating traffic towards each destination. In other words, a single flow entry is added to the Group Table of each switch, as shown by the solid green arrow. Such a mechanism is very efficient in video multicasting scenarios with a large number of receivers. Ideally, the forwarding rules that install a flow should be placed in the switches along the shortest path to the destination. In the actual network, switch memory is a limited resource, and it might not be possible to install all the necessary flow entries into the shortest paths. Longer paths mean a higher use of the links whose capacity is also limited. Thus, choosing the right nodes to receive the flow entries is crucial, allocating paths that satisfy both the endpoint and routing policies under the network constraints.

Group Table Placement

The specification of an OpenFlow Logical Switch [18] states that a switch may have one or more Flow Tables and one Group Table. The former matches the packet headers, and the latter can be used to forward multicast traffic (Group Type 'ALL'). Group Tables receive the packets from the Flow Tables when the flow that matches the packets has an action *Group*, and the Group ID indicates the group pointed by the flow entry. Depending on the implementation, the size of the Flow and Group Tables can be either fixed or defined during the compilation process when the switch has a programmable data plane. Since each flow entry can only send traffic to one output port, we can use the Group Table type 'ALL' to forward this traffic if the same matched flow needs to be sent to multiple ports. They can be a powerful tool to reduce the number of table entries since the flows with multiple destinations (i.e., multiple output ports need to be assigned) can be installed using a single entry in the Group Table instead of one entry per output port.

The optimized placement of flow entries in Group Tables of switches in a network is challenging, as illustrated in an example in Figure 2. This example shows a network composed of 7 OpenFlow switches and a list of 2 flows with a total bandwidth of 6 to be routed. Each flow has requested bandwidth, ingress, and egress points. For the sake of simplicity, we denote the ingress and egress links as the switches to which they are connected. In this example, every link has a maximum capacity of 5, each Flow Table stores at most five flow rules, and the Group Tables, two entries at most. Each thick line denotes a flow entry combined with a group entry. The numbers next to the lines refer to the flow being allocated to that specific rule entry. On the left-hand side, multicast flow #1 from A toward F and G is allocated in a Group Table of switch A, replicating the traffic to links A-C and A-B. Since the destinations of flow #1 are F&G, flow entries are installed into intermediate switches B, C, and E, as denoted by the number 1 in their tables. Totally, the option of placing a group table entry of flow #1 at switch A results in 4 entries and 15 bandwidth in the network. On the right-hand side, another option of placing the entries of the same flow #1 in the switches is shown. At first, flow #1 is sent through link A-C with a simple entry in the flow table of switch A. Then, a Group Table entry is inserted into switch C, which replicates the traffic to links C-F and C-G. Totally, the option of placing a group table entry of flow #1 at switch C results in 2 entries and 9 bandwidth in the network. A similar result can be achieved for flow #2 by choosing the Group Table placement on switches D or E.

This example shows that by placing the Group Table entry for flow #1 into switch C, more traffic can be allocated in the network

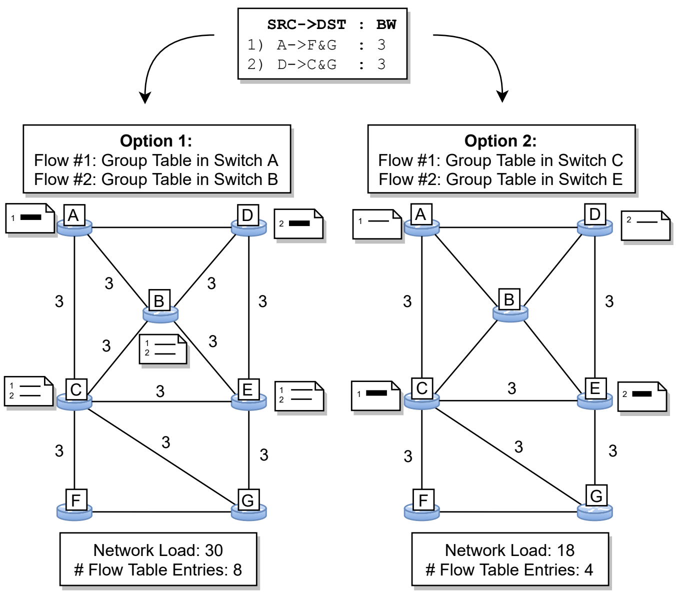

Figure 2: Group Table placement can increase the efficiency of the network.

since links B-E and E-G remain free. Therefore, the optimal placement of the Group Table entries can maximize the number of flows in the network and, at the same time, save resources.

