Map-Cache Synchronization and Merged RLOC Probing Study for LISP

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 Separation Protocol is alternative routing paradigm, which tries to solve limitations (cumbersome support of mobility, multihoming, inbound traffic engineering, renumbering and rapid growth of default-free zone routing tables) of traditional TCP/IP routing model. The presented work deals with a map-cache synchronization and merged RLOC probing, which are outlined and evaluated as possible solutions improving performance and reducing the overhead of LISP. The proposed extension is evaluated using simulation model built for OMNet++ tool.**

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

I. INTRODUCTION

This paper extends our previous work published in [1] and [2]. We refined implementation of previously developed simulation modules; we designed and investigated two merged RLOC algorithms; and we conducted new measurements.

Locator/ID Separation Protocol (LISP) development started after IAB Workshop in 2006 as a response to problems described in RFC 4984 [3] and RFC 6227 [4]. LISP should reduce default-free zone (DFZ) routing table growth, stop prefix deaggregation, allow easier multihoming and mobility without the Border Gateway Protocol (BGP). LISP can be deployed transparently without any changes needed for hosts or domain-name system (DNS). LISP is agnostic to any network protocol and support not only IPv4 and IPv6 but any other future protocol operating on L3. LISP provides communication with the legacy non-LISP world because transition mechanism is an integral part of LISP specification. Nevertheless, the enterprise is always skeptical and slow when adopting a new technology. Hence, it is a great research challenge to investigate LISP features using modeling and simulation as the referential testbed tools producing meaningful outcomes.

IP address functionality is dual. It serves for identification ("which device is it?") and localization ("where is the device?") purposes. The consequence of this overloading is the inability to build scalable and long-term effective DFZ routing system. 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:

- **Endpoint Identifier (EID)** namespace (so called LISP site), where each device has unique address;
- **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 the 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 [5]). 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. Figure 1 shows LISP architecture components including xTRs.

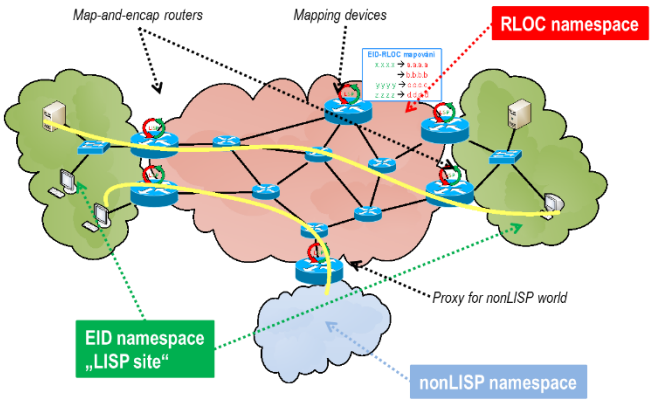

Figure 1. LISP reference model

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 a mapping is retrieved. This behavior is analogous to the DNS protocol and allows LISP to operate a decentralized database of EID-to-RLOC mappings.

Replication of the 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 a 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 and 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 (a 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 forwards clients' traffic within each group, where backup routers are checking master's liveness ready to substitute it. Switching to a new active router is transparent from the host's perspective thus no additional configuration or special software is needed.

The Automated Network Simulation and Analysis (ANSA) project aims to develop a variety of RFC-compliant simulation models to provide researchers and network administrators with a reliable verification tool. This paper provides more detail description of implemented and further refined simulation models, which create a part of the ANSA project and which extend the functionality of the INET framework version 2.4 in OMNeT++.

This paper has the following structure. The next section describes the design of relevant LISP and VRRP models. Section III 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 IV presents validation scenarios for outlined implementations and shows promising results backing up improvement's impact on LISP operation. The paper is summarized in Section V together with the unveiling of our plans.

II. IMPLEMENTATION

A. *LISP – A Theory of Operation*

LISP is being codified within IETF [6]. The main core and functionality are 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 the 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 handles registering all LISP sites (their EID addresses) and by which RLOCs they are accessible. If we inspect structure of LISP packet somewhere in RLOC space then:

- Inner header source IP = sender's EID address;
- Inner header destination IP = receiver's EID address;
- Outer header source IP = ITR's RLOC address;
- Outer header destination IP = ETR's RLOC address.

LISP mapping system consists of two components – **Map Resolver (MR)** and **Map Server (MS)**. Looking for EID-to-RLOC mapping is an analogous process as DNS name resolution (see Figure 2). In the case of DNS, the host asks its DNS resolver (configured within OS) which IP address belongs to a given FQDN. DNS server responds with a cached answer or delegates the question recursively or iteratively to another DNS server according to the name hierarchy. For LISP, querier is ITR that needs to find out, which RLOCs could be used to reach a given EID. ITR has preconfigured MR, which is bothered each time mapping is needed. Just as in the case of DNS, mapping queries are data-driven. This means that data transfer between LISP sites initiates mapping process and data itself are postponed until a mapping is discovered. Map-cache on each ITR holds only those records that are actively needed for ongoing traffic.

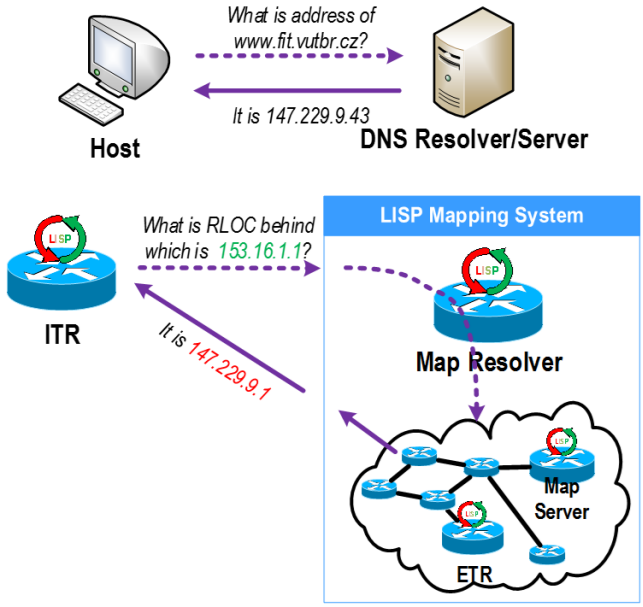

Figure 2. Comparison between DNS and LISP mapping system

The list below contains all LISP control messages responsible for mapping systemsignalization.

