Implementation and Evaluation of LISP Publish/Subscribe

Tom Piron
Montefiore Institute
Université de Liège
Liège, Belgium
tompiron@hotmail.be

Luigi Iannone
Datacom Department
Huawei Technologies
Paris, France
luigi.iannone@huawei.com

Benoit Donnet
Montefiore Institute
Université de Liège
Liège, Belgium
benoit.donnet@uliege.be

***Abstract*—Future 6G network will push even further the convergence of different types of mobile networks, integrating space, aerial and terrestrial access. Mobility, remains one of the most difficult aspects to tackle in this context. One approach under consideration is the use of an overlay solution able to cope with new mobility requirements. LISP (Locator/ID Separation Protocol) being one candidate overlay protocol. LISP separates the addressing space in two orthogonal spaces, one to identify end points, the other to locate them. End-to-end-Communication is guaranteed by a mapping system allowing to associate location with identities. Mapping resolution is done at communication setup, opening the question: how to guarantee that, in case of changes, the latest mapping is used? Originally, there was no mechanism to explicitly express the interest in updates of specific mappings. LISP Publish-Subscribe has been introduced in order to provide such a feature. This paper provides an implementation of LISP Publish-Subscribe in NS-3 and quantitatively analyze its benefits.**

***Index Terms*—LISP, Mapping System, NS-3, Publish, Subscribe, Mobility, 6G.**

I. INTRODUCTION

The Locator/ID Separation Protocol [1] – LISP– has been proposed a decade ago to tackle what was felt as the Internet scalability problem ([2], [3]). Despite its technical merit, LISP has not been deployed in the open Internet, however, it is still under continuous development for use cases related to virtualization and mobility [4]. In particular the mobility management aspects of LISP make the protocol suitable to be considered as part of the future 6G architecture ([5], [6], [7]).

LISP separates the addressing space in two separate spaces: (i) end-point identifiers addressing space, used in stub networks, like for instance a 6G access network; (ii) routing locator addressing space, used in the interconnecting infrastructure, i.e., the 6G backhaul and the open Internet. The latter is in charge to correctly deliver IP packets to the right place. In order to deliver packet from one end of the interconnecting infrastructure to the other end without having to advertise the whole end-point identifier addressing space, a tunneling approach is used, so the interconnecting infrastructure just forwards packets using routing locators as source-destination addresses. However, this approach raises the need to map end-point identifiers to routing locators to allow the ingress router to tunnel the packet to the correct egress router. Mapping distribution is done on demand, meaning that, when needed, the ingress router will perform a query to the LISP Mapping System in order to retrieve the required mapping.

The *on-demand* approach used by LISP to handle mappings allows to achieve great scalability, since tunnel routers do not need to store the entire database of all existing mappings. However, this comes at the price of having to find a mechanism to explicitly signal any change in mappings (due to mobility), so to avoid traffic disruptions. The original mechanisms that LISP included to deal with this issue, namely mapping lifetime [1], mapping versioning [8], and Solicit Map Request (SMR) [9], blindly rely on the data plane, without an explicit way to express interest in mapping updates. This shortcoming is being fixed by the Publish/Subscribe functionality for LISP [10], actively discussed in the LISP WG. The main idea being to let tunnel routers express interest in mapping updates, and receive such updates through the LISP control plane whenever they are published.

Intuitively, the Publish/Subscribe technique allows one to improve responsiveness and update mappings in remote tunnel routers more quickly. However, to the best of our knowledge, no work evaluating the benefits of such a mechanism has been published sofar. As such, the work present in this paper is the first attempt to quantify the improvements introduced by Publish/Subscribe. We implemented the mechainsm on NS-3 [11], including a realistic time model, in the aim of evaluating the gain in a relatively simple mobility scenario, where one of the communicating host changes its routing locator. Our simulations clearly show that, indeed, LISP Publish-Subscribe does provide a better handover delay, meaning that the time between the mobility event and the time that the updated mapping is communicated to the ingress tunnel router is shortened compared to other potential solutions. Simulations also show that performance remains quite good even when increasing the number of communicating hosts at the cost of the delay distribution becoming a bit more heavy tailed.

The reminder of the paper is organized as follows. Sec. II provides the necessary background on LISP, which can be safely skipped if familiar with this technology. Sec. III gives the details of the Publish/Subscribe mechanism, whose evalution is described in Sec. IV, before concluding the paper in Sec. V.