4 PROBLEM FORMULATION

We model our network as a undirected graph $V(S, L)$, where S is the set of switches and L is the set of links, with capacity B_l, interconnecting them. A set of flows F needs to be mapped into flow rules in the switches in the network. Each flow $f \in F$ specifies a packet rate p_f, a source $l_f \in I$ which is the ingress link connecting the AR application server to the Aggregation switch where the packet enters the backhaul network. The same flow $f \in F$ has one or more destinations $d \in E(f)$ ($E(f) \subseteq E$), which are the egress links where the packets leave the backhaul network to one or more destination Base Stations (BS). $L^+ = L \cup I \cup E$ is the set of all directed links. Each switch $s \in S$ has a Flow Table and a Group Table with limited capacity, respectively, C_s and G_s. The Flow Table performs the matching of packets for each flow $f \in F$ and forwards them to a specific port towards the destination $d \in E(f), |E(f)| = 1$. When a flow has more than one destination (multicast), the action in the Flow Table is to steer the packets to the Group Table, which clones the packets and send them to the list of ports towards the egress links $d \in E(f), |E(f)| > 1$. In this case, two entries are necessary: one in the Flow Table, where the matching happens, and the other in the Group Table, where the action is performed (clone packets and send them out). Table 1 summarizes the notations used in this paper.

We present the placement solution by two matrices of binary variables: $A = (a^d_{f,l})$, where $a^d_{f,l} = 1$ when a flow f, with destination $d \in E(f)$, passes through the directional link $l = (u, v)$; $u, v \in S^+$ from node u to node v; and $G = (g_{f,s})$, where $g_{f,s} = 1$ when a flow f allocated to switch s is also put in its Group Table.

The allocation matrix is a source of information for an operator as it provides at the same time the forwarding table, switch memory occupation, and link usage for a given high-level objective and endpoint policy.

Constraints (1) and (2) verify that $a^d_{f,l}$ and $g_{f,s}$ are binary variables.

$$\forall f \in F, \forall l \in L^+, \forall d \in E(f) : a^d_{f,l} \in \{0, 1\} \tag{1}$$

$$\forall f \in F, \forall s \in S : g_{f,s} \in \{0, 1\} \tag{2}$$

Bandwidth constraint (3) ensures that the sum of all flows (for all destinations) allocated to link l does not exceed its capacity B_l. Two terms are used here because when a flow has more than one destination, but the Group Table is not allocated, all $d \in E(f)$ sharing link l will consume one unit of packet rate p_f.

$$\forall s \in S, \forall l \in L^+ :$$
$$\sum_{f\in F}\sum_{d\in E(f)} \frac{p_f a^d_{f,l}}{|E(f)|}(1 - g_{f,s}) + \sum_{f\in F}\sum_{d\in E(f)} p_f a^d_{f,l} g_{f,s} \leq B_l \tag{3}$$

We also need to make sure not to exceed the memory limitation of each switch s. Constraints (4) indicates that all destinations $d \in E(f)$ will match the same flow table entry, while (5) is the Group Table capacity constraint.

$$\forall s \in S : \sum_{d\in E(f)}\sum_{v\in N^{\leftarrow}(s)}\sum_{f\in F} \frac{a^d_{f,l}}{|E(f)|} \leq C_s \tag{4}$$

$$\forall s \in S : \sum_{f\in F} g_{f,s} \leq G_s \tag{5}$$

Constraint (6) indicates that packets belonging to flow f will only traverse their ingress link l_f.

$$\forall f \in F, \forall d \in E(f) : a^d_{f,l} = \begin{cases} 0 & \text{if } l \in I \setminus \{l_f\} \\ 1 & \text{if } l = l_f \end{cases} \tag{6}$$