- *LISP Map-Register* – Each ETR announces LISP site(s) to the map server utilizing this message. Each registration contains authentication data and the list of mappings and their properties.
- *LISP Map-Notify* – UDP cannot guarantee message delivery. The map server may optionally (when the proper bit is set) confirm a 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 mappings of RLOCs to a certain EID together with their attributes. Each ITR has its map-cache where the reply information is stored for a limited time and used locally to reduce the signalization overhead of mapping system. Moreover, the mapping system generates *LISP Negative Map-Reply* as a response whenever a given identifier is not the EID, and thus proxy routing for non-native LISP communication must occur.

A map resolver processes ITR's *LISP Map-Requests*. Either the map resolver 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 a mapping systemto the appropriate map server.

Every map server maintains **mapping database** of LISP sites that are advertised by *LISP Map-Register* messages. If the map server receives *LISP Map-Request* then: a) either the map server responds directly to querying ITR; or b) the map server forwards request towards designated ETR that is registered to a map server for the 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.

The following demonstration should help the reader to get more familiar with LISP data traffic. Figure 3 depicts two LISP sites ("Site A" using EID prefix 100.0.0.0/24 and "Site B" with prefix 200.0.0.0/24) that are interconnected via RLOC space composed of five ISP networks. `PC-A` with address 100.0.0.99 wants to perform unicast data transfer to `PC-B` with address 200.0.0.99. EIDs are transparent for hosts and end-stations do not concern about LISP routing. The steps necessary to complete this scenario are following:

#1) DNS resolver returns EID as IP address associated with `PC-B` (e.g., pc.siteb.com A 200.0.0.99);

#2) The packet traverses "Site A" until it reaches `xTR-A2`. This router acts as ITR and prepares appropriate outer header for encapsulation. RLOC is looked up in map-cache based on destination EID 200.0.0.99, and RLOC 4.0.0.1 is chosen due to the lowest priority;

#3) Packet traverses RLOC space with 2.0.0.1 as source and 4.0.0.1 as destination RLOC in the outer header and with 100.0.0.99 as the source and 200.0.0.99 as destination EID in the inner header.

#4) The packet is routed via ISPs until it reaches `xTR-B2`'s interface with address 4.0.0.1. This router performs decapsulation and forwards packet to "Site B" based on destination EID address;

#5) The packet is delivered to `PC-B` having the same structure (single IP header, EIDs as addresses) as it was in #1. LISP functionality is transparent to end-point devices and non-LISP routers.

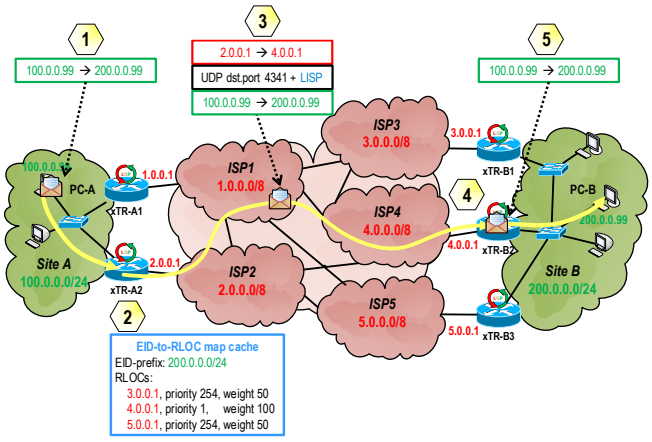

Figure 3. Illustrative LISP unicast data transfer

B. LISP – Design of a Simulation Module

A 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 4 and described in Table I below. Implementation is in full compliance with definitions from RFC 6830 [7] and RFC 6833 [8].

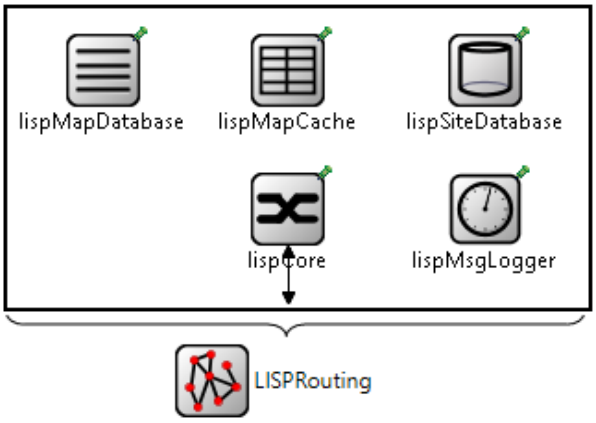

Figure 4. LISPRouting module structure

All LISP abstract data structures and settings can be configured statically (using XML file before simulation beginning). Map-cache and map/site database are implemented using generic class `LISPMapStorage` that is extended via C++ inheritance to accommodate different requirements of each control plane component. Every `LISPMapStorage` contains the ordered list of `LISPMapEntry` instances.

TABLE I. DESCRIPTION OF LISP ROUTING SUBMODULES

Name	Description
`lispCore`	The module handles LISP control and data traffic. It independently combines the 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 the 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 the 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 [9] describes IPv4-only VRRPv2 and RFC 5798 [10] 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 a superior willingness to become Master, zero priority causes the 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). Preemption 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 Backup routers 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.

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

D. VRRP – Design of Simulation Module

VRRP version 2 is implemented as `VRRPv2` compound module connected with `networkLayer`. The 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 5, 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 [9]. VRRP FSM's states *Init*, *Backup* and *Master* reflect VRRP router role and govern control message generation and processing.

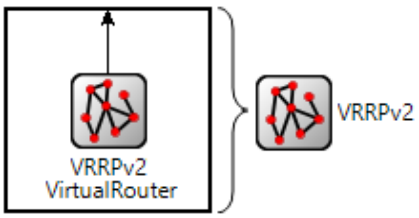

Figure 5. VRRP modules structure

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, the transition between states and directs ARP for a single VRRP group.

