Enhanced Interior Gateway Routing Protocol with IPv4 and IPv6 Support for OMNeT++

Vladimír Veselý, Vít Rek, Ondřej Ryšavý

Brno University of Technology, Brno, Czech Republic
{ivesely, rysavy}@fit.vutbr.cz, xrekvi00@stud.fit.vutbr.cz

Abstract. Enhanced Interior Gateway Routing Protocol seized to be Cisco proprietary since its release in form of IETF's informational draft. EIGRP has history of hybrid routing protocol widely deployed by Cisco's customers due to its performance and advanced features. This paper introduces free-available simulation module that complies with RFC specification and offers platform for EIGRP's further testing and evaluation within OMNeT++ simulator.

Keywords: EIGRP · routing protocol · hybrid distance vector · simulation module · OMNeT++.

1 Introduction

The ANSA project (Automated Network Simulation and Analysis) running at our university 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. It might be used by public as the routing/switching baseline for further research initiatives using simulator for verification. This paper extends our previous work involving EIGRP [1] by updated version of our simulation module, which is a part of the ANSA project and which extends functionality of the INET framework [2] in OMNeT++ [3].

Network layer serves the purpose of end-to-end data delivery. Routers employ routing tables to make correct routing decisions in order to pass packet closer to receiver. Dynamic routing protocols maintain up-to-date content of routing tables by exchanging updates about known networks.

Routing protocols for traditional wired networks could be divided into three categores: a) **distance-vector** where routing is based on information provided by neighbors and each route has one attribute representing distance of network from a given router; b) **path-vector** which is the same as distance vector but routes have more than one attribute; and c) **link-state** where every router maintains independent view on topology and computes the shortest path tree towards all other nodes. Additional typology of routing protocol is according to type of deployment: a) **interior gateway protocols (IGP)** for routing within one administrative domain; b) **exterior gateway protocols (EGP)** for routing between autonomous systems (AS). Among typical representants belong:

Advances in Intelligent Systems and Computing, p. 1, 2014.

- Routing Information Protocol (RIPv2 for IPv4 [4], RIPng for IPv6 [5]) – Distance-vector routing protocol that works with hop-count as the metric. Routes with metric 16 or more are considered unreachable;
- Babel [6] – Babel is distance-vector protocol specialized (but not exclusively) for wireless networks that have different metric criteria than wired networks. Metric may represent cost, number of host or any other implementation dependent route atribute. Nevertheless, routes with infinity metric 0xFFFF are considered unreachable. Babel currently supports both IPv4 and IPv6;
- Intermediate System to Intermediate System (IS-IS) [7] – The first link-state protocol ever, which is also capable of working with different metrics simultaneously. IS-IS was originally intended to be used with Connection-less Mode Network Service Protocol (concurrent of IP) for ISO/OSI networks, however, later was developed implementation for both IPv4 and IPv6 protocols. IS-IS is by design agnostic to used address-family and single instance can carry routing updates for various network protocols. Formerely used IS-IS metrics were delay and link errorness, current revision employs only speed of the link;
- Open Shortest Path First (OSPFv2 for IPv4 [8], OSPFv3 for IPv4/6 [9]) – OSPF started as IP alternative to IS-IS and later become industrial standard link-state routing protocol that has wide-spread deployment. OSPF uses cost as the metric, where cost is derived from the interface bandwidth. OSPF supports only IP routing updates;
- Border Gateway Protocol (BGPv4) [10] – Extends distance-vector idea by having multiple attributes acompanying the single prefix update. BGPv4 is currently the only one EGP that is being used and it is often refered as policy-control routing protocol.

Enhanced Interior Gateway Protocol (EIGRP) is the backward compatible successor of previous Cisco proprietary Interior Gateway Protocol (IGRP). It is categorized as a hybrid routing protocol which means that it is a crossover between distance-vector (topology is known based on announcement from neighbors) and link-state protocols (instead of periodic updates, topology changes are propagated immediately). Down below follows the list of main beneficial features of EIGRP:

- EIGRP employs **Diffusing Update Algorithm (DUAL)** [11] that effectively propagates any topology change and minimizes path recomputational time;
- Currently EIGRP is the only routing protocol that guarantees loop-free topology even during the time when topology is actively converging towards a new routing state;
- EIGRP leverages its own reliable transport protocol (even for multicast data transfer);
- In the contrary to other distance-vector protocols, EIGRP is capable of sending event-driven partial bounded updates;
- It has neighbor discovery and recovery mechanism to determine route reachability via particular adjacent node;
- EIGRP contains protocol-dependent modules that allow operation over different network protocols (including IPv4 and IPv6).

The EIGRP was introduced in 1993 as a result of joint effort of Cisco and SRI International [12]. Initial and later measurements revealed that it outperforms other routing protocols (i.e., speed of convergece, network bandwidth utilization, queing delay) [13]. Despite its beneficial aspects (or maybe because of them) it had been protected as one of the major Cisco intellectual properties by a bunch of patents for nearly twenty years. In the beginning of 2013, basic EIGRP design and functionality were submitted as a publicly available IETF informational draft [14].

This paper has the following structure. The next section covers a quick overview of existing EIGRP implementations (either real or simulation ones). Section 3 deals with our contribution, mainly necessary theory, proposed design and subsequent implementation. Section 4 presents validation scenarios proving corectness of the implementation. The paper is summarized in Section 5 together with unveiling our future plans.

2 State of the Art

Currently none of vendors other than Cisco supports EIGRP in its active network devices. Despite positive campaign targeting wider EIGRP acceptance, many manufacturers and customers remain skeptical and rely on a long-time proven open solutions like OSPF or IS-IS. The one of the first publicly available open-source EIGRP routing demon is being developed at the University of Žilina [15] within the scope of Quagga project [16].

A freely available demonstration tool called Easy-EIGRP [17] exists rather for educational purposes.

