Supporting Future Internet Services with Extensible In-band Processing (EIP) - extended version

Stefano Salsano
Univ. of Rome Tor Vergata
Italy
stefano.salsano@uniroma2.it

Giulio Sidoretti
Univ. of Rome Tor Vergata
Italy
giulio.sidoretti@uniroma2.it

Carmine Scarpitta
Univ. of Rome Tor Vergata
Italy
carmine.scarpitta@uniroma2.it

Hesham El Backoury
Consultant
US
helbakoury@gmail.com

Diego R. Lopez
Telefónica
Spain
diego.r.lopez@telefonica.com

Lorenzo Bracciale
Univ. of Rome Tor Vergata
Italy
lorenzo.bracciale@uniroma2.it

Pierpaolo Loreti
Univ. of Rome Tor Vergata
Italy
pierpaolo.loreti@uniroma2.it

ABSTRACT

Networking architectures need to evolve to support the requirements of future Internet services and 6G networks. In this paper we propose an evolutionary solution that extends the IPv6 networking architecture fully supporting the needs of future services, called Extensible In-band Processing (EIP). The EIP solution considers a feature-rich networking layer, in which hosts and routers can cooperate by reading and writing the EIP information in the IPv6 packet headers, to support a number of use cases. Example use case are: advanced monitoring, semantic routing, deterministic networking, slicing and so on. This list of use cases is not exhaustive, as a key feature of EIP is to be extensible for the support of new use cases. In the paper we describe the initial design of the EIP header and protocol mechanisms. We have released an open source prototype implementation of EIP for Linux OS, based on the eBPF packet processing framework. The implementation includes tools for the generation of test EIP packets and for the protocol dissection of packets. A replicable testbed provides an early demonstration of the advanced monitoring and semantic routing use cases.

1 INTRODUCTION

Networking architectures need to evolve to support the requirements of future Internet services and 6G networks. The networking research and standardization communities are considering different approaches for this evolution, which can be broadly classified in 3 different categories. A first category includes “clean slate” and “revolutionary” solutions, which aims at replacing the legacy networking layer (IP protocol) with something different and new. A second category of solutions operate at layers above the networking layer (i.e. transport and application layer) and do not require to modify the IP layer. The third category includes “evolutionary solutions” that extend the IP layer trying to preserve backward compatibility.

In this paper we propose the Extensible In-band Processing (EIP) solution, which belongs to the third category. EIP extends the current IPv6 architecture fully supporting the needs of Future Internet services and 6G networks, but without requiring a clean-slate revolution. In section 2 we discuss the overall approach and architecture of EIP. In section 3 we present three use cases for EIP: advanced monitoring, semantic routing, deterministic networking. Clearly, this list is not exhaustive, as EIP is meant to be extensible and further use cases can be added to the framework. Section 4 provides a review of the three categories of solutions for the evolution of the networking architectures thet we have identified. In particular, a short review is presented for the first two categories (the revolutionary solutions and those operating above the networking layer), as these solutions are different from the one we are proposing. A more comprehensive review is presented for the evolutionary solutions, our goal