



EPiC Series in Computing

Volume 56, 2018, Pages 43–51

Proceedings of the 5th International
OMNeT++ Community Summit



Simulating LISP-Based Multilink Communications in Aeronautical Networks

Chiara Caiazza, Antonio Viridis, Giovanni Stea,
Dipartimento di Ingegneria dell'Informazione, University of Pisa
Largo Lucio Lazzarino 1, I-56122, Pisa, Italy
c.caiazza@studenti.unipi.it
{antonio.viridis, giovanni.stea}@unipi.it

Abstract

Over the next few years, future aeronautical communications are expected to move from the OSI protocol stack to the TCP/IP one. Multiple data-link channels (e.g., satellite, terrestrial) will be available to each aircraft to communicate with ground, realizing the so-called multilink. In order to ensure a seamless handover between different data-links, the LISP protocol has been defined. This paper describes the modelling, design and implementation of LISP on the SAPIENT simulator, an OMNeT++-based simulator for aeronautical communications. It includes models of the aircrafts, including their mobility, multiple data links and ground-network elements.

1 Introduction

AIR-Traffic-Management (ATM) operations include heterogeneous services, such as Communication, Navigation and Surveillance (CNS), Air Traffic Services (ATS), Airline Operational Communications (AOC), etc., having different data rates and performance requirements for the communication among aircrafts (ACs) and ground nodes. Moreover, the above services involve different stakeholders, e.g. airport authorities, Air Navigation Service Providers (ANSP), air-traffic-flow managers, etc..

In this context, the Single European Sky Air Traffic Management Research (SESAR) program in Europe, and the NextGen project in the US, have focused on facilitating the implementation of the above concepts for the next generation of aviation communication systems. One of SESAR's objectives, in particular, is to define a novel communication architecture to improve performance, security and safety in ATM. Such system is called Future Communication Infrastructure (FCI) and is designed to support ATS and AOC communications in an end-to-end fashion, including both air-to-ground and air-to-air segments. FCI services will have to meet tight operational requirements, in terms of safety, communication capacity, flight and cost efficiency [1].

ATM networks include several data-links (DL). Satellite DLs are needed to ensure coverage over ocean space, and several terrestrial DLs have been standardized for land connectivity, such as VDL [2], which is currently operational, or LDACS, [3], which is expected to be deployed in the future. Each DL has its own domain network, and the FCI will be responsible for connecting them all together, thus realizing a system of systems.

In such a context, Multilink (ML) refers to the ability to use different DLs at different times for communication, possibly in a seamless way. The problem lies in the fact that an AC is a whole network, with its own address space, and its hosts must be reachable through different DLs. In order to achieve this, proper *routing and forwarding mechanisms* must be in place within the whole FCI network that connects the ground stations and the AC, so that packets destined to the same AC host (hence having the same IP address) are routed to different DL domains at different times, ML also requires *decision policies*: some entity in the network should be able to decide when to change DL (a decision sometimes called *vertical handover*, VH), and to have the network routing subsystem react accordingly to enforce such decisions in a coherent fashion. *Scalability* is a major requirement for a ML system, since the number of ACs involved may range in the thousands. Moreover, the speed of routing convergence is paramount, since connectivity should be preserved to enhance safety-of-life as well as to enable remotely-piloted ACs.

A possible solution to implement ML routing is the Locator/Identifier Separation Protocol (LISP), standardized by the IETF [4]. With LISP, an AC keeps its own fixed IP address space (such addresses are called *Endpoint Identifiers*, EIDs), and EIDs are associated to temporary, varying *Routing Locators* (RLOCs) within the network. The mapping between EIDs and RLOCs is stored at a central server, and routing and forwarding in the core domain is based on RLOCs, which act as external tunnel identifiers. This way, a VH decision translates to a modification of the EID to RLOC association in the server.

