Locator/Id Split Protocol Improvement for High-Availability Environment

Full-fledged LISP and VRRP simulation modules for OMNeT++

Vladimír Veselý, Ondřej Ryšavý
Department of Information Systems
Faculty of Information Technology, Brno University of Technology (FIT BUT)
Brno, Czech Republic
e-mail: {ivesely, rysavy}@fit.vutbr.cz

Abstract—**Locator/Id Split Protocol is a currently discussed alternative to deal with the traditional IP drawbacks (like cumbersome support of device mobility or more importantly default-free zone routing table growth due to the increased demand for multihoming and traffic engineering). This work outlines LISP and its properties for high-availability environments employing first-hop redundancy protocols. This paper also suggests LISP improvement for map-cache synchronization that should impact its routing performance. For this cause, two new simulation models (LISP and VRRP) are introduced that are behaviorally fully RFC compliant.**

Keywords-LISP; VRRP; map-cache synchronization; OMNeT++

I. INTRODUCTION

The Automated Network Simulation and Analysis (ANSA) project running at our university is dedicated to developing the variety of simulation models compatible with RFC specifications or referential implementations. Subsequently, these tools allow a formal analysis of real networks and their configurations. They may be publicly used as the routing/switching baseline for further research initiatives, i.e., in simulations for proving (or disproving) certain aspects of technologies and/or related protocols.

Locator/ID Split Protocol (LISP) emerged as one of the routing alternatives for Internet Protocol (IP) networks. LISP is the response to problems described in RFC 4984 [1] and RFC 6227 [2]. It should solve Internet architecture problems such as unscalable default-free zone (DFZ) routing, cumbersome mobility and prefix deaggregation caused by multihoming and ingress traffic engineering.

IP address functionality is dual. It serves for identification (“which device is it?”) and localization (“where is the device?”) purposes. The main idea behind LISP is to remove this duality so that there are networks doing routing either based on locators (i.e., transit networks like DFZ) or identifiers (i.e., edge end networks). LISP accomplishes this by splitting the IP addresses into two distinct namespaces: a) **Endpoint Identifier (EID)** namespace (so called LISP site), where each device has unique address; b) **Routing Locator (RLOC)** namespace with addresses intended for localization. There is also a non-LISP namespace where direct LISP communication is (even intentionally) not supported.

Apart from namespaces, there also exist: a) specialized routers (called **tunnel router** a.k.a. **xTR**) spanning between different namespaces; b) dedicated devices maintaining mapping system; and c) proxy routers allowing communication between LISP and non-LISP world.

A LISP mapping system performs lookups to retrieve a set of RLOCs for a given EID. Tunnel routers between namespaces utilize these EID-to-RLOC mappings to perform map-and-encapsulation (see RFC 1955 [3]). The original (inner) header (with EIDs as addresses) is encapsulated by a new (outer) header (with RLOCs as addresses), which is appended when crossing borders from EID to RLOC namespace. Whenever a packet is crossing back from RLOC to EID namespace, the packet is decapsulated by stripping outer header off.

Queries performing EID-to-RLOC mapping are data-driven. This behavior means that a new data transfer between LISP sites may require a mapping lookup, which causes that data dispatch is stopped until mapping is retrieved. This behavior is analogous to the domain-name system (DNS) protocol and allows LISP to operate decentralized database of EID-to-RLOC mappings. Replication of whole (potentially large-scale) database is unnecessary because mappings are accessed on-demand, just like as in DNS a host does not need to know complete domain database. Tunnel routers maintain **map-cache** of recently used mappings to improve performance of the system.

LISP is being successfully deployed in enterprise networks, and one of its most beneficial use-cases is for data-centers networking. An important feature of any data-center is its ability to maintain high-availability of provided services. This goal is accomplished mainly with redundancy. In the case of the outage, service delivery is not affected because of redundant links, devices or power sources. **Virtual Router Redundancy Protocol (VRRP)** is among related protocols and technologies guaranteeing redundancy and helping to achieve high-availability.

VRRP is widely adopted protocol providing redundancy of default-gateway (crucial L3 device that serves as exit/entry point to a given network). VRRP is IETF’s response for

Cisco's proprietary Hot Standby Routing Protocol (HSRP) and Gateway Load Balancing Protocol (GLBP) delivering same goals.

VRRP combines redundant first hop routers into virtual groups. One master router actively forwards clients traffic within each group, where others in the group are backing its functionality. Backup routers are periodically checking liveness of the master waiting ready to substitute it in case of failure. Switching to a new active router is transparent from the host's perspective thus no additional configuration or special software is needed.

This paper introduces two new simulation modules, which create a part of the ANSA project and which extend the functionality of the INET framework in OMNeT++. Subsequently, they are employed as measurement tools supporting proposed LISP map-cache synchronization technique.

This paper has the following structure. The next section covers a quick overview of existing simulation modules. Section III describes the design of relevant LISP and VRRP models. Section IV deals with a map-cache synchronization mechanism – how synchronization works, how it is implemented and how it should aid devices to run LISP and VRRP simultaneously. Section V presents validation scenarios for outlined implementations and shows promising results backing up improvement's impact on LISP operation. The paper is summarized in Section VI together with unveiling of our plans.

II. State of the Art

This section outlines the current state of the art of available LISP and VRRP implementations for simulator environments.

Limited LISP implementation was created [4] to support LISP MobileNode NAT traversal [5]. However, it is intended for outdated INET-20100323 and OMNeT++ 4.0. Previously, LISP map-cache performance have been evaluated employing high-level simulation that is not taking into account protocol implementation specifics [6].

