

Protocol Independent Multicast in OMNeT++

Vladimír Veselý, Ondřej Ryšavý, Miroslav Švéda

Department of Information Systems

Faculty of Information Technology, Brno University of Technology (FIT BUT)

Brno, Czech Republic

e-mail: {ivesely, rysavy, sveda}@fit.vutbr.cz

Abstract—Multicast transmission for one-to-many data delivery and for online media streaming are becoming more and more popular. Interest in proper simulation and modeling has increased together with those two trends. This paper introduces simulation modules for dynamic multicast routing, namely Protocol Independent Multicast and its variants Dense Mode and Sparse Mode. Both of them are now parts of our ANSA extension developed within the INET framework for OMNeT++ discrete event simulator.

Keywords—Multicast Routing; PIM-DM; PIM-SM.

I. INTRODUCTION

The multicast transfers prove to be more efficient for one-to-many data delivery if there is one (or more) known source(s) and a number of unknown destinations ahead [1]. Multicast spares network resources, namely bandwidth. Sender and receivers communicate indirectly instead of many separate connections between them. Because of that, multicast traffic is carried across each link only once and the same data are replicated as close to receivers as possible. However, this effectiveness goes concurrently with increased signalization and additional routing information exchange which is done by the following protocols:

- IGMP (Internet Group Management Protocol) [2] /MLD (Multicast Listener Discovery) [3] – End-hosts and first hop multicast-enable routers are using IGMP and MLD protocols for querying, reporting and leaving multicast groups on local LAN segments – they announce their willingness to send or receive multicast data. IPv6 MLD is descendent of IPv4 IGMP, but both protocols are identical in structure and message semantic.
- DVMRP (Distance Vector Multicast Routing Protocol) [4], MOSPF (Multicast Open Shortest-Path First) [5], PIM (Protocol Independent Multicast) – All of them are examples of multicast routing protocols that build multicast topology in router control plane to distribute multicast data among networks. DVMRP and MOSPF are closely tight to the particular unicast routing protocol (RIP, OSPF), whereas variants of Protocol Independent Multicast (PIM) are independent by design and they are using information inside unicast routing table more generally.

End-hosts and routers maintain multicast connectivity with the help of previously mentioned protocols. Nowadays,

computer network design suggests deployment of IGMP/MLD on the access layer and PIM on the distribution layer of hierarchical internetworking model.

The project ANSA (Automated Network Simulation and Analysis) running at the Faculty of Information Technology is dedicated to develop the variety of software tools that can create simulation models based on real networks and subsequently allow for formal analysis and verification of target network configurations. One of our future goals is to model multicast flows in the Brno University of Technology network and thus implementing PIM models is one of our milestones. This report outlines two simulation modules (first one revisited and second one new), which are part of the ANSA project and which are extending functionality of the INET framework in OMNeT++.

This paper has the following structure. The next section covers a quick overview of existing OMNeT++ simulation modules relevant to the topic of this paper. Section 3 treats about design of the relevant PIM models. Section 4 presents validation scenarios for our implementations. The paper is summarized in Section 5 together with unveiling our future plans.

II. STATE OF THE ART

The current status of multicast support in OMNeT++ 4.3.1 and INET 2.2 framework is according to our knowledge as follows. We have merged functionality of generic IPv4 Router and IPv6 Router6 nodes, so that we created the dual-stack capable router – **ANSARouter**.

The module `RoutingTable` has been updated and since INET 2.0 it supports multicast routes and appropriate functions enabling to find the best matching record for the target multicast group.

The basic goal behind our research is to support native multicast transfers together with dynamic multicast routing using PIM. Hence, we have decided to start with IPv4 multicast and to add missing functionality in form of simulation modules directly connected to `networkLayer` module, as depicted on Figure 1.

We have searched in scientific community around simulation and modeling for other PIM implementations prior to our work. Limited versions exist for NS-2 [6] or OPNET [7]. However, none of them provide robust implementation. Also, existing OMNeT++ multicast attempts proved to be depreciated [8].

OMNeT++ state of the art prior to this paper is the result of our ongoing research covered in our other articles about IGMPv2 and initial PIM Dense Mode multicast support [9].