In this paper, we describe the modeling, implementation and evaluation of the LISP protocol within the SAPIENT simulator. The SAPIENT simulator has been developed within the framework of the SAPIENT EU project [2], framed into the EU SESAR (Single European Sky ATM Research) funding framework. SAPIENT aims at using ACs as crowdsensing entities that report geo- and time-referenced measures of the state and quality of their DL to a ground server (the SAPIENT Solution Owner). The latter can then build a map of the state of the DL over the European sky, and use that knowledge for different purposes, among which suggesting ACs which DL to use when, based on global considerations such as overall load or impairments ahead along their trajectory. The SAPIENT simulator is based on OMNeT++ [5], and leverages INET models [6], and it includes models of the ACs, the DLs, and the core network interconnecting them. We model the LISP protocol entities and nodes in the simulator, and the interactions between them. To the best of our knowledge, there is a substantial lack of system-level performance analysis for LISP in the literature, hence this paper provides an initial performance evaluation of LISP capabilities, taking into account multiple network domains. Notably, we show that – depending on network conditions and mode of usage – overhead of LISP on the network and on the management nodes can vary significantly.

The rest of the paper is organized as follows: Section 2 reports background on LISP. Section 3 describes the SAPIENT simulator, Section 4 describes the modeling and implementation of LISP within the latter. In Section 5 we report performance evaluation results, and Section 6 concludes the paper.

2 Background

In this section, we provide a brief background on the LISP protocol, focusing on the aspects that will be modeled in the following sections. The LISP protocol, which runs on UDP, provides a set of

mechanisms to allow the use of two namespaces instead of a single IP address, decoupling the information used to identify a node from the knowledge of its location. To achieve this, LISP distinguishes between “core-routable” and “core-unroutable” address spaces, as follows:

- EID (Endpoint IDentifier) space: an EID is a 32-bit or 128-bit value assigned regardless of the network topology. This does not depend on the mobility of the host’s network. An EID address is syntactically equivalent to an IP address but it is not routable through the core network.
- RLOC (Routing LOCator) space: an RLOC is an IPv4 or IPv6 address of a LISP router’s interface. RLOCs are routable in the core.

Each LISP node manages both address spaces and performs translation from one to the other. Figure 1 provides a graphical representation of the two namespaces.

Two types of nodes are defined: first, an *Ingress Tunnel Router* (ITR) encapsulates packets sent by hosts outside the core network. The ITR uses the destination EID address to perform an EID-to-RLOC lookup using a local mapping. If that fails, it sends a Map-Request to the mapping server. Second, an *Egress Tunnel Router* (ETR) accepts packets whose destination address is one of its own RLOCs. It receives LISP packets from the network and decapsulates them to be forwarded to the network beyond it. The acronym *xTR* can be used to refer to both nodes when common functions are addressed. The linchpin of the LISP system is the LISP Mapping-Server/Mapping-Resolver (LISP Ms/Mr): the latter is located in the Core network, and it must be reachable using only RLOC address. It learns the EID-to-RLOC mapping from the ETRs, and stores them in a *global mapping table*. It accepts and responds queries from ITRs whenever a mapping for an EID is required [7].

Consider Figure 1 as reference. Node ES1 starts to send messages to MN1. As soon as the LISP router xTR41 receives the first packet, it uses the EID destination address to perform a lookup in its local-mapping cache. If a match is found, the xTR41 encapsulates the packet using the RLOC 7.7.1.1 as destination address. When the encapsulated packets reaches xTR11, it is decapsulated and sent through the LDACS network. When a miss in the xTR41’s local mapping occurs, it sends a Map-Request to the LISP Ms/Mr. The latter then check in the global mapping for an ETR that can answer the request. If no correspondence is found, the LISP Ms/Mr sends to xTR41 a Negative Map-Reply, i.e. a Map-Reply without RLOCs. Otherwise, the LISP Ms/Mr chooses an ETR among those that have registered at least one RLOC for a prefix contains the destination EID.