We are not aware that any VRRP (or another first-hop redundancy protocol) implementation is supported by other major simulators like NS-2/3 or OPNET.

According to our knowledge, OMNeT++ 4.6 (discrete event simulator) and INET 2.4 (framework for wired networks simulation) do not support VRRP simulation modules at all. LISP is supported partially as the result of our previous research effort [7].

Thus, we have implemented LISP and VRRP modules by ourselves in order to have reliable components for subsequent research (i.e., evaluation of proposed improvements).

III. Implementation

A. LISP – Theory of Operation

LISP is being codified within IETF [8]. The main core and functionality is described in RFCs 6830-6836.

LISP supports both IPv4 and IPv6. Moreover, LISP is agnostic to address family thus it can seamlessly work with any upcoming network protocol. Transition mechanisms are part of the protocol standard. Hence, LISP supports communication with legacy non-LISP world. LISP places additional UDP header succeeded by LISP header between inner and outer header. LISP uses reserved port numbers – 4341 for data traffic and 4342 for signalization.

Basic components of the LISP architecture are **Ingress Tunnel Router (ITR)** and **Egress Tunnel Router (ETR)**. Both are border devices between EID and RLOC space; the only difference is in which direction they operate. The single device could be either ITR-only or ETR-only or ITR and ETR at the same time (thus abbreviation xTR). ITR is the exit point from EID space to RLOC space, which encapsulates the original packet. This process may consist of querying mapping system followed by updating local map-cache, where EID-to-RLOC mapping pairs are stored for a limited time to reduce signalization overhead. ETR is the exit from RLOC space to EID space, which decapsulates packet. Outer header, auxiliary UDP, and LISP headers are stripped off. ETR is responsible for registering all LISP sites (their EID addresses) and by which RLOCs they are accessible.

LISP mapping system consists of two components – **Map Resolver (MR)** and **Map Server (MS)**. The list below contains all LISP control messages responsible for mapping system signalization. They are without inner header – just outer header, followed by UDP header (with source and destination ports set on 4342), followed by appropriate LISP message header.

- *LISP Map-Register* – Each ETR announces LISP site(s) to the MS with this message. Each registration contains authentication data and the list of mappings and their properties.
- *LISP Map-Notify* – UDP cannot guarantee message delivery. MS may optionally (when proper bit is set) confirm reception of *LISP Map-Register* with this message.
- *LISP Map-Request* – ITR generates this request whenever it needs to discover current EID-to-RLOC mapping and sends a message to preconfigured MR.
- *LISP Map-Reply* – This is a solicited response from the mapping system to the previous request and contains all RLOCs to a certain EID together with their attributes.

MR processes ITR's *LISP Map-Requests*. Either MR responds with *LISP Negative Map-Reply* if queried address is from a non-LISP world (not EID), or *LISP Map-Requests* is delegated further into mapping system to appropriate MS.

Every MS maintains **mapping database** of LISP sites that are advertised by *LISP Map-Register* messages. If MS receives *LISP Map-Request* then: a) either MS responds directly to querying ITR; or b) MS forwards request towards designated ETR that is registered to MS for target EID. xTRs perform **RLOC probing** (checking of non-local locator liveness) in order to always use current information.

Each RLOC is accompanied by two attributes – priority and weight. **Priority** (one byte long value in the range from 0 to 255) expresses each RLOC preference. The locator with the lowest priority is preferred for outer header address. Priority value 255 means that the locator must not be used for traffic forwarding. Incoming communication may be load-balanced based on the **weight** value (in the range from 0 to

100) between multiple RLOCs sharing the same priority. Zero weight means that RLOC usage for load-balancing depends on ITR preferences.

B. LISP – Design of a Simulation Module

Simulation model of LISP xTR, MR and MS functionality is currently implemented as `LISPRouting` compound module. It consists of five submodules that are depicted in Figure 1 and described in Table I below. `LISPRouting` exchanges messages with `UDP`, IPv4 `networkLayer`, and IPv6 `networkLayer6` modules. Implementation is fully in compliance with RFC 6830 [9] and RFC 6833 [10].

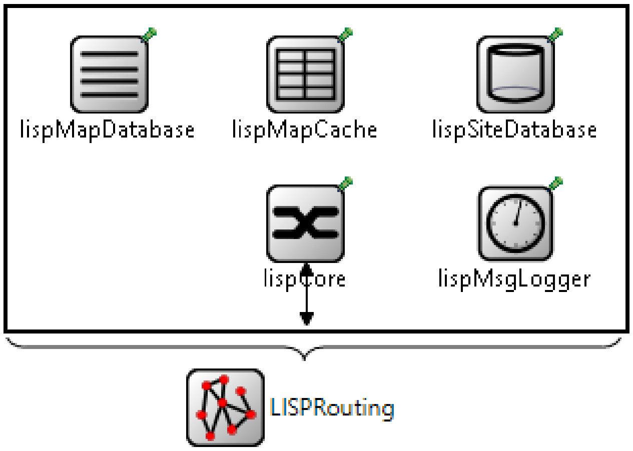


Figure 1. LISPRouting module structure

TABLE I. DESCRIPTION OF LISPROUTING SUBMODULES

Name	Description
`lispCore`	Module handles LISP control and data traffic. It independently combines functionality of ITR, ETR, MR and MS. This involves: encapsulation and decapsulation of data traffic; ETR site registration and MS site maintenance; ITR performing lookups; MR delegating requests.
`lispMapDatabase`	Each xTR maintains configuration of its LISP sites (i.e., which RLOCs belong to a given EID or which local interfaces are involved in LISP) that is used by control-plane during registration or for RLOC probing.
`lispMapCache`	Local LISP map-cache that is populated on demand by routing data traffic between LISP sites. Each record (EID-to-RLOC mapping) has its separate handling (i.e., expiration, refreshment, availability of RLOCs).
`Lisp SiteDatabase`	MS's database that maintains LISP site registrations by ETRs. It contains site-specific information (e.g., shared key, statistics of registrars and most importantly known EID-to-RLOC mappings).
`lisp MsgLogger`	This module records and collects statistics about LISP control plane operation, e.g. number, types, timestamps and length of messages.