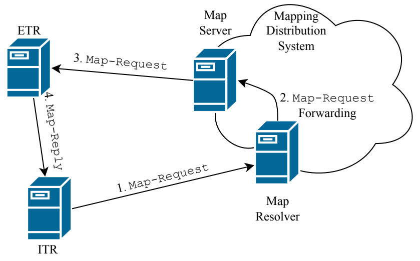

Fig. 1. LISP Control Plane message sequence for mapping retrieval.

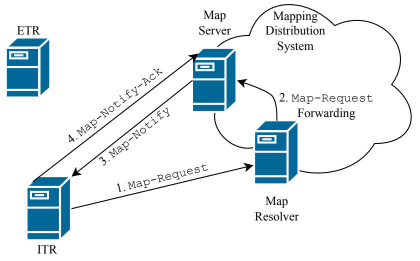

Fig. 2. LISP Control Plane message sequence for mapping subscription.

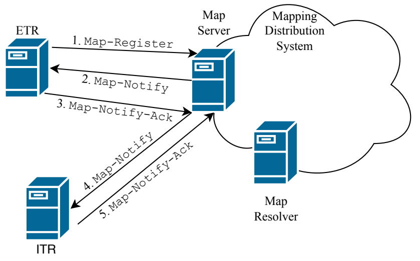

Fig. 3. LISP Control Plane message sequence for mapping publication.

II. LISP BACKGROUND

In order to provide a LISP overview, we first show how the LISP data plane (i.e., packet forwarding) works (Sec. II-1), before describing the LISP control plane (Sec. II-2).

1) LISP Data Plane: The *Locator/IDentifier Separation Protocol* (LISP) [1] separates the identification and localization roles of IP addresses by introducing two logical addressing spaces: (i) the *Routing LOCator space* (RLOC), which is globally routable; (ii) the *Endpoint IDentifier space* (EID), which is only locally routable and whose main purpose is to identify the communication endpoints. With this separation, the Internet core, also known as *Default Free Zone* (DFZ), handles RLOCs addresses like it is done today, i.e., maintaining routes so that packets can be forwarded between any router within the DFZ. Stub networks instead use the EID addressing space. The implication of such a separation lies in stub networks not needing anymore a full knowledge of the Internet routing information, whereby the DFZ does not need anymore to advertise the EID space in its routing infrastructure. Nonetheless, in order to provide end-to-end communication, another level of indirection is required.

The LISP data plane provides this level of indirection through a tunneling mechanism over the DFZ. More specifically, any communicating host generates regular IP packets using its EID as the source address and the destination EID as the destination address. Forwarding towards the border router is done as usual in the local domain. The border router, now called *Ingress Tunnel Router* (ITR), will encapsulate the packets using the RLOC addressing space, i.e., using its RLOC address as the source address and the destination RLOC as the destination address in the tunnel header encapsulating the original packet [1]. The encapsulated packets can now be forwarded over the DFZ. The border router at the destination site, now called *Egress Tunnel Router* (ETR), will decapsulate the LISP packets so that the original packet can be forwarded to its final EID destination.

2) LISP Control Plane: In order to perform the data plane operations, tunnel routers need to be able to associate EIDs to RLOCs. The binding between the two addressing spaces is named *mapping*. A mapping enables a tunnel router (generally referred to as an XTR) to retrieve the RLOCs associated to a given EID, to be used in the outer header when encapsulating. Mappings are stored in two data structures present on XTRs: (i) the LISP *Database* storing the mappings for EID prefixes for which the XTR itself is an RLOC; (ii) the LISP *Cache* storing mappings for EID prefixes used in ongoing communication towards/from distant LISP sites.

The LISP control plane introduces a *Mapping Distribution System (*MDS*)* providing a lookup infrastructure from where mappings can be retrieved upon an explicit query. The initial design of the MDS was based on an *on-demand* approach. From an abstract point of view, the MDS works as follows. The ITR that needs a mapping for a new flow first sends a query, consisting of a `Map-Request` message to a *Map-Resolver* (MR) [9]. The query is forwarded by the Map-Resolver inside the MDS according to the specific protocol/architecture used, to reach the *Map-Server* (MS) where the site using the requested EID has registered the mapping. The Map-Server then forwards the query to the XTR that registered such a mapping. In turn, the XTR will send the reply, consisting of a `Map-Reply` message containing the requested mapping, directly to the ITR that, in the first place, sent the query. The prcedure is depicted in Fig. 1. The Map-Resolver and Map-Server elements represent respectively where to ask for a mapping and where to register a mapping so as to make it available to other LISP sites. They provide a general front-end for any mapping system, *"hiding"* the specific MDS in use to the LISP tunnel routers.