All xTR periodically sends Map-Register messages containing the list of EID prefixes to be recorded in the global mapping database and the set of RLOCs that can be used by the LISP Ms/Mr to reach them. When the LISP Ms/Mr receives a Map-Register message it stores the information contained within the global mapping, eventually updating existing entries.

The LISP protocol defines two modes to discover EID-RLOC mapping, namely *direct* and *indirect* modes. With reference to Figure 2, in the *direct mode* the Ms/Mr sends a Map-Reply on behalf of the ETR. Otherwise, (*indirect mode*) the LISP Ms/Mr forwards the Map-Request to the selected best ETR.

When the ITR receives a Map-Reply, it stores the received RLOCs into its local mapping. The LISP Map-Request messages have to be rate-limited. The ITR should not send request more than once per second [8]. The ETRs are responsible for gathering announcements for EID prefixes announced by the Airborne Routers (A-R) and assign them a LISP Priority. Although the A-R is able to control the path used for Air-to-Ground traffic, the reverse route is driven by the LISP priority values announced by the ETRs. These values are governed by the preference parameters contained in the radio advertisement sent by the A-R. This means that the responsibilities for enforcing traffic symmetry rely on the A-R.

With mobility, an aircraft may want to transfer part of its traffic from a link to a different one. When LISP priorities change the best ETR could change as well, this means that some entries in the

ITRs' local mapping must be removed. The old ETR sends a Solicit-Map-Request message to its ITRs to ask them to send a new query for that EID prefix to the mapping system.

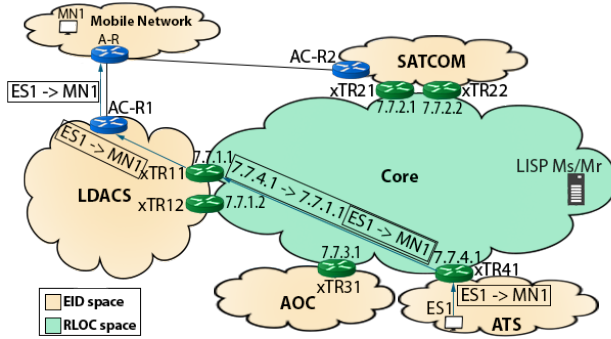


Figure 1: High-level view of the EID and RLOC address spaces and example of communication.

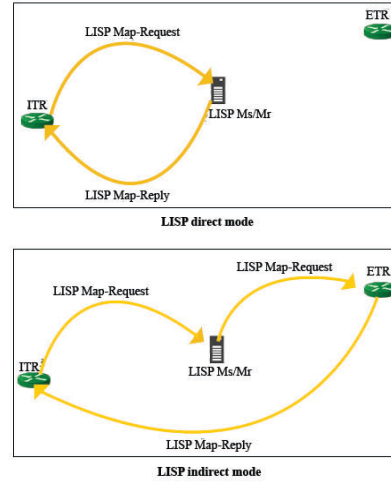


Figure 2: Message exchanges using the direct and indirect modes

3 SAPIENT Simulator

This section describes the SAPIENT simulator architecture and main functions. The SAPIENT simulator models a communication network that includes several components of the aeronautical network: the ACs themselves, the DL infrastructure, and a core network. In Figure 3 we represent the main actors involved in the communication network and the interaction among them. The architecture can be divided into several domains, each one further composed of a number of nodes. In the following, we provide a brief description of each domain. We refer the interested reader to [9] for further details. The *Application domain* is composed of all the application nodes communicating with ACs. Each application node is implemented as an INET standard host, having a configurable number of TCP and UDP applications. The *Core Network (CN) domain* provide connectivity among the ground application domain and the DLs, and includes the communication elements of the ground network, e.g. IP routers. The model of the CN includes IP-based routers from the INET library, whose number, position and connection can be configured. The CN serves two main purposes: QoS support and routing. First, it provides QoS to traffic belonging to applications having diverse priorities. This is implemented via the Differentiated Service (DiffServ) architecture provided by INET, which relies on traffic classification and priority scheduling. Second, it implements routing mechanism to support multilink, allowing the ground nodes to reach an AC using either DLs. This can be done either using static routes or via dedicated routing protocols. In the next section, we will describe how LISP can be used for this purpose.

