IPv6 Unicast and IPv4 Multicast Routing in OMNeT++

Vladimír Veselý, Ondřej Ryšavý, Miroslav Švéda
Faculty of Information Technologies
Brno University of Technology
Božetěchova 2, Brno – Královo Pole, 612 66, Czech Republic
+420 541 141321
{ivesely, rysavy, sveda}@fit.vutbr.cz

ABSTRACT
The deployment of IPv6 is accelerating with depletion of IPv4 address space in the last year. Audio/video streaming and overall effectiveness of multicast one-to-many data delivery are becoming popular nowadays. Interest in proper simulation and modeling has increased together with those two trends. This paper introduces two brand-new simulation modules for RIPng and PIM-DM dynamic routing protocols, which are now parts of our ANSA extension, built over the INET framework.

Categories and Subject Descriptors
C.2.1 [**Network Architecture and Design**]: Network Communications and Network Topology—*multicast communication*; I.6.5 [**Model Development**]: Modeling methodologies—*OMNeT++, INET Framework and ANSA extension*;

General Terms
Design, Experimentation.

Keywords
IPv6 unicast routing, multicast routing, RIPng, PIM-DM, ANSA extension, OMNeT++, INET framework.

1 INTRODUCTION
Despite the fact that IPv6 is more than 16 years old [1], the major public interest in it has been raised quite recently since official IPv4 address space was exhausted on 31st January 2011. It seems that IPv6 is currently the most widely accepted as possible solution for current problems of the Internet (i.e. mobility, growth of DFZ routing tables, multihoming), even though it is not the only option. Thus, it is more and more deployed by ISPs and web service operators (like Google, Facebook, Microsoft, etc.) providing native IPv6 connectivity to their customers and users.

Dynamic IPv6 unicast routing protocols have origins in IPv4 routing protocols. Below are following representatives commonly used in networks:

- RIPng – Routing Information Protocol: New Generation (RIPng) is distance-vector interior gateway protocol (IGP) based on messages and features of RIPv2;
- OSPFv3 – The link-state IGP based on messages and topology database of OSPFv2 but with the new link-state advertisement types;
- EIGRP – Flexible Cisco proprietary hybrid IGP where transition to IPv6 meant to create a new type-length-value (TLV) record;
- IS-IS – The link-state IGP that is agnostic to address family. IPv6 support is done by specifying new TLV record (the same principle as in the case of EIGRP);
- M-BGP – The path-vector exterior gateway protocol capable of carrying data from different address families by changing MP_REACH_NLRI and MP_UNREACH_NLRI attributes.

It appears that the main difference between IPv4 and IPv6 is in the structure of those protocols so that they are prepared for and could carry the different address family. But reality is a bit more complex because all of those protocols must also integrate the specifics of IPv6 – adjacency formed using link-local addresses, missing broadcast in IPv6, special multicast addresses, etc.

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. Multicast spares network resources, namely bandwidth. Sender and receivers communicate indirectly instead of maintain 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. But this effectiveness goes concurrently with increased signalization and additional routing information exchange which is done by following protocols:

- IGMP/MLD – End-hosts and first hop multicast-enable routers are using IGMP and MLD protocols for querying, reporting and leaving multicast groups on local LAN segment – they announce their willingness to send or receive multicast data. IPv6 MLD is descendent of IPv4 IGMP, but in reality both protocols are identical in structure and semantics of messages.
- DVMRP, MOSPF, PIM – 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 depend on used unicast routing protocol (RIP resp. OSPF), whereas variants of Protocol Independent Multicast (PIM) are independent by design and they are using information inside unicast routing table more generally.

Project ANSA (Automated Network Simulation and Analysis) running at our faculty 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 IPv6 multicast flows in the Brno University of

OMNeT++ 2013, March 5, Cannes, France.

DOI 10.4108/icst.simutools.

Technology network and thus implementing models of RIPng and PIM-DM is our first milestone. This report outlines two new simulation modules, which are part of the ANSA project and which are extending functionality of the INET framework in OMNeT++.

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

2 STATE OF THE ART

The current status of IPv6 and multicast support in OMNeT++ 4.2.2 and INET 2.0 framework is according to our knowledge as follows.