III. LISP PUBLISH-SUBSCRIBE

LISP Publish-Subscribe [10] allows the mapping system for notifying ITRs for changes, enabling faster mobility events (with a minimal amount of packet drop), and for keeping connections (such as TCP) through events.

Two procedures are proposed with LISP Publish-Subscribe: (i) a subscription procedure allowing an ITR to express interest in updates for changes of a specific EID prefix and, (ii), a publish procedure allowing the Map-Server to notify mappings' updates to ITRs that made a subscription.

The subscription procedure, illustrated in Fig. 2, requires the ITR to send a `Map-Request` to retrieve the EID prefix to subscribe to. To perform the subscription, the `Map-Request` message is slightly modified compared to the original format. In particular, the message now includes a *Site ID* and XTR *ID* fields in order to keep track of which site and which

XTR is requesting subscription. The `Map-Request` is then forwarded through the mapping-system to the corresponding MS. Then, the MS creates a subscription state to remember the request. The state stores the nonce received in the `Map-Request` to limit the risk of replay attack (possibly also denial-of-service attacks [12]). The MS responds with a `Map-Notify` to confirm the subscription and respond to the `Map-Request`. Finally, the ITR responds to the `Map-Notify` with a `Map-Notify-Ack`. If, for any reason the subscription fails, the MS simply responds with a traditional `Map-Reply`.

The publish procedure, shown in Fig. 3, starts with an update to the mappings in the MS, e.g., a time-to-live expiration or an entry withdrawal in the MS database. The MS then fetches the subscription state associated to the updated prefix, increase all nonces by one, and send a `Map-Notify` to all subscribers. At reception, each ITR checks if the nonce of the `Map-Notify` is greater than the previous one and, if so, updates its cache and sends a `Map-Notify-Ack`. If not, the `Map-Notify` is simply dropped.

IV. EVALUATION

In order to perform an evalution through simulations, we decided to use ns-3 [11], as prior work had been done with LISP on this simulator and the simulator itself is widely adopted in the research community [13]. We extended the existing implementation so that XTRs, Map-Servers, and Map-Resolvers support LISP Publish-Subscribe.

Our evaluation focused on the effectiveness of the different procedures to notify an ITR of a mapping change, which can be achieve in four different manners:

Nothing: Doing nothing means waiting the mapping to expire in the ITR cache. One option to make LISP more reactive in this scenario is to use very short TTL for the mappings.[1] In our simulation we shrinked the TTL down to 1 minute.

SMR: Using the SMR (Solicit Map Request) procedure, making the ETR send a `Solicited Map-Request` message to the ITR in order to trigger a `Map-Request` from the ITR, which will go through the Mapping Distribution System.

SMR+Proxy: Using the SMR procedure and the Map-Servers proxy service. This is similar to the previous solution but relying on the MS, rather than the ETR, to send back the updated mapping.

PubSub: Using LISP Publish-Subscribe (see Sec. III).

We run 1,000 simulations of a simple mobility scenario. On one side a sending host placed behind an ITR. On the other side a receiving host behind an ETR, moving behind a different ETR, hence requiring to update the mapping since it basically changed its RLOC. In order to maintain the communication the ITR needs to be updated with the new mapping. The mobility event happens at a random time in the simulation, following an uniform distribution. We measured the *Handover Delay*, namely the time between the moment the mapping is updated on the ETR and the instant the ITR receives such new mapping. It is important to mention that, in order to have realistic results, the transmission delay between XTRs has been modeled using the measurements results from Saucez et al. [14]. For the same reason, the mapping system resolution delay has been modeled upon the DNS system resolution delay. This is done because data are widely available and the LISP mapping system in its LISP–DDT [15] proposal is very similar to DNS. In particular, the dataset from OpenINTEL Project [16] has been used.

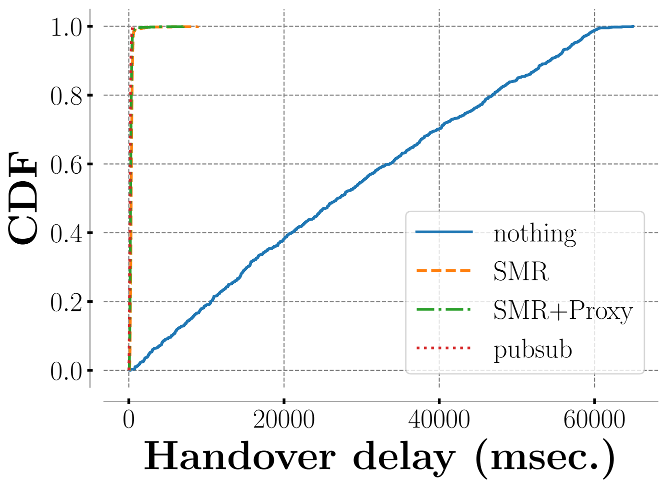

Fig. 4. Handover delay for all evaluated mechanisms for the single sender single receiver scenario.