OPNET simulator has contained EIGRP simulation modules even before its public IETF release. However, its functionality is limited and it lacks IPv6 support for EIGRP. Nevertheless, OPNET and its simulation models were used to conduct several measurement studies comparing different routing protocols including EIGRP [18].

Previously described state of EIGRP deployment affirmed our decision to offer academic and enterprise community with a full-fledged EIGRP implementation with all usually employed features.

The current status of unicast routing support in OMNeT 4.5 and INET 2.3 framework is according to our best knowledge as follows: a) the IPv4 (named `networkLayer`) and IPv6 (pragmatically called `networkLayer6`) layers are already parts of INET framework; b) the framework contains OSPFv2 as the only available IGP routing protocol.

During ANSA project development we have extended original simple router module to be dual-stack capable and enhanced it with a variety of dynamic routing protocols (RIP, RIPng, IS-IS, OSPFv3, PIM), thus creating **ANSARouter** as the compound simulation module based on the standard behavior of Cisco routers.

The basic goal behind our effort is to support EIGRP dynamic routing protocol. Hence, we have decided to add missing functionality in form of simulation module directly connected to `networkLayer` and `networkLayer6` as depicted in Fig. 1.

OMNeT++ state of the art prior to this paper is the result of ongoing research covered in our other articles.

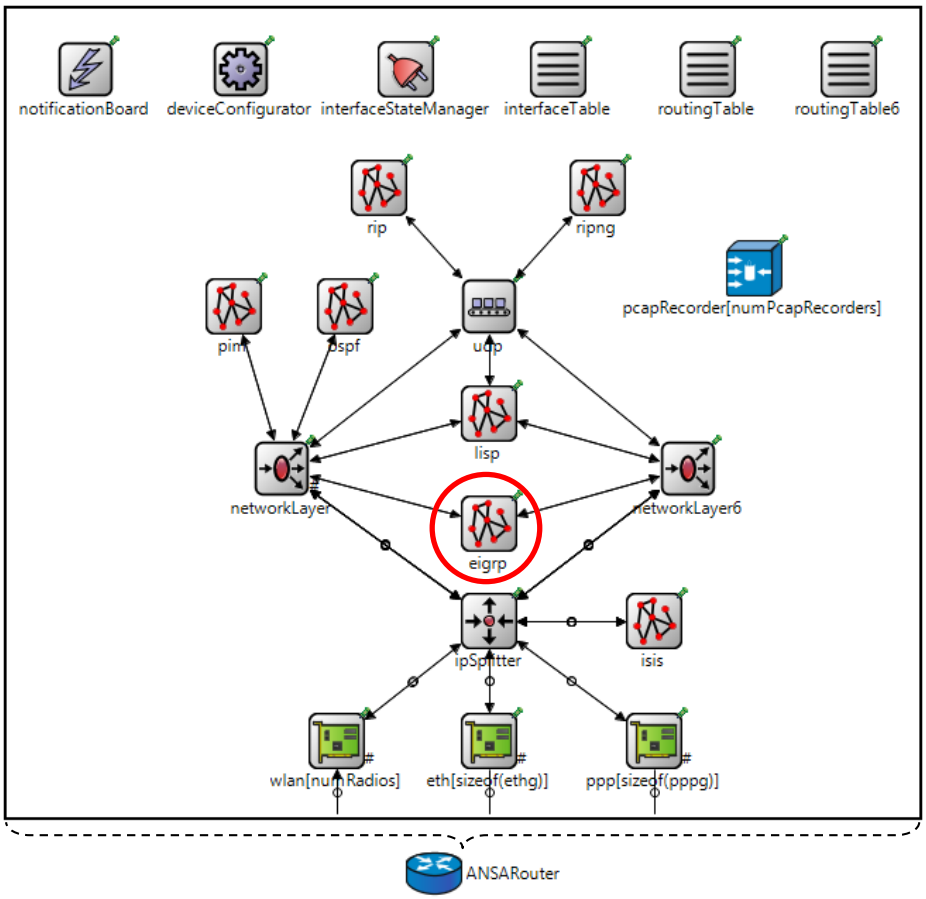

Fig. 1. ANSARouter structure with highlighted contribution.

3 Contribution

We have implemented OMNeT++ compound simulation module supporting EIGRP behavior and functionality. This section provides a short theoretical background, overview of design and some implementation notes.

3.1 Theory of Operation

An EIGRP process computes a successor for every destination. A **successor** represents the next-hop router where the route to the destination via successor is loop-less and with the shortest distance. **Feasible successor** (FS) or so called backup next-hop is the router that provides loop-less route but with higher distance. To determine whether particular router is a feasible successor, the router is working with two parameters – a feasible and a reported distance. **Feasible distance** (FD) is the best known distance from a destination network to a given EIGRP router (historical minimum). **Reported**

distance (RD) is distance from destination network advertised by a given EIGRP router neighbor. The router is using FD and RD to decide whether the feasible condition is satisfied or not. **Feasible condition** (FC) assumes that any route with $RD < FD$ is without any doubts loop-less. The **passive state** is the state of the destination network when the successor is known and the route is converged and usable. **Active state** is in contrast to the previous definition when the destination network does neither have a successor nor FS and the router is actively searching and computing a new successor.