III. CONTRIBUTION

This section identifies our contribution that we made to LISP specification. The contribution consists of providing efficient solutions to two of LISP known problems, namely, Site-Based State Synchronization Problem and Locator Path Liveness Problem.

Assume multiple redundant routers are acting as first hops in the high-availability scenario like in Figure 6. 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. This fact is illustrated in Figure 6 for EID address y.y.y.y. On the one hand, packets (with y.y.y.y as destination) can traverse `ITR1` without any problem (locator is present in map-cache) but on the other hand, same packets are discarded on `ITR2`, which misses the mapping. 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 [11], where any triggering traffic should be discarded to protect control-plane processing and prevent superfluously recurrent mapping system queries.

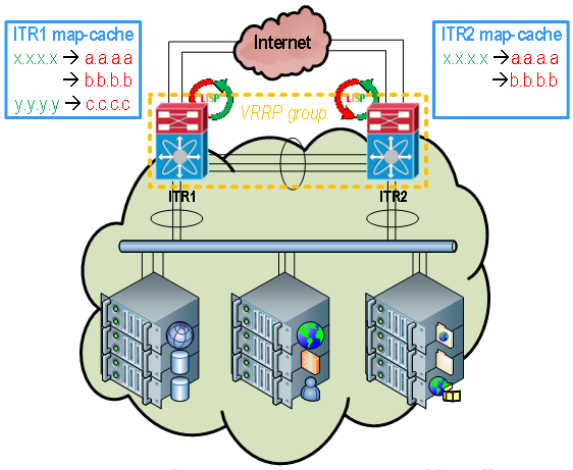

Figure 6. Site-Based State Synchronization Problem illustration

Each xTR has own map-cache, and its content may differ even within the same LISP site because different traffic may initialize various cache record. 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 is known as **Site-Based State Synchronization Problem**. If we have two or more redundant xTRs, then we want to reduce packet drops as much as possible during the intermittent phase of switching to a new active device. xTR outage leads to the off-site signalization storm (lots of *LISP Map-Request/Reply* messages being exchanged) and dispatching delay for ordinary traffic.

This problem is described as the one of LISP weak-points in [12] and theoretically investigated in [13]. The viable solution would be to provide map-cache content synchronization that should minimize map-cache misses upon failure. We present our solution addressing this problem based on this assumption.

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.

SS members are identified and reached using the IP address. Following strategies might be used when choosing appropriate SS member address:

- SS address comes from non-LISP world – Either IP address should be loopback or address of dedicated interconnection shared by all SS members. In the first case, unique device loopbacks need to employ additional routing. In the second case, the additional port for the dedicated connection is seldom available. Also, tracking of SS member needs additional LISP control plane updates;
- SS address comes from LISP world:
  - SS address is RLOC – SS membership is bound to the operability of a given RLOC interface, but this has negative implications for the situation, where xTR has more than one RLOC available. Although, it is easy to track SS member status using return value of RLOC probing;
  - SS address is EID – The best option reflecting LISP's ideology. EID as SS address should be reachable via direct routing (xTRs share common EID segment) or unless all RLOCs to this EID are down (which could be also used to track peer synchronization status).

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. By default, every record uses the same initial TTL value. 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 traffic demands). Either SS membership may be completely stateless, or SS member may maintain a state of its synchronization peers. This allows sending of partial synchronization updates. 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 a record that caused synchronization is transferred. Moreover, we adhere to the following policy. When TTL expires, the ITR must check record usage during the last minute (one minute should be a period long enough to detect ongoing communication). If the mapping has not been used, then, it is removed from the cache. Otherwise, its state is refreshed in a query followed by the necessary data synchronization.

Both approaches guarantee that devices within SS could forward rerouted LISP data traffic without a packet loss because they share the same content as ITR's map-cache of the former Master device.

The proposed solution employing synchronization defines a new mechanism that introduces two new control messages (one carries synchronization data, another optionally acknowledges successful synchronization):

- *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.

The following discussion explains the issues related to the reuse of existing LISP mechanism and advocates the proposed extension. The first approach would be to alter existing *LISP Map-Requests* by forcing included map-reply record field to contain more than one record. However, this approach is unreliable because it lacks acknowledgment scheme. The second approach would be to leverage so-called **Solicit-Map-Request (SMR)**. SMR is a mechanism how ETRs may rate-limit requests and notify ITRs about mapping change. When mapping changes, ETR starts to send *LISP Map-Request* (with the SMR-bit set) to ITRs with which it recently exchanged data. Then, ITR generates SMR-invoked *LISP Map-Request* to discover new mapping. If we want to use SMR to push new mappings into ITR's map-cache, then the best way seems to be extending the functionality of MR (see [13]). However, this approach yields significant off-site signalization.

Locator Path Liveness Problem concerns whether a destination locator is reachable via a particular source locator. This ensures the existence of bi-directional connectivity between a given pair of locators. A problem relevant to LISP is depicted in , where *xTR-A1* asks for "Site B" locators. In this case, two locators are available (1.0.0.1 and 2.0.0.1). *xTR-A1* chooses the second one as a destination address for packets. If the link between *ISP1* and *ISP2* goes (un)intentionally down, 2.0.0.1 is not reachable anymore, and *xTR-A1* must somehow find out this fact.

The simple method for Locator Path Liveness detection does not scale well in large networks because the reachability of every destination locator must be probed against every source locator of a given device. Complexity of such a task is generally $O(n \times m)$, where n is a number of source and m a number of destination locators. However, instead of brute-force probing some hints might be used to mitigate (but not to avoid) such complexity, e.g., piggybacking, timeouts, knowing of underlying routing, or positive feedback from control protocol messages.