C. VRRP – Theory of Operation

VRRP specification is publicly available as RFC standard – RFC 3768 [11] describes IPv4-only VRRPv2 and RFC 5798 [12] describes dual IPv4+IPv6 VRRPv3. VRRPv2 routers send control messages to multicast address 224.0.0.18. VRRPv3 routers use ff02::12 for IPv6 communication. VRRP has its own reserved IP protocol number 112.

Clustered redundant routers form a VRRP group identified by **Virtual Router ID** (**VRID**). Within the group, a single router (called **Master**) is elected based on announced **priority** (a number in the range from 1 to 255). Higher priority means superior willingness to become Master, zero priority causes router to abstain from being Master. In the case of equal priority, binary higher IP address serves as tie-breaker. VRRP election process is always preemptive (unlike to non-preemptive HSRP or GLBP). Peemption means that the router with the highest priority always wins to be the Master no matter whether the group already have other Master elected. Only Master actively forwards traffic. Remaining routers (called **Backup**s) are just listening and checking for Master's keep-alive messages.

Hosts have configured virtual IP address as their default-gateway. Only Master responds to *ARP Requests* for this IP. This IP address has assigned reserved MAC address – 00:00:5e:00:01:$$ for VRRPv2 and 00:00:5e:00:02:$$ for IPv6 (where $$ is VRID). Whenever VRRP group changes to a new Master, *ARP Gratuitous Reply* is generated in order to rewrite association between the interface and reserved MAC in CAM table(s) of switch(es). This allows transparent changing of Masters for hosts during the outage.

VRRP has only one type of control message – *VRRP Advertisement*. If Master is not elected, then, VRRP routers exchange advertisements to determine which one is going to be a new Master. If Master is already elected, then, only Master is sending *VRRP Advertisements* to inform Backups that it is up and correctly running. *VRRP Advertisement* is generated whenever advertisement timer (AT) expires (by default every 1 second). If this interval is set to a lower value, then, Master's failure is detected faster but protocol overhead increases. Master down interval (MDI) resets with each reception of an advertisement message. Backup, which expires the MDI sooner, becomes a new Master. Value of MDI depends on priority of each VRRP router according to (1). The highest (best) priority Backup times out first (because of the lowest $skew\ time$) and thus takes over role as a new Master before others.

$$MDI = 3 \times AT + \frac{\overbrace{256-priority}^{skew\ time}}{256} \qquad (1)$$

D. VRRP – Design of Simulation Module

VRRP version 2 is implemented as `VRRPv2` compound module connected with `networkLayer`. Module is a container for dynamically created instances of `VRRPv2VirtualRouter` during simulation startup. Each instance handles particular VRRP group operation on a given interface. Its structure is depicted in Figure 2, and a brief description of the functionality follows in Table II. Both modules together implement full-fledged VRRPv2 with the same finite-state machine (FSM) as in [11]. VRRP FSM's states *Init*, *Backup* and *Master* reflect VRRP router role and govern control message generation and processing.

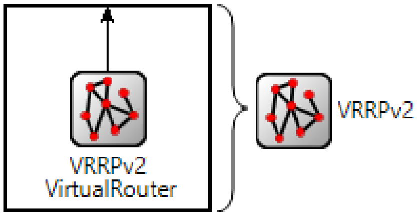


Figure 2. VRRP modules structure

TABLE II. DESCRIPTION OF VRRP MODULES

Name	Description
`VRRPv2`	Responsible for the creation of `VRRPv2VirtualRouters` according to the startup configuration and forwarding VRRP messages to/from them between appropriate gates.
`VRRPv2 VirtualRouter`	This module governs *VRRP Advertisements* processing, transition between states and directs ARP for a single VRRP group.

IV. CONTRIBUTION

Assume multiple redundant routers are acting as first hops in high-availability scenario. Those routers are simultaneously clustered into VRRP groups and act as LISP's xTRs – they run LISP and VRRP at the same time.

The performance of map-and-encap depends on the fact whether xTR's map-cache contains valid EID-to-RLOC mapping or not. Dispatched data traffic drives Map-cache record creation. If map-cache misses the mapping, then, a mapping system needs to be asked and initiating data traffic is meantime dropped. Packet dropping is a valid step as long as the mapping is not discovered because map-and-encap cannot occur without proper information. The rationale behind this behavior is the same as in the case of ARP throttling [13], where any triggering traffic should be discarded to protect control-plane processing and prevent superfluously recurrent mapping system queries.

Each xTR has its own map-cache and its content may differ even within the same LISP site because each cache record may be initialized by different traffic. Hence, xTRs can easily experience severe packet drops and LISP control message storms due to the map-cache misses when Master change occurs within VRRP group.

This problem is described as the one of LISP weak-points in [14] and theoretically investigated in [15]. The viable solution would be to provide map-cache content synchronization that should minimize map-cache misses upon failure. Inspired by that, we present our solution addressing this problem.