The EIGRP employs composite metric which takes into account multiple route attributes. The basic composite metric consists of following four parameters: a) bandwidth (abbr. Bw is minimal bandwidth enroute); b) delay (abbr. Dl is accumulative sum of route delays), c) load (abbr. Lo is maximal traffic load in range from 1 to 255 on the links towards destination where lower is considered better), d) reliability (abbr. Re is minimal reliability in range from 1 to 255 on the links towards destination where higher is considered better). Parameters a) and b) are static, parameters c) and d) are dynamically recomputed every 5 minutes on certain EIGRP versions. Parameters are accompanied with K-values called weights which are unsigned byte long values, where $K_4 = K_5$. Usually, Cisco routers are using default composite formula (1) for metric computation without dynamic parameters:

$$K_1 \cdot Bw + K_3 \cdot Dl \tag{1}$$

Complete composite formula (2) including all parameters looks like this:

$$\left(K_1 \cdot Bw + \frac{K_2 \cdot Bw}{256 - Lo} + K_3 \cdot Dl\right) \cdot \frac{K_5}{Re + K_4} \tag{2}$$

The new revision of EIGRP establishes two new parameters: a) jitter (abbr. Ji is accumulative delay variation enroute measured in microseconds where lower is preferred); b) energy (abbr. En is accumulative energy consumption in watts per transferred kilobit where lower is preferred). Both parameters are accompanied with K_6 weight. A new wide metric is 64 bit long in opposite to older 32 bit long standard metric and it also solves problem of standard metric when taking into account delay on links faster than 1 Gbps. Wide metric composite formula (3) is then:

$$\left(K_1 \cdot Bw + \frac{K_2 \cdot Bw}{256 - Lo} + K_3 \cdot Dl + K_6 \cdot (En + Ji)\right) \cdot \frac{K_5}{Re + K_4} \tag{3}$$

When employing multicast for communication on local segment, EIGRP has either reserved address 224.0.0.10 for IPv4 or FF02::A for IPv6. EIGRP routers exchange following messages during operation:

- *EIGRP Hello* – Detects EIGRP neighbors with their settings (K-values, autonomous system number, timers and authentication) and checks their aliveness. Sent periodically every 5 seconds by default. Hold timer (period after which neighbor is considered dead) is 3× longer, and by default it is 15 seconds. Neighbor announces its own hello and hold intervals which will obey during its operation;

- *EIGRP Update* – Carries routing information that might cause receivers to start DUAL. Sent either as unicast or multicast;
- *EIGRP Ack* – Used for acknowledging *EIGRP Update*, *Query* and *Reply* messages. It is reused *EIGRP Hello* message with empty structure;
- *EIGRP Query* – If network transits to active state and router starts to search for a new successor then router starts DUAL and sends *EIGRP Queries* to neighbors usually as multicast;
- *EIGRP Reply* – This message contains the routing answer to previous *EIGRP Query*.

DUAL functionality could be described in form of finite-state machine that reacts on events and messages and transits from one state to another with accompanied response action (change of distance D). Basic version (shown in Fig. 2) consists of 5 states and 18 transitions depending on the fact whether router is successor (S) or not (¬S) and whether feasible condition is satisfied or not (more details in [14]).

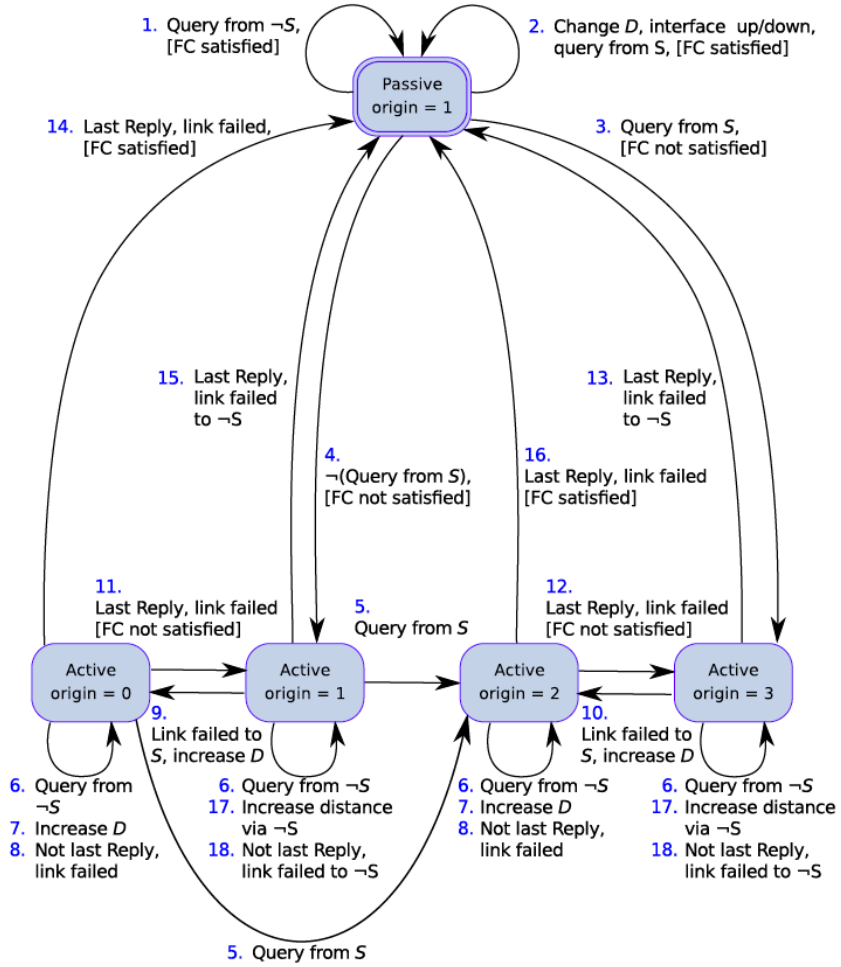

Fig. 2. Basic DUAL in form of finite-state machine

3.2 Design and Implementation

The EIGRP implementation works with three tables:

- **Neighbor table (NT)** – Stores information (e.g., IP address, router-id, uptime, hold-time, query count, etc.) relevant to all adjacent EIGRP routers;
- **Topology table (TT)** – The main routing information base from point of view of a given router. It contains each known network and relevant routes, their states and next-hop addresses together with their FD and RD;
- **Routing table (RT)** – A routing table is the gathering place of best routes from different routing sources, thus the best EIGRP routes are installed here from TT.

