Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institut für
Technische Informatik und
Kommunikationsnetze

Patrick Leu

SDN-assisted IP Multicast

Semester Thesis SA-2014-30
September 2014 to December 2014

Tutor: Dr. Panagiotis Georgopoulos
Co-Tutor: Dr. Bernhard Ager
Supervisor: Prof. Bernhard Plattner

Acknowledgements

First of all, I would like to thank my supervisors Dr. Panagiotis Georgopoulos and Dr. Bernhard Ager for all the support and valuable feedback given during my work on this semester project. Also, a special thank goes to Prof. Stephan Neuhaus for his valuable input. I would like to express my gratitude to Prof. Bernhard Plattner and the Communication Systems Group in general for giving me the opportunity to work on such an interesting topic.

Abstract

IP multicast provides efficient one-to-multipoint delivery of IP packets. However, IP multicast has seen a rather slow deployment in the open Internet. A problem content providers face is the lack of group management offered by the service model. The goal of this project is to propose a receiver access control scheme for IP multicast with SDN-assistance in the receiver domain. We therefore describe requirements on such a scheme with en emphasis on the commercial perspective. We then propose a design for a receiver access control scheme and evaluate it with regard to our requirements as well as security threats. In addition, we provide estimates on the complexity imposed by our scheme on both senders and controller.

Contents

Chapter 1

Introduction

IP multicast is a technology for efficient point-to-multipoint IP packet delivery [1]. In the years after its introduction in 1990 IP multicast has received a lot of research attention and resulting standards found support by most of the routers and end devices in the Internet. Still, applications relying on IP multicast have only been rarely deployed in the open Internet [2].

This is rather surprising as streamed video content makes up an ever increasing share of overall Internet traffic. Globally, video traffic was 66% of all consumer Internet traffic in 2013 and will be 79% in 2018 [3]. Services around multimedia streaming e.g. for video conferencing were among the originally envisioned applications of IP multicast and still could greatly benefit from multicast packet delivery. Emerging IP-TV services can be considered especially well-suited because of a) their one-to-many delivery pattern and b) their high bandwidth requirements per stream. In combination, these two factors possibly enable high efficiency gains in the network compared to individual unicast delivery.

One explanation why IP multicast has not been widely deployed is due to the service model not addressing commercial requirements of content providers [2]. These neglected requirements also encompass group management and security features. According to the service model, any host is able to join any multicast group and receive traffic sent to this group. From the perspective of an IP-TV content provider this lack of access control is far from ideal. It is very likely that a content provider instead requires the set of recipients to be restricted to the set of paying customers who previously subscribed to the service. In addition, the set of legitimate recipients may change dynamically. Whenever a customer unsubscribes, she should also not be able to receive respective traffic anymore.

A straightforward solution to this problem could consist of having multicast edge routers perform receiver access control in a fully decentralized way. However, this approach exhibits serious drawbacks. Firstly, having the routers perform access control individually is based on the assumption that they can be trusted. Placing trust in a highly distributed and huge set of devices is problematic. Commercial best-practice in this context seems therefore to be the deployment of closed and statically configured customer-facing routers [4]. Of course, that approach exhibits bad scaling properties in dynamic settings as the routers would require individual reconfiguration. The scaling problem can be alleviated by having the router look up a distributed receiver authentication service as proposed in [8]. However, this lookup requires changes to existing distributed multicast protocols. Furthermore, it increases the amount of control messages for a receiver join, each of them competing with actual data for bandwidth.

Centralized network control provided by Software Defined Networking (SDN) allows a new perspective on the problem of receiver access control. The network controller has a global view over a possibly large set of receivers. The OpenFlow [5] protocol allows the group membership information to be pushed to the controller and the reception state being managed in a logically centralized way.

With CastFlow [11], an SDN-driven approach to IP multicast routing has already been proposed. GroupFlow [12] is based on CastFlow and adds the interoperability with legacy group management.

1.1 Goal

The goal of this project is to design an SDN-assisted receiver access control scheme for IP multicast. The steps taken in order to achieve this goal are the following. First, we define requirements that govern the subsequent design process. The second step is the actual design and proposal of a receiver access control scheme. An implementation based on GroupFlow serves then as proof of concept for said scheme.

The remainder of this report is organized as follows. Section 2 presents the background work this project is based on. Section 3 explains requirements on a receiver access control scheme leading up to the design in Section 4. In Section 5, the implementation of a proof of concept is presented and the general scheme is evaluated. Section 6 draws conclusions and outlines possible future work.

Chapter 2

Background

2.1 IP Multicast

IP multicast allows the transmission of IP packets to a group of receivers [1]. Compared to unicast delivery, multicast can reduce transmission overhead at the sender as well as overhead in the network and decrease the latency. Furthermore, IP multicast serves as a rendez-vous service as a sender is not required to know specific receiver IP addresses. Similarly, the recipient of multicast data does not necessarily need to know the address of the very source the data originated from.

The basic principle behind IP multicast packet delivery is depicted in Figure 2.1. The core observation is that any packet destined for multiple receivers is transmitted at most once over the same link. At branches in the multicast routing tree, the packet is duplicated and forwarded individually.

In order to receive traffic, hosts need to announce their membership to a multicast group. Senders do not need to be members of the multicast group they are sending traffic to. The architecture of IP multicast is fully decentralized. This means, there is no central administration of group membership and routes are established on the basis of control information exchanges between multicast routers.

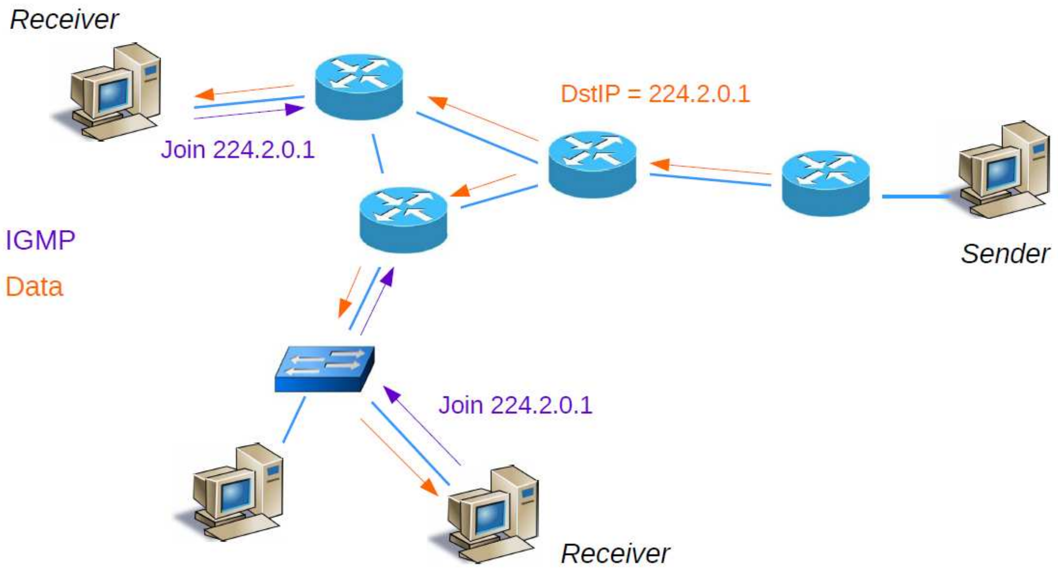


Figure 2.1: A multicast sender transmitting to two receiving hosts

2.1.1 Group Membership

A receiver joins a group by sending an IGMP membership report to the well-known destination IP address 224.0.0.22 [13]. The packet format of an IGMPv3 membership report is depicted in Figure 2.2. IGMP adds its own header fields to IP packets with protocol number 2. For each multicast group a receiver intends to join, the packet contains the associated class D multicast address. Moreover, the latest IGMP version 3 allows for source-specific joins. By providing a list of IP addresses, the host can either announce to only receive traffic from these sources or to receive traffic from all but these sources. The record-type-field indicates if the pertaining list of sources should be included in or excluded from the reception state. The membership report has a TTL of 1 and does not traverse any router supporting IP multicast. The router receiving the membership report is called the designated router (DR) and maintains the reception state. This means, for each downstream interface the DR keeps (S,G)-state, i.e. multicast source- and group-specific forwarding state.