We have decided to implement it as a technique maintaining synchronized map-caches within a predefined **synchronization set (SS)** of ITRs. Any solicited *LISP Map-Reply* triggers synchronization process among SS members.

Each record in the map-cache is equipped with a time-to-live (TTL) parameter. TTL expresses for how long the record is considered to be valid and usable for map-and-encap. Map-caches within SS must maintain the same TTL on shared records; otherwise a loss of synchronization might occur (on some ITRs, identical records could expire because of no demand by traffic).

We have implemented two modes of synchronization:

1) *Naïve* – The whole content of map-cache is transferred to SS. All mappings are then updated according to the new content and TTLs are reset. This approach works fine, but it obviously introduces significant transfer overheads;
2) *Smart* – Only record that caused synchronization is transferred. Moreover, we bound this mode with following policy. When TTL expires, the ITR must check record usage during the last minute (one minute should be a period enough long to detect ongoing communication). If the mapping has not been used, then, it is removed from the cache. Otherwise, its state is refreshed by query followed by synchronization.

Synchronization itself is done with the help of two new LISP messages:

- *LISP CacheSync* – Message contains map-cache records that are being synchronized and authentication data protecting SS members from spoofed messages;
- *LISP CacheSync Ack* – Because LISP leverages UDP, it cannot guarantee message delivery. However, we decided to employ the same principle as for *LISP Map-Register* and *LISP Map-Notify*. Hence, *LISP CacheSync* delivery may be optionally confirmed by echoing back *LISP CacheSync Ack* message.

This approach guarantees that devices within SS could forward rerouted LISP data traffic without packet loss or interruption because they share the same content as ITR's map-cache of malfunctioned former VRRP Master.

V. TESTING

In this section, we provide information regarding validation of LISP and VRRP simulation models. This is necessary in order to build up reliability of used tools for subsequent evaluation of proposed map-cache synchronization technique.

We have built exactly the same real network topologies as for simulations. We captured and analyzed (using transparent switchport analyzers and packet sniffers) relevant messages exchanged between devices for both LISP and VRRP functionality validation. We compared the results with the behavior of a referential implementation running on Cisco routers (namely C7200 with IOS version c7200-adventerprisek9-mz.152-4.M2) and host stations.

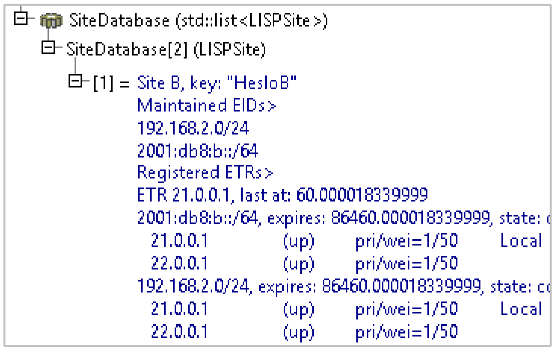
A. LISP Functionality

We have verified LISP implementation on the topology depicted in Figure 6. Simulation network contains two sites – green areas "Site A" (interconnected by switch `S1`, bordered by `xTR_A1` and `xTR_A2`) and "Site B" (interconnected by `S2`, bordered by `xTR_B1` and `xTR_B2`). The topology contains router `MRMS`, which acts as MR and MS for both sites. IPv4 only capable core (red area) is simulated by a single `Core` router. Static routing is employed to achieve mutual connectivity across core. `HostA` and `HostB` are dual-stack devices, where `HostA` is scheduled to ping `HostB` after second successful site registration (at `t=70s`). `MRMS` is allowed to proxy-reply on mapping requests for “Site A”. All RLOCs are configured with priority 1 and weight 50 to achieve equal load balancing for incoming traffic.

Testing scenario beginning is aligned with initialization of xTR_A1’s LISP process that freshly starts after the reboot. The list of important phases is briefly described below:

#1) First of all, each ETR starts RLOC probing, which is polling mechanism that checks reachability of announced locators. Each ETR sends *LISP Map-Request* with probe-bit set on to queried RLOC address (e.g., xTR_A1 is probing xTR_A2’s locator 12.0.0.1). Neighboring xTR_* then responds with *LISP Map-Reply* with probe-bit set announcing state of its RLOC interface. This process repeats by default every minute. The lower RLOC-probe timer is, the sooner RLOC outage is detected but protocol’s overhead increases. Also Cisco’s LISP implementation queries same RLOC for each assigned EID.

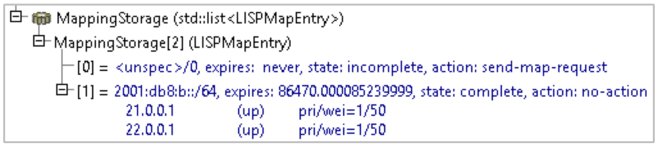
#2) ETRs sends registration about their EID sites towards MS. Each xTR_* generates *LISP Map-Register* message. Registration process repeats every 60 seconds in order to keep mappings up-to-date. *LISP Map-Register* contains all EID-to-RLOC mapping properties (i.e., EID, TTL, RLOC statuses, and attributes). For phase #2 illustration, figure 3 shows xTR_B1’s “Site B” registration after #1.



```
⊟ SiteDatabase (std::list<LISPSite>)
  ⊟ SiteDatabase[2] (LISPSite)
    ⊟ [1] = Site B, key: "HesloB"
            Maintained EIDs>
            192.168.2.0/24
            2001:db8:b::/64
            Registered ETRs>
            ETR 21.0.0.1, last at: 60.000018339999
            2001:db8:b::/64, expires: 86460.000018339999, state: c
              21.0.0.1          (up)      pri/wei=1/50      Local
              22.0.0.1          (up)      pri/wei=1/50
            192.168.2.0/24, expires: 86460.000018339999, state: c
              21.0.0.1          (up)      pri/wei=1/50      Local
              22.0.0.1          (up)      pri/wei=1/50
```