The overall handover delay for the four mechnisms we evaluate is presented in Fig.4. We can readily see difference between the mechanism that uses a notification the the one without. The latter has a uniform distribution of delay between 0 and 60 seconds. This is an artefact of the 1-minute TTL used in our simulations. Indeed, since the mobility event is triggered following an uniform disribution, it means that it is uniformly distributed with the mappings TTL, hence its linear shape.

Fig.5 shows a zoom that allows one to appreciate the performance of the other solutions, including LISP Publish-Subscribe. As expected LISP Publish-Subscribe is performing the best, since the notifications sequence is triggered directly upon the mapping update, with the MS directly contacting the ITR. This is not the case for the SMR-based mechanism. Indeed, in this case the ETR will send the SMR directly to the ITR, but then, the corresponding `Map-Request` will go through the MDS, hence experiencing longer delay. The SMR+Proxy scenario performs better than the pure SMR scenario since, in this case, the MS is responsible to send back the updated mapping with a `Map-Reply` message, thus shortening the procedure.

The previous scenario, with one sender and one receiver helps in having a good idea of ideal (best) case and the performance difference of the four mechanism. In order to

---

[1] Every mapping is delivered with a *Time-to-live*, expressing for how long the mapping can be used. Once this time is expired, the mapping must be deleted from the local cache and a new request has to be sent if the ITR wants to continue to use the mapping.

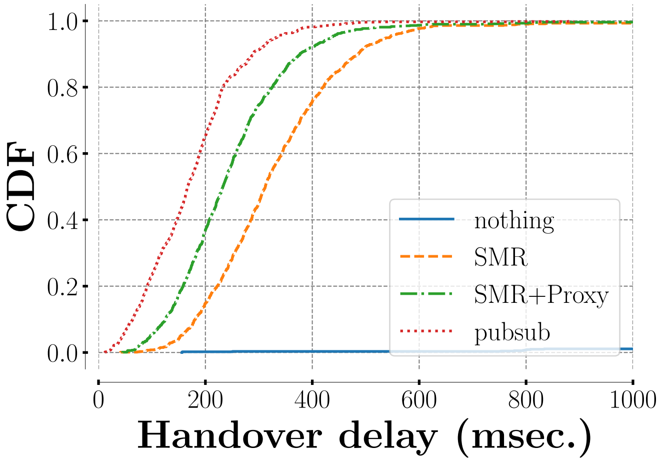

Fig. 5. Zoom on the mechanisms using an explicit notification for the single sender single receiver scenario.

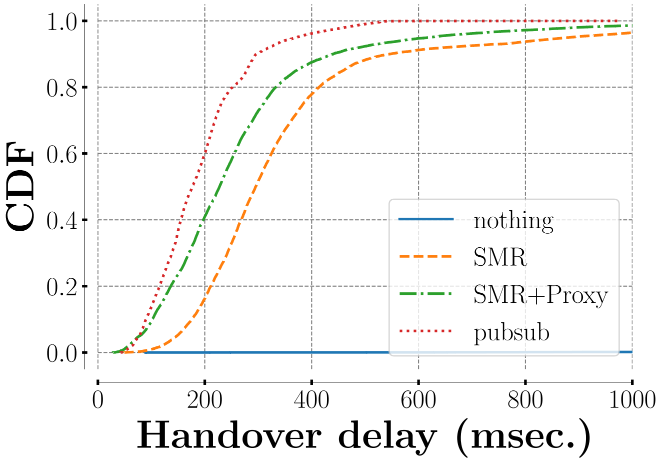

Fig. 6. Handover delay for all evaluated mechanisms for the multiple senders single receiver scenario.

move forward, we started to have a look at the mechanisms scalability, meaning exploring scenarios where more than one ITR need to be updated. In particular, Fig.6 shows the handover delay measurements in the case of 1,000 sending hosts. In this scenario, the Handover delay is the time between the moment the mapping is updated and the moment when all of the ITRS have received the updated mapping. We can observe that, on average, the difference is not huge. Focusing on LISP Publish-Subscribe, 60% of the handover delay is smaller than 200 msec, down from 65% in the single sender-single receiver case. However, the distribution becomes now more heavy-tailed. In particular, SMR and SMR+Proxy now experience less than 600 msec handover delay around 90% of the cases, down from almost 100%, going beyond one second in some cases. This higher variability is clearly due to the higher number of ITR that needs to be updated generating an higher volume of signaling traffic.