The compound EIGRP simulation module is divided into components depicted in Fig. 3 and their brief description is in Table 1.

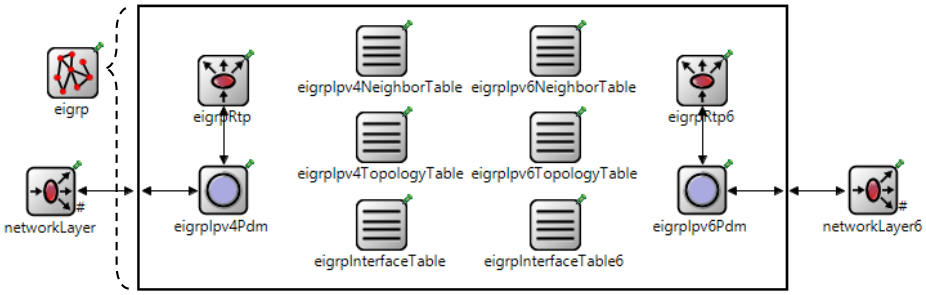

Fig. 3. EIGRP simulation module structure

Table 1. Description of EIGRP submodules.

Name	Description
`eigrpIpv*Pdm`	The protocol-dependent module (PDMs) sends and receives EIGRP messages that contain routing information. It mediates control exchange between routing table and topology table. It leverages different network protocols as carriers for EIGRP messages. Currently IPv4 and IPv6 is supported.
`eigrpRtp*`	EIGRP uses Cisco Reliable Transport Protocol (RTP) to ensure reliable transfer of EIGRP messages. It uses sequence number and positive acknowledgement scheme to detect any gaps in transfers. There is separate RTP module for each PDM.
`eigrpIpv* NeighborTable`	This module is model of neighbor table. It maintains state of all EIGRP adjacencies (i.e., neighbor address, state, hold timer, RTP sequence number)
`eigrpIpv* TopologyTable`	EIGRP routing information base, which includes all learned routes, their state (either active or passive), FDs and computed successors (in form of their addresses).
`eigrp InterfaceTable*`	Simulation module keeps settings relevant to any interface on which EIGRP is enabled (i.e., separate hello and hold timers, query count).

4 Testing

In this section, we provide information on testing and validation of our implementation. Only two scenarios are described here really thoroughly because of limited space. Nevertheless a rich set of test scenarios (including the one verifying our newly implemented stub feature) is accompanied with the published source codes.

We compared results with the behavior of the referential EIGRP implementation running at Cisco routers. For this reason, we built exactly the same topology and observed (using Switched Port Analyzer and Wireshark) relevant message exchanges between real devices (Cisco 7204 as routers with c7200-adventerprisek9-mz.152-4.M5 IOS implementing EIGRP rev. 10 with EIGRP TLV 2.0 and host stations with Windows 7 OS).

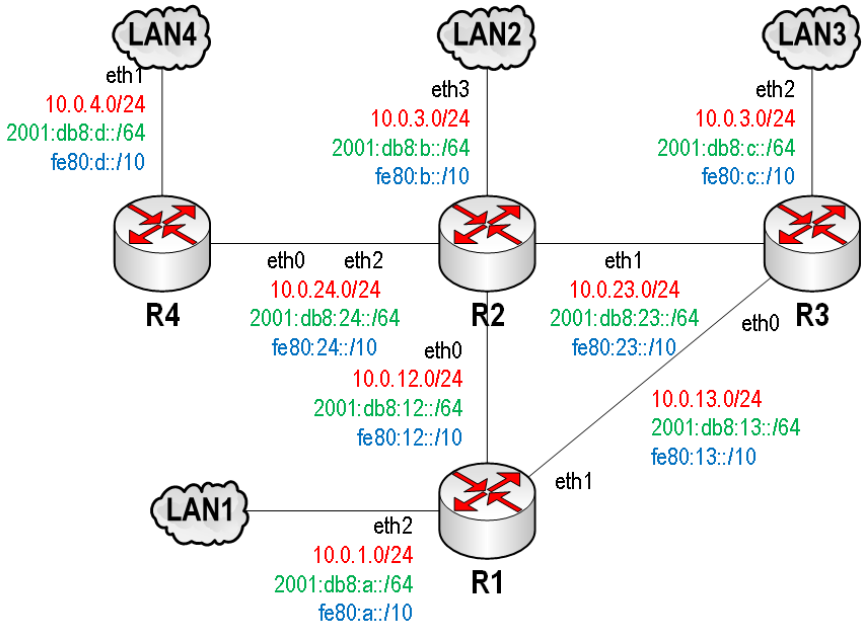

Fig. 4. EIGRP testing topology

Testing topology (see above Fig. 4) consists of four `EIGRPRouters` (marked `R1`, `R2`, `R3` and `R4`) and four `ANSA_DualStackHosts` (`LAN1`, `LAN2`, `LAN3` and `LAN4`) which substitutes whole separate LAN segment with dedicated IP networks. `EIGRPRouter` is `ANSARouter` equipped with only above described `eigrp` routing module. Router interfaces are marked "eth*". Each one of them has single IPv4 address, one IPv6 global unicast address and IPv6 link-local address.

In the first scenario, we would like to show how metric changes are being propagated. In the second scenario, we focus on topology changes. IPv4 and IPv6 routing events were recorded and evaluated for both scenarios. Following thorough description contains EIGRP events for both IPv4 and IPv6 routes.