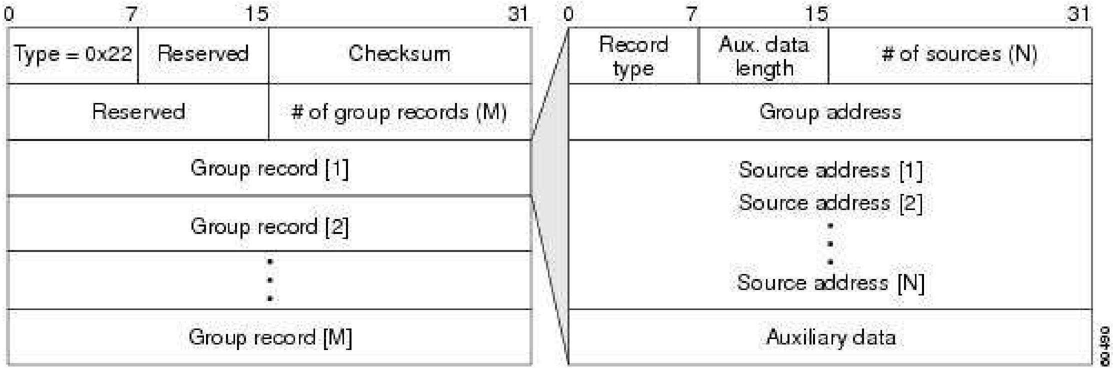


Figure 2.2: Packet Format of an IGMPv3 Membership Report [9]

2.1.2 Routing

The current IP multicast architecture uses distributed routing protocols to establish routing trees that carry multicast packets to the destination host. These protocols instantiate group-specific or group-and-source-specific routing state at multicast routers, based on reception state found in DRs. A prominent example for such a routing protocol is Protocol Independent Multicast (PIM). PIM relies on routing information provided by an arbitrary underlying unicast routing protocol. Within a certain domain, PIM constructs shared trees centered at a preconfigured router called rendezvous-point (RP). There are two different varieties of the PIM routing protocol. PIM dense mode (PIM-DM) assumes a high density of recipients in the network and operates with a flood-and-prune technique. PIM sparse mode (PIM-SM) [14] on the other hand only establishes routes to DRs reactively to group joins. In PIM-SM, multicast trees are thereby generated as follows. Upon instantiating reception state for a certain multicast group, a DR contacts the RP on the shortest path based on unicast routing information. Multicast routing state is instantiated in any router along this reverse-path (the path in the opposite direction of subsequent multicast traffic). Inter-domain operability is achieved by extensions to BGP provided by the Multicast Boarder Gateway Protocol (MBGP). The list of multicast sources located in a given domain is advertised to other domains with the multicast source discovery protocol (MSDP).

2.1.3 Problems

Causes for the sparse real-world deployment of IP multicast can be found in the following problems. Unless stated otherwise, these points are made with reference to [2].

- Multicast groups are not managed, arbitrary hosts can theoretically join any multicast group as well as send to any multicast address.
- IP multicast relies on UDP and therefore generally does not offer reliable transport.

- Multicast routing tends to place additional complexity on edge routers. This can be in conflict with the tendency of ISPs to migrate older routers to the edge of the network.
- Security problems. Especially can IP multicast be misused to amplify DoS attacks on the network.
- Routing issues in practice: In practice, shared tree routing protocols like PIM operate with inefficient source trees due to badly configured routers and inter-domain routing.
- Inter-domain routing is not handled well. After learning about a source in another domain, a RP directly joins to the multicast source.

Lack of Group Management

The current IP multicast service model does not include mechanisms for group management. Most prominently, there is no way to authorize receivers and senders. Also considered to be part of group management are aspects of accounting and billing as well as address discovery. Of course, for content providers generating content for paying customers the lack of such mechanisms is an unsatisfactory situation. This because services relying on IP mutlicast are vulnerable to attacks like unauthorized transmission and reception, causing service disruption as well as lost revenue for content providers [2].

2.1.4 Receiver Access Control

Receiver access control in IP multicast refers to the problem of limiting multicast traffic delivery to the set of legitimate receivers. There are two different perspectives on this problem. First, there is the perspective of the network and its notion of multicast group membership. From this perspective, the problem becomes one of multicast group access control [8]. From the perspective of the content provider, on the other hand, receiver access control means to limit the scope of hosts who receive content without considerations for network state. Existing efforts under this premise use traffic encryption and group key management [6], [7].

Group Access Control

This section refers to controlling access to the network's notion of group membership and subsequent instantiation of routing state only for admitted receivers. Selectively instantiating reception state at DRs has already been proposed in [8] as a means for receiver access control. To get access to a restricted multicast group, a host adds an authentication token to the IGMP report. This token, in turn is used by the DR to query an authentication server (AS). Only if the receiver is accepted by the AS, reception state is instantiated at the DR. In consequence, illegitimate hosts only have a minor influence on network control overhead as no routing attempt is made. Therefore, this approach mainly meets the needs of ISPs to save bandwidth and storage only for admitted receivers and especially could defuse the susceptibility of IP multicast to DoS attacks.

Group Key Management

Receiver authentication without trust in intermediate infrastructure has to be done end-to-end which suggests the use of traffic encryption. In the better understood unicast scenario, secure traffic delivery usually involves a security association between the two involved parties. Typically, this security association is represented by a shared cryptographic key which is used to encrypt the traffic between the communicating parties. Whenever two parties initiate a communication session, a new security association is established and maintained until one of the two parties leaves the session. The goal is to exclude everyone except the two communicating parties from understanding the data in transit for the whole duration of the session.

In contrast, secure multicast communication requires a security association between a sender and a dynamically changing set of receivers. The core principle in IP multicast is to deliver duplicates of the very same IP packet to different receivers. Therefore, multicast traffic encryption is based on a group-wide secret traffic encryption key. The major challenge here is to account for the dynamic behaviour of multicast groups and the implications thereof for re-keying. Whenever a new receiver joins, he should not be able to understand past traffic. Whenever a receiver leaves, she should not be able to understand future transmissions. This requires frequent group keying which is hard to scale up to large groups. In Iolus [6], the multicast group is partitioned into hierarchically organized subgroups. However, this approach also relies on a single central entity as well as trusted infrastructure at transition points between subgroups. VersaKey [7] offers different group keying schemes ranging from centralized to fully decentralized key management.

2.2 Software Defined Networking

Software Defined Networking is an emerging networking approach that relies on a logically centralized controller pushing instructions to simple forwarding devices (switches). Therefore, the central controller needs to maintain a global view of the network. The result is a complete separation between control and data plane allowing more efficient switching by reducing processing, management and control overheads. The OpenFlow protocol [5] is a prominent way for the controller to communicate with the forwarding hardware.