V. CONCLUSION

The LISP protocol, by separating the the addressing space in EIDS and RLOCS, requires a Mapping Distribution System in order to map the former in the latter. A mapping associating an EID to its RLOC is retrieved from such a system at the beginning of the communication on an on-demand fashion. Existing mechanism to notify changes in the mapping when communication are ongoing does not allow to express explicitly an interest in changes in specific mappings. LISP Publish-Subscribe fills this gap by introducing such explicit signaling.

This paper is a first attempt to quantitatively assess the benefits that the publish subscribe model bring to the over all LISP architecture. AS expected, our early simulation-based results show that indeed LISP Publish-Subscribe does provide an advantage by shorting the handover delay in case of mappings changes. Future work on this subject includes a more thorough analysis in more complex scenario, as well as exploring its scalability properties.

REFERENCES

[1] D. Farinacci, V. Fuller, D. Meyer, D. Lewis, and A. Cabellos-Aparicio, "The locator/ID separation protocol (LISP)," Internet Engineering Task Force, RFC 9300, October 2022.

[2] B. Quoitin, L. Iannone, C. De Launois, and O. Bonaventure, "Evaluating the benefits of the locator/identifier separation," in *Proc. 2nd ACM/IEEE International Workshop on Mobility in the Evolving Internet Architecture*, August 2007.

[3] T. Li, "Recommendation for a routing architecture," Internet Engineering Task Force, RFC 6115, February 2011.

[4] "Locator/ID separation protocol (LISP) working group," https://datatracker.ietf.org/wg/lisp/about/, last Accessed: 2023-01-09.

[5] P. Xie, Q. Wang, and J. Chen, "A survey of mobility management for mobile networks supporting leo satellite access," in *2022 IEEE/CIC International Conference on Communications in China (ICCC Workshops)*. IEEE, 2022, pp. 59–64.

[6] L. Cai, J. Pan, W. Yang, X. Ren, and X. Shen, "Self-evolving and transformative (set) protocol architecture for 6g," *IEEE Wireless Communications*, 2022.

[7] G. Hong, B. Yang, W. Su, Q. Wen, X. Hou, and H. Li, "Decentralized vehicular mobility management study for 5g identifier/locator split networks," *Wireless Communications and Mobile Computing*, 2022.

[8] L. Iannone, D. Saucez, and O. Bonaventure, "Locator/ID separation protocol (LISP) map-versioning," Internet Engineering Task Force, RFC 9302, October 2022.

[9] D. Farinacci, F. Maino, V. Fuller, and A. Cabellos-Aparicio, "Locator/ID separation protocol (LISP) control plane," Internet Engineering Task Force, RFC 9301, October 2022.

[10] A. Rodriguez-Natal, V. Ermagan, A. Cabellos-Aparicio, S. Barkai, and M. Boucadair, "Publish/Subscribe Functionality for the Locator/ID Separation Protocol (LISP)," Internet Engineering Task Force, Internet-Draft draft-ietf-lisp-pubsub-15, Feb. 2023, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-lisp-pubsub/15/

[11] T. R. Henderson, M. Lacage, G. F. Riley, C. Dowell, and J. Kopena, "Network simulations with the NS-3 simulator," August 2008.

[12] M. Gabriel, L. Iannone, and B. Donnet, "LISP mapping system as dos amplification vector," *IEEE Networking Letters*, vol. 3, no. 1, 2021.

[13] E. Marechal, "Simulating LISP with NS-3," Master's thesis, Université de Liège, Belgium, 2019.

[14] D. Saucez and B. Donnet, "On the dynamics of locators in LISP," in *Proc. Networking*, May 2012.

[15] V. Fuller, D. Lewis, V. Ermagan, A. Jain, and A. Smirnov, "Locator/ID separation protocol delegated database tree (LISP–DDT)," Internet Engineering Task Force, RFC 8111, May 2017.

[16] R. van Rijswijk-Deij, M. Jonker, A. Sperotto, and A. Pras, "A high-performance, scalable infrastructure for large-scale active DNS measurements," *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 34, no. 6, pp. 1877–1888, June 2016.