Flow conservation constraints assure that the incoming traffic of switch s leaves through the egress links. While (7) is the usual flow conservation constraint used for the unicast flows, (8) has a per-destination approach which is necessary since, in a multicast transmission, one incoming packet at an intermediate node might produce one or more outgoing packets, which violates the flow conservation principle.

$$\forall f \in F, \forall s \in S :$$
$$\sum_{d\in E(f)}\sum_{v\in N^{\rightarrow}(s)} (1 - g_{f,s}) p_f a^d_{f,(v,s)} - \sum_{d\in E(f)}\sum_{v\in N^{\leftarrow}(s)} (1 - g_{f,s}) p_f a^d_{f,(s,v)} = 0 \tag{7}$$

$$\forall f \in F, \forall s \in S, \forall d \in E(f), \forall v \in N^{\leftarrow}(s) :$$
$$\sum_{u\in N^{\rightarrow}(s)} g_{f,s} p_f a^d_{f,(u,s)} - g_{f,s} p_f a^d_{f,(s,v)} = 0 \tag{8}$$

Table 1: Notation

Notation	Description
F	Set of flows.
p_f	Packet rate of flow $f \in F$.
S	Set of Openflow switches composing the network.
C_s	Flow Table capacity of switch s.
G_s	Group Table capacity of switch s.
S_e	Set of external nodes directly connected to the network.
S^+	Set of all nodes ($S^+ = S \cup S_e$).
$N^{\rightarrow}(s) \subseteq S^+$	Set of incoming neighboring nodes of switch $s \in S$ (i.e., neighbors from which s can receive packets).
$N^{\leftarrow}(s) \subseteq S^+$	Set of outgoing neighboring nodes of switch $s \in S$ (i.e., neighbors towards which s can send packets).
L	Set of directed links, defined by $(u, v) \in S \times S$, where u is the origin of the link and v is its termination.
I	Set of directed links connecting the AR application servers to the aggregation switches (ingress links). The ingress link of a flow $f \in F$ is written l_f by abuse of notation.
E	Set of directed links connecting the aggregation switches to the destination Base Stations (egress links).
$E(f) \subseteq E$	Set of egress links for flow $f \in F$ according to the endpoint policy.
L^+	Set of all directed links (i.e., $L^+ = L \cup I \cup E$).
B_l	Capacity of link $l \in L^+$.
Variables	**Description**
$a^d_{f,l}$	Equals 1 if flow f to destination d passes through link l, 0 otherwise.
$g_{f,s}$	Equals 1 if flow f allocated to switch s is also put in its group table.

As described in Equation 9, the objective function aims to maximize the volume of traffic satisfying the Endpoint Policy, and it is NP-hard [17].

$$\text{Maximize} \quad \mathbb{F}(A, F, E) = \sum_{f \in F} \sum_{l, d \in E(f)} a^d_{f,l} p_f \tag{9}$$

where p_f is the packet rate of flow $f \in F$.

OpenFlow Multicast Allocation Algorithm

As discussed above, the optimization problem (Equation 9) is NP-hard, so the optimal solution cannot be obtained in real-time when the number of flows and nodes in the network is large. Therefore, we design an algorithm named OFMAA (Algorithm 1) which aims to install the flows with higher packet rate p_f, so we can approximate the optimal goal of maximizing the traffic that satisfies the Endpoint Policy.

Line 5 sorts the flows according to their packet rate, starting with the flows with the highest requested bandwidth. The algorithm follows a greedy approach in the sense that it tries to allocate larger flows first and fills the remaining resources with smaller flows.

Algorithm 1 OpenFlow Multicast Allocation Algorithm

1: **INPUT**: Set of flows F, set of switches S, and set of links L^+
2: **OUTPUT**: Allocation matrices $A = (a^d_{f,l})$ and $G = (g_{f,s})$
3: $A \leftarrow [0]_{F,L^+,E(f)}$
4: $G \leftarrow [0]_{F,S}$
5: $M \leftarrow \texttt{sort}(F, p_f, descending)$
6: **for all** $f \in M$ **do**
7:   $P(f) \leftarrow \texttt{paths}(l_f, E(f))$
8:   **if** $\texttt{groupTableNode}(P(f))$ **then**
9:     $\texttt{allocate}(a^d_{f,l}, g_{f,s})$
10:     $\texttt{update}(A, G)$
11:     **break**
12:   **end if**
13: **end for**