The IPv6 layer and datagrams together with static routing are already parts of the INET framework, namely in the modules `IPv6` and `RoutingTable6`. We have merged functionality of generic IPv4 `Router` and IPv6 `Router6` nodes so that we created the dual-stack capable router – **ANSARouter**.

`NetworkLayer` contains an interface for the IGMP module. Thus, one can use either official INET IGMPv2 or our own IGMPv2/v3 implementation [2].

The module `RoutingTable` has been recently updated to support multicast routes and appropriate functions enabling to find the best matching record for the target multicast group.

The basic motivation behind our work is to introduce dynamic IPv6 unicast and IPv4 multicast routing to the INET framework. Hence, we have decided to start with the simplest IPv6-enable unicast routing protocol, which is RIPng. We want to take another step in present multicast support and allow users to create more complex (and more real-world) simulations by extending the existing functionality with widely used PIM protocol family.

3 DESIGN AND IMPLEMENTATION

We have implemented compound modules and relevant submodules to support RIPng and PIM-DM functionality. From all PIM protocols PIM-DM is the simplest one (and also the least one to be deployed in current networks) but we decided to implement it first and the rest of PIM protocols in our multicast framework later. This section gives an overview of design and briefs some implementation specifics of each newly added features.

3.1 RIPng

RIPng [3] is the successor of RIP for IPv4. The main principle is still the same. RIPng-enabled routers periodically exchange routing information and run Bellman-Ford algorithm to determine the shortest path to the destination network.

The protocol uses the UDP port 521 and RIPng routers use reserved multicast address FF02:9 for communication on the LAN segments. The hop-count is used as a metric where the maximum value is 15. The routes with metric 16 are considered unreachable.

RIPng uses only two message types:

- *RIPng Request* message is generated whenever the router needs a routing information from its neighbors.
- *RIPng Response* message is further differentiated to:
  - *RIPng Regular Update* – Inside this message is the whole RIP content of routing table from the neighbor using the multicast address FF02::9 as a destination;
  - *RIPng Triggered Update* – Send to neighbor in case of topology change where the main goal is to inform about particular routes only.

RIPng employs the following three timers:

- *Regular Update Message Timer* – The interval between two consecutive *RIPng Regular Update* messages which is by default 30 seconds long;
- *Timeout* – The maximum time of waiting for fresh *RIPng Responses* of the same network after which the route is considered unreachable. It is by default 180 seconds long and it resets upon receive of relevant *RIPng Response*;
- *Garbage-Collection Time* (GCT) – It starts after expiration of *Timeout*. The target route is deleted from routing table when GCT expires. By default it is 120 seconds long.

We have added RIPng support (that extends entire functionality covered in RFC) in form of the `RIPngRouting` module which is connected to `UDP` and cooperates with `RoutingTable6` and `InterfaceTable`. Figure 1 depicts the overall ANSARouter architecture and how newly created modules are connected to existing ones:

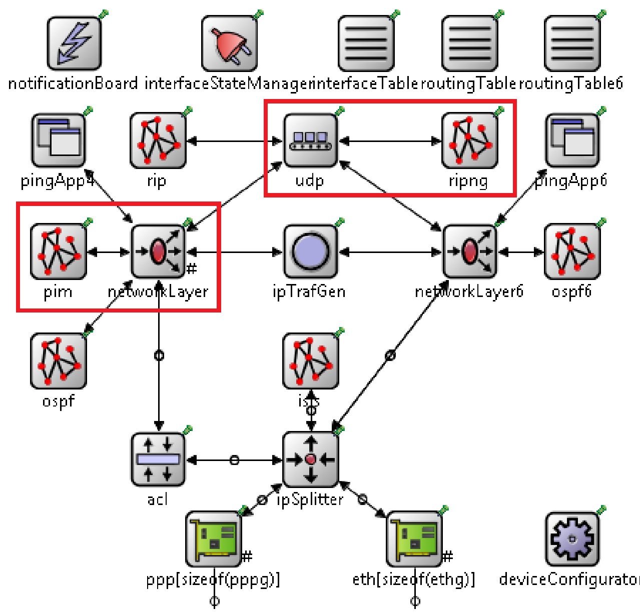

Figure 1. ANSARouter with highlighted `pim` and `ripng`

We have extended functionalities of the INET `RoutingTable` and `RoutingTable6` to prefer routing information based on a trustworthiness of the source (a.k.a. **administrative distance**) so that the same destination network is present in the table only once and there is a strict ladder of preference similar to Cisco devices (static > OSPF > RIP > BGP).