Figure 3. xTR_B1's registration of "Site B"

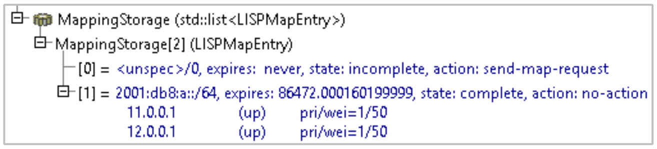
#3) HostA initiates ping to HostB’s address 2001:db8:b::99. *ICMP Echo Request* is delivered to xTR_A1 (hosts default-gateway), where it triggers LISP query because that particular EID-to-RLOC mapping is currently unknown. First ping is dropped due to that. xTR_A1 sends *LISP Map-Request* to MS. MRMS performs lookup on its site database and delegates request to one of the designated ETRs, in this case, xTR_B1. xTR_B1 responds with *LISP Map-Reply* with current mapping (to EID 192.168.2.0/24 belongs two RLOCs 21.0.0.1 and 22.0.0.1). Figure 4 illustrates this result.



```
⊟ MappingStorage (std::list<LISPMapEntry>)
  ⊟ MappingStorage[2] (LISPMapEntry)
     ├[0] = <unspec>/0, expires:  never, state: incomplete, action: send-map-request
     ⊟[1] = 2001:db8:b::/64, expires: 86470.000085239999, state: complete, action: no-action
            21.0.0.1          (up)      pri/wei=1/50
            22.0.0.1          (up)      pri/wei=1/50
```

Figure 4. Content of xTR_A1's map-cache after phase #3

#4) Second ping arrives on xTR1_A1. Because mapping is known, it is encapsulated with outer header as LISP carrying data (marked *LISP Data* message) and sent to one of xTR_B* after random selection of equally preferred locators. In our case, *LISP Data* is delivered to xTR_B2 where original ping is decapsulated and forwarded further to end destination. HostB responds with *ICMP Echo Reply* that is passed to its default-gateway (xTR_B1). Over here the same process as in #4 repeats – ping is dropped and mapping query triggered. Only this time, MS replies directly to *LISP Map-Request*. MRMS is allowed to send *LISP Map-Reply* instead of designated ETR because of proxy-reply for “Site A”. Figure 5 shows the result.



```
⊟ MappingStorage (std::list<LISPMapEntry>)
  ⊟ MappingStorage[2] (LISPMapEntry)
     ├[0] = <unspec>/0, expires:  never, state: incomplete, action: send-map-request
     ⊟[1] = 2001:db8:a::/64, expires: 86472.000160199999, state: complete, action: no-action
            11.0.0.1          (up)      pri/wei=1/50
            12.0.0.1          (up)      pri/wei=1/50
```

Figure 5. Content of xTR_B1's map-cache after phase #4

#5) Third and other consecutive pings pass without experiencing any drop because both default-gateways have proper EID-to-RLOC mappings.

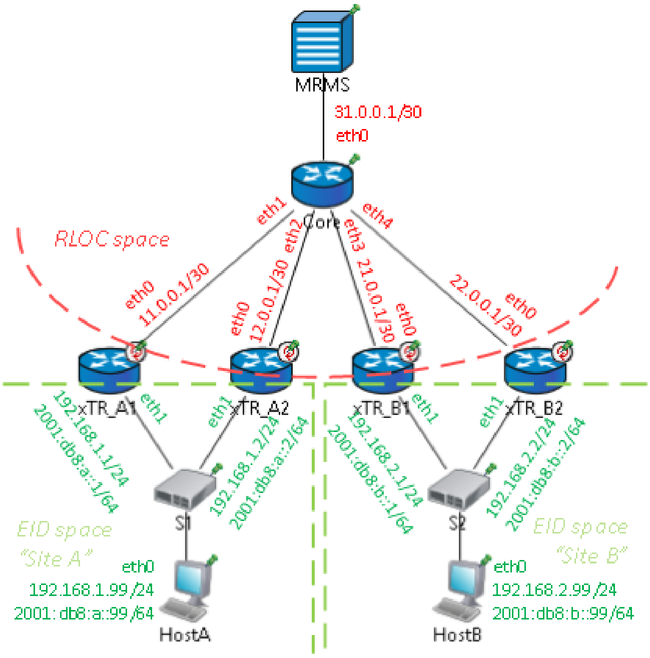


Figure 6. LISP testing topology

Phases of LISP operation are compared to simulation and real network in Table III. For clarity and due to limited space, only some messages are recorded for #1, #2 and #3. Nevertheless, omitted messages do not show significant deviations.