Based on the Depth-First Search (DPS) algorithm, the function $\texttt{paths}(l_f, E(f))$ finds all the single paths, with enough available capacity for p_f, from the ingress link l_f to each egress link $d \in E(f)$ of flow f and stores them in $P(f)$. Line 8 calls the function `groupTableNode`, which receives set of paths $P(f)$ and checks if there is one node s that intercepts the paths to all the destinations. If so, the function $\texttt{allocate}(a^d_{f,l}, g_{f,s})$ installs the Group Table entry in the node s by setting $g_{f,s} = 1$. It also installs flow table entries (setting $a^d_{f,l} = 1$) from the source l_f to the node s, as well as from s to all $d \in E(f)$ by choosing the paths with minimum average load over all links. Finally, the allocation matrices A and G are updated and the process is repeated for every flow in M. The asymptotic time complexity of the OFMAA is driven by the loop in line 6 and runs in $O\left(|M| \cdot (|S| + |L^+|)\right)$.

The outputs of OFMAA are the allocation matrices A and G. The matrix A provides the information necessary for defining Flow Table entries since it informs which links each flow will pass through. Likewise, the matrix G shows the switches in which the Group Tables must be installed for each flow. The operator can quickly generate FLOW_MOD messages from the combination of both matrices to push the appropriate rules in each switch.

5 NUMERICAL RESULTS

We evaluate the performance of the proposed model in two simulated topologies: a 10-node mesh topology, the NSF network (depicted in Figure 3) both with equal link capacity and also on a nationwide testbed at TELUS Lab in Edmonton, AB, which is a packet-optical network with SDN capabilities. Telus Lab link capacities are depicted in Figure 4. All three topologies were assessed with unidirectional links, except for Telus topology, where we added one bidirectional link per layer to provide more options for the Group Table placement.

We compare our OFMAA solution with two baselines: i) a prior work that optimally allocates OpenFlow rules with no multicast consideration, and ii) an optimal solution of the model in Equation (9) obtained by a mathematical solver. We generate a set of flows from two different ingress points and four different egress points in each topology. They are ordered by decreasing requested bandwidth

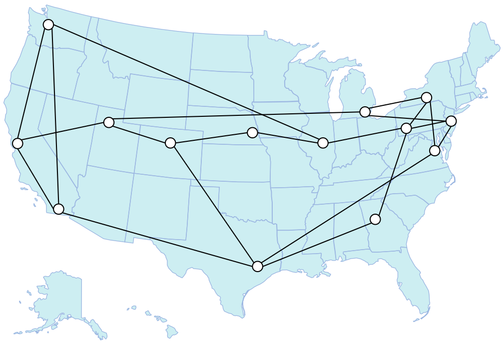

Figure 3: NSFNET - The National Science Foundation Network [13]

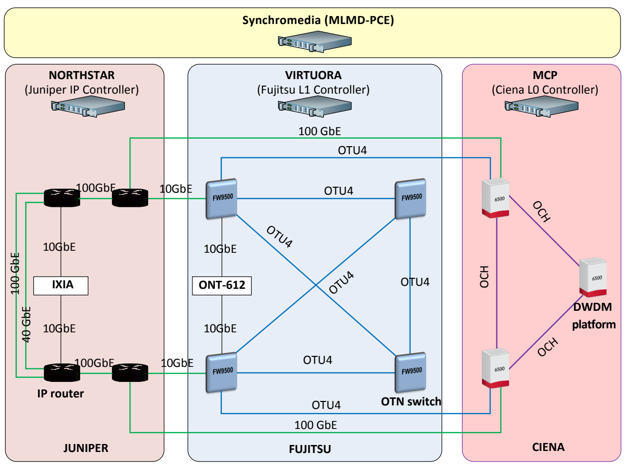

Figure 4: Testbed at TELUS-Ciena Laboratory

with the highest values up to 5x the lowest ones to have flows of different sizes to allocate.

The flows are mixed multicast and unicast to reflect the nature of AR application traffic traversing the network alongside other types of streaming data.