4.1 Scenario 1: Metric Change

A typical EIGRP message exchange of freshly booted router consist of following phases (hereafter numbered with #X):

#1) Routers establish neighborship by sending and receiving *EIGRP Hello* messages. Whenever a new neighbor is discovered, all relevant information is recorded and stored in NT. `EigrpInterfaceTable` with its settings is in Fig. 6. We can observe `eigrpIpv*NeighborTable` content on router R2 prior to Scenario 1 events (few seconds before the metric change) in Fig. 5. Please notice that neighborship is bound to link-local addresses in case of IPv6 adjacencies;

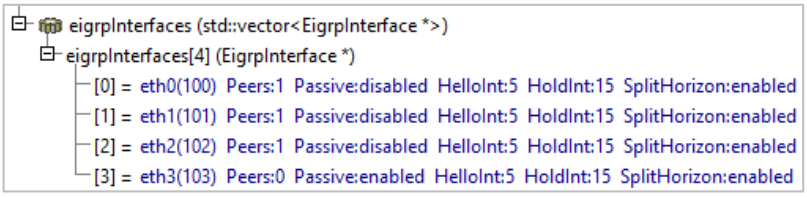

Fig. 6. R2's Interface Table settings

Fig. 5. R2's IPv4 and IPv6 Neighbor Tables prior to Scenario 1 events

#2) Whenever neighborship is established, routers exchange *EIGRP Updates* containing routing information to build their TTs and determine best routes towards known destinations. Reception and processing of any update is confirmed by *EIGRP Ack*. Fig. 7 shows converged state of the topology from the router R2's `eigrpIpv*TopologyTable` point of view. Routes have known *FD*, successors and are in passive states. Please notice that addresses of successors for IPv6 are link-local ones, same applies also for next-hops in IPv6 RT.

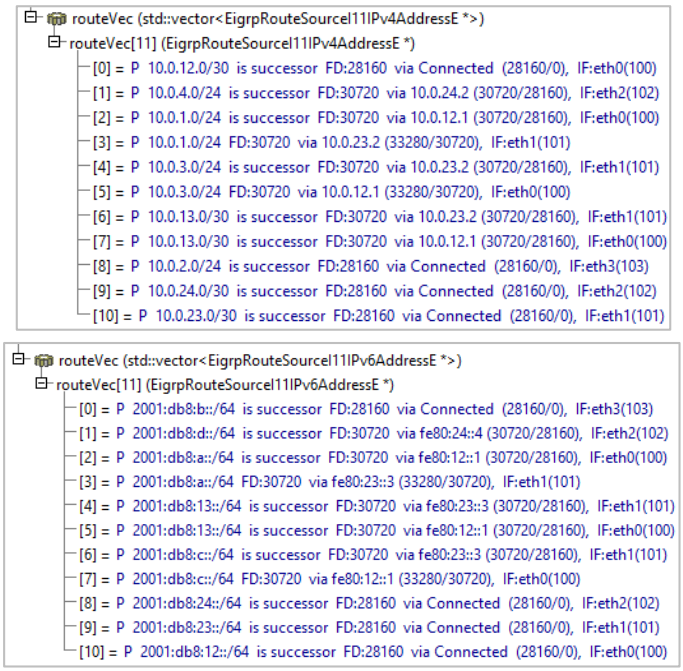

Fig. 7. R2's IPv4 and IPv6 Topology Tables prior to Scenario 1 events

We scheduled bandwidth alternation R3's eth2 interface facing LAN3 changes its *Dl* attribute from 1 ms to 100 ms in order to show how the change of metric influences topology (for instance content or R2's RT is depicted in Fig. 8). In simulator, we uses `scenarioManager` to accomplish this goal, in case of real-network, we change interface configuration.

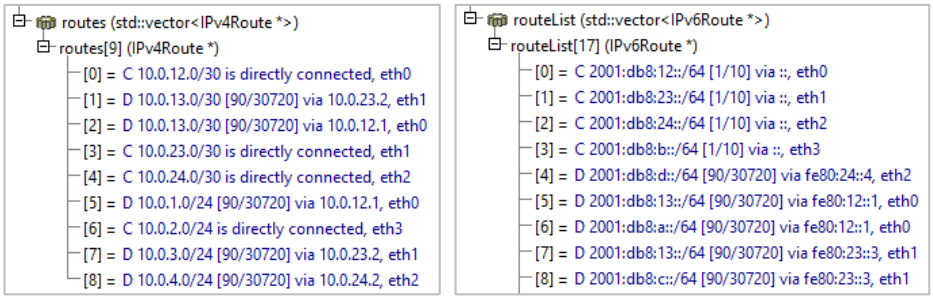

Fig. 8. R2's IPv4 and IPv6 Routing Tables prior to Scenario 1 events

#3) `R3` initiates DUAL, which discovers that `LAN3` is only reachable via eth2 and propagates metric change to its neighbors `R2` and `R1` by sending *EIGRP Update* for `LAN3`'s network (either 10.0.3.0/24 or 2001:db8:c::/64);

#4) `R2` acknowledges update with *EIGRP Ack*. `R2`'s DUAL is unable to find FS, hence route transits to active state and router sends ordinary *EIGRP Query* to `R1` and `R4` and poison reverse *EIGRP Query* with maximal metric towards `R3`. Same

previous steps apply also for R1 where situation is similar – acknowledgment towards R3, DUAL marks network as active, query to R2 and poison reverse query to R3;

#5) R1 receives *EIGRP Query* from R2 and it acknowledges it with *EIGRP Ack*. Following next, R1 responds with *EIGRP Reply* with a new metric via successor R3. Same situation repeats on R2 when replying to R1 query;

#6) R3 receives queries from R1 and R2 and it acknowledges them. Following next, R3 finds out FS (itself) and responds with *EIGRP Replies* to R2 and R1;

#7) R4 receives *EIGRP Query* from R2 and confirms it with *EIGRP Ack*. DUAL is unable to determine FS, thus route transits to active state. Because of split-horizon rule, there is no neighbor to query. Hence, R2 is marked as a successor due to infinity *FD*. The network transits back to passive state with a changed metric via new and old successor R2. R4 sends poison reverse *EIGRP Reply* back to R2;

#8) R1 and R2 receive and acknowledge *EIGRP Replies* which they exchanged and store a new metric in TT;

#9) R1 and R2 receive *EIGRP Reply* from R3 and store a new metric in TT. Because all neighbors of R1 and R2 responded to their queries, DUAL stops. Next, they both R1 and R2 update records in RTs to reflect changed metric situation of LAN3's network (for IPv4 10.0.3.0/24 or for IPv6 2001:db8:c::/64). Topology is converged and state of R2's routing table is depicted in Fig. 9.