3.2 PIM-DM

All multicast routing protocols function to answer the question, "How to create routing path between sender(s) and receivers?" Baselines for this are distribution trees of 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. But memory and computation overhead causes this type to be not scalable in the case of a network with many sources of multicast. In these situations usually the Shared tree is being used.

Shared trees – A router called *Rendezvous Point* (RP) exist in a topology, which serves as a meeting point for the traffic from multiple different sources towards destinations. The shared tree interconnects RP with all multicast 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 signalization, employed distribution trees and suitable applications.

PIM-DM idea consists in initial deliver of data 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;
- *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 solicitated *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 receive of this message;
- *PIM Assert* – In case of multi-access segment with multiple multicast-enabled routers one must be elected as an authoritative spokesman. Mutual exchange of *PIM Asserts* accomplishes this operation.

The proper PIM-DM implementation must store these vital items:

- "For each PIM-enable interface": Hello Timer, State-Refresh capability, LAN delay capability, Propagation Delay, Override Interval, neighbor state.
- "For each source tree and…"
  - "…each interface": Prune State, Prune Pending Timer and Prune Timer + Asser Winner State, Assert Timer and winner information;
  - "…uplink interface": Graft/Prune State, Graft Retry Timer, Override Timer and Prune Limit Timer + Originator State, Source Active Timer and State Refresh Timer.

The detailed description of the previous items and the thorough survey on PIM-DM message exchange scenarios are out of scope of this paper. More can be found in RFC3973 [4]; let us state that our implementation fully complies with standard.

Brief description of implemented components:

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 behavior and logic of PIM-DM 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 neighbor information.
pimSM, pimSSM, pimBiDir	Prepared future submodule interfaces that yet to be implemented.

Figure 2 shows implemented architecture of the pim module:

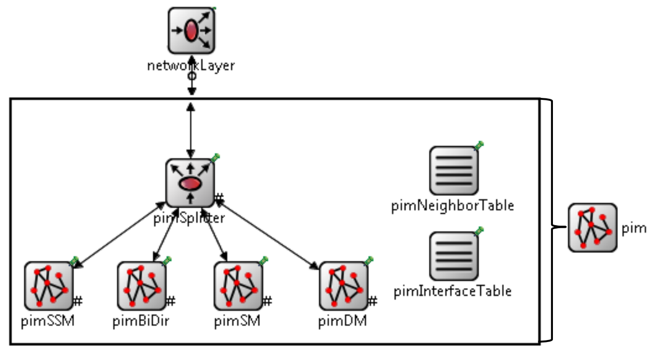

Figure 2. Proposed PIM module design

4 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 SPAN and Wireshark) relevant messages exchange between real devices (Cisco 2811 as routers and host stations with FreeBSD 8.2 OS).

4.1 RIPng scenario

Testing topology (see Figure 3) consists of three routers (R1, R2 and R3) and six StandardHost6 (LAN1-LAN6) which substitute whole separate LAN segments with dedicated IP networks.

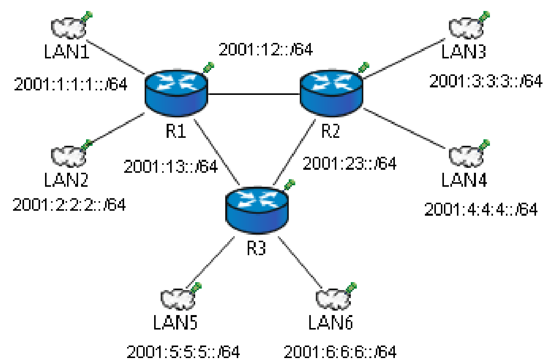

Figure 3. RIPng testing topology

Typical message exchange of freshly booted router R1 is:

#1) Router starts with sending *RIPng Request* asking its neighbors to reply back with all available RIP routes;
#2) Following this it generates *RIPng Regular Update* with directly connected networks (metric 1);
#3) Meantime *RIPng Requests* arrive from neighbors (R2/R3 on a link between R1-R2/R1-R3) querying R1 for routing information;
#4) R1 replies back to the unicast address of each neighbor with *RIPng Response* with all known RIP routes.