In order to compare the performance of our multicast model to the unicast approach, we create different scenarios where the ratio of multicast to the total requested traffic increases and, we compare different key performance indicators in all scenarios. The ratio increases when more users receive traffic from the AR application, thus demanding more resources from the network compared to other unicast traffic. We assume that the flow of any user is set up only if the bandwidth requirements (policy) are met. We show how our method consumes the network resources (links and flow table entries) in the different scenarios where the amount of multicast traffic traversing the network rises. We also compare our solution with a unicast-only model used to deliver the flows to the same egress points in each scenario.

We evaluate the Network-wide Average Link (NAL) utilization in the three topologies, and we compare the NAL when the different demands with multicast/unicast ratio are allocated using the optimization model, OFMAA (Algorithm 1) and the equivalent unicast-only scheme. The initial point in each curve is the lowest load, in which each flow is delivered to one destination only,

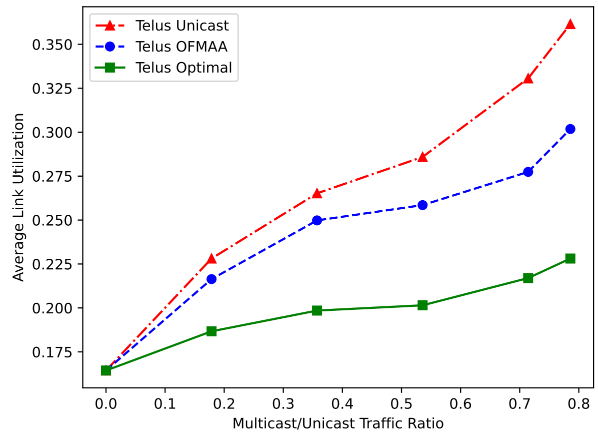

Figure 5: Network-wide Average Link (NAL) Utilization - Telus Topology

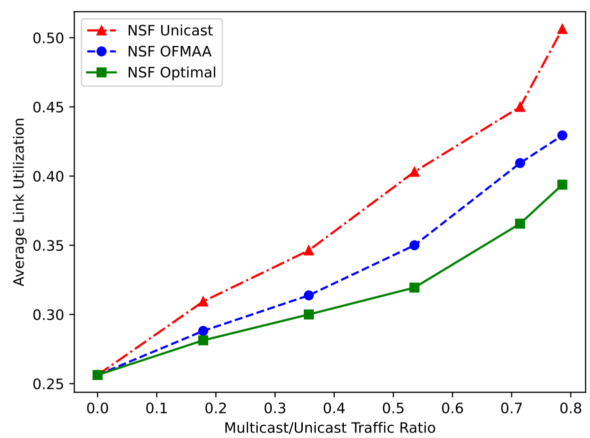

Figure 6: Network-wide Average Link (NAL) Utilization - NSF Topology

and they are different for each topology due to their design. For instance, the average link utilization in Telus topology (Figure 5) is the lowest because we added the bidirectional links increasing the number of links. So more options are available, and the flows can be more distributed across the layers and links. NSF is the more restrictive topology (fewer links available), so the link utilization is always higher than the other ones (Figure 6). Mesh topology NAL is illustrated in Figure 7. Though it has the highest number of links available, since they are all unidirectional, effectively there are less allocation options than the Telus network, so its NAL is slightly higher.

The results show a similar trend: increasing the number of multicast flows by adding egress points (a.k.a. destinations) increases the traffic ratio since these flows require more resources. However, the number of unicast flows also increases because a new flow is required for each new destination added. Nevertheless, when allocated using our model, the same flows require less link capacity overall. The path chosen by the optimization model will consume the same amount of resources for all destinations along the path from the source to the node where the Group Table is installed for that particular flow.