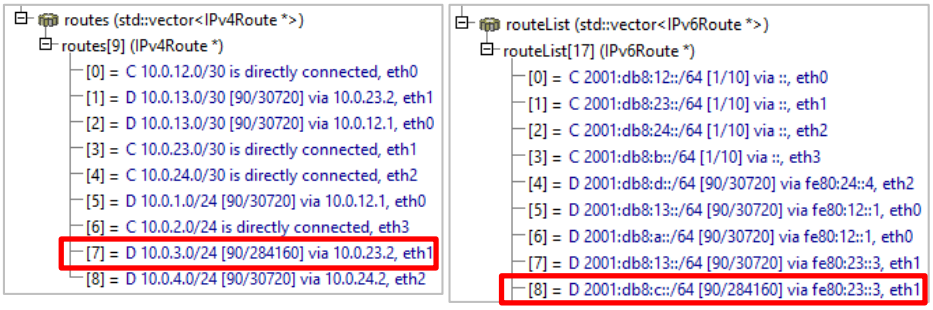

Fig. 9. R2's Routing Table after Scenario 1 events

4.2 Scenario 2: Topology Change

This scenario begins exactly same as the previous one with phase #1, when neighbors are discovered, and phase #2, when topology converges by initial routing information exchange (same content of R2's NT, TT and RT as on Fig. 6, Fig. 5, Fig. 7 and Fig. 8).

We scheduled link failure (R2's eth1) of interconnection between routers R2 and R3 for this scenario. The goal is to show how topology change is propagated from the source to other routers. Once again we accomplish this with the help of scenarioManager in simulator. In case of real network, we just shut down the interface. In both cases, R3's eth1 remains operational.

We have decided to omit all acknowledgements in subsequent text in order to make it clearer and easier to read. Nevertheless, all routers correctly confirm reception of *EIGRP Update*, *Query* and *Reply* messages by sending *EIGRP Ack*. Scenario continues in following manner:

#3) Eth1 comes down on R2. EIGRP process goes through TT and transits all networks reachable via successor, i.e., R2-R3 interconnection and LAN3 (10.0.23.0/30 or 2001:db8:23::/64 and 10.0.3.0/24 or 2001:db8:c::/64) on eth1 interface to active state. R2 sends *EIGRP Queries* to neighbors R1 and R4. Load balancing is enabled, thus R1-R3 interconnection (i.e., 10.0.13.0/30 or 2001:db8:13::/64) is reachable via two routes in the RT – the one that leads through R3 is removed and neighbors are notified by *EIGRP Update* messages;
#4) R4 receives *EIGRP Query* from R2. DUAL cannot find FS for routes and because of split-horizon rule there is no other neighbor to ask. Hence, R4 sends *EIGRP Reply* stating that R2-R3 interconnection and LAN3 are unreachable from its perspective;
#5) R1 receives *EIGRP Query*. Dual finds out FS and responds back with *EIGRP Reply*. Moreover, the route to 10.0.23.0/30 or 2001:db8:23::/64 via R2 is removed from RT and *EIGRP Update* about this is sent to neighbors R3 and R2. Routes on this router remain in passive state;
#6) Integrated optimization prevents information from particular updates to be passed to DUAL. Namely previously sent *EIGRP Update* from R1 to R3, from R2 to R1, from R1 to R2 and from R2 to R4;
#7) R2 receives *EIGRP Reply* from R4 and from R1. All replies has been received, thus routes to R2-R3 interconnection and LAN3 have a new successor in R2's TT and that is R1. Those routes are propagated to R2's RT and information about change is sent to neighbors as *EIGRP Update*;
#8) R4 receives *EIGRP Update* from R2 and inserts R2 as a new successor to its RT. Because of RT change, poison reverse *EIGRP Update* is sent back to R2;
#9) Same optimization as in case of phase #6. EIRGP Updates from R2 to R1 and from R4 to R2 are omitted from DUAL processing. Content of R2's NT, TT and RT does not change for the rest of scenario and it shown in Fig. 10, Fig. 11 and Fig. 12);
#10) Hold timer expires on R3, thus neighborship is terminated and R2 is removed from R3's NT. Also R3 sends goodbye *EIGRP Hello* as a preventive notification. All affected networks reachable via R2 (i.e., 10.0.24.0/30, 10.0.2.0/24, 10.0.4.0/24 or 2001:db8:24::/64, 2001:db8:b::/64 and 2001:db8:d::/64) transit to active state and *EIGRP Query* is sent to remaining neighbor R1. Only exception is R1-R2 interconnection (i.e., 10.0.12.0/30 or 2001:db8:12::/64) that has another FS due to load balancing. However, its second route is removed from R3's RT and *EIGRP Update* is sent to R1;
#11) R1 receives *EIGRP Query* and *Update* from R3. DUAL finds FS for all queried routes in R1's TT and thus no network transits to active state. *EIGRP Reply* is sent to R3 as response;
#12) R3's DUAL collects all (single) *EIGRP Replies* (from R1). R3's TT is updated with a new successor and affected networks transit back to passive state. The best routes are introduced to R3's RT and *EIGRP Update* is sent to R1;
#13) Processing of update is optimized just as in case of phase #6 and #9 on R1. Topology is converged.