We scheduled link failure between routers R2 and R3 sometime later during network operation. Following events happen:

#5) R2 sends *RIPng Triggered Update* stating that network 2001:23::/64 has metric 16;
#6) Later the network converges (for instance the route between LAN5 and LAN3 goes via routers R3, R1 and R2). All routers

(including R1) start to exchange *RIPng Regular Updates* every 30 seconds.

Table 2 compares the timestamps (rounded to three decimal places) of selected messages (column "Message"), namely by whom (col. "Sender") and when they were generated in OMNeT++ simulation (col. "Simul.") and in real network (col. "Real"):

Table 2. Timestamp comparison of RIPng messages

Phase	Message	Sender	Simul. [s]	Real [s]
#1	*RIPng Request*	R1	0.000	0.000
#2	*RIPng Response*	R1	0.000	0.321
#3	*RIPng Request*	R2	0.001	0.620
#4	*RIPng Response*	R1	0.001	0.640
#5	*RIPng Response*	R2	211.000	211.888

4.2 PIM-DM scenario

In this testing network (topology is shown on Figure 4) we have three routers (R1, R2 and R3), two sources of multicast (Source1 and Source2) and three receivers (Host1, Host2 and Host3).

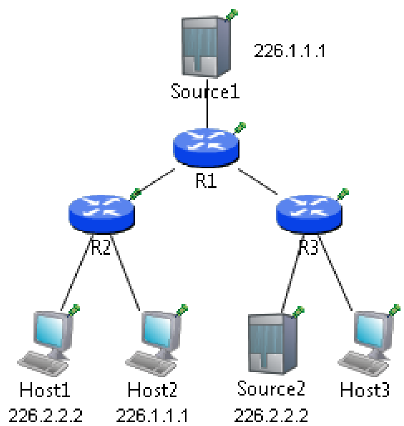

Figure 4. PIM-DM testing topology

We scheduled some actions and summarized them in Table 3.

Table 3. 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 receiving	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* generated by R1.
- #4) The new source starts to send multicast data. All routers are part of the source distribution tree with R3 as the root.
- #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, then R2 disconnects itself from distribution tree with *PIM Prune/Join* message.
- #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 proving correctness of our PIM-DM implementation from simulation as well as real network can be observed in Table 4.

Table 4. Timestamp comparison of PIM messages

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

5 CONCLUSION

In this paper we discuss options for modeling IPv6 dynamic routing and IPv4 multicast transfers. We present an overview of currently existing modules relevant to above topics in OMNeT++. The main contribution are simulation models for RIPng and PIM-DM 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.

5.1 Future work

We plan to carry on our work on IPv6 dynamic routing modules and to add support for OSPFv3. Moreover we would like to wrap up our native IPv4 multicast implementation and complete it with the PIM-SM module. After finishing this we would like to focus on IPv6 multicast.

5.2 Additional information

Some parts of this paper are based on work done by Jiří Trhlík, Tomáš Procházka and Veronika Rybová, students of Brno University of Technology.

More information about project is available on webpage http://nes.fit.vutbr.cz/ansa. Source codes could be downloaded via GitHub repository https://github.com/kvetak/ANSA.

6 REFERENCES

[1] Deering, S. and Hinden, R. *RFC 1883 - Internet Protocol, Version 6 (IPv6), Specification*. http://tools.ietf.org/html/rfc1883, IETF, December 1995.

[2] Veselý, V., Matoušek, P. and Švéda, M. Multicast Simulation and Modeling in OMNeT++. In *Proceedings of the IEEE 5th International ICST Conference on Simulation Tools and Techniques* (Desenzano del Garda, Italy March 2012), ICST, p. 298-301.

[3] Malkin, G. *RFC 2080 - RIPng for IPv6*. http://tools.ietf.org/html/rfc2080, IETF, January 1997.

[4] Adams, A., Nicholas, J. and Siadak, W. *RFC 3973 - Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification (Revised)*. http://tools.ietf.org/html/rfc3973, IETF, January 2005.