TABLE III. TIMESTAMP COMPARISON OF LISP MESSAGES

Phase	Message	Sender	Simul. [s]	Real [s]
#1	*LISP Map-Req. Probe*	xTR_A1	0.000	0.000
	LISP Map-Rep. Probe	xTR_A2	0.000	0.063
#2	*LISP Map-Register*	xTR_A1	60.000	60.567
#3	*ICMP Echo Request*	HostA	70.000	70.000
	LISP Map-Request	xTR_A1	70.000	70.361
	LISP Map-Reply	xTR_B1	70.000	70.460
#4	*ICMP Echo Request*	HostA	72.000	71.931
	LISP Data	xTR_A1	72.000	71.944
	ICMP Echo Reply	HostB	72.000	71.962
	LISP Map-Request	xTR_B1	72.001	72.852
	LISP Map-Reply	MRMS	72.001	72.889
#5	*ICMP Echo Request*	HostA	74.000	74.011
	ICMP Echo Reply	HostB	74.001	74.177

B. *VRRP Functionality*

We have verified VRRP functionality on the topology depicted in Figure 7. Simulation network contains two VRRP routers (GW1 and GW2) clustered in VRID 10, one switch (SW) interconnecting devices on local segment, one host (Host) and one router (ISP) substituting communication outside LAN. Both VRRP routers are configured with the default priority, default *AT* value and virtual default-gateway IP address set to 192.168.10.254.

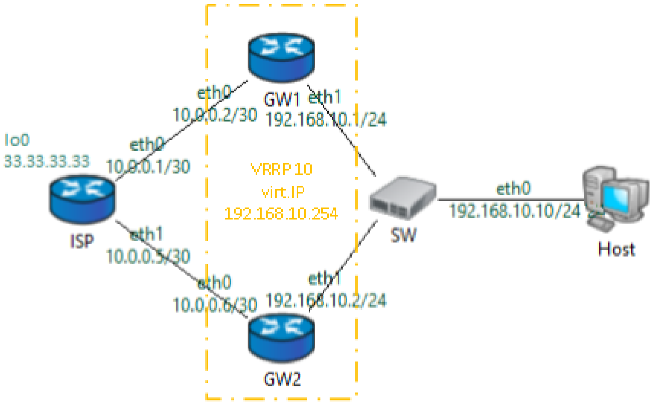


Figure 7. VRRP testing topology

For this test, we scheduled that original Master (GW2) would go down (at t=20s) and back up (at t=30s). Meantime, Host starts pinging (at t=10s) Internet address 33.33.33.33 every second where traffic goes via virtual default-gateway. Scenario beginning (phase #1 at t=0s) is aligned with initialization of VRRP process.

Test goes through following phases:

#1) Both GW1 and GW2 immediately transit from *Init* state to *Backup* and are waiting to hear *VRRP Advertisement* from potential Master.

#2) They both expire *MDI* at the same time (t=3.609275, equation (1) yields the same result) and transit to *Master* state. This allows them to send their own *VRRP Advertisement* and discover each other. They compare announced properties in advertisement with their own VRRP settings. GW2 becomes a new Master. Despite having same priority (value 100), GW2 address 192.168.10.2 is higher.

#3) If Host wants to ping 33.33.33.33, then, the traffic needs to go via default-gateway and Host requests IP-to-MAC mapping with the help of *ARP Request*. Message is delivered to GW1 and GW2, but only GW2 responds with *ARP Reply* because it is Mater. Subsequently, endless ping passes through GW2.

#4) GW2 failure occurs and GW1 seizes to receive *VRRP Advertisements*. GW1's *MDI* expires and next GW1 becomes a new Master sending its own *VRRP Advertisements*. But before that, GW1 sends *ARP Gratuitous Reply* in order to change CAM of SW. Meantime, pings are being dropped since moment of failure until GW1 is elected.

#5) Pings pass through SW towards GW1 and ISP.

#6) GW2 goes up and transits after *MDI* from *Init* to *Backup*. Then, GW2 transits from *Backup* to *Master* state. GW2 sends its own *VRRP Advertisement*, which is superior to ones from GW1, and *ARP Gratuitous Reply* for virtual default-gateway 192.168.10.254. Immediately when GW1 hears GW2's advertisement, GW1 abdicates for being Master router and transits to *Backup* state.

The comparison between timestamps and message confluence can be observed in Table IV.

TABLE IV. TIMESTAMP COMPARISON OF VRRP MESSAGES

Phase	Message	Sender	Simul. [s]	Real [s]
#2	*VRRP Advertisement*	GW1	3.609	3.612
	VRRP Advertisement	GW2	3.609	4.367
	VRRP Advertisement	GW2	4.609	5.286
#3	*ARP Request*	Host	10.000	10.000
	ARP Reply	GW2	10.000	10.034
	IMCP Echo Request	Host	10.000	10.986
#4	*VRRP Advertisement*	GW1	23.219	23.655
	ARP Gratuitous Reply	GW1	23.219	23.643
#6	*VRRP Advertisement*	GW2	33.718	33.612
	ARP Gratuitous Reply	GW2	33.718	33.611

Please notice that Cisco's VRRP implementation sends two *ARP Gratuitous Replies* before any VRRP advertisement. After we had observed this, we implemented another FSM in our VRRP module to accommodate this behavior. However, the routing outcome from Host perspective is same no matter on chosen FSM.

C. *Map-Cache Synchronization*

The goal of the following test is to show the impact of our synchronization technique on a packet drop rate and a number of map-cache misses. A scenario is focused on cache misses due to the missing mapping rather than expired ones because of default TTL (1 day). Five minutes time slot with the single

VRRP Master outage is the simplest illustration of how to compare the impact of map-cache synchronization.

We prepared simulation topology that contains a single LISP site with two routers (xTR1 and xTR2), which provide highly-available VRRP default-gateway for two hosts interconnected by switch SW. Host1 and Host2 are pinging IPv4 EIDs randomly thus generating traffic that triggers LISP mapping system queries. All routing is done statically. Hence, there is no need to employ routing protocol on Core router. We prepared special xTR called xTR_Responder that: a) registers destination EIDs; and b) responds to hosts ICMP messages. The whole topology is depicted in Figure 8.