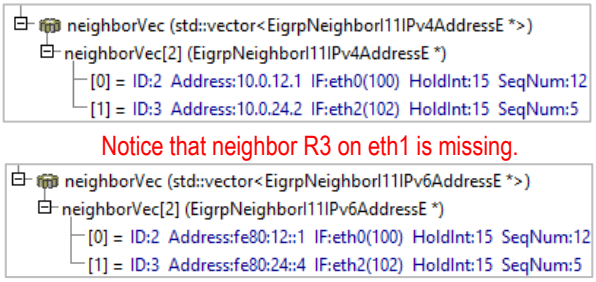

Fig. 10. R2's IPv4 and IPv6 Neighbor Tables after Scenario 2 events

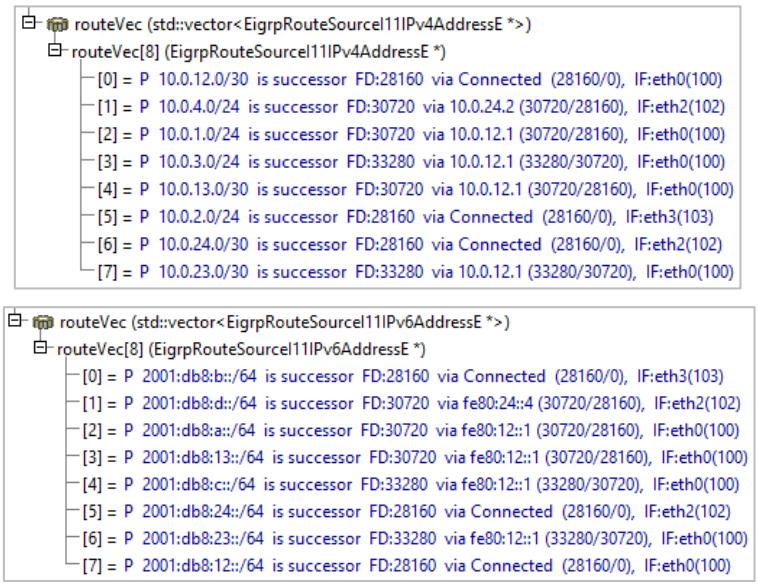

Fig. 11. R2's IPv4 and IPv6 Topology Tables after Scenario 2 events

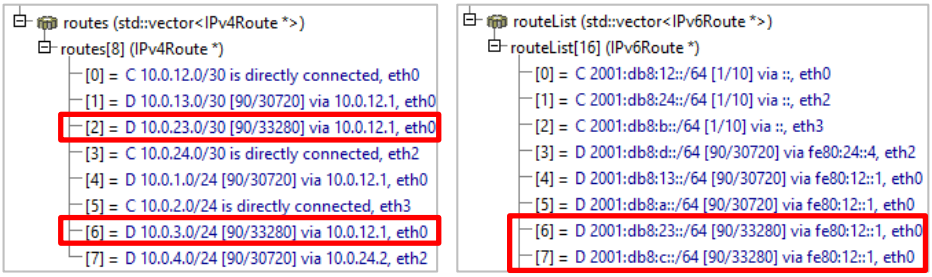

Fig. 12. R2's IPv4 and IPv6 Routing Tables after Scenario 2 events

4.3 Test Summary

Comparison for Scenario 1 can be observed in Table 2 and Table 3 for IPv4 and IPv6. Similarly description for Scenario 2 is in Table 4 and Table 5 for IPv4 and IPv6. In completely revisited comparisons, we have focused on messages processed mostly by router `R2`. Nevertheless, messages that are not shown and were processed by other routers are also in correct order and without any significant deviations between simulation and real time.

The correlation of messages between simulation and real network suggests correctness of our EIGRP implementation.

Table 2. Timestamp comparison for IPv4 routing in Scenario 1

Phase	Message	Sender → Receiver	Simulation [s]	Real [s]
#3	*Update*	R3 → R2	0.000	0.000
#4	*Query*	R2 → R1	0.000	0.030
#5	*Reply*	R1 → R2	0.000	0.057
#7	*Reply*	R4 → R2	0.001	0.074

Table 3. Timestamp comparison for IPv6 routing in Scenario 1

Phase	Message	Sender → Receiver	Simulation [s]	Real [s]
#3	*Update*	R3 → R2	0.000	0.000
#4	*Query*	R2 → R1	0.000	0.062
#5	*Reply*	R1 → R2	0.000	0.097
#7	*Reply*	R4 → R2	0.001	0.132

Table 4. Timestamp comparison for IPv4 routing in Scenario 2

Phase	Message	Sender → Receiver	Simulation [s]	Real [s]
#3	*Query*	R2 → R1	0.000	0.000
#4	*Reply*	R4 → R2	0.000	0.021
#5	*Reply*	R1 → R2	0.000	0.046
#7	*Update*	R2 → R1	0.000	0.074
#8	*Update*	R4 → R2	0.001	0.124
#10	*Hello*	R3 → R2	10.924	10.277
	Query	R3 → R1	10.924	10.299
#11	*Reply*	R1 → R3	10.924	10.349

Table 5. Timestamp comparison for IPv6 routing in Scenario 2

Phase	Message	Sender → Receiver	Simulation [s]	Real [s]
#3	*Query*	R2 → R1	0.000	0.000
#4	*Reply*	R4 → R2	0.000	0.059
#5	*Reply*	R1 → R2	0.000	0.033
#7	*Update*	R2 → R1	0.000	0.121
#8	*Update*	R4 → R2	0.001	0.179
#10	*Hello*	R3 → R2	14.587	14.564
	Query	R3 → R1	14.587	14.575
#11	*Reply*	R1 → R3	14.587	14.617

Validation testing against the real-life topology shows just reasonable time variations. Slight difference could be observed in case of Scenario 2 for IPv6. Phase #5 precedes phase #4. The vindication is that phases #4 and #5 run parallel and are independent. Hence, in real topology message for phase #5 may be dispatched earlier by IOS.