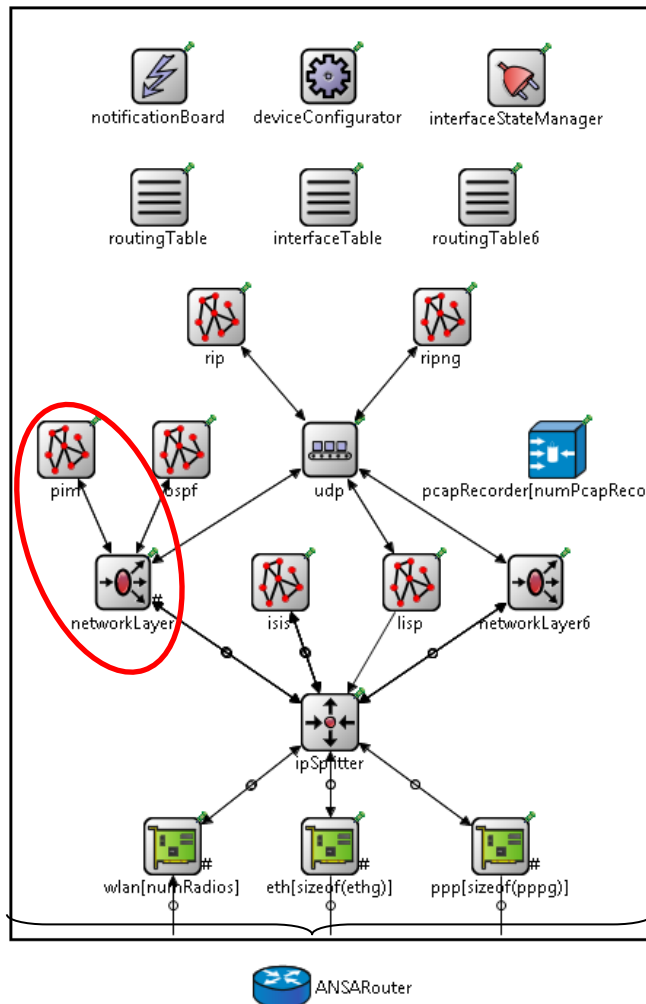


Figure 1. ANSARouter structure with highlighted contribution

III. IMPLEMENTATION

A. Theory of Operation

All multicast routing protocols provide a function to answer the question, “How to create routing path between sender(s) and receivers?” Baselines for this functionality are distribution trees of the following two types:

Source trees – The separate shortest path tree is built for each source of multicast data. A sender is the root and receivers are the leaves. However, memory and computation overhead causes this type is not scalable in the case of a network with many sources of multicast. In these situations usually the Shared tree is used.

Shared trees – A router called Rendezvous Point (RP) exist in a topology that serves as a meeting point for the traffic from multiple sources to reach destinations. The shared tree interconnects RP with all related receivers.

There are four PIM operational modes: PIM Dense Mode (PIM-DM), PIM Sparse Mode (PIM-SM), Bidirectional PIM (BiDir-PIM) and PIM Source-Specific Multicast (PIM-SSM). All of them differ in signaling, employed distribution trees and suitable applications.

Multicast routing support is performed by one dedicated router on each LAN segment elected based on *PIM Hello* messages. This router is called **designated router (DR)** and it is the one with highest priority or highest IP address.

PIM-DM is recommended for topologies with only one multicast source and lots of receivers. PIM-DM can be easily deployed without burdening configuration on active devices. However, PIM-DM does not scale well when number of sources increases. For this situation or for topologies with sparsely connected receivers, PIM-SM is suggested to be employed. Sparse mode scales much better in large topologies comparing to Dense mode, but configuration and administration is more complicated. PIM-SSM suits for multicast groups containing multiple sources providing the same content where client using IGMPv3 or MLDv2 may specify from which particular source it wants to receive data. BiDir-PIM is intended for topologies where many-to-many communication occurs. Currently, PIM-DM and PIM-SM are widely deployed PIM variants. Hence, we decided to implement them as the first.