The *DL Domain* includes the models of two DLs, namely a terrestrial and a satellite one, that can be used to transport data to/from the application domain. The *terrestrial* DL is a model of the LDACS data link, i.e. a cellular-based communication technology for aeronautical communications. The *satellite* DL instead (SATCOM), models the ANTARES satellite DL, composed of orbiting satellite relays communication between the AC and a ground station. For both DLs, the network can be arbitrarily complex. More specifically, each DL network may have several connection points to the core network. Finally, the *AC Domain* models the ACs as mobile communication end-points. Each

AC runs ATM/AOC applications, and has an associated mobility model. The ACs can communicate with the rest of the network via either or both the two available Datalink networks, i.e. SATCOM or Terrestrial. A model of the internal structure of the AC is shown in Figure 4. Hosts on the AC communicate through an *airborne router*, whose internal structure is shown to the right of the same figure. The *link selector* enforces ML decisions in the air-to-ground communication, by selecting which link to forward packets to, whenever such a choice exists.

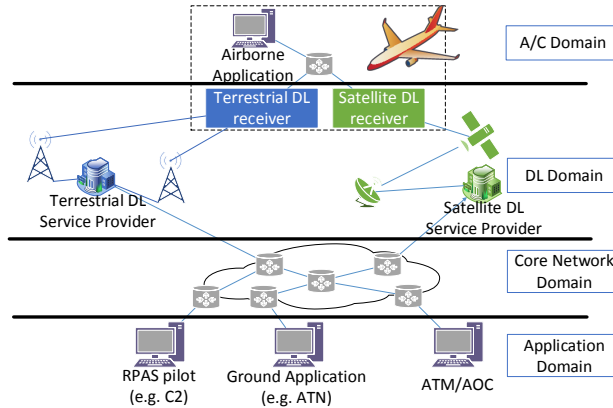


Figure 3: Domains of the SAPIENT communication network.

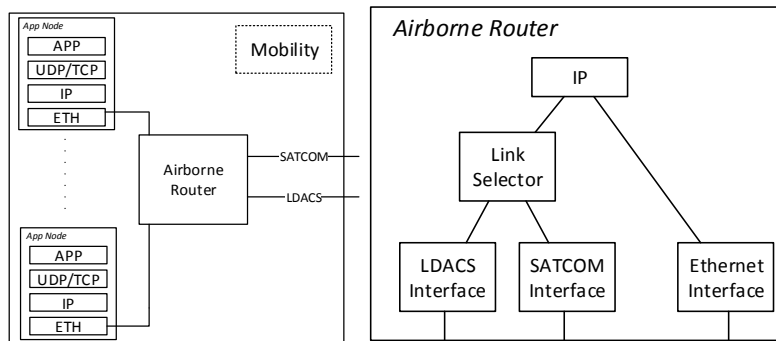


Figure 4: Internal structure of the AC (left) and of the airborne router (right).

4 Modeling and implementing the LISP protocol

LISP mainly runs in the core network, while ACs and edge networks (i.e., DL or ground networks) are largely unaffected by it. As far as LISP is concerned, the only interface between edge and core network is at the xTR. We do not simulate intradomain routing in the edge networks, since this is largely immaterial to our discussion. However, we need to model its *effects*, i.e. to endow xTRs with (un)reachability information related to ACs: whenever an AC becomes (un)reachable through a DL network, some xTR of that network must inform the LISP Ms/Mr. To achieve this, we model a dedicated per-network *route reflector* (RR), to which xTRs register at initialization. Whenever the latter gets to know that a new AC has entered the network, the RR forwards that information to all registered xTRs, which can then use it to run LISP.

Within the core network, both xTRs and the LISP Mr/Ms are implemented. With reference to Figure 5, both ground and air xTRs are implemented as compound modules containing an xTRMgr application and a LISPGroundLinkSelector module. Each xTR may act as both ETR and ITR (also

simultaneously), depending on the direction of the communications, so they contain exactly the same modules, having only slightly different internal connections.

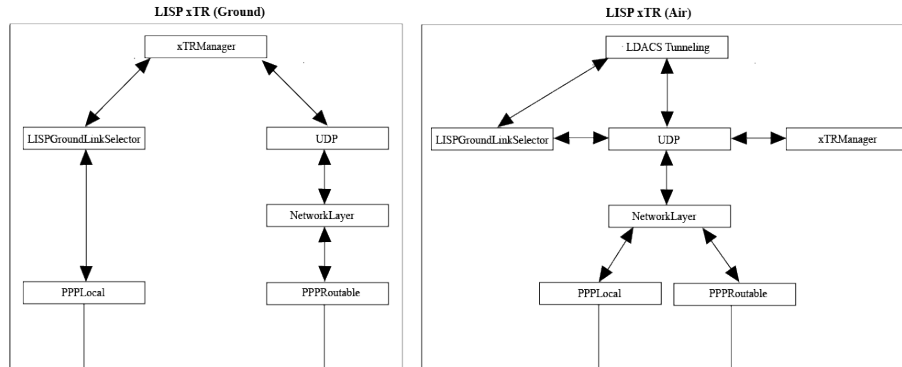


Figure 5: Internal structure of xTRs nodes on the ground (left) and air (right) sides.

The xTRManager module maintains a local table with the EID prefixes for which it is responsible. Whenever it receives from the RR a *PriorityUpdate* message, advertising the fact that an AC has changed its priority in that network (e.g., has just connected to it), it updates its table accordingly. The xTRManager sends Map-Register messages to the LISP Ms/Mr server once per minute. Each message includes the whole table stored by the module. However, when the xTRManager receives a *withdraw* from the RR, it forwards it immediately to the LISP Ms/Mr server, to allow it to clean up the global mapping as well. In the *indirect mode*, the xTRManager receives Map Requests and sends MapReply accordingly. Each time the optimal ETR changes, the xTRManager sends Solicit-Map-Request messages to all relevant ITRs.

The LISPGroundLinkSelector module parses packets coming from end-systems addressed to an EID. It searches its own local mapping for an EID-to-RLOC mapping and, if it finds one, it encapsulates the packets with the correspondent RLOC. Otherwise, it sends a Map-Request to the LISP Ms/Mr module, and stores all the mappings contained in the reply. Packets whose EID is waiting for an RLOC can be either dropped (which is the standard LISP behavior) or buffered until the relevant reply is received. It handles withdraw, solicit and push messages so that the local mapping is up to date.

The LISP Ms/Mr node is implemented as a compound module containing a *LISP manager* application. It stores the global EID-to-RLOC mapping, which is obtained from the received LISP Map-Register messages. When an existing EID-to-RLOC mapping entry doesn't receive any Map-Register updates within 180s, the LISP Ms/Mr Manager removes the entry from the global mapping.

Each time a withdraw update is received, the LISP Ms/Mr Manager immediately cleans up the global mapping. If the *direct mode* is used the LISP Ms/Mr Manager immediately sends a Solicit-Map-Request to all ITRs that have requested a mapping for the announced EID.

Whenever a mapping entry is updated a change about the best ETR may occur. If *direct mode* is used, the LISP Ms/Mr Manager sends a Solicit-Map-Request messages to all the connected ITRs allowing them to ask for a new mapping or, alternately, the LISP Ms/Mr Manager sends a Push-Message directly (direct mode with Push messages). When an ITR receives a Push message for a mapping it immediately discards the stored correspondences installing the new ones. Otherwise, if the *indirect mode* is applied, a notify message is sent to all ETRs who announces a mapping for the EID, allowing them to send the solicit messages to the ITRs. When the LISP Ms/Mr Manager receives a Map-Request from an ITR, it chooses the ETR with the minimum LISP priority (lower is better). If *indirect mode* is applied the LISP Ms/Mr Manager forwards the request to the selected node. Otherwise, in the *direct mode*, the LISP Ms/Mr Manager sends the Map-Reply by himself.

5 Performance evaluation

In this section, we evaluate the performance of the LISP protocol in terms of the overhead introduced on the system. The network deployment in use is shown in Figure 6 and includes three LISP routers, one for each network, namely ground, LDACS and SATCOM; a LISP Ms/Mr Manager; a configurable number of ACs, two DLs networks, a core-network router, and two application servers. Only Ground-to-Air traffic is taken into account, so the Ground network's LISP router is the ITR while the remaining two LISP routers are ETRs. Each AC dynamically selects the DL to be used for communication: for low-priority traffic it prefers SATCOM to LDACS, switching to the latter only if the former is not available. For high-priority traffic instead, it selects the best DL in terms of received signal power. ACs move at constant speed within a floorplan of 2400km x 2400km, following a random-waypoint mobility model. A summary of the main simulation parameters in shown in Table 1.

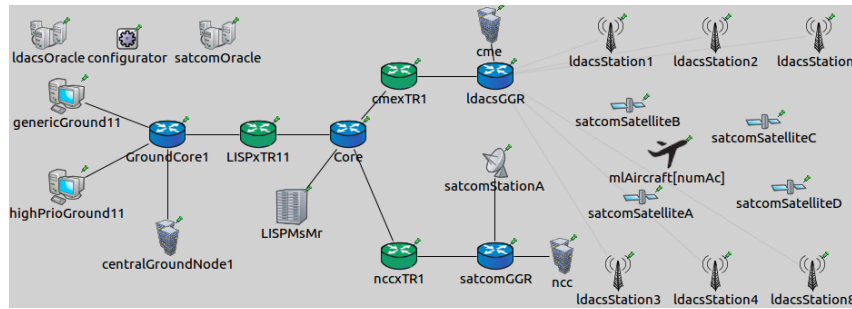


Figure 6: Exemplary deployment of a LISP-enabled scenario.

Table 1: Main Simulation parameters

Parameter	Value
Simulation duration	24 h
Warm-up durations	100 s
# repetitions	4
# AC (N_{AC})	{10 , 30 , 50}
# per-AC Applications (N_{ios})	2
Validity Time(t_v)	180 s
Register Time (t_r)	60 s

In Figure 7, we show the number of entries stored by an ITR in its local and global mapping database, respectively on the left and right of the figure. The number of stored entries allows one to evaluate the overhead of LISP operations on storage and processing of a ITR. In the first case, this number is independent from the request mode used. In the second case instead, the number of entries contained in the global mapping using the direct mode is larger than the indirect one's. In fact, when direct mode is used, the LISP Ms/Mr needs not only the information required to choose the best ETR, but also further correspondences to be included in the reply. Using push messages does not modify the performance.

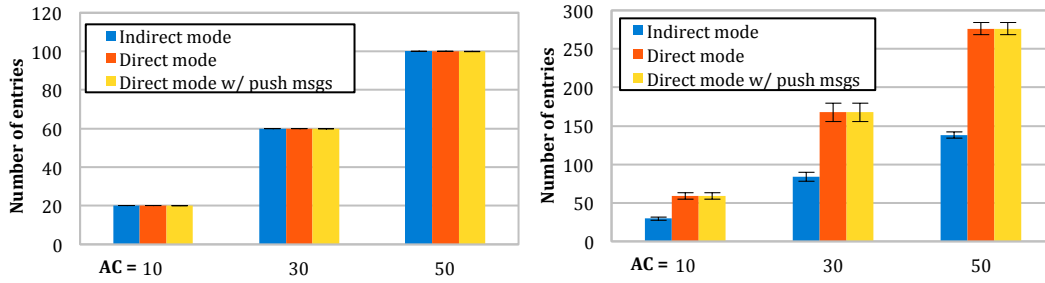


Figure 7: Number of entries stored in the local (left) and global (right) mappings database by an ITR.

In Figure 8, we compare the number of mapping requests that are received by the LISP Ms/Mr in the direct and indirect modes, and we show how it is affected by the network conditions. To this aim, we consider four different values of delay for the link between the xTRs and the core-network router, namely 0ms, 10ms, 15ms, 20ms and 40ms, representing different level of congestion. As we can see, the two modes exhibit the same behavior in case of no delay, having a number of requests that proportional to the number of ACs. As the delay increases, the indirect mode is unaffected, as the ETR always provides the ITR with a fresh mapping information. In the direct mode instead, the number of requests grows with the network delay, until 20ms, i.e. the application traffic period, and then stabilizes. In the direct mode in fact, map registers may be delayed and reach the LISP Ms/Mr *after* a mapping request is performed. The probability that this occurs increases with the network delay.

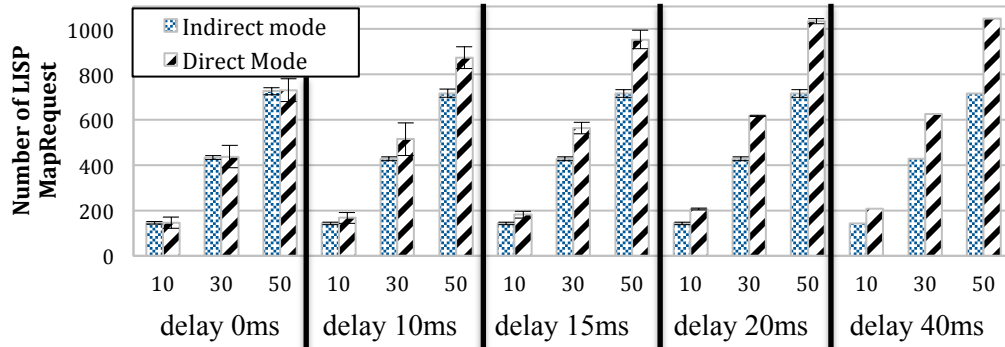


Figure 8: Comparison of the number of mapping requests when using indirect and direct modes.

6 Conclusion

In this paper, we described a model for the LISP protocol for the SAPIENT simulator. LISP permits the decoupling of the information used to identify a node from the knowledge of its location, allowing the implementation of the Multilink Concept. We first provided a background on aeronautical communication and on the LISP protocol. Then, we described the modeling and implementation of LISP nodes and function on the SAPIENT simulator. Finally, we showed performance-evaluation results over an exemplary network, comparing the performance of two LISP operational modes in terms of system overhead, and in various network conditions.

7 References

- [1] B. Haindl, M. Lindner, "Ground based lisp for multilink operation in ATN/IPS communication infrastructure", DASC, 25-29 Sept., 2016.
- [2] VDL mode 2 Capacity and Performance Analysis, PDF available at <http://tinyurl.com/y97rtd6k>.
- [3] M. Sajatovic, et al., "Updated LDACS1 System Specification", SESAR JU, Brussels, Belgium.
- [4] W. Kampichler, M. Lindner, B. Haindl, D. Eier and B. Gronau, "LISP: A novel approach towards a future communication infrastructure multilink service," DASC 2013, East Syracuse, NY, 2013.
- [5] OMNeT++ website, <https://omnetpp.org>, accessed Sept. 2017.
- [6] INET-Framework website, <https://inet.omnetpp.org>, accessed Sept. 2017
- [7] Locator/Identifier Separation Protocol (LISP). RFC 6830, IETF.
- [8] Locator/ID Separation Protocol (LISP) Map-Server Interface, RFC 6833, IETF.
- [9] A. Viridis, G. Stea, S. La Barbera, R. Winkler, "SAPIENT-Simulator Modelling and Architecture", in Proceedings of the 23rd Ka and Broadband Communications Conference, Trieste, Italy, October 4-6 2017