Chapter 3

Design Requirements

This chapter aims at laying down relevant design requirements on a receiver access control scheme. Having already gained some insights into why the existing IP Multicast service model is not widely used, the commercial perspective shall be given an emphasis in Section 3.1. Requirements dictated by the existing networking architecture are then part of Section 3.2. An access control scheme obviously needs to take into account malicious activities of hosts. Section 3.3 is therefore devoted to security threats.

3.1 Commercial Application

Originally envisioned applications for IP multicast can be classified as follows:

- One-to-many: IP-TV, audio and video distribution in general
- Few-to-few: Audio and video conferencing
- Few-to-many: Educational services, public conferences
- Many-to-many: Online gaming, collaborative simulations

This broad spectrum of applications nicely reflects the open service model of IP multicast. However, having in mind the deployment issues of the last two decades, it is worth focusing on the requirements of these applications which are most likely to be of commercial interest. The following discussion of commercial requirements shall for that reason be fully devoted to one-to-many applications.
The absence of market motivations has been an important factor why IP multicast has not found broad deployment so far [2]. On the parts of all the receiving customers, sending customers (content providers) and ISPs, the promised benefit has to be worth the effort and risk of a deployment. For instance, a customer subscribing for an IP-TV service does not have any direct benefits from multicast traffic delivery compared to standard unicast delivery. The bandwidth savings associated with IP Multicast obviously do not hold for the last link to the receiver. Therefore charging the receiving customer for IP multicast deployment is not a valid business case for the ISP. Multicast senders, however, can greatly benefit from multicast delivery in the presence of large audiences. So far, the service model was to such an extent unsuitable for most one-to-many applications, that IP multicast still was not considered an alternative. The goal therefore is to adapt the IP multicast service model in such a way, that it fits the needs of content providers in one-to-many applications better and renders IP multicast in such scenarios a more attractive delivery method. We argue that the following requirements should be met by an augmented IP multicast service model to accommodate for this:

- *Access Control:* Any group-and-source-specific set of receivers should be controlled on behalf of the respective multicast source. So at any point in time, only receivers authorized by the sender should be able to receive the multicast stream. The access control scheme should account for dynamic changes of the multicast group as receivers join and leave groups. This is the core requirement for video streaming applications.

- *Usage Model:* The usage model of IP Multicast services should be restricted as little as possible by the receiver access control scheme. This entails that the access control scheme should not make any assumptions about the authentication procedure in place between sender and receiver. This is also to accommodate for the increasing variety and density of different end devices.
- *Backwards Compatibility:* If receiver access control is not required by the multicast source, the overall scheme should be fully backwards compatible with legacy IP multicast group management.
- *Customer-friendly:* The amendments to the IP multicast service model should be easy to install and manage by ISPs. The reason being that whenever an ISP struggles to set up and configure new services duly, its customers might switch to another service provider. As receivers have in general no incentive to prefer multicast over unicast services, this is an important economic criterion. Especially avoided should be additional expenses on the part of receivers for using IP multicast services, e.g. for routers.

3.2 Network Architecture

- *Interoperability:* The authentication scheme should be interoperable with multicast sources located in domains with legacy (non-SDN) infrastructure.

3.3 Security Threats

It is worthwhile to consider the following attacks on the receiver access control scheme.

- *Malicious Replay:* A malicious host should not be able to gain unauthorized access to a multicast group by intercepting packets of a legitimate receiver and injecting them into the network at a later time and possibly at a different location in the network.
- *IP Address Spoofing:* Possibly intending to enforce membership to a certain multicast group, a malicious host could use a bogus source IP address.
- *Eavesdropping:* Traffic observation should not enable any host to acquire information enabling unauthorized multicast group access.
- *Denial of Service:* IP Multicast protocols are considerably more prone to DoS attacks than their unicast counterparts [8]. Most prominently, arbitrary (illegitimate) hosts sending to multicast groups can congest the network.

Chapter 4

Design of a Receiver Access Control Scheme

This chapter elaborates on the design of the proposed receiver access control scheme. Firstly, the direct consequences of our stipulated design requirements are documented. Then, the remaining design decisions are explained and motivated. The last section introduces the proposed design.

4.1 Consequences of the Requirements

Leaving the authentication procedure to the source as well as limiting the usage model as little as possible implies moving the authentication to the application layer. Backwards compatibility with IGMP suggests piggybacking of authorization information onto IGMP membership reports. The source being responsible for receiver authentication suggests that the source also maintainss per-receiver state.

4.2 Remaining Design Space and Decisions

A few design questions do not directly follow from our requirements and need further consideration.

4.2.1 Decentralized vs. Centralized Authorization

Traffic reception requiring an authorization step was earlier found to be a commercial design requirement. There are two basic approaches towards receiver authorization for dynamically changing groups. Firstly, authorization could be performed in a distributed way, i.e. by the network controller located in the receiver domain, on behalf of the sender. This has the advantage of minimal join latencies as the source does not need to be contacted for authorization purposes. The difficulty hides in the fact that either the receiver has to provide specific information (i.e. multicast groups, lifetimes), signed by the source, to the network controller, *or* the sender has to push this information close to the receiver domain in advance. Related to the latter scenario is the proposal to use distributed authentication servers in [8] which constitute a multicast group themselves. A problem with this approach is that a sender can not know in advance in which domain a receiver intends to join a multicast group, causing a lot of redundancy in the distributed state. In addition, installing and managing distributed infrastructure may put off content providers. On the other hand, providing authorization information to the controller via the receiver limits the sender's flexibility in making authorization decisions, as these decisions have to be made well in advance to the receiver joining the group and can not be revoked.

Centralized Authorization, on the other hand, requires the network controller to query the source whenever a receiver joins a multicast group. This allows for flexible authorization policies by the sender and facilitates accounting and billing at the sender. The main drawback of this approach is an increased load on the sender. However, considering that sender complexity in IP multicast one-to-many scenarios already is greatly reduced compared to unicast, this is assumed to be manageable by an accordingly provisioned sender. The sender load was furthermore considered a smaller disadvantage than requiring the sender to manage distributed authentication servers. The centralized authorization is also promising because of the level of indirection offered by the controller. This means, a sender and network controller are likely maintain a session key during the exchange of authorization information for many receivers. This reduces the number of asymmetric cryptographic evaluations compared to the sender authenticating the receivers more frequently.

As a consequence of these considerations, we make the decision in favour of a centralized authorization at the sender.

4.2.2 Enforcing Access Control

After deciding who is allowed to receive traffic, this decision needs to be enforced. There are two main motivations behind this. Firstly, it is about keeping unauthorized hosts from receiving unpaid content. This was identified as an important commercial requirement earlier. Secondly, enforcing access control to multicast groups may also be of interest to reduce vulnerabilities to DoS attacks and network congestion. This by only performing routing operations towards legitimate receivers.

Advantages of an approach using traffic encryption are a) the source needs to place a reduced level of trust into intermediate infrastructure and b) access control can be enforced at a fine granularity. However, end-to-end encryption alone does not decrease the load on the network. This because any host is still allowed to join a multicast group, however not necessarily able to decrypt received content. Furthermore, group key management would need to be in place to support dynamically changing groups. One could therefore imagine the receiver-side network controller to assist the sender in distributing the keys or even to perform re-encryption on behalf of the sender, similar to [6]. However, we consider network assistance for traffic encryption to be in contrast to the main motivation behind end-to-end encryption, namely to reduce trust into intermediate infrastructure. A cryptographic approach called proxy re-encryption could possibly offer a way for a network controller to re-encrypt traffic without the having the ability to decrypt it [10]. However, this approach seems to be at a very early stage and might not withstand complexity considerations.