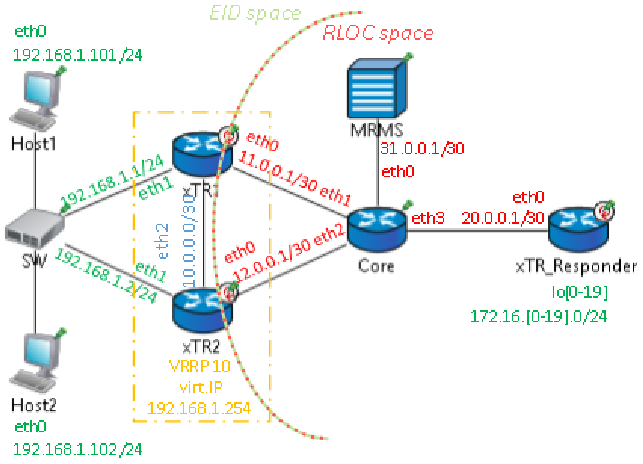


Figure 8. LISP map-cache synchronization testing topology

We scheduled following phases for the test run:

#1) At first, all xTRs register their EIDs. In the case of xTR_Responder, EID space is modeled with the help of loopback interfaces – twenty of them ranging with addresses from 172.16.0.0/24 to 172.16.19.0/24 reachable via single RLOC 20.0.0.1. In case of xTR1 and xTR2, EID 192.168.1.0/24 is reachable via two RLOCs 11.0.0.1 and 12.0.0.1.

#2) xTR1 and xTR2 form VRRP group with VID 10 and virtual address 192.168.1.254, which is used by Host1 and Host2 as default-gateway. xTR1 is Master because of higher priority (xTR1 has 150, xTR2 only 100) as long as it is operational.

#3) Host1 starts pinging ten random EIDs in range from 172.16.0.0/24 to 172.16.9.0/24. Because EIDs are chosen randomly, they may be duplicate. Each first ICMP packet causes mapping query and is dropped.

#4) Then, xTR1 failure occurs right before a new LISP registration (at t=119s). Hosts traffic is diverted to a new VRRP Master, which is xTR2.

#5) After #4, also Host2 starts to ping ten random EIDs from 172.16.10.0/24 to 172.16.19.0/24. Same duplicity rule as in #3 applies.

#6) xTR1 recovers from outage at t=235s and once again all hosts traffic goes through it.

Depending on map-cache synchronization type, additional map-cache misses might occur. xTR1 and xTR2 synchronized themselves via 10.0.0.0/30 connection, which forms dedicated SS. xTR1 uses address 10.0.0.1 and xTR2 address 10.0.0.2.

The scenario has been tested with three simulation configurations each representing different map-cache synchronization technique: α) no synchronization at all (default LISP behavior); β) naïve mode; and γ) smart mode. Impact on map-cache is summarized in Table V for all previously mentioned different configuration runs.

Before interpreting the results, please note that Host1 randomly (using same seeds for all three runs) chose 8 different EIDs, Host2 6 EIDs, totally 14 distinct ping destinations.

TABLE V. MAP-CACHE MISSES FOR DIFFERENT CONFIGURATIONS

Phase	α cache misses		β cache misses		γ cache misses	
	xTR1	*xTR2*	*xTR1*	*xTR2*	*xTR1*	*xTR2*
#3	8	0	8	0	8	0
#5	0	14	0	6	0	6
#6	14	0	0	0	0	0
Total	22	14	8	6	8	6

Without any synchronization, traffic diversion to a new VRRP Master always causes misses due to unknown mappings. We can see this in phases #5 and #6 for α-run, when router starts to dispatch LISP data with the empty map-cache.

If the synchronization is employed, then, only new destinations lead to map-cache miss. The reason is that a new VRRP Master already have mappings discovered by neighbor xTR. Hence, there is a difference in phase #5 for α-run (empty cache) and β/γ-runs (cache in sync with SS member). β-and γ-runs are equal in the number of cache misses but γ- run is more effective in protocol overhead. The difference (36 cache misses versus 14) would be even more significant in the case of multiple VRRP Master outages. Please note that every map-cache miss is also connected with the data packet drop.

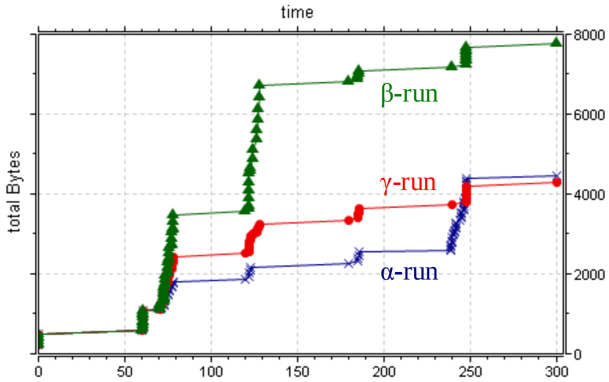


Figure 9. Total size of processed LISP control messages by xTR1

In order to compare synchronization modes, we conducted measurement taking into account all LISP control messages processed by lispCore module, namely their packet sizes. We assume that larger size is always greater burden for router’s control plane processing. Figure 9 shows the results (α-run = blue crosses, β-run = green triangles, γ-run = red circles), where is visible that smart outperforms naïve. The reason for being less intensive is that only single mapping is transferred during synchronization, not a whole map-cache. Moreover, smart mode is even better than no synchronization because it decreases number of mapping queries. It is even more apparent in the same scenario but with more VRRP outages (see [16]).