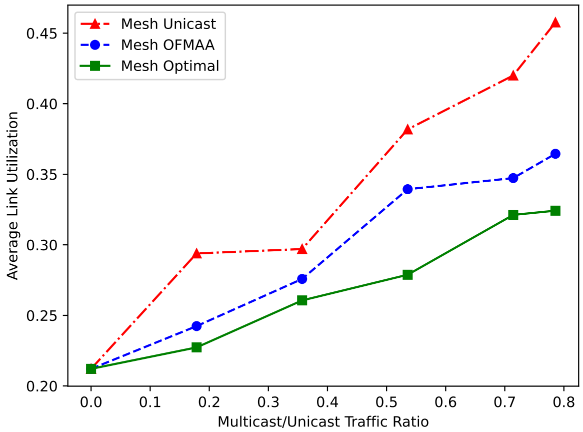

Figure 7: Network-wide Average Link (NAL) Utilization - Mesh Topology

Figure 8 shows the savings in link utilization when we allocate the flows optimally and with OFMAA when compared to the baseline unicast case. We can see that for the extreme case where almost 80% of the traffic is multicast, the optimal allocation can consume, on average, 30% less link capacity. The OFMAA algorithm, therefore, saves up to 17% for the same case. When we compare Figure 8 and the NAL for each topology, it is clear that for the NSF network, the reduction is lower due to the limited number of options for the Group Table (see Figure 6). Telus testbed, in turn, can benefit significantly, as displayed by the difference between the optimal and the unicast curves in Figure 5.

We also evaluate the increasing number of installed flow table entries necessary to deliver the traffic to one destination Base Station (egress link). As the number of destinations increases (and also bandwidth traversing the network), the proposed method consumes fewer flow entries than the unicast equivalent. This behavior

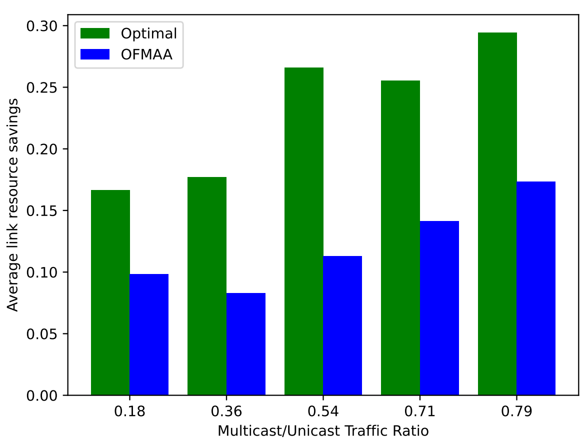

Figure 8: Link utilization reduction

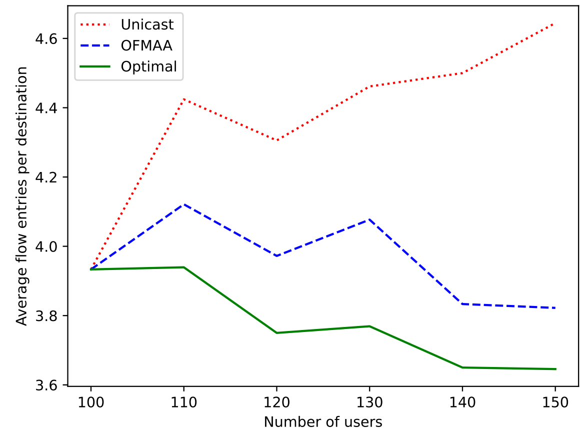

Figure 9: Average Flow Table entries per destination Base Station

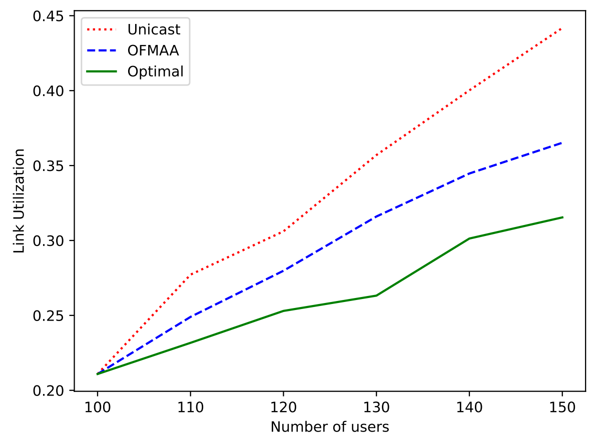

Figure 10: Link Utilization increases as the number of users grows

is possible because the model installs one flow entry per node along the path until the Group Table is installed. From there, the traffic is replicated to the different egress links. Figure 9 shows that, as the number of users in the application (and also the multicast/unicast traffic ratio) increases, our approach consumes, on average, a smaller number of flow table entries and, with this, saves memory in the switches.