Simulation results are influenced by the fact that `EIGRPRouter` has simpler control-plane and it is not delayed by any other traffic. Hence, some timestamps have sub 0 ms differences (same tables with milliseconds accuracy are available at [19]).

Time variation observable on real Cisco devices is caused by three factors: a) control-plane processing delay and internal EIGRP optimizations; b) packet pacing that guarantees constant bandwidth consumption by EIGRP process and avoids potential race conditions between EIGRP instances; and c) inaccuracy in timing of certain event in real-life network.

Nevertheless, the routing outcomes of simulated and real network are exactly same when taking into account accuracy in order of seconds and EIGRP messages are in confluence on phases that depends on each other.

5 Conclusion

We presented an overview of the theory behind EIGRP routing protocol. The main contribution of this work is a new OMNeT++ compound module routing both IPv4 and IPv6. Module mimics Cisco's EIGRP protocol implementation based on the available specification and from a reverse-engineering observations. We introduce a simulation scenario and relevant results to demonstrate its compliance with the reference Cisco IOS implementation. EIGRP is beneficial namely for large enterprise networks because it generally consumes less resources than link-state IGPs. It is the one of the best distance-vector IGPs available and with its public release we can expect that more companies will tend to use it. For such entities, we offer polished simulation models for a reliable comparison on their network functionality which now includes also EIGRP.

We plan to carry on work towards extending simulation module with stuck-in-active support and further tune EIGRP. Additional plan is to conduct comparative evaluation of our models against those in OPNET simulator.

More information about the ANSA project is available on homepage [20]. All source codes including EIGRP implementation could be downloaded from GitHub [21]. Real packet captures, which serve as a baseline for results reproduction, could be downloaded from Wiki of above mentioned GitHub repository.

Acknowledgement

This work was supported by following research grants and institutions:

- FIT-S-14-2299 supported by Brno University of Technology;
- VG20102015022 supported by Ministry of the Interior of the Czech Republic;
- IT4Innovation ED1.1.00/02.0070 supported by Czech Ministry of Education Youth and Sports.

References

1. Veselý, V., Jan, B., Ryšavý, O.: Enhanced Interior Gateway Routing Protocol for OMNeT++. In SciTePress, ed. : Proceedings of the 4th International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH 2014), Wien, pp.50-58 (2014)
2. OMNeTpp/INET: INET Framework | Main / Welcome to the INET Framework. (Accessed 2014) Available at: http://inet.omnetpp.org/
3. OMNeTpp: OMNeT++ Network Simulation Framework. (Accessed 2014) Available at: http://www.omnetpp.org/
4. Malkin, G.: RFC 2453: RIP Version 2. (Accessed November 1998) Available at: https://tools.ietf.org/html/rfc2453
5. Malkin, G., Minnear, R.: RFC 2080: RIPng for IPv6. (Accessed January 1997) Available at: https://tools.ietf.org/html/rfc2080
6. Chroboczek, J.: RFC 6126: The Babel Routing Protocol. (Accessed April 2011) Available at: http://tools.ietf.org/html/rfc6126
7. Oran, D.: RFC 1142: OSI IS-IS Intra-domain Routing Protocol. (Accessed February 1990) Available at: http://tools.ietf.org/html/rfc1142
8. Moy, J.: RFC 2328: OSPF Version 2. (Accessed April 1998) Available at: http://tools.ietf.org/html/rfc2328
9. Coltun, R., Ferguson, D., Moy, J., Lindem, A.: RFC 5340: OSPF for IPv6. (Accessed July 2008) Available at: http://tools.ietf.org/html/rfc5340
10. Rekhter, Y., Hares, S.: RFC 4271: A Border Gateway Protocol 4 (BGP-4). (Accessed January 2006) Available at: http://tools.ietf.org/html/rfc4271
11. Garcia-Lunes-Aceves, J. J.: Loop-Free Routing Using Diffusing Computations. IEEE/ACM Transactions on Networking Vol. I(No. 1), 130-141 (1993)
12. Albrightson, R., Garcia-Luna-Aceves, J., Boyle, J.: EIGRP a fast routing protocol based on distance vectors. Proceedings Networld/Interop Vol. XCIV, 136-147 (May 1994)
13. Xu, D., Trajkovic, T.: Performance Analysis of RIP, EIGRP, and OSPF Using OPNET. (Accessed August 2011) Available at: http://summit.sfu.ca/item/10841
14. Savage, D., Slice, D., Ng, J., Moore, S., White, R.: Enhanced Interior Gateway Routing Protocol. (Accessed October 7, 2013) Available at: https://tools.ietf.org/html/draft-savage-eigrp-01
15. GitHub/janovic: janovic/Quagga-EIGRP. (Accessed 2013) Available at: https://github.com/janovic/Quagga-EIGRP
16. nonGNU: Quagga Software Routing Suite. (Accessed 2013) Available at: http://www.nongnu.org/quagga/

17. SourceForge: Easy-EIGRP | Free software downloads at SourceForge.net. (Accessed 2013) Available at: http://sourceforge.net/projects/easy-eigrp/
18. Wu, B.: Simulation Based Performance Analyses on RIPv2, EIGRP, and OSPF Using OPNET. In: Math and Computer Science. (Accessed August 2011) Available at: http://digitalcommons.uncfsu.edu/macsc_wp/11
19. Veselý, V., Vít, R.: EIGRP comparison tables between simulation and real topology with milliseconds accuracy. (Accessed 2014) Available at: http://nes.fit.vutbr.cz/ivesely/PreciseEigrpTables.pdf
20. Brno University of Technology In: ANSAWiki | Main / HomePage. (Accessed January 2014) Available at: http://nes.fit.vutbr.cz/ansa/pmwiki.php
21. GitHub/kvetak In: kvetak/ANSA. (Accessed December 2013) Available at: https://github.com/kvetak/ANSA