PIM-DM idea consists of initial data delivery to all multicast-enable destinations (to flood multicast traffic everywhere), where routers prune themselves explicitly from the distribution tree if they are not a part of the multicast group. PIM-DM is not taking advantage of RP; thus, it is using source trees only.

PIM-DM routers exchange following messages during operation:

- *PIM Hello* – Used for neighbor detection and forming adjacencies. It contains all settings of shared parameters used for DR election;
- *PIM Prune/Join* – Sent towards upstream router by downstream device to either explicitly prune a source tree, or to announce willingness to receive multicast data by another downstream device in case of previously solicited *PIM Prune*;
- *PIM Graft* – Sent from a downstream to an upstream router to join previously pruned distribution tree;
- *PIM Graft-Ack* – Sent from an upstream to a downstream router to acknowledge *PIM Graft*;
- *PIM State Refresh* – Pruned router refreshes prune state upon receiving this message;
- *PIM Assert* – In case of multi-access segment with multiple multicast-enabled routers one of them must be elected as an authoritative spokesman. Mutual exchange of *PIM Asserts* accomplishes this operation.

On the contrary to PIM-DM, **PIM-SM** works with different principle where initially no device wants to receive multicast. Thus, all receivers must explicitly ask for multicast delivery and then routers forward multicast data towards end-hosts. PIM-SM employs both types of multicast

distribution trees. Sources of multicast are connected with RP by source trees – source of multicast is the root of a source tree. RP is connected with multicast receivers by shared trees – RP is the root of shared tree. Multicast data are traversing from sources down by source tree to RP and from here down by shared tree to receivers. PIM-SM cannot work properly as long as all PIM routers in a network do not know exactly which router is RP for a given multicast group.

PIM-SM exchanges subsequent message types:

- *PIM Hello* – same as PIM-DM;
- *PIM Register* – Sent by source’s DR towards RP whenever new source of multicast is detected;
- *PIM Register-Stop* – Solicited confirmation of *PIM Register*. It is sent by RP in reverse direction that source’s DR can stop registering process of a new source. RP is aware of multicast data and may send them to receivers via shared tree;
- *PIM Prune/Join* – This message forms the shape of source and shared distribution trees. Multiple sources could provide data to the same multicast group – each one of them sends data via own source tree towards RP, from here data are reflected to receivers via shared tree;
- *PIM Assert* – same as PIM-DM.

The thorough survey on PIM-DM and PIM-SM message exchange scenarios are out of scope of this paper. More can be found in RFC 3973 [10] and RFC 4601 [11]; let us state that our implementations (i.e., finite-state machines, message structure, etc.) fully comply with IETF’s standards.

B. Design

We have synthesized multiple finite-state machines that describe behavior of PIM-DM and PIM-SM with reference to used timers and exchanged PIM messages [12].

Figure 2 shows implemented architecture of the pim module:

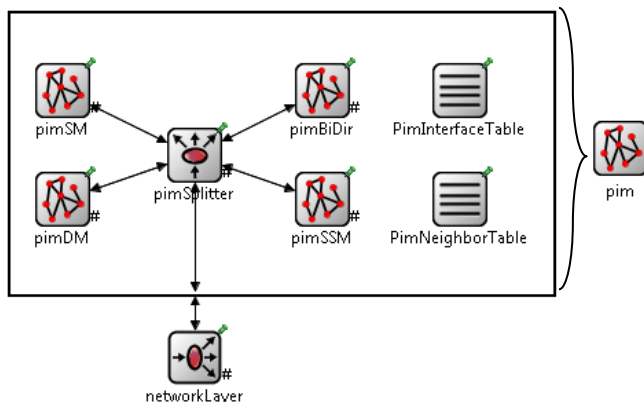


Figure 2. Proposed PIM module design

Besides previous modules, there were also some minor alternations to IPv4 networkLayer as well as to IPv4 routingTable module.

Implementation is done in NED (model design) and C++ (model behavior) languages. Brief description of implemented components is summarized in following table:

TABLE 1. DESCRIPTION OF PIM SUBMODULES

Name	Description
pimSplitter	This submodule is connected with INET networkLayer. It inspects all PIM messages and passes them to appropriate PIM submodules.
pimDM	The main implementation and logic of PIM-DM protocol is over here.
pimSM	The main implementation and logic of PIM-SM protocol is over here.
pim InterfaceTable	Stores all PIM relevant information for each router’s interface.
pim NeighborTable	Keeps state of formed PIM adjacencies and information about neighbors (PIM version they are using, priorities, neighbors IPs).
pimSSM, pimBiDir	Prepared as a placeholder for upcoming implementations of BiDir-PIM and PIM-SSM variants.

IV. TESTING

In this section, we provide information on testing and validation of our implementations using several test scenarios. We compared the results with the behavior of referential implementation running at Cisco routers. We have built exactly the same topology and observed (using transparent switchport analyzers and Wireshark) relevant messages exchange between real devices (Cisco 2811 routers with IOS c2800nm-advipservicesk9-mz.124-25f.bin and host stations with FreeBSD 8.2 OS).

A. PIM-DM

In this testing network (topology is shown on Figure 3), we have three routers (R1, R2 and R3), two sources of multicast (Source1 and Source2) and three receivers (Host1, Host2 and Host3).

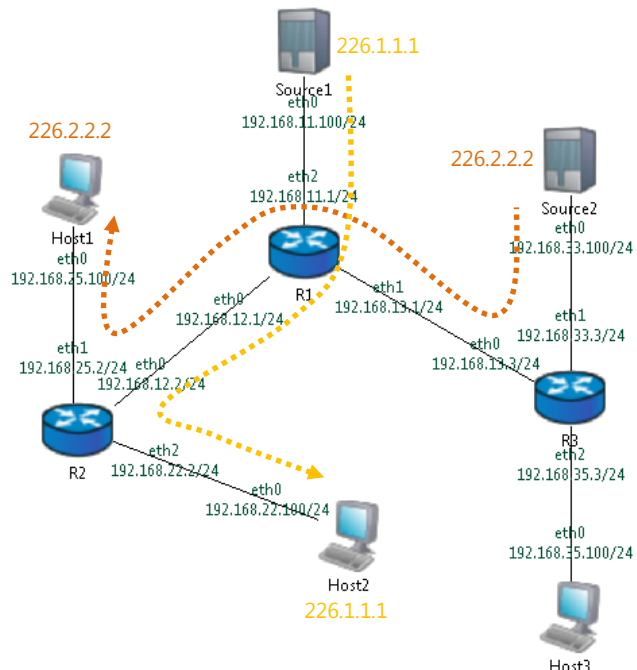


Figure 3. PIM-DM testing topology

We scheduled actions like source’s start and stop sending and host’s start and stop receiving of multicast data. Scheduled scenario is summarized in Table 2.

TABLE 2. PIM-DM EVENTS SCENARIO

Phase	Time [s]	Device	Multicast action	Group
#1	0	Host1	Starts receiving	226.2.2.2
#2	87	Source1	Starts sending	226.1.1.1
#3	144	Host2	Starts receiving	226.1.1.1
#4	215	Source2	Starts sending	226.2.2.2
#5	364	Host2	Stops receiving	226.1.1.1
#6	399	Source2	Stops sending	226.2.2.2

Hosts sign themselves to receive data from particular multicast group via *IGMP Membership Report* message during phases #1 and #3. Similarly, the host uses *IGMP Leave Group* message to stop receiving data during phases #5 and #6.

- #1) There are no multicast data transferred. Only *PIM Hellos* are sent between neighbors.
- #2) First multicast data appear but, because of no receivers, routers prune themselves from source distribution tree after initial flooding.
- #3) Host2 starts to receive data from group 226.1.1.1 at the beginning of #3. This means that R2 reconnects to source tree with help of *PIM Graft* which is subsequently acknowledged by *PIM Graft-Ack*.
- #4) The new source starts to send multicast data. All routers are part of the source distribution tree with R3 as the root. R3 acts as RP that is illustrated on Figure 4.

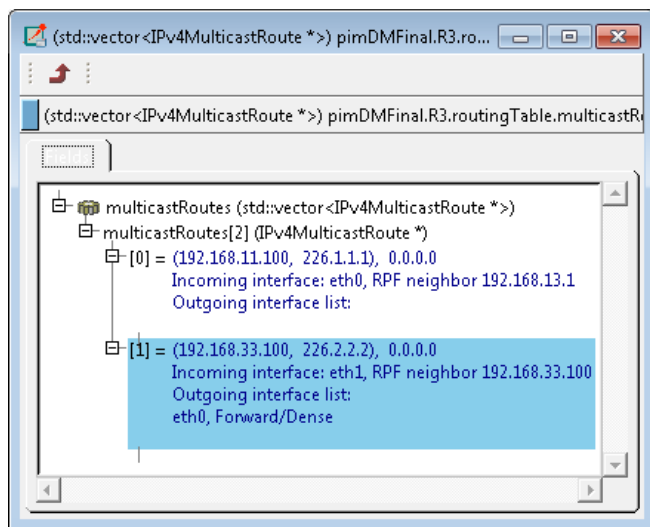


Figure 4. R3 multicast routing table after phase #4

- #5) Host2 is no longer willing to receive multicast from 226.1.1.1 and, because Host2 is also the only listener to this group, R2 disconnects itself from distribution tree with *PIM Prune/Join*.
- #6) Finally Source2 stops sending data to the group 226.2.2.2 at the beginning of #6. Subsequent to this, no PIM message is generated. Routers just wait for

180 seconds and then wipe out an affected source tree from the multicast routing table.

The confluence of messages proved correctness of our PIM-DM implementation by simulation as well as by real network monitoring, which can be observed in Table 3.

TABLE 3. TIMESTAMP COMPARISON OF PIM-DM MESSAGES

Phase	Message	Sender	Simul. [s]	Real [s]
#1	<i>PIM Hello</i>	R1	30.435	25.461
#2	<i>PIM Prune/Join</i>	R3	87.000	87.664
#3	<i>PIM Graft</i>	R2	144.000	144.406
	<i>PIM Graft-Ack</i>	R1	144.000	144.440
#5	<i>PIM Prune/Join</i>	R2	366.000	364.496

A. PIM-SM

For testing purposes of PIM-SM, topology is more complex. We have two designated routers (DR_R1, DR_R2) for receivers (Receiver1, Receiver2), two DRs (DR_S1, DR_S2) for sources (Source1, Source2) and one rendezvous point (RP). The scenario is depicted on Figure 5.

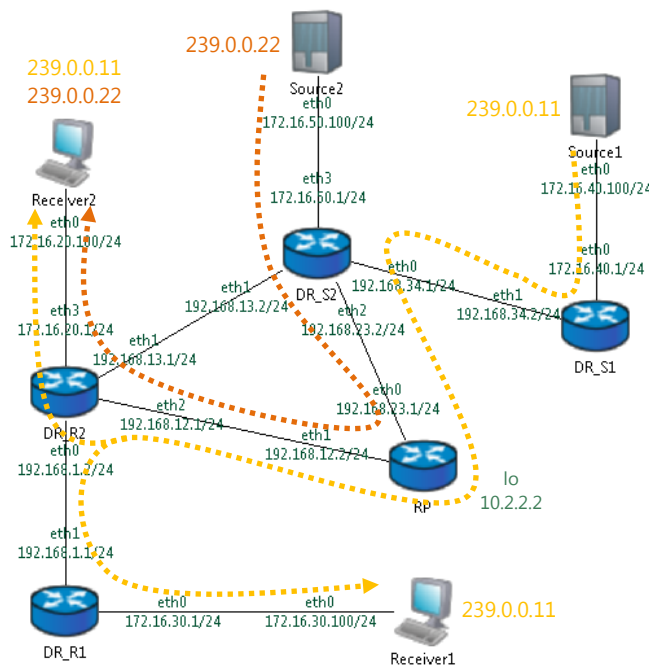


Figure 5. PIM-SM testing topology

A scenario for PIM-SM is summarized in Table 4 and additional description of actions follows below:

TABLE 4. PIM-SM EVENTS SCENARIO

Phase	Time [s]	Device	Multicast action	Group
#1	10	Source1	Starts sending	239.0.0.11
#2	20	Receiver1	Starts receiving	239.0.0.11
#3	25	Receiver2	Starts receiving	239.0.0.11
#4	40	Receiver2	Starts receiving	239.0.0.22
#5	60	Source2	Starts sending	239.0.0.22
#6	90	Receiver1	Stops receiving	239.0.0.11
#7	120	Receiver2	Stops receiving	239.0.0.11
#8	220	Receiver2	Stops receiving	239.0.0.22
#9	310	Source1	Stops sending	239.0.0.11
#10	360	Source2	Stops sending	239.0.0.22

Just as in PIM-DM scenario, receivers send *IGMP Membership Report* and *IGMP Leave Group* messages to sign on and off the multicast groups during phases #2, #3 and #6-#8.

- #1) Source1 starts to send multicast data. Those data are encapsulated into *PIM Register* message sent by DR_S1 via DR_S2 towards RP. Following next RP responds with *PIM Register-Stop* back to DR_S1, thus registration of new source is finished.
- #2) *IGMP Membership Report* for multicast group 239.0.0.11 by Receiver1 turns on joining process of DR_R1 and DR_R2 to shared tree and joining of RP and DR_S2 to source tree by sending *PIM Join/Prune*.
- #3) DR_R2 is already connected to a shared tree, thus *IGMP Membership Report* only adds another outgoing interface to shared tree as could be seen on Figure 6.

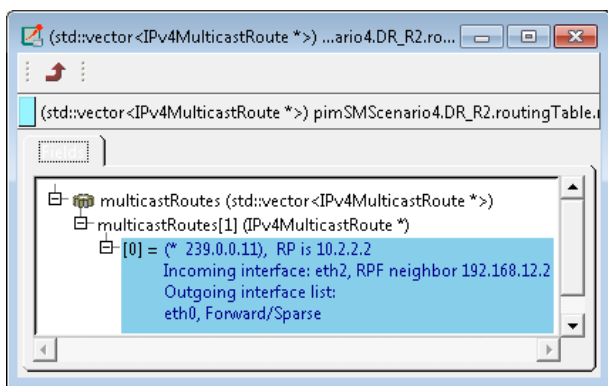


Figure 6. DR_R2 multicast routing table after phase #3

- #4) Whenever Receiver2 starts receiving multicast group 239.0.0.22, new multicast route is added on DR_R2 (see Figure 7). Subsequently DR_S2 joins to shared tree via *PIM Join/Prune* sent towards RP.

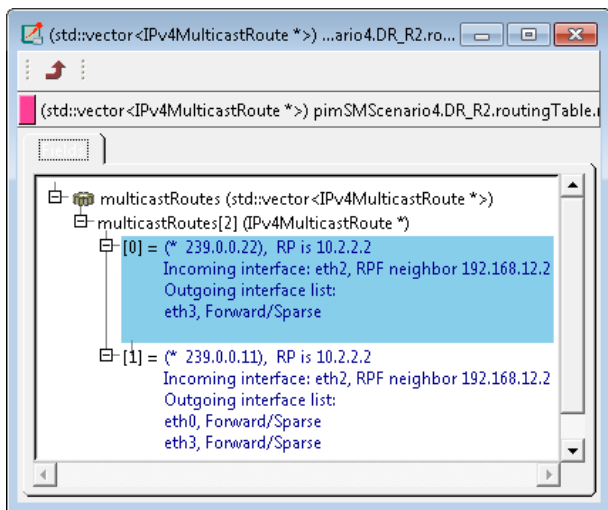


Figure 7. DR_R2 multicast routing table after phase #4

- #5) Source2 starts sending multicast data to 239.0.0.22 after Receiver1 already joined the shared tree. DR_S2 registers source with *PIM Register* that contains also multicast data. Those data are decapsulated and sent down via shared tree to receivers. As a next step RP joins the source tree via *PIM Prune/Join* message and a moment later it confirms registration via *PIM Register-Stop* sent towards DR_S2. Multicast routes on RP converged and they could be observed on Figure 8.

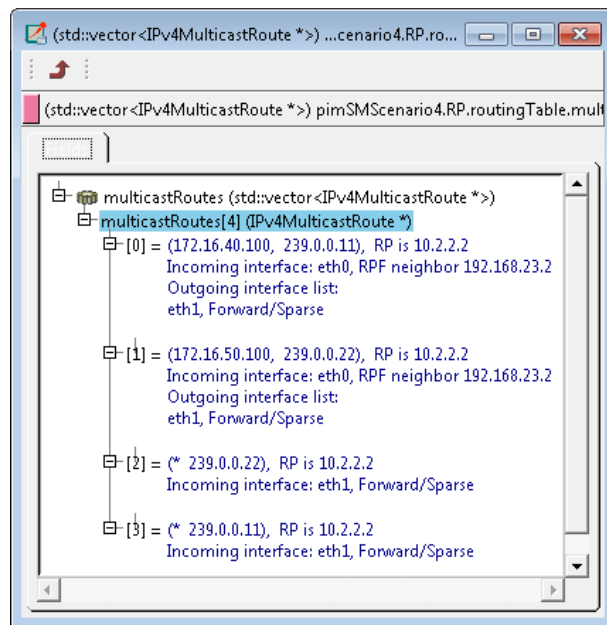


Figure 8. RP multicast routing table after phase #5

- #X) Every 60 second after successful source registration, the given DR and RP exchange empty *PIM Register* and *PIM Register-Stop* messages to confirm presence of multicast source. Also every 60 seconds after last receiver joined multicast group, PIM router refreshes upstream connectivity to any tree via *PIM Prune/Join* message. This phase cannot be planned or scheduled; it is default behavior of PIM-SM protocol finite-state machine. It is illustrated only once for Source1 distribution trees but the same message exchange happens also for Source2.
- #6) Upon receiving *IGMP Leave Group*, DR_R1 prunes itself from shared tree via *PIM Prune/Join* message sent upstream to DR_R1. DR_R1 then removes interface eth0 as outgoing interface for multicast group 239.0.0.11.
- #7) Receiver2 decides not to receive multicast from Source1. Its *IGMP Leave Group* starts pruning process that goes from DR_R2 up to DR_S1. On each interim PIM router, multicast route for 239.0.0.11 is removed via *PIM Prune/Join* message.
- #8) Later Receiver2 signs off from receiving 239.0.0.22 which causes similar exchange of *PIM Prune/Join* as in case of #7.

#9) Whenever Source1 stops sending multicast, elimination process starts for a given multicast route. As time goes by, ExpireTimer times out on every PIM router and multicast distribution tree for 239.0.0.11 is wiped out from routing table. The same approach applies for #10.

Validation testing against real-life topology shows just reasonable time variations (around ±3 seconds). This variation observable on real Cisco devices is caused by two factors: a) control-plane processing delay; b) stochastic message jitter to avoid potential race conditions in similar processes. Table 5 outlines results:

TABLE 5. TIMESTAMP COMPARISON OF PIM-SM MESSAGES

Phase	Message	Sender	Simul. [s]	Real [s]
#1	<i>PIM Register</i>	DR_R1	10.005	10.127
	<i>PIM Register-Stop</i>	RP	10.006	10.380
#2	<i>PIM Prune/Join</i>	DR_R1	20.001	20.422
	<i>PIM Prune/Join</i>	DR_R2	20.002	20.813
	<i>PIM Prune/Join</i>	RP	20.003	21.117
	<i>PIM Prune/Join</i>	DR_S2	20.005	21.320
#4	<i>PIM Prune/Join</i>	DR_R2	40.001	43.524
#5	<i>PIM Register</i>	DR_S2	60.000	61.459
	<i>PIM Prune/Join</i>	RP	60.003	61.970
	<i>PIM Register-Stop</i>	RP	60.004	62.758
#X	<i>PIM Register</i>	DR_S1	70.008	74.304
	<i>PIM Register-Stop</i>	RP	70.009	75.671
	<i>PIM Prune/Join</i>	DR_R1	80.000	83.041
	<i>PIM Prune/Join</i>	DR_R2	80.001	83.647
	<i>PIM Prune/Join</i>	RP	80.003	83.950
	<i>PIM Prune/Join</i>	DR_S2	80.003	84.004
#6	<i>PIM Prune/Join</i>	DR_R1	90.000	92.909
#7	<i>PIM Prune/Join</i>	DR_R2	120.001	122.311
	<i>PIM Prune/Join</i>	RP	120.002	122.704
	<i>PIM Prune/Join</i>	DR_S2	120.003	123.296

V. CONCLUSION AND FUTURE WORK

In this paper, we discussed options for dynamic multicast routing. We presented an overview of currently existing modules relevant to above topics in OMNeT++. The main contributions are simulation models for PIM-DM and PIM-SM that extend functionality of our ANSARouter and overall INET framework. Also, we introduce simulation scenarios and their results, which show that our implementations comply with relevant RFCs and referential behavior on Cisco devices.

We plan to wrap up our native IPv4 multicast implementation and complete it with integrating MLD support. After finishing this, we would like to focus on IPv6 and Source-Specific Multicast simulations.

More information about ANSA project is available on webpage [13]. Source codes of simulation modules could be downloaded via GitHub repository [14].

ACKNOWLEDGMENT

This work was supported by the Brno University of Technology organization and by the research grant IT4Innovation ED1.1.00/02.0070 by Czech Ministry of Education Youth and Sports.

REFERENCES

- [1] G. Phillips et al., "Law, Scaling of Multicast Trees: Comments on the Chuang-Sirbu Scaling", in Proceedings of ACM SIGCOMM '99, Cambridge-Boston, 1999.
- [2] B. Cain et al., "Internet Group Management Protocol, Version 3", October 2002. [Online]. Available: <https://tools.ietf.org/html/rfc3376>. [Accessed: 2013].
- [3] R. Vida and K. Costa, "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", June 2004. [Online]. Available: <http://tools.ietf.org/html/rfc3810>. [Accessed: 2013].
- [4] D. Waitzman, C. Partridge and S. Deering, "Distance Vector Multicast Routing Protocol", November 1988. [Online]. [Accessed: <http://tools.ietf.org/html/rfc1075> 2013].
- [5] J. Moy, "Multicast Extensions to OSPF", March 1994. [Online]. Available: <http://tools.ietf.org/html/rfc1584>. [Accessed: 2013].
- [6] T. Henderson, November 2011. [Online]. Available: <http://www.isi.edu/nsnam/ns/doc/node338.html>. [Accessed: 2013].
- [7] C. Adam, Y. Chien, May 1998. [Online]. Available: <http://www.cs.columbia.edu/~hgs/teaching/ais/1998/projects/pim/report.html>. [Accessed: 2013].
- [8] R. Leal, J. Cacinero and E. Martin, "New Approach to Inter-domain Multicast Protocols", ETRI Journal, vol. 33, issue 3, pp. 355-365, June 2011.
- [9] V. Veselý, O. Ryšavý and M. Švéda, "IPv6 Unicast and IPv4 Multicast Routing in OMNeT++", in SimuTools '13 Proceedings of the 6th International ICST Conference on Simulation Tools and Techniques, Cannes, France, 2013.
- [10] A. Adams, J. Nichols and W. Siadak, "Protocol Independent Multicast - Dense Mode (PIM-DM)", January 2005. [Online]. Available: <http://tools.ietf.org/html/rfc3973>. [Accessed: 2013]
- [11] B. Fenner et al., "Protocol Independent Multicast - Sparse Mode (PIM-SM)", August 2006. [Online]. Available: <http://tools.ietf.org/html/rfc4601>. [Accessed: 2013]
- [12] V. Rybová and T. Procházka, Brno University of Technology, October 2013. [Online]. Available: <https://nes.fit.vutbr.cz/ansa/uploads/Main/pim-fsm.pdf>. [Accessed: 2014].
- [13] Brno University of Technology, January 2014. [Online]. Available: <http://nes.fit.vutbr.cz/ansa>. [Accessed: 2014].
- [14] GitHub, December 2013. [Online]. Available: <https://github.com/kvetak/ANSA>. [Accessed: 2014].