VI. CONCLUSION

In this paper, we presented a detailed description of LISP and VRRP technologies. We proposed LISP improvement aimed to achieve a better routing performance primarily in high-availability use-cases, e.g. data-centers with mission-critical applications sensitive to packet drops. We evaluated the impact of our improvement using OMNeT++ simulator. In order to achieve this objective and relevant results, we first thoroughly developed LISP and VRRP simulation modules that mimic behavior of real implementations.

Validation testing against a real-life topology shows very reasonable time variations for LISP and VRRP functionality. However, simulation results are affected by simpler simulated control-plane and the simulated processing time does not include all processes running on a real router. Hence, some simulation timestamps in Table III and IV are below one millisecond accuracy. Time variation observable on real Cisco devices is caused by three factors: a) control-plane processing delay and internal optimizations; b) packet pacing avoiding race conditions; and c) inaccuracy in timing of certain events 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.

During our tests, we closely observed RLOC-probing algorithm that Cisco devices are using to verify locator reachability. ITR is checking assigned locators for each configured EID. Although this often leads to repeated check of the same locator multiple times, which represents scalability issue in larger networks. Therefore, we already implemented enhancement that reduces LISP protocol overhead and its precise evaluation is a future research task.

We plan to carry on the work on simulation modules to further test them in more realistic network simulations. Proxy ITR/ETR capability, solicit map-requests and different mapping distribution systems (e.g., LISP-ALT, LISP-DDT) are on our LISP development roadmap. We would like to upgrade VRRP to support IPv6 addresses and all features of VRRP version 3.

All source codes could be downloaded from GitHub repository [16]. Real packet captures and simulation datasets for the results reproduction could be downloaded from Wiki of the repository mentioned above. More information about ANSA project is available on its homepage [17].

ACKNOWLEDGMENT

The VRRP simulation core is work by Petr Vítek, a former student of the Brno University of Technology. This work was supported by following organizations and research grants:

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

REFERENCES

[1] D. Meyer, L. Zhang, and K. Fall, "RFC 4984: Report from the IAB Workshop on Routing and Addressing," September 2007. [Online]. Available: http://tools.ietf.org/html/rfc4984.

[2] T. Li, "RFC 6227: Design Goals for Scalable Internet Routing," May 2011. [Online]. Available from: http://tools.ietf.org/html/rfc6227.

[3] R. Hinden, "RFC 1955: New Scheme for Internet Routing and Addressing (ENCAPS) for IPNG," June 1996. [Online]. Available: http://tools.ietf.org/html/1955.

[4] D. Klein, M. Hoefling, M. Hartmann, and M. Menth, "Integration of LISP and LISP-MN into INET," in Proceedings of the IEEE 5th International ICST Conference on Simulation Tools and Techniques, Desenzano del Garda, Italia, March 2012, pp. 299-306, ISBN 978-1-4503-2464-9.

[5] D. Klein, M. Hartmann, and M. Menth, "NAT Traversal for LISP Mobile Node," July 2010. [Online]. Available: http://tools.ietf.org/html/draft-klein-lisp-mn-nat-traversal.

[6] J. Kim, L. Iannone, and A. Feldmann, "A deep dive into the LISP cache and what ISPs should know about it," NETWORKING 2011, May 2011, vol. 6640, pp. 367-378.

[7] V. Veselý, M. Marek, O. Ryšavý, and M. Švéda, "Multicast, TRILL and LISP Extensions for INET," International Journal On Advances in Networks and Services, 2015, vol. 7, no. 3&4, unpublished.

[8] IETF, "Locator/ID Separation Protocol (lisp)," January 2015. [Online]. Available: http://datatracker.ietf.org/wg/lisp/charter/. [Retrieved: January 2015].

[9] D. Farinacci, V. Fuller, D. Meyer, and D. Lewis, "RFC 6830: The Locator/ID Separation Protocol (LISP)," January 2013. [Online]. Available: http://tools.ietf.org/html/rfc6830.

[10] V. Fuller, "RFC 6833: Locator/ID Separation Protocol (LISP) Map-Server Interface," January 2013. [Online]. Available: https://tools.ietf.org/html/rfc6833.

[11] R. Hinden, "RFC 3768: Virtual Router Redundancy Protocol (VRRP)," April 2004. [Online]. Available: https://tools.ietf.org/html/rfc3768.

[12] S. Nadas, "RFC 5798: Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6," March 2010. [Online]. Available: https://tools.ietf.org/html/rfc5798.

[13] R. Froom, E. Frahim, and B. Sivasubramanian, CCNP Self-Study: Understanding and Configuring Multilayer Switching, Cisco Press, 2005.

[14] D. Saucez, O. Bonaventure, L. Iannone, and C. Filsfils, "LISP ITR Graceful Restart," December 2013. [Online]. Available: https://tools.ietf.org/html/draft-saucez-lisp-itr-graceful-03.

[15] D. Saucez, J. Kim, L. Iannone, O. Bonaventure, and C. Filsfils, "A Local Approach to Fast Failure Recovery of LISP Ingress Tunnel Routers," NETWORKING 2012, May 2012, vol. 7289, pp. 397-408, ISBN: 978-3-642-30044-8.

[16] GitHub, December 2013. [Online]. Available: https://github.com/kvetak/ANSA/wiki/ICNS-2015. [Retrieved: January 2014].

[17] Brno University of Technology, January 2014. [Online]. Available: http://nes.fit.vutbr.cz/ansa/pmwiki.php. [Retrieved: January 2014].