Compared to traffic encryption, selective instantiation of membership and routing state for legitimate receivers is a simple approach towards receiver access control. An SDN controller can act as single interface for the authorizing entity to validate the reception state. We argue that a mechanism to filter routing state is required by the ISP anyway for DoS prevention. Access control to the granularity of the network located behind the respective DR port can be considered as positive side-effect of DoS prevention efforts, thereby not adding much overhead at the ISP.

Although traffic encryption may at first seem desirable from the perspective of the content provider, third party involvement in encryption may render it less attractive. Furthermore, per-receiver state required for key-distribution assistance at the controller may cause scalability problems. ISPs therefore may lack motivation to provide this assistance, especially because they have no direct economic incentive to do so. As a purely reception state based access control mechanism offers benefits to both ISPs (DoS prevention) and content providers (access control), we decide in favour of such an approach.

4.2.3 Usage Model, Authentication and Authorization Frequencies

Our first design requirement states that a receiver needs to be authorized whenever new content is accessed. As discussed in Section 4.2.1, this could possibly also be done based on information provided during authentication, which in turn would require authentication to happen more frequently. However, the authentication frequency can be considered a design decision itself as it affects the usage model. Furthermore, a high authentication frequency places a lot of complexity on the source. This due to the fact that any authentication requires expensive asymmetric cryptographic evaluations. This could jeopardize the good intentions behind IP Multicast regarding source complexity. However, there is also an advantage of frequent authentications.

Assuming the source adds a certificate to the reply, a third party could possibly carry out subsequent authorization on behalf of the source. This would obviate the need to contact the source for authorization purposes. When authenticated frequently, the receiver could always present a fresh certificate to the authorizing entity. At low authentication frequencies, this might be problematic, as the source's access control list now is likely to deviate from the certificate issued earlier. From a usage model perspective, frequent authentication also has serious drawbacks. For the user, authentication typically involves entering user name and password. Considering the example of IP-TV, this is certainly not something a user appreciates to do upon switching the program, for example.

Authentication is at least then required when a user subscribes and pays for new content. The design decision we make is to not have the receiver subscribe at the source more frequently than required for subscription purposes.

4.3 Proposal

In this Section, we propose a receiver access control scheme resulting from the previously made design decisions.

4.3.1 Scheme

Overview

A host desiring to receive a multicast stream is assumed to authenticate at the source prior to reporting its multicast group membership to the SDN controller via its designated router. Choosing an appropriate unicast authentication mechanism for the receivers is left to the respective source of a multicast stream. Authorization information provided by the source to the receiver is then piggybacked onto an IGMP membership report and relayed to the network controller.

Receiver Joining

Figure 4.1 depicts the complete process of a receiver joining a multicast stream. The steps are:

1. *Authentication:* The receiver is authenticated by the source. The authentication procedure as well as the details of the user ID are not defined by the scheme. Imaginable user IDs are username-password or a hardware identifier for closed receiver devices. The result of the authentication is an authentication token being sent to the receiver. The source internally maps its own notion of the user ID to the value of the authentication token.

2. *IGMP membership report:* The receiver transmits a source-specific IGMP membership report. This report contains a list of source IPs the receiver wishes to receive traffic from. In the auxiliary data field of the IGMP membership report, the list of pertaining authentication tokens is added.

3. *OpenFlow PacketIn:* An IGMP membership report arriving at an SDN switch is sent to the network controller as part of an OpenFlow PacketIn event.

4. *Authorization query:* For any source being part of newly added reception state the controller checks if this source is a) known to require authentication *or* b) not known to the controller at all. If this is the case, the controller sends an authorization request to this source. The query message contains the authentication token as well a hash of the network location ID of the pertaining DR. The message is signed with a secret key shared between controller and sender.

5. *Authorization response:* Upon performing the receiver authorization, the source replies appropriately to the network controller with either an approval or denial message. The message is signed with a secret key shared between sender and controller.

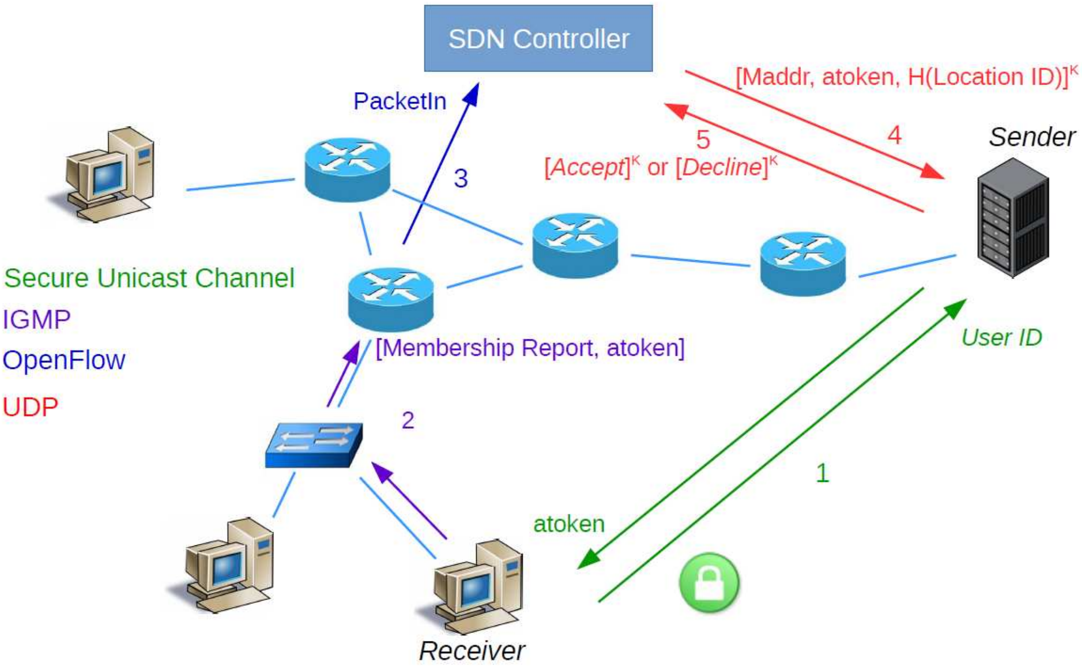


Figure 4.1: Receiver Access Sontrol Scheme Handling a Receiver Join

Term	Meaning
atoken	Authentication token
Maddr	Multicast address
PacketIn	An OpenFlow PacketIn event
Location ID	A string of the format [dpid,port] containing the data path ID of the DR as well as the DR port number.
$[m]^K$	A message m, cryptographically signed with a symmetric key K.
$H(m)$	A cryptographic hash of message m

Table 4.1: Terms Used in the Scheme

Receiver Leaving

IGMPv3 is a soft-state protocol [13]. Therefore, the reception state of any DR times out after a certain time. This behaviour is not changed by the access control scheme. A receiver leaving the session will therefore just cease to reply to IGMP queries issued by the controller and the reception state kept at the controller will time out. As a consequence, multicast traffic will not be routed to the receiver's DR anymore (assuming no other host behind the same DR port still wishes to receive said traffic).

Source excluding a Receiver

When a source decides to stop sending to a certain receiver, it will reply to the next authorization query with a deny message. The controller will then delete the pertaining reception state. The time interval a receiver is able to receive content after being excluded by the source therefore depends on the authorization interval.

Authorization

As a decision base for the source performing the authorization, the controller provides the multicast address, the authentication token as well as a hash of the location ID to the source. The purpose of the location hash is for the source to detect location conflicts indicating a replay of an authentication token from a different network location. In this case, the source is assumed to deny the authorization.

4.3.2 Communication

Extending IGMPv3

Group membership changes at a receiving host are signalled to the network controller with IGMP membership reports. The list of authentication tokens is thereby added to an IGMP membership report in the auxiliary data field. This field is not used in IGMPv3 [13].

Controller-Source

Authorization queries and replies constitute the messages exchanged between the network controller and the multicast source. Queries are sent by the controller as UDP packets destined to the respective sender IP address at a well-known port. Assuring reliable transmission needs to be done by the controller.

4.3.3 Senders without Receiver Access Control

Sources are free to omit receiver access control. If the controller knows that a certain source does not require receiver access control, reception state is instantiated directly without requiring an authentication token (i.e. a legacy IGMP membership report is sufficient).

Therefore, the controller maintains separate lists of multicast sources known to require receiver access control and of sources known to not require receiver access control. If a receiver joins a source not found in any of the two lists at the controller, the controller queries the source without providing an authentication token. Subsequently, the source signals to the controller to not require receiver access control by accepting the query, or conversely to require access control by responding with a deny message.

IGMPv3 allows receivers to join multicast groups without providing source-specific information. In this case, the controller forwards traffic from all sources that do not require receiver access control to the respective receiver. Whenever the controller receives traffic from a previously unknown source, the source is queried to find out if receiver access control is required.

Chapter 5

Implementation and Evaluations

This chapter consists of two main parts. Section 5.1 introduces the implementation that serves as a proof of concept for the receiver access control scheme. In Section 5.2, the design in general is evaluated with respect to the previously stated requirements and security concerns. Furthermore, we provide estimates on the complexity placed on network controller and sender.

5.1 Proof of Concept

In this section, we provide insights into the architecture of our implemented proof of concept.

5.1.1 Simplifications

In comparison to the proposed design, the implemented proof of concept was subject to a few simplifications. However, theses simplifications only encompass the out-of-band part as well as features which can be implemented with comparably small effort. Therefore, the validity of the evaluation is not negatively affected.

A receiver authenticating and subscribing for services at the source (step 1 in the scheme) is not modelled as an information exchange in mininet. Instead, the authentication and subscription part is implemented as a receiving host directly writing state to a file. This file models receiver-specific sender-side state and is accessed by the source in order to authorize receivers. This approach was chosen because of its simplicity as all hosts in mininet share the same file system. This abstraction can be motivated by the fact that receiver authentication is most likely based on a secure unicast channel, which is a well-understood principle. In addition, the authentication procedure is not defined in our scheme. Therefore out-of-band authentication is theoretically possible, even though it may not be a very realistic scenario.

As a second simplification, the UDP packets exchanged between the network controller and the sender are not cryptographically signed.

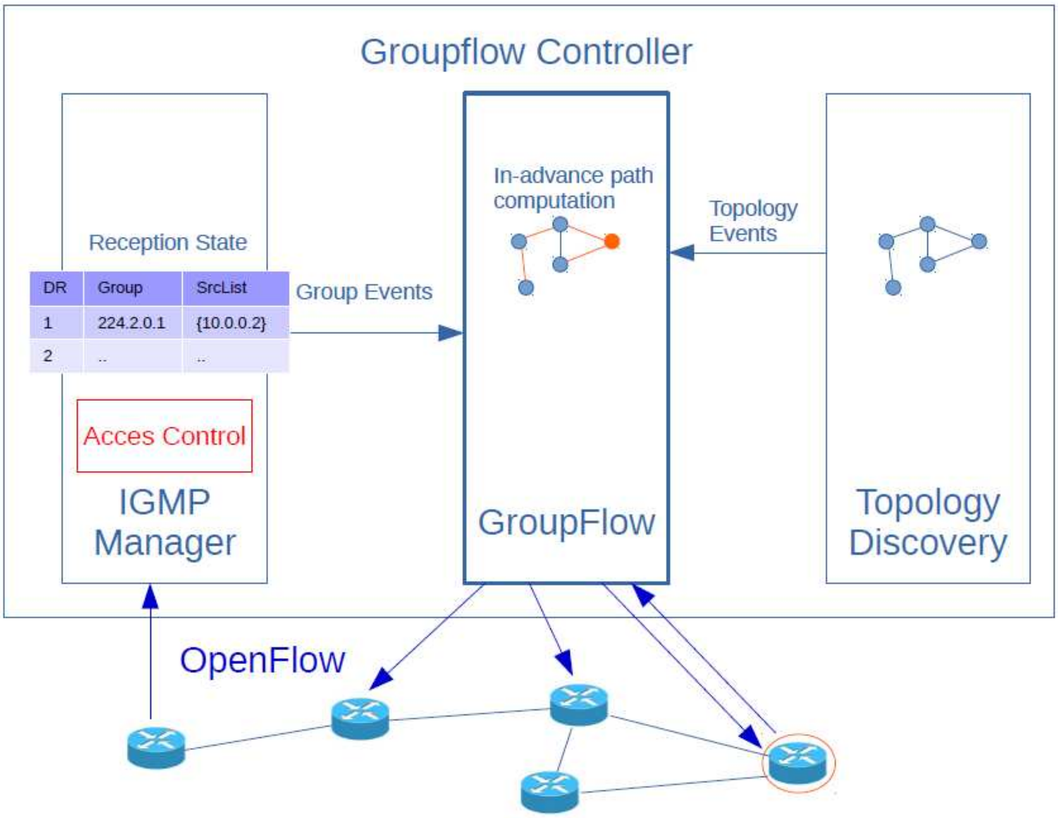
Time-outs for the reception state have not been implemented.

Furthermore, the controller does not assure the reliable transmission of the authorization queries.

Worth a note is also the length of the authentication token. For the sake of simplicity, the length of the authentication token is chosen to be only 32 bits. To avoid brute-forced replays in a realistic scenario, its length would have to be significantly increased.

5.1.2 Controller Design

The core of this implementation builds on the existing POX-based [15] GroupFlow network controller providing multicast routing, different flow-replacement strategies as well as compatibility with IGMPv3 [12]. The interaction between different modules is depicted in Figure 5.1. The internals of the controller shall be briefly explained with respect to the main building blocks.



DR	Group	SrcList
1	224.2.0.1	{10.0.0.2}
2	..	..

Figure 5.1: Controller Design

Topology Discovery

This module is included in the original POX controller. The purpose of this module is to provide the global view over the network to the GroupFlow module responsible for routing decisions. To do so, link layer discovery protocol (LLDP) messages are periodically sent out from the ports of the controlled switches. This way, links are discovered in the network. Upon the discovery of a new link, a Topology Event is triggered at the GroupFlow module.

Reception State Management

Managing the IGMP reception state of every DR in the network is done by GroupFlow's IGMP Manager module. Furthermore, this module is responsible for periodically transmitting IGMP queries to all hosts in the network. Any reception state change is signalled to the GroupFlow module with a Group Event.

Multicast Routing

As GroupFlow itself is based on CastFlow [11], routing paths are computed in advance for known sources to all other switches in the network. This means that whenever a new multicast source starts sending, paths are computed only based on the topology information provided by the topology discovery module. Actual flows are then installed on the switches based on the pre-computed routes as well as the group membership information provided by the IGMP Manager module.

Receiver Access Control

Receiver Access Control implemented as part of this project mainly comprises additions to the IGMP Manager module responsible for reception state management. The idea is to only allow the instantiation of new source specific reception state for sources that authorize the respective receiver. As depicted in Figure 5.1, receiver access control can be understood to receive as input the membership reports coming in from DRs and output reception state of authorized receivers.

5.1.3 Receiver and Source Logic

In addition to the controller logic, the proof of concept also comprises simple sender and receiver programs. Receiver-side logic is implemented with three different python scripts. The first script, `rcv_authentication.py`, is run whenever a receiver subscribes for new content and services at the source. The user passes her username as well as the source IP address and a multicast IP address as command line arguments. A fresh authentication token is then returned to the host as well as written to the sender-side state. The second script, `rcv_dm.py` is meant to run in the background. Its purpose is to respond to IGMP queries. Finally, `group_membership.py` allows the receiving host to join a multicast group.

Source behaviour is implemented in `source.py`. This script is run in the background and replies to authorization queries originating from the receiver-side network controller. The script accesses the sender-side state file `src_state.pkl` and sends out appropriate authorization replies back to the controller. The source IP address as well as the query port number is passed to the script as command line argument.

Capturing and transmitting IGMP and UDP packets at receiver and source, respectively, is done with Scapy [16]. Scapy is a packet sniffing and generation tool implemented in python and supports a variety of packet formats. However, the official Scapy contribution for IGMPv3 is still under development and had to be improved. Our extension to IGMPv3 of course needed to be added to this contribution, as well.

5.1.4 Source and Receiver State

Due to our receiver access control scheme, any multicast source holds the following receiver-specific state:

- Authentication token: A 32 bit value identifying an authenticated receiver.
- Controller address: The IP address of the network controller responsible for the receiver domain.
- Group list: A list of all the IPv4 multicast destination addresses of groups the receiver has subscribed to at this source.
- Location hash: A 32 bit hash over the receiver Location ID.

For any joined multicast group the receiving host maintains the following state:

- IP group address
- IP source list: IP addresses of sources the host wishes to receive content from.
- List of authentication tokens: A 32 bit value for each source.

5.2 Evaluation

This sections aims at evaluating our design with regard to the requirements stated in Section 3. This also includes assessing our design with regard to possible security threats. In a last step, the additional burden placed on the network controller and sender by our access control scheme shall be estimated.

5.2.1 Addressing the Requirements

Access Control

Our scheme enforces access control to multicast groups and includes the sender into the authorization decision. Dynamic group changes are accounted for by reception state timing out at the network controller. However, leaving receivers are able to receive traffic illegitimately for up to the length of the authorization interval (which is typically a few minutes). In addition, receiver access control is only enforced to the granularity of networks behind the same DR-port and not to the granularity of individual users. Our scheme can be considered to capture the individual aspects of the requirement (authentication, authorization, access control), but in a less stringent way than stated.

Usage Model

An underlying assumption in the design was that the authentication procedure may have a considerable impact on the usage model. The only restriction our scheme puts on the authentication procedure is the fact that its result should be an authentication token stored at the receiving host. Therefore, we argue that the usage model is limited minimally. A possible limitation to the usage model stems from the fact that a receiver changing the network is required to re-authenticate in our scheme. This because the source will try to prevent replays of same authentication tokens from different network locations.

Backwards Compatibility

Our receiver access control relies on a slightly adapted version of IGMPv3 for group management. For sources requiring receiver access control, only source-specific joins (i.e. include lists) are possible. However, a receiver joining with an exclude list is still served with traffic from all the sources of the respective group that do not require receiver access control. Therefore, our access control scheme can be considered fully backwards compatible for sources omitting receiver authentication.

Customer-friendly

The main challenge behind the proposed scheme for the ISP is possibly the transition to an SDN-controlled receiver domain in customer-friendly way. However, under the premise of a receiver domain already being SDN-controlled, our access control scheme can be considered customer-friendly in the sense that no dedicated hardware is required by the end user and changes to existing protocols (IGMP) are minimal.

Interoperability

This requirement is met in the sense that authorization queries are based on legacy unicast. Therefore, the source domain does not need to be under SDN control. However, we did not address how the receiver domain controller interacts with legacy inter-domain multicast source announcement and routing.

5.2.2 Design vs. Security Threats

In order to evaluate the proposed design with regard to security threats, IGMP replay attacks as well as IP address spoofing attacks were staged in the proof-of-concept-implementation. Furthermore, the threat of eavesdropping attacks and DoS attacks was assessed qualitatively with regard to the proposed design.

IGMP Replay

A staged IGMP replay attack on our implementation was found to be unsuccessful. The reason being that the sender keeps track of the network location of authorized receivers. Therefore, the same authentication token appearing from a different network location results in the authorization query being denied at the source.

IP Address Spoofing

A host spoofing his IP address has not interfered with our implementation of the access control scheme in a staged attack. As the network only enforces access control on the granularity of DR ports, a means to identify receivers is not required for access control purposes at the network controller. Therefore, the access control scheme does not rely on host IP addresses.

Eavesdropping

Assuming the authentication of the receiver at the sender relies on a secure unicast channel, a session key can be established to prevent eavesdropping on secret authentication token by traffic encryption. The authentication token does not traverse a communication link in the clear before the receiver joins the multicast group. At this stage, the authentication token is linked to a receiver network location by the multicast sender. Therefore, an attacker can not gain illegitimate access to a multicast group by eavesdropping on the authentication token.

However, the access control scheme does not impose any measure to mitigate direct eavesdropping on multicast traffic. An attacker is not prevented by our scheme from illegitimately listening to a multicast stream if he is located behind the same DR port as a legitimate receiver.

Denial of Service

Our receiver access control scheme allows to only perform routing operations towards legitimate receivers. This mitigates some DoS attacks. However, other DoS threats still remain. Unsolved is the problem of unauthorized senders congesting the network or unauthorized receivers overwhelming the controller, and subsequently the sender, with IGMP messages.

5.2.3 Scalability Considerations

In the following, we derive an estimate on the complexity our access control scheme imposes on the receiver-side network controller as well as the sender. Therefore, we consider the scenario of one multicast group comprising a total of N_R receiving hosts [1]. Each of the receivers is located in a domain controlled by one of N_C network controllers. N_S senders transmit to this multicast group and each receivers subscribes on average to $\mathbf{E}[N_S^r]$ different sources. As a simplification, we assume that the receivers are distributed evenly across different domains.

An overview of all the parameter assignments used for the complexity estimates is provided in Table 5.1.

Parameter	Variable	Assignments
Number of receivers	N_R	Up to 800000
Number of senders	N_S	-
Number of controllers	N_C	4
Number of receivers per domain	N_R/N_C	Up to 200000
Expected number of senders joined per receiver	$\mathbf{E}[N_S^r]$	{1, 3, 10, 20}
Average authorization interval of a receiver	I_{Az}^r	10 min
Average authentication interval of a receiver	I_{An}^r	{3, 5, 8}h
Probability of a source being included in a receiver join	$\mathbf{E}[N_S^r]/N_S$	{0.3, 0.5, 1}

Table 5.1: Parameter assignments used for the complexity estimates

Our estimates are based on worst case receiver authorization rates given a specific set of parameter choices. Therefore, a receiver triggering an authorization query is modelled as a Poisson process [17] with an average arrival rate $\lambda_{Az}^r = 1/I_{Az}^r$. Similarly, the process of receivers

[1] We could also consider multiple disjoint groups, using statistical values for N_R and N_S, however in our simplified scenario this adds nothing to the statement.

authenticating at the sender has a rate $\lambda^r_{An} = 1/I^r_{An}$. For the results, we consider the 99th percentile rates of the events in question. Table 5.2 presents our estimates for complexity arising at sender and receiver depending on the introduced parameters. We make the following core observations:

- The rate of authorization queries a controller issues increases proportionally to the number of receivers in the pertaining domain N_R/N_C
- The rate of authorization queries sent to a specific sender is proportional to the number of receivers multiplied with the probability that the receiver includes this source, $\mathbf{E}[N^r_S]/N_S$.
- Although receiver-specific state only has to be kept at the controller for the duration of an authorization message exchange with the source, this state still increases on the order of the number of receivers in the respective domain.

	Authentication	**Authorization**
Load on sender		
Arrival Rate	$\lambda^s_{Ac} \sim N_R \cdot \mathbf{E}[N^r_S]/N_S/I^r_{An}$	$\lambda^s_{Az} \sim N_R \cdot \mathbf{E}[N^r_S]/N_S/I^r_{Az}$
State	$\mathcal{O}(N_R \cdot \mathbf{E}[N^r_S]/N_S)$	$\mathcal{O}(N_R \cdot \mathbf{E}[N^r_S]/N_S)$
ACE[a]	$\mathcal{O}(\lambda^s_{Ac})$	$\mathcal{O}(N_C)$
SCE[b]	-	$\mathcal{O}(\lambda^s_{Az})$
Load on controller		
Arrival Rate	-	$\lambda^c_{Az} \sim N_R/N_C/I^r_{Az}$
State	-	$\mathcal{O}(N_R \cdot \mathbf{E}[N^r_S]/N_C)$ (transient session state only)
ACE	-	$\mathcal{O}(N_S)$
SCE	-	$\mathcal{O}(\lambda^c_{Az} \cdot \mathbf{E}[N^r_S])$

[a] Asymmetric cryptographic evaluations

[b] Symmetric cryptographic evaluations

Table 5.2: Estimates of the complexity incurred by both sender and controller due to our receiver access control scheme

Figure 5.2 illustrates how increasing the number of receivers in its domain affects the load on the network controller in terms of sent authorization queries. It becomes evident that many-source scenarios place a multiple of the computational load on the controller.

Figure 5.3 depicts the computational load placed on senders for an increasing number of receivers. It can be observed that asymmetric cryptographic evaluations are about ten times less frequent than symmetric cryptographic evaluations. This can be considered a desirable ratio as asymmetric cryptography is much more costly in terms of complexity.

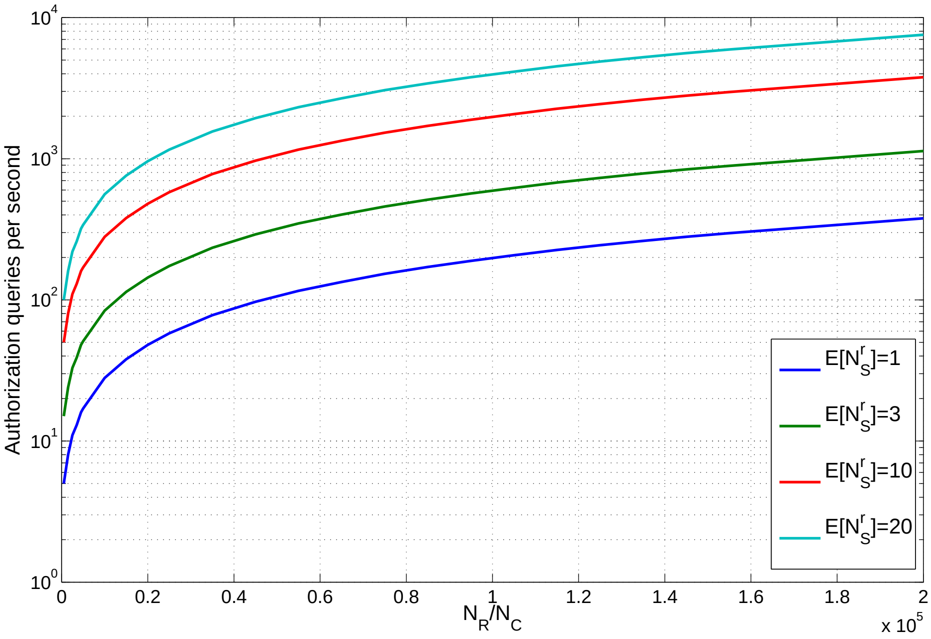


Figure 5.2: Expected controller load in terms of transmitted authorization queries per second with the number of receivers in the controlled domain on the x-axis. The colors represent different numbers of expected sources per receiver.

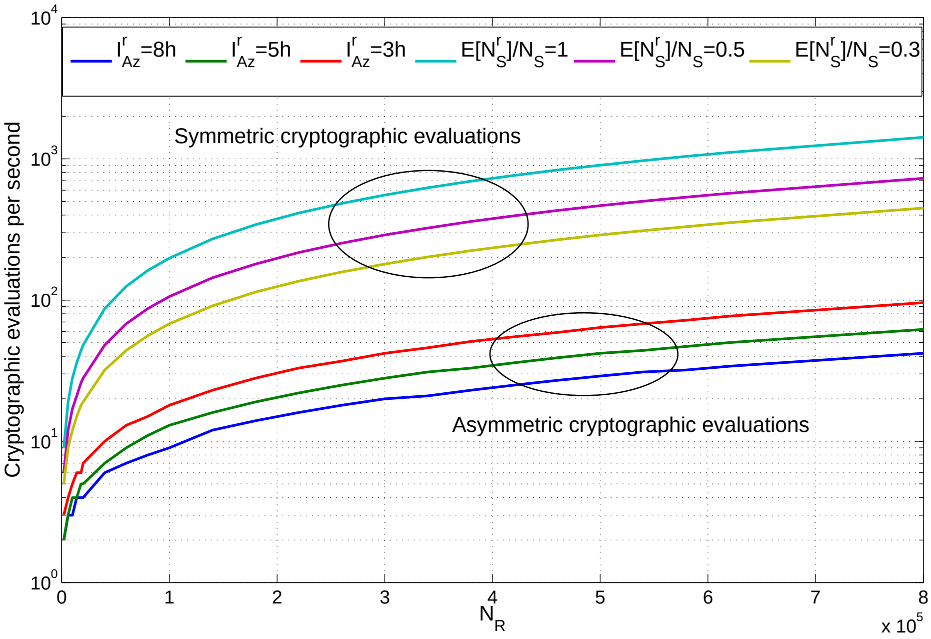


Figure 5.3: Expected sender load in terms of a) the expected number of symmetric cryptographic evaluations in (top three lines) and b) the expected number of asymmetric cryptographic evaluations (bottom three lines). The colors represent different values for the probability for a receiver to subscribe to this source (top three lines) and the mean receiver authentication intervals (bottom three lines), respectively.

Chapter 6

Conclusion

During the course of this project, we proposed a receiver access control scheme for IP Multicast with the following properties:

- Source-controlled authentication and authorization of multicast receivers
- Per-DR-port granularity of receiver access control
- Backwards compatibility with legacy multicast group management
- Interoperability with multicast sources located in non-SDN-controlled domains

Staged attacks on the proof-of-concept-implementation suggest that the access control scheme is resilient to IP address spoofing attacks as well as IGMP replay attacks. Possibilities for DoS attacks are limited as multicast traffic is only forwarded towards legitimate receivers. Moreover, eavesdropping attacks on the access control scheme can be mitigated by using a secure unicast channel for receiver authentication. However, direct eavesdropping on multicast traffic is not prevented by our scheme. Complexity estimates suggest that the scalability may be limited in many-source scenarios due to the load on the network controller. We conclude that an application of the proposed access control scheme may be more suitable in a one-to-many or few-to-many scenario.
Possible improvements or further work could be concerned with:

- *Scalability:* Locality of reference exhibited by users accessing multicast traffic could be leveraged. Authorization information of joined sources could be cached at the controller for different multicast sources. The trade-off between decreased sender complexity and increased receiver-specific state at the controller could be explored with respect to application requirements.
- *Fine-grained access control:* Access control to the granularity of individual users could possibly consist of traffic (re-)encryption at the edge router.
- *Receiver mobility:* The proposed scheme favours resilience to IGMP replay attacks over user mobility. However, re-authentication upon network changes can have negative implications for the usage model.
- *Source authentication:* Receivers authenticating multicast sources by time-delayed disclosure of symmetric keys is a promising approach [18]. However, DoS caused by illegitimate senders is not mitigated this way. Statically assigning blocks of IP Multicast addresses to domains and verifying multicast traffic accordingly could be a thinkable approach.

Appendix A

Running the Implementation

In the following, we provide the instructions to run the implementation of the proof of concept. The commands preceded by # are meant to be executed on a host in the mininet simulation environment. To start mininet with a simple provided topology example, execute the following command in the mininet VM:

```
$ sudo mn --custom mctopo.py --topo mctopo --controller remote --mac
```

In a separate terminal, the controller can then be started with the following command:

```
./pox.py openflow.discovery openflow.flow_tracker --query_interval=1
     --link_max_bw=30 --link_cong_threshold=30 --avg_smooth_factor=0.65
     --log_peak_usage=True openflow.igmp_manager openflow.groupflow
     --util_link_weight=10 --link_weight_type=linear --auth_port=35272
     openflow.l3_learning --arp_for_unknowns=True
     log.level --WARNING --openflow.igmp_manager=DEBUG
     --openflow.groupflow=DEBUG --openflow.l3_learning=DEBUG
```

In the above command, only the `auth_port` argument for the IGMP Manager module has relevance for receiver access control. It denotes the destination port to which the controller sends the authorization queries.
Next we will test the functionality of the implementation by setting up a host as sender and one as receiver. Then we will authenticate the receiver and subsequently join a multicast group.
First of all, make sure that a default route for IP packets is installed at all hosts in mininet:

```
# route -n add net 224.0.0.0 netmask 240.0.0.0 dev [interface]
```

At the sending host in mininet, start the source daemon with the following command including the source IP address (-i) as well as the port number to listen for authorization queries (-p):

```
# python source.py -i 10.0.0.2 -p 35727
```

To perform the simulated authentication of a receiver, run the following command with appropriate values for source address (-s), multicast address (-m) and user name (-u) at the receiving host:

```
# python rcv_authentication.py -s 10.0.0.2 -m 224.2.0.1 -u testuser add

Setting new state...
Authentication token for user testuser at source 10.0.0.2:
1105306673
```

Upon allocating receiver-specific state, the script returns the assigned value for the authentication token. We can check the state held at the multicast source `10.0.0.2` any time as follows:

```
# python rcv_authentication.py -s 10.0.0.2 show

Users:
{’10.0.0.2’: {’testuser’: 1105306673}}
Receiver state at source 10.0.0.2 for all authentication tokens:
{1105306673: {’ca’: None, ’ma’: [’224.2.0.1’], ’lh’: None}}
```

Note that the controller address (ca) as well as the location hash (lh) are not known by the source at this point.
Before joining the multicast group with the receiver, we start the receiver daemon that takes care of sending out replies to IGMP queries:

```
# python rcv_dm.py
```

We can then join the multicast group with the following command, providing the multicast group address (-m), the source list (-s) as well as the list of pertaining authentication tokens (-a) as arguments. As a convention, an authentication token has the same list index as the respective source. In order to explicitly not provide an authentication for a certain source, the respective entry is set to zero.

```
# python group_membership.py -m 224.2.0.1 -s ['10.0.0.2']
    -a [1105306673] add
```

The above command will generate multicast reception state at the host as well as send out an IGMP Membership Report. The receiver authorization can now be tested for instance by sending a ping to the multicast group address from the sender:

```
# ping 224.2.0.1
```

By using `tcpdump` or `wireshark` we are then able to check if the ping is received at the authenticated receiver. Conversely, the third host should not be able to receive the ping.

Bibliography

[1] "Multicast routing in datagram internetworks and extended LANs."
Deering, Stephen E., and David R. Cheriton. ACM Transactions on Computer Systems (TOCS) 8.2 (1990): 85-110.

[2] "Deployment issues for the IP multicast service and architecture."
Diot, Christophe, et al. Network, IEEE 14.1 (2000): 78-88.

[3] "Cisco VNI Global Forecast (2014)."
http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.html (accessed 08.12.14)

[4] "Securing IP Multicast Services in Triple-Play and Mobile Networks."
http://www.cisco.com/c/en/us/products/collateral/ios-nx-os-software/ip-multicast/prod_white_paper0900aecd80557fd4.html (accessed 04.12.14)

[5] "OpenFlow: enabling innovation in campus networks."
McKeown, Nick, et al. ACM SIGCOMM Computer Communication Review 38.2 (2008): 69-74.

[6] "Iolus: A Framework for Scalable Secure Multicasting."
Mittra, Suvo. ACM SIGCOMM Computer Communication Review. Vol. 27. No. 4. ACM, 1997.

[7] "The VersaKey framework: Versatile group key management."
Waldvogel, Marcel, et al. Selected Areas in Communications, IEEE Journal on 17.9 (1999): 1614-1631.

[8] "Multicast-specific security threats and counter-measures."
Ballardie, Tony, and Jon Crowcroft. Network and Distributed System Security, 1995., Proceedings of the Symposium on. IEEE, 1995.

[9] "IP Multicast Technology Overview."
http://www.cisco.com/c/en/us/td/docs/ios/solutions_docs/ip_multicast/White_papers/mcst_ovr.html (accessed 15.12.14)

[10] "Identity-based proxy re-encryption."
Green, Matthew, and Giuseppe Ateniese. Applied Cryptography and Network Security. Springer Berlin Heidelberg, 2007.

[11] "CastFlow: Clean-Slate Multicast Approach using In-Advance Path Processing in Programmable Networks."
Marcondes, Cesar AC, et al. Computers and Communications (ISCC), 2012 IEEE Symposium on. IEEE, 2012.

[12] "GroupFlow: Multicast over SDN with OpenFlow."
https://github.com/alexcraig/GroupFlow (accessed 22.09.14)

[13] "Internet group management protocol, version 3."
Cain, Brad, et al. (2002).

[14] "Protocol independent multicast-sparse mode (PIM-SM): Protocol specification."
Farinacci, Dino, et al. (1998).

[15] "POX Wiki."
https://openflow.stanford.edu/display/ONL/POX+Wiki (accessed 22.09.14)

[16] "Welcome to Scapy's documentation!"
http://www.secdev.org/projects/scapy/doc/introduction.html#about-scapy (accessed 07.11.14)

[17] "Poisson process."
http://en.wikipedia.org/wiki/Poisson_process (accessed 12.12.14)

[18] "Efficient and secure source authentication for multicast."
Perrig, Adrian, et al. Network and Distributed System Security Symposium, NDSS. Vol. 1. 2001.