Figure 10 illustrates the average link utilization growth as the number of users in the application increases. The proposed method shows a much lower slope in the curve, which means we can accommodate a much higher number of users under the same network when compared with the unicast baseline.

Finally, we calculate the number of flows required to afford a total demand in three cases: no multicast, allocating multicast using OFMAA, and optimally allocating multicast, as displayed in Figure 11. As the number of users increases, so grows the number of paths to be allocated to the increasing number of destinations and the number of flow table entries necessary to deliver the flows to the egress points. The proposed method consumes between 6% and 22%

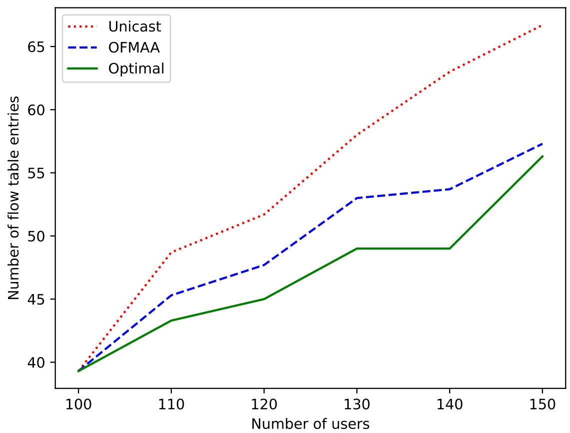

Figure 11: Flow Table entries - users

fewer flow table entries for the same demand than the unicast-only allocation.

Our results show that the proposed model and algorithm increase the network's overall efficiency, which can accommodate a higher number of receivers compared to the unicast-only solution. Furthermore, our solution supports both unicast, and multicast flows by design, making it suitable for the AR applications that share the same network with many other types of traffic.

6 CONCLUSIONS

This paper proposes a model that leverages the Group Table for AR-generated multicast traffic. Experimental results showed that our model could reduce the link utilization up to 30% in the extreme case where almost 80% of the traffic is multicast. We also showed that the number of flow table entries needed for each destination is reduced, which allows the accommodation of a higher number of users in the application.

However, we do not cover the strategies to handle the new arriving policies. One approach would be to enforce the new policy with the current ones, which is challenging because the constraints are the residual bandwidth and memory capacity. The placement will not be optimal, thus decreasing the overall efficiency. In contrast, when a new policy arrives, the controller might enforce the old and new policies as a set, which would result in better use of the overall resources, but it is more costly and can lead to traffic disruption. In this context, we could leverage the use of predictors (e.g., Machine Learning tools) based on the historical data about the incoming requests in an hourly fashion. With this, we can minimize the changes in the network, which will cause less disruption and save costs.

In the future, we will extend our work by studying the strategies mentioned above, and we will also consider multicast in different scenarios, such as software-defined radio.

ACKNOWLEDGMENTS

The authors thank NSERC, Ciena, and Telus for their support in this work.

REFERENCES

[1] Hamed Ahmadi, Omar Eltobgy, and Mohamed Hefeeda. 2017. Adaptive multicast streaming of virtual reality content to mobile users. In *Proceedings of the on Thematic Workshops of ACM Multimedia 2017*. 170–178.

[2] Samaresh Bera, Sudip Misra, and Mohammad S Obaidat. 2018. Mobi-flow: Mobility-aware adaptive flow-rule placement in software-defined access network. *IEEE Transactions on Mobile Computing* 18, 8 (2018), 1831–1842.

[3] Ming-Hung Chen, Kai-Wen Hu, I-Hsin Chung, and Cheng-Fu Chou. 2019. Towards VR/AR multimedia content multicast over wireless LAN. In *2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 1–6.

[4] Ming-Hua Cheng, Wen-Shyang Hwang, Yan-Jing Wu, Cheng-Han Lin, and Jhen-Siang Syu. 2020. An Effective Flow-Rule-Reducing Algorithm for Flow Tables in Software-Defined Networks. In *2020 International Computer Symposium (ICS)*. IEEE, 25–30.

[5] Leonardo C Costa, Alex B Vieira, Erik de Britto e Silva, Daniel F Macedo, Geraldo Gomes, Luiz HA Correia, and Luiz FM Vieira. 2017. Performance evaluation of OpenFlow data planes. In *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*. IEEE, 470–475.

[6] David Fernández, Luis M Contreras, Ricardo Flores Moyano, and Sandra García. 2021. NFV/SDN Based Multiple Upstream Interfaces Multicast Proxy Service. In *2021 24th Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*. IEEE, 159–163.

[7] Yingya Guo, Huan Luo, Zhiliang Wang, Xia Yin, and Jianping Wu. 2021. Routing optimization with path cardinality constraints in a hybrid SDN. *Computer Communications* 165 (2021), 112–121.

[8] Huawei Huang, Peng Li, Song Guo, and Baoliu Ye. 2014. The joint optimization of rules allocation and traffic engineering in software defined network. In *2014 IEEE 22nd International Symposium of Quality of Service (IWQoS)*. IEEE, 141–146.

[9] Salekul Islam, Nasif Muslim, and J William Atwood. 2017. A survey on multicasting in software-defined networking. *IEEE Communications Surveys & Tutorials* 20, 1 (2017), 355–387.

[10] Nanxi Kang, Zhenming Liu, Jennifer Rexford, and David Walker. 2013. Optimizing the "one big switch" abstraction in software-defined networks. In *Proceedings of the ninth ACM conference on Emerging networking experiments and technologies*. 13–24.

[11] Seiki Kotachi, Takehiro Sato, Ryoichi Shinkuma, and Eiji Oki. 2019. Multicast routing model to minimize number of flow entries in software-defined network. In *2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS)*. IEEE, 1–6.

[12] Seiki Kotachi, Takehiro Sato, Ryoichi Shinkuma, and Eiji Oki. 2020. Multicast Routing Model to Minimize Number of Flow Entries in Software-Defined Network. *IEICE Transactions on Communications* (2020).

[13] Zikang Li, Changming Yang, Zhenrong Zhang, and Sifeng Hu. 2020. Inter-domain resource optimization method of software-defined multi-domain optical network. *EURASIP Journal on Wireless Communications and Networking* 2020, 1 (2020), 1–9.

[14] Bommareddy Lokesh and Narendran Rajagopalan. 2021. Orchestrator for synchronizing network events in SDNs. *IEEE Transactions on Network and Service Management* (2021).

[15] Akbar Majidi and Ahmed H Zahran. 2020. Optimized Joint Unicast-Multicast Panoramic Video Streaming in Cellular Networks. In *2020 IEEE 28th International Conference on Network Protocols (ICNP)*. IEEE, 1–6.

[16] M Makolkina, A Vikulov, and A Paramonov. 2017. The augmented reality service provision in D2D network. In *International Conference on Distributed Computer and Communication Networks*. Springer, 281–290.

[17] Xuan-Nam Nguyen, Damien Saucez, Chadi Barakat, and Thierry Turletti. 2015. OFFICER: A general optimization framework for OpenFlow rule allocation and endpoint policy enforcement. In *2015 IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 478–486.

[18] Shrinivas Petale and Jaisingh Thangaraj. 2020. Link Failure Recovery Mechanism in Software Defined Networks. *IEEE Journal on Selected Areas in Communications* 38, 7 (2020), 1285–1292. https://doi.org/10.1109/JSAC.2020.2986668

[19] Xiuquan Qiao, Pei Ren, Guoshun Nan, Ling Liu, Schahram Dustdar, and Junliang Chen. 2019. Mobile web augmented reality in 5G and beyond: Challenges, opportunities, and future directions. *China Communications* 16, 9 (2019), 141–154.

[20] Pei Ren, Xiuquan Qiao, Yakun Huang, Ling Liu, Calton Pu, Schahram Dustdar, and Jun-Liang Chen. 2020. Edge AR X5: An Edge-Assisted Multi-User Collaborative Framework for Mobile Web Augmented Reality in 5G and Beyond. *IEEE Transactions on Cloud Computing* (2020).

[21] Jiulong Wu, Quanxin Zhao, Ning Yang, and Jingshan Duan. 2015. Augmented reality multi-view video scheduling under vehicle-pedestrian situations. In *2015 International Conference on Connected Vehicles and Expo (ICCVE)*. IEEE, 163–168.