To make Locator Path Liveness Problem even more complicated, let us imagine a situation when the LISP site has two or more ITRs with different destination locator reachability. One ITR has connectivity, and another has not

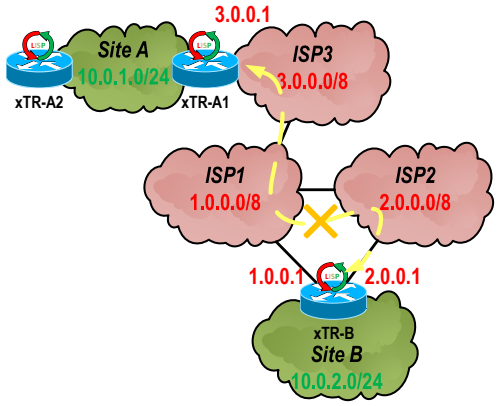

Figure 7. Locator Path Liveness Problem illustration

(e.g., *xTR-A1* and *xTR-A2* on). Hence, all packets processed by that ITR is going to be discarded somewhere in topology. Because of LISP transparency, neither routing protocol nor hosts have capabilities to detect this issue and inform LISP devices accordingly.

In order to find a remedy for this problem, we focused on the behavior of Cisco referential implementations and their RLOC probing algorithm checking locator reachability. ITR is probing assigned locators for each configured EID. This is in compliance with [7], but it leads to repeated check of the same locator multiple times, which represents a significant overhead larger networks.

We decided to reduce protocol overhead by merging EIDs to check locator liveness with single RLOC probe that we call **merged RLOC probing**. This method if based on the following assumption: "If the same locator is reachable for one EID then it would also be reachable for other EID." Hence, the router can generate only one RLOC probe during a single liveness checking period. If it receives a positive *LISP Map-Reply Probe*, it may consider probed locator as alive for all EIDs in map-cache that are using it. More *sophisticated* approach is to perform the following steps:

1) On the sender side, check liveness of a given locator with a single *LISP Map-Request Probe* containing one or more query records. Each query record specifies cached EID that uses probed RLOC;
2) On the receiver side, respond with *LISP Map-Reply Probe* that includes locator status updates for all queried EIDs contained in request (or only subset of those EIDs that are in up state);
3) Back on the sender, refresh a locator status of relevant EIDs in map-cache according to answer(s) in reply.

Above described mechanism is compatible with RFC description and does not need any protocol extensions. Yet, it preserves the accuracy of Cisco's RLOC probing algorithm but with only single RLOC probe exchanged.

We have integrated all above described algorithms – *Cisco's*, *Simple* and *Sophisticated* – in our LISP simulation module and perform their evaluation as described further.

IV. TESTING

In this section, we provide information regarding: a) validation of LISP and VRRP simulation models (the goal

is to build reliability of implemented simulation models); and b) evaluation of map-cache synchronization and merged RLOC probing (the goal is to show the impact of deployed techniques on LISP operation).

Validation is based on the comparison (i.e., message order and timestamps) of behavior with the referential implementation. Therefore, we have built exactly the same real network topologies as for simulations. We captured and analyzed relevant messages exchanged between devices for both LISP and VRRP functionality. We compared the results with the behavior of an 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 8. 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 the 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.

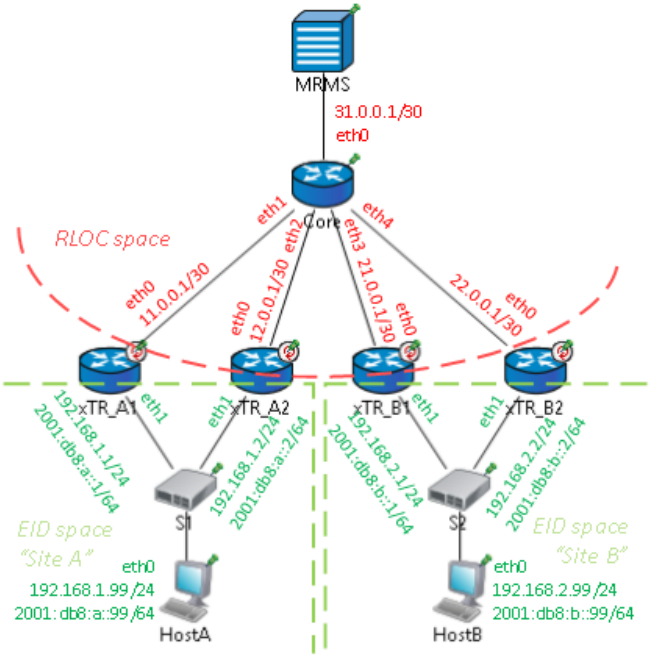

Figure 8. LISP testing topology

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 a polling mechanism that checks the reachability of announced locators. Each ETR sends *LISP Map-Request* with a 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 a 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 9 shows `xTR_B1`'s "Site B" registration after #1.

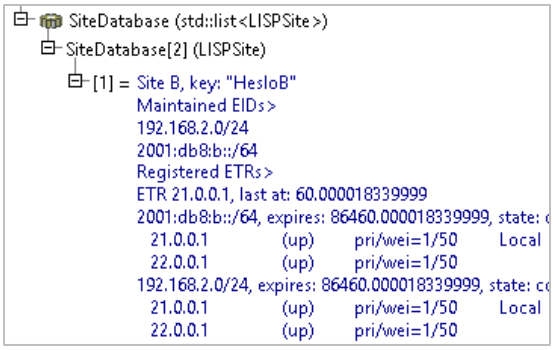

Figure 9. xTR_B1's registration of "Site B"

#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. The first ping is dropped due to that. `xTR_A1` sends *LISP Map-Request* to MS. MRMS performs a 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 (two RLOCs 21.0.0.1 and 22.0.0.1 belong to EID 192.168.2.0/24). Figure 10 illustrates this result.

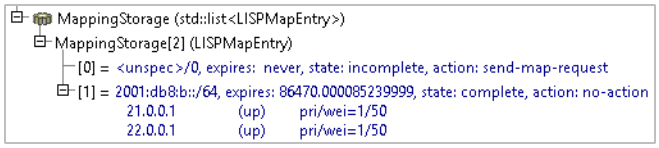

Figure 10. Content of xTR_A1's map-cache after phase #3

#4) The second ping arrives on `xTR1_A1`. Because the mapping is known, it is encapsulated with an 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 11 shows the result.

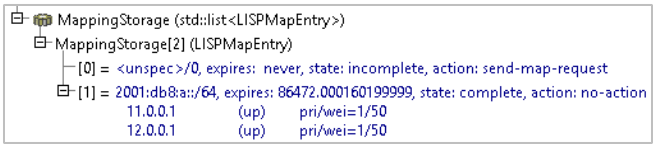

Figure 11. Content of xTR_B1's map-cache after phase #4

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

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 12. Simulation network contains two VRRP routers (`GW1` and `GW2`) clustered in VRID 10. A single switch (`SW`) interconnects devices on the local segment. Host (`Host`) and router (`ISP`) pair substitute 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.

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.

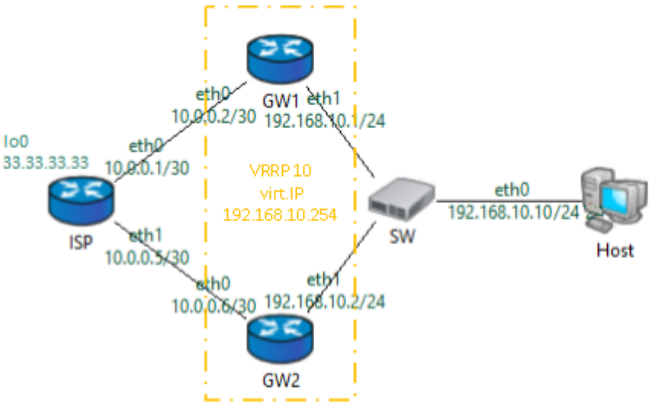

Figure 12. VRRP testing topology

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*. The 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. Impact of Map-Cache Synchronization

There are two goals behind the following test. The first one is to show the impact of synchronization on a packet drop rate and map-cache misses. The second one is to enumerate the burden of deploying map-cache synchronization on the control plane. 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 (172.16.[0-19].0/24) 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 13.

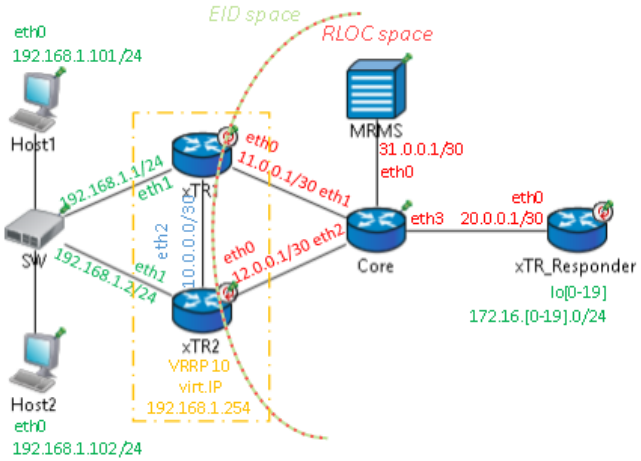

Figure 13. LISP testing topology for map-cache synchronization

During the outage, all xTR1's interfaces shutdown (i.e., they are physically disconnected from the network). Yet, the xTR1's control plane is operational – generating scheduled LISP messages that are not being delivered. 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 the 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 via their RLOCs (11.0.0.1 for xTR1 and 12.0.0.1 for xTR2).

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. Fewer map-cache misses are considered better.

We do not employ LISP synchronization acknowledgment scheme for β/γ-runs, the impact of acks is analyzed later. The scenario offers testing of all three kinds of addressed for SS member identification – e.g., nonLISP with 10.0.0.0/30; RLOC with 11.0.0.1 and 12.0.0.1; and EID with 192.168.1.1 and 192.168.1.2) with same results. Nevertheless, we use EIDs as the most feasible options. Before interpreting the results, please note that Host1 randomly (using same seeds for all three runs) chose eight different EIDs, Host2 six EIDs, totally fourteen distinct ping destinations.

TABLE V. MAP-CACHE MISSES IN SCENARIO WITH ONE OUTAGE

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 the 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. This is because a new VRRP Master already has 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.

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 a greater burden for router's control plane processing. Figure 14 shows the results (α-run = blue crosses, β-run = green triangles, γ-run = red circles), where each symbol represents one LISP control message.

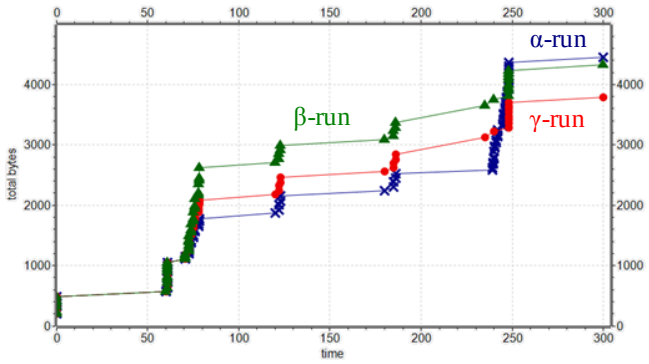

Figure 14. xTR1's LISP control messages occurrence and total processed byte size in scenario with single outage

We can see that *smart* outperforms *naïve* because it is less intensive while only single mapping is transferred during synchronization, not a whole map-cache. Moreover, both synchronization modes are better than no synchronization on protocol overhead because they decrease the number of mapping queries (i.e., exchanged messages count). The difference is not that significant on Figure 14, especially between naïve and no sync mode. However, it is getting more obvious as the number of VRRP outages increases. Following table and figure prove this claim for the same topology but with two xTR1 outages – phases #4 and #6 repeat twice.

TABLE VI. MAP-CACHE MISSES IN SCENARIO WITH TWO OUTAGES

Phase	α cache misses		β cache misses		γ cache misses	
	xTR1	*xTR2*	*xTR1*	*xTR2*	*xTR1*	*xTR2*
#3a	8	0	8	0	8	0
#5a	0	14	0	6	0	6
#6a	14	0	0	0	0	0
#5b	0	0	0	0	0	0
#6a	14	0	0	0	0	0
Total	36	14	8	6	8	6

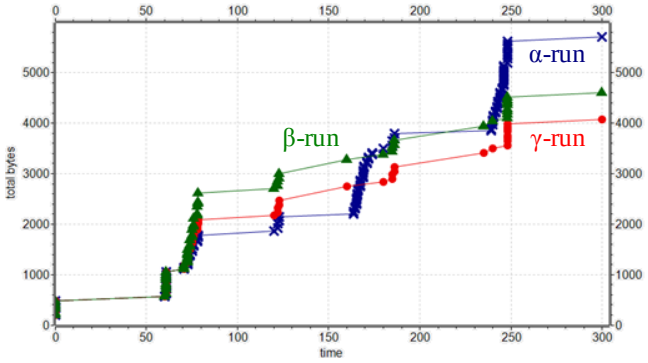

Figure 15. xTR1's LISP control messages occurrence and total processed byte size in scenario with two outages

Repetition of phases 4), 5) and 6) is denoted in Table VI with letters: "a" for the first outage; and "b" for the second outage. In Table VI, we can observe that a total number of cache misses for α-run has increased by 14. `xTR1` had gone down (losing its map-cache content), then went back (repopulating map-cache once again with 14 EIDs) and then this cycle repeats once again. For β-run and γ-run, additional outages pose no change, because `xTR1` completely synchronizes itself with `xTR2` (`xTR2` sends the whole map-cache as soon as it detects the status of the one of `xTR1`'s RLOCs up), when it is once again operational. Figure 15 shows an increase in a number of processed LISP control message for no synchronization, where impacts of other synchronization techniques remain same.

LISP synchronization acknowledgment mechanism poses an additional control plane burden. In order to evaluate acknowledgment impact, we conducted measurement on the same topology with two outages. The results in a number of processed LISP control messages bytes are depicted in Figure 16 and can be compared with Figure 14.

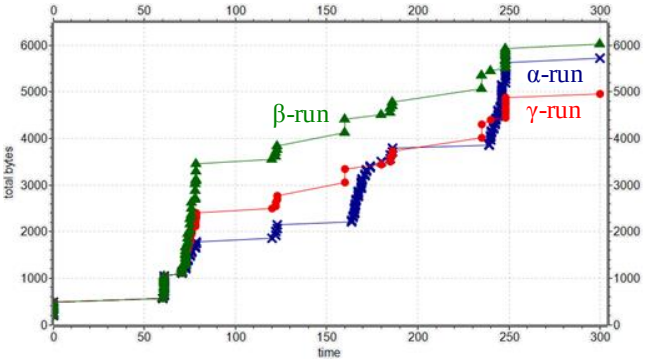

Figure 16. xTR1's LISP control messages occurrence and total processed byte size in scenario with two outages + acknowledgments

It is apparent that protocol overhead on the number of messages has increased. In the case of no synchronization, it slightly outperforms naïve mode by a total size of processed bytes. However, the smart mode still has the best characteristic even with enabled acknowledgments. Once again, we can expect that additional outages or more EID ping destinations would influence results in favor of β/γ-runs over α-run (see Figures 17 and 18).

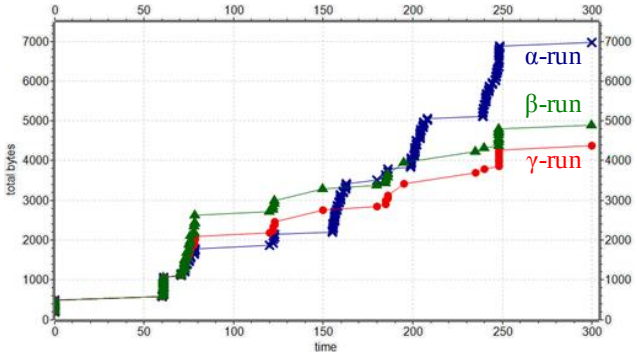

Figure 17. xTR1's LISP control messages occurrence and total processed byte size in scenario with three outages

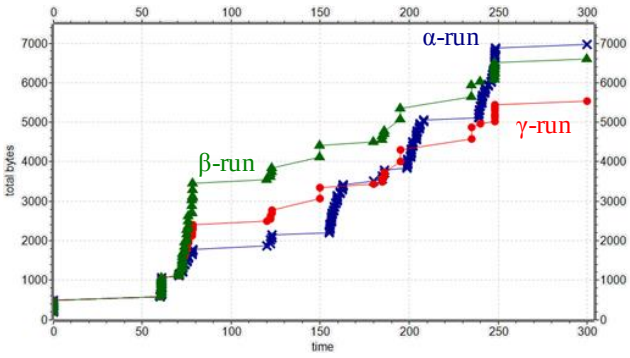

Figure 18. xTR1's LISP control messages occurrence and total processed byte size in scenario with three outages with ack

TABLE VII. XTR1'S STATISTICS FOR DIFFERENT MAP-CACHE SYNCHRONIZATION SCENARIOS

single xTR1 outage scenario								
α			β			γ		
miss	cnt	size	miss	cnt	size	miss	cnt	size
22	81	4 458	8	62	4 328	8	62	3 796
two xTR1 outages scenario								
α			β			γ		
miss	cnt	size	miss	cnt	size	miss	cnt	size
36	109	5 718	8	63	4 614	8	63	4 082
three xTR1 outages scenario								
α			β			γ		
miss	cnt	size	miss	cnt	size	miss	cnt	size
50	137	6 978	8	64	4 900	8	64	4 368

single xTR1 outage with sync ack scenario								
α			β			γ		
miss	cnt	size	miss	cnt	size	miss	cnt	size
22	81	4 458	8	71	5 458	8	71	4 394
two xTR1 outages with sync ack scenario								
α			β			γ		
miss	cnt	size	miss	cnt	size	miss	cnt	size
36	109	5 718	8	73	6 030	8	73	4 966
three xTR1 outages with sync ack scenario								
α			β			γ		
miss	cnt	size	miss	cnt	size	miss	cnt	size
50	137	6 978	8	75	6 602	8	75	5 538

Table VII summarizes the evaluation of map-cache synchronization techniques. The table shows α/β/γ-run (i.e., none, *naïve* and *smart* sync) statistics for different scenarios (one/two/three outage(s) with or without acknowledgment). `xTR1`'s statistic numbers are depicted with following column meanings: "miss" as the number of map-cache miss occurrence; "cnt" as the total count of LISP control plane messages sent and received; "size" as processed messages count by LISP control plane measured in total byte size. We added to Table VII also same statistics section for a scenario with three outages in order to analyze trends. Results show a linear growth in complexity.

D. Impact of Merged RLOC Probing

The goal of the following subsection is to measure the impact of merged RLOC probing on control plane processing.

We took the previous topology and adjusted it; see the result in Figure 19. Currently, it contains a LISP site with just one `xTR` router and one end-device called `Host1`. More important are LISP sites that are reachable via `xTR_Responder1` and `xTR_Responder2`. We simulate multiple EID networks reachable via the same xTRs with the help of loopback interfaces. Each xTR_Responder has forty loopbacks with EID addresses in the range of 172.16.[0-39].0/24. Each EID is being registered towards MRMS as reachable via `xTR_Responder1`'s RLOC 21.0.0.1 and `xTR_Responder2`'s RLOC 22.0.0.1. VRRP functionality on xTR is disabled because it is not needed for this scenario. `Host1` might randomly generate ICMP traffic towards destination EIDs, but this is not necessary for merged RLOC probing analysis. All communicating parties are interconnected via `Core` employing static routing configuration.

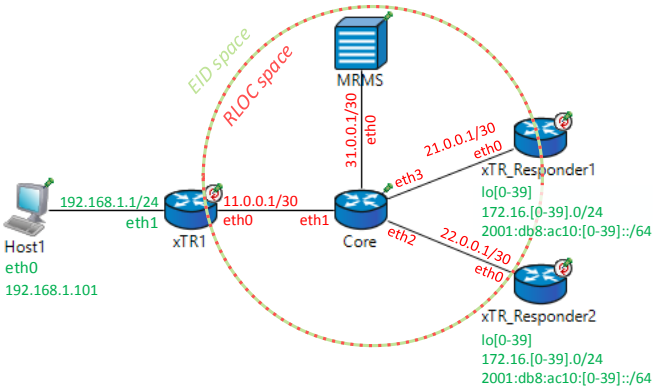

Figure 19. LISP testing topology for merged RLOC probing

RLOC probing starts immediately after LISP routing control plane is initialized. Following phases occur no matter on used RLOC probing algorithm:

#1) Probing xTR sends *LISP Map-Request Probe* to RLOC address for a given set of EIDs;
#2) Probed xTR responds with *LISP Map-Reply Probe* announcing that RLOC is up;
#3) In case that *LISP Map-Request Probe* was not replied, probing xTR repeats the probe at time $t_{next} = t_{last} + 2^{numOfRetries}$, where t_{last} is the last time probe was sent and $numOfRetries$ is number of retry attempts to send this probe. After by default three

unsuccessful *LISP Map-Request Probe*, mark RLOC as down and schedule next probe after 60 seconds.

Optional phase #3) behavior is solely based on Cisco implementation observations. Also Cisco's LISP implementation has some other specifics: a) postponed start of first EID registration ($t + 60$ seconds since control plane initialization); b) postponed start of RLOC probing for IPv6 RLOCs ($t + 30$ since the first IPv4 probe). We have integrated this behavior into the LISP simulator. However, we are not employing it in order to provide better readability of this scenario's results.

Those phases repeat by default every minute in order to keep RLOC reachability up-to-date. This interval could be decremented to a lower value, but protocol overhead increases in an inverse relationship.

Measurement is focused on a number of *LISP Map-Request/Reply Probes* exchanged between `xTR_Responder1` and `xTR_Responder2` and the amount of corresponding bytes processed by `xTR_Responder1`'s LISP control plane. We assume that five minutes simulation time is a period long enough to show the trend of each RLOC probing algorithm. During this period, five RLOC probe batches occur. Except mandatory EID registrations, no other LISP control traffic is spoiling the results.

We have conducted two simulation scenarios in order to observe complexity trends. The first one is for the topology with forty different EIDs (twenty IPv4 172.16.[0-19].0/24 and twenty IPv6 2001:db8:ac10:[0-19]::/64) on xTR_Responders reachable via RLOCs 21.0.0.1 and 22.0.0.1, the second with eighty different EIDs (forty IPv4 172.16.[0-39].0/24 and forty IPv6 2001:db8:ac10:[0-39]::/64). All three algorithms are evaluated separately as different configuration simulation runs - Cisco's default algorithm as δ-run, simple as ε-run and sophisticated as λ-run algorithm variants of merged RLOC probing.

TABLE VIII. XTR_RESPONDER1'S STATISTICS FOR DIFFERENT RLOC PROBING ALGORITHM SCENARIOS

40 EIDs scenario					
δ		**ε**		**λ**	
cnt	**size**	**cnt**	**size**	**cnt**	**size**
805	55 500	25	8 520	25	28 530
80 EIDs scenario					
δ		**ε**		**λ**	
cnt	**size**	**cnt**	**size**	**cnt**	**size**
1 605	110 900	25	15 920	25	56 330

Total count of sent and received LISP control messages are shown in Table VIII with following meaning of columns: "cnt" as the total count of LISP control plane messages sent and received; "size" as the amount processed messages by LISP control plane measured in total byte size.

Apart from five *LISP Map-Register*, *xTR_Responder1* five times: a) sends *LISP Map-Request Probe* and receives *LISP Map-Reply Probe*; b) receives *xTR_Responder2*'s probes and responds to them with replies. It is apparent that count of exchanged messages is drastically lower when using any merged RLOC probing algorithm. Cisco's algorithm generates RLOC probe for each EID-to-RLOC mapping, which means forty/eighty *LISP Map-Request Probe* and forty/eighty *LISP Map-Reply Probe* messages per single phases #1 and #2 occurrences. Opposite to that any merged RLOC algorithm exchanges only single *LISP Map-Request/Reply Probe* pair between xTR_Responders.

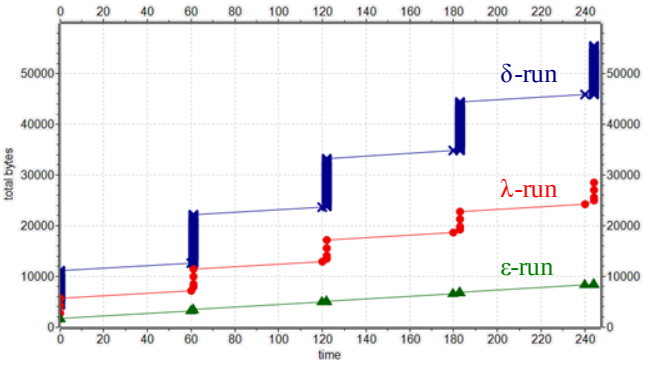

Figure 20. xTR1's LISP control messages occurrence and total processed byte size in scenario with two outages

In Figure 20, we can see that *simple* algorithm (ε-run = green triangles) has the lowest protocol overhead measured in the total amount of bytes processed by *xTR_Responder1*. This is because each probe carries only single EID chosen in a round-robin fashion, where successful reception of *LISP Map-Reply Probe* refreshes RLOC state for all EIDs that are using it. In case of *sophisticated* algorithm (λ-run = red circles), all relevant EIDs are packed in a single probe, thus, (significantly) increasing its size but still half of Cisco's (δ-run = blue crosses) total processed byte size. On the other hand, *simple* merged RLOC probing algorithm might seem to be too simple and lacking of accuracy if we want the use-case where the same RLOC is up for some EIDs, and down for another EIDs. In that case, *sophisticated* variant offers the same functionality but with better granularity. Because scenarios are linearly dependent, the only difference between forty and eighty EIDs scenario graphs is in Y-axis values and a higher amount of RLOC probe (symbol) occurrences.

V. CONCLUSION

In this paper, we presented a detailed description of LISP and VRRP technologies. We proposed and tested two LISP improvements – map-cache synchronization and merged RLOC probing – aimed to achieve a better routing performance (primarily in high-availability use-cases).

We evaluated proposed improvements using newly implemented models in OMNeT++ simulator tool. Validation of these models against a real-life topology shows the acceptable precision in terms of time accuracy. However, simulation results are affected by a simpler simulated control-plane (without any potential processing delay of the real router). Hence, some simulation timestamps in Table III and IV are below one millisecond difference.

Previously, LISP map-cache performance have been evaluated employing high-level simulation that is not taking into account protocol implementation specifics [14]. Hence, one of the goals of our work was to provide the community

with a simulation tool with a near-real implementation behavior. Results from simulations show that deployment of map-cache synchronization techniques has a positive impact on data packet-loss and a total number of map-cache misses. Moreover, synchronization decreases LISP signalization overhead (i.e., no need to query mapping system for forgotten map-cache entries). *Smart* mode map-cache synchronization yields the best results.

We investigated Cisco's RLOC probing algorithm used to verify locator reachability. Cisco's algorithm has disputable scalability mostly because of periodical polling. We developed two new RLOC probing variants that aim to be more efficient (even though they still use polling). Simulation tests show that both of them significantly reduces the number of exchanged *LISP Map-Reply/Request Probes*, thus reducing LISP protocol overhead. *Sophisticated* merged RLOC probing algorithm provides the same reliability as Cisco's version, but it utilizes only the half of Cisco's bandwidth in bytes processed by the control plane.

Among our plans with further investigation of LISP is to add support for proxy xTR functionality and recognize more LISP control flags (like SMR bits). We would like to use further our LISP simulation modules and test effectiveness of different distributed mapping systems (e.g., LISP-ALT, LISP-DDT). Also, 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 [15]. Real packet captures and simulation datasets for the results reproduction could be investigated from Wiki of the repository mentioned above. More information about ANSA project is available on its homepage [16].

ACKNOWLEDGMENT

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.

BIBLIOGRAPHY

[1] V. Veselý and O. Ryšavý, "Locator/Id Split Protocol Improvement for High-Availability Environment," in ICNS 2015 - Proceedings of 11th Conference on Networking and Services, pp. 61-67, Rome, Italy, 2015.

[2] V. Veselý, M. Marek, O. Ryšavý and M. Švéda, "Multicast, TRILL and LISP Extensions for INET," International Journal On Advances in Networks and Services, vol. 7, no. 3&4, pp. 240-251, 2014.

[3] 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. [Accessed 6th December 2015].

[4] T. Li, "RFC 6227: Design Goals for Scalable Internet Routing," May 2011. [Online]. Available: http://tools.ietf.org/html/rfc6227. [Accessed 6th December 2015].

[5] R. Hinden, "RFC 1955: New Scheme for Internet Routing and Addressing (ENCAPS) for IPNG," June 1996. [Online]. Available: http://tools.ietf.org/html/1955. [Accessed 6th December 2015].

[6] IETF, "Locator/ID Separation Protocol (lisp)," 15th September 2015. [Online]. Available: http://datatracker.ietf.org/wg/lisp/charter/. [Accessed 15th September 2015].

[7] 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. [Accessed 6th December 2015].

[8] V. Fuller and D. Farinacci, "RFC 6833 - Locator/ID Separation Protocol (LISP) Map-Server Interface," January 2013. [Online]. Available: https://tools.ietf.org/html/rfc6833. [Accessed 6th December 2015].

[9] R. Hinden, "RFC 3768: Virtual Router Redundancy Protocol (VRRP)," April 2004. [Online]. Available: https://tools.ietf.org/html/rfc3768. [Accessed 6th December 2015].

[10] S. Nadas, "RFC 5798: Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6," April 2010. [Online]. Available: https://tools.ietf.org/html/rfc5798. [Accessed 6th December 2015].

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

[12] 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. [Accessed 6th December 2015].

[13] 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, vol. 7289, no. ISBN: 978-3-642-30044-8, pp. 397-408, May 2012.

[14] J. Kim, L. Iannone and A. Feldmann, "A deep dive into the LISP cache and what ISPs should know about it," NETWORKING 2011, vol. 6640, no. ISBN: 978-3-642-20756-3, pp. 367-378, 2011.

[15] GitHub, "kvetak/ANSA," December 2013. [Online]. Available: https://github.com/kvetak/ANSA/wiki. [Accessed 6th December 2015].

[16] Brno University of Technology, "ANSAWiki | Main / HomePage," January 2014. [Online]. Available: http://nes.fit.vutbr.cz/ansa/pmwiki.php. [Accessed 6th December 2015].