H2020 iP Over IcN– the betTer IP (POINT)

Design-IGMP-IPM

POINT IGMP-IP Multicast Handler Design

List of Authors:

Yannis Thomas

Dirk Trossen

Xenofon Vasilakos

George Xylomenos

1.Architecture

IP multicast services are a special case of IP-based services, in which a sender transmits IP packets to an IP multicast address; these packets are delivered to any receiver that has previously subscribed to that IP multicast group. For example, live IPTV can be implemented by transmitting each channel to a multicast group, inside RTP/UDP packets, and viewers subscribing to the multicast group corresponding to a desired channel.

IP multicast handling in POINT assumes unmodified user terminals, which use IGMPv2 as the control protocol for joining and leaving multicast groups [1]. The actual implementation of IP Multicast in POINT replaces a network of IGMPv2 snooping switches (L2 switches that process L3 messages generated by IGMPv2), a common configuration for live IPTV.

To support IP multicast, POINT uses two co-operating handlers, the IGMP handler for control and the IP multicast handler for data. The IGMP operations for joining and leaving an IP multicast group are realized via publications to an appropriate ICN name, representing a control channel. A potential sender (data source) to an IP multicast address acts as a subscriber to the appropriate ICN name, in order to notified about join and leave messages to that group.

On the other hand, the receivers of the IP multicast address act as subscribers to another appropriate ICN name, representing a data channel. When a join or leave message is sent by a receiver, it is translated to a publication with implicit subscription, where the publication of the message to the control channel implies a subscription (or an un-subscription) to the corresponding data channel. A sender to the IP multicast group simply acts as a publisher to the data channel of the IP multicast group.

The configuration settings for both handlers are provided in the documentation of the NAP, as they are included in the NAP configuration file.

2.IGMP Handler

Figure 1 shows the ICN namespace used by the IGMP handler. A separate root identifier is used for IGMP, since each IGMP message is an independent message sent in unicast mode. A flat namespace of control channel identifiers is used under this scope, created by hashing the corresponding IP Multicast addresses. Each IP Multicast group requires one such IGMP control channel, as well as an IP multicast data channel (described in the next section).

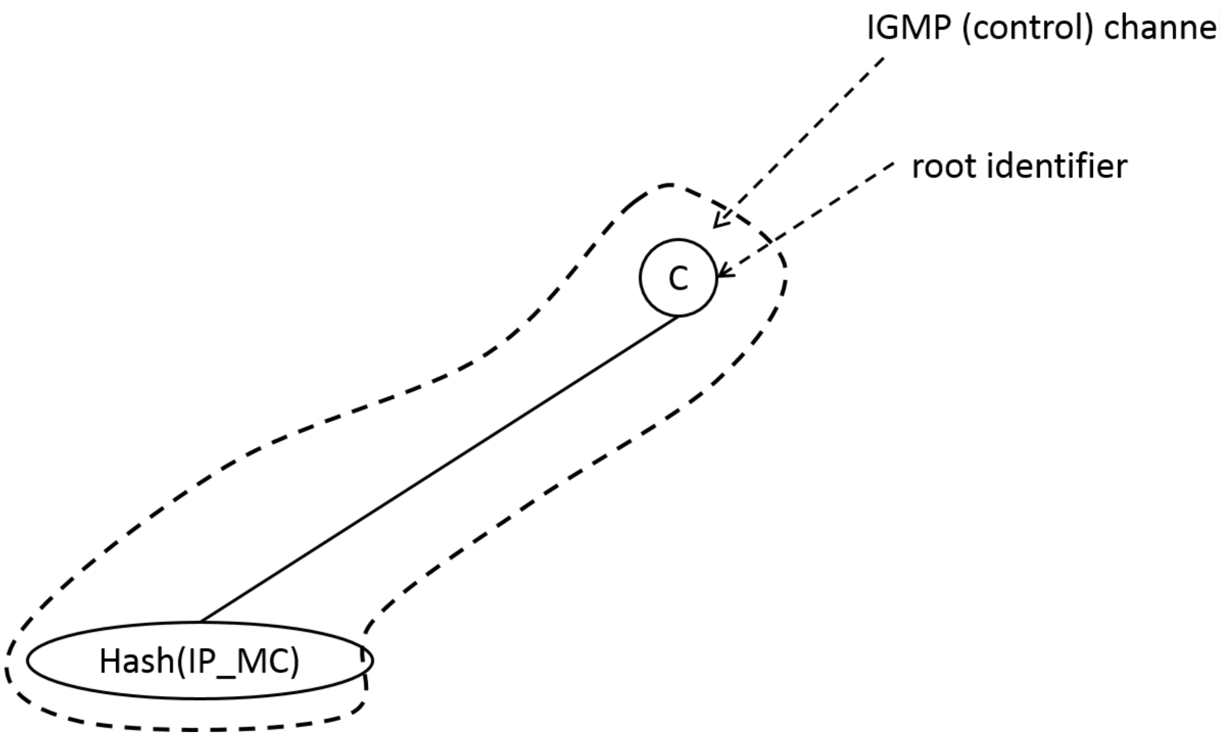

Figure 1: Namespace for IGMP-over-ICN.

The IGMP handler running at the NAPs differentiates its behaviour and state based on whether it is running on the sNAPs (for multicast senders) or the cNAPs (for multicast receivers), with respect to a multicast group. On the sNAP side, we maintain a multicast database containing the advertised data scope IDs (CIDd), that is, the groups that have senders attached to the sNAP, and the Node IDs (NID) of the cNAPs that have joined each group on behalf of their attached terminals. On the cNAP side, we maintain a multicast database containing the control scope IDs (CIDc) corresponding to groups that currently have joined terminals, as well as an inactivity counter for each group (explained below). The sNAP/cNAP distinction does not imply that a NAP cannot concurrently hold both roles; it can be the sNAP for some groups and the cNAP for others.

While the IP multicast receivers indicate their interest in a group via IGMPv2 join and leave messages, allowing cNAPs to auto-configure themselves, there is no standard way for IP multicast senders to indicate their presence. This prevents the sNAPs from subscribing to the appropriate control channels. As our target application for the IGMP handler is live IPTV, we use a static configuration file at each sNAP, which allows it to be preconfigured with the addresses of the IP multicast groups for which it will act as a sender.

Based on the above, an ICN NAP performs the following operations at startup, due to its potential role as an sNAP:

- Load the served IP multicast group addresses from a static configuration file.
- For every IP address in the configuration file:

- ○ Create a database row with a matching multicast IP group column and insert an empty list of NIDs to that row. This list will be used to track the cNAPs that want to receive data for that group.
- ○ Subscribe to the corresponding control channel CID, i.e. to /root/C/hash(CIDc). This indicates to the network that this sNAP is a potential source for the group.

The NAP only initializes an empty multicast database for its role as a cNAP; everything else will be automatically configured during its operation.

In order for an IP terminal to join an IP multicast group as a receiver, it sends a join request (formally, an IGMPv2 <MEMBERSHIP_REPORT> to that IP multicast group address) towards its local cNAP. Upon receiving the <MEMBERSHIP_REPORT>, the cNAP checks its internal database for a row for the IP multicast address in the request. If a row is found, this means that another UE is already receiving multicast packets from this cNAP, therefore there is no need to do anything else. If no such row is found, a new row is created and the cNAP sends a publish_data_isub() control message for the CIDc name; this is received by the sNAP subscribed to that CIDc on behalf of the corresponding IP multicast sender. Within the control message, the cNAP includes its own NID information, as well as the IP multicast group address it intends to join.

The inclusion of the NID and the implicit subscription may allow IP multicast packets to flow from the sNAP to the cNAP without involving the RV and TM components. Specifically, the pub_data_isub() will only involve the domain-local RV/TM functions in cases where the NID has not before been published to. Note, however, that this takes place at the POINT core and not at the handler.

On the sNAP side, the NAP looks for an entry in its own DB for the provided IP multicast address upon receiving a publish_data_isub(). If no such row is found, the request is ignored, as it means that the sNAP does not have any attached senders for this group. Otherwise, the NID provided in the publish_data_isub() message is checked against the NIDs column of the appropriate IP multicast address row. If it is already included, the message is ignored as a duplicate; if not, then it is added to the NIDs, signifying that a new cNAP will receive multicast data for that group.

In order for an IP UE to leave an IP multicast group, is sends a leave request (i.e, an IGMPv2 <LEAVE> message to a fixed address). Upon receiving a <LEAVE>, the cNAP checks its internal DB for the appropriate IP multicast address row. If no such row is found, the request is ignored; the group is not active at this cNAP. If it is found, the cNAP sends a number of IGMPv2 <GROUP_SPECIFIC_QUERY> messages to the multicast address of the group. If a <MEMBERSHIP_REPORT> arrives for the group, then the leave request is ignored; there are still local IP UEs that want to receive the group. If no reports arrive, then the cNAP deletes the DB entry for the group and it forwards the leave request, along with its own NID, to the subscribed sNAP(s) for the appropriate CIDc, again via a publish_data_isub() message.

At the sNAP side, receiving such a publish_data_isub() message triggers a search for an entry in the DB of the sNAP for the provided IP multicast address. If no such row is found, the request is ignored. Otherwise, the NID provided in the publish_data_isub() message is removed from the NIDs column of the appropriate IP multicast address row.

We assume only single sender cases for each group. However, different groups can use different sNAPs as their entry points to the network, by simply configuring each sNAP to subscribe to the corresponding control channel names. When a cNAP sends a join or a leave request by publishing it to the control channel, the proper sNAP receives it and acts accordingly.

Finally, the NAPs need to generate and consume additional IGMPv2 control messages to properly integrate with the rest of the network. In our particular setup, we assume that the POINT network replaces a network of IGMPv2 snooping switches, that is, the sNAP is downstream from such a switch and the cNAP is upstream of one or more such switches.

At the sNAP side, the sNAP receives IGMPv2 <MEMBERSHIP_QUERIES>, which are sent periodically. It responds with one <MEMBERSHIP_REPORT> for each group in its local DB that has at least one cNAP active, to ensure that the IP multicast packets will keep arriving. When a group becomes active for the first time (the first cNAP is added to the multicast DB for this group), the sNAP sends a number of IGMPv2 <MEMBERSHIP_REPORT> messages upstream, to trigger transmission of that group. When a group becomes inactive (the last cNAP is removed from the multicast DB for this group), the sNAP sends an IGMPv2 <LEAVE> message upstream, to trigger removal of that group; it is expected that a number of <GROUP_SPECIFIC_QUERY> messages will then arrive for the group, which will be ignored, as the group is no longer active.

At the cNAP side, we periodically send IGMPv2 <MEMBERSHIP_QUERIES> and process the corresponding <MEMBERSHIP_REPORT> messages. Specifically, each active group has a counter in the multicast DB row, which is incremented every time a query is sent and zeroed when a report arrives. If the counter reaches a predefined number, meaning that the corresponding number of queries was sent with no response, the group is assumed inactive, and is deleted, triggering the sending of a leave request to the sNAP, but not the group specific queries used when a normal <LEAVE> arrives, as we have already probed the group.

3.IP Multicast Handler

The IP Multicast Handler is quite simple, since all the IGMPv2 and ICN signalling, takes place at the IGMP handler. Essentially, the IP Multicast Handler is only in charge of receiving IP multicast packets coming to the sNAP from a server, forwarding them through the POINT network if their IP multicast address has active subscribers and having the receiving cNAPs forward those packets via IP multicast towards the UEs.

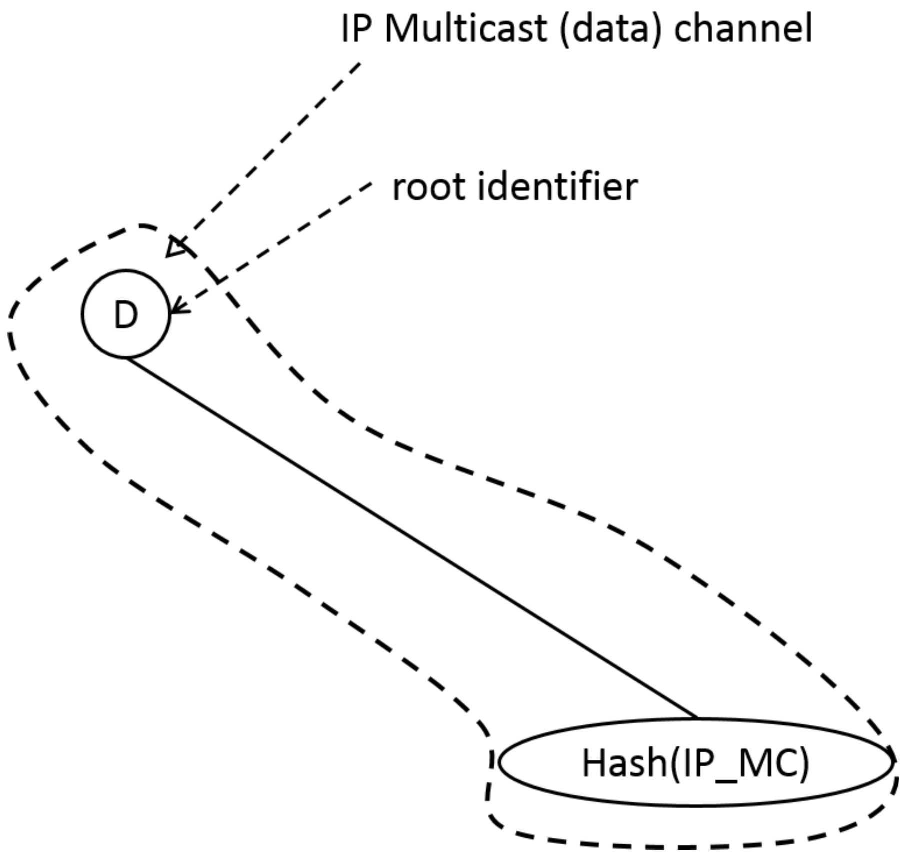

Figure 2: Namespace for IP multicast-over-ICN.

Figure 2 shows the ICN namespace used by the IP multicast handler. A separate root identifier is used for IP multicast, since packets are sent over a long-lived channel via multicast from the sNAP to the cNAPs. A flat namespace of data channel identifiers is used under this scope which is created by hashing the corresponding IP Multicast addresses, mirroring the control channels in the IGMP handler.

IP-enabled UEs can send IP multicast packets to a registered IP multicast group via their sNAP. Upon receiving an IP multicast packet, the sNAP checks its internal DB for the row with the appropriate IP multicast group. If no such row exists or if it exists but the number of NIDs in it is zero, the packet is ignored. If the number of cNAP NIDs is non-zero, the sNAP publishes the packet under the CIDd ICN name corresponding to its IP address. The FID used for the publication is calculated by the POINT core, transparently to the handler.

When the FIDs need to be recalculated due to a new join or leave, data packet forwarding to cNAPs is temporarily paused by the POINT core. As a result, multicast packets are buffered until the POINT core signals that the channel is available again, at which point data forwarding resumes with the buffered packets, but now also reaching the new cNAP. Finally, packet forwarding from the cNAP to the client UEs takes place with ordinary IP multicast.

5.References

[1] Xylomenos, George, et al. “”Scenarios, Requirements, Specifications and KPIs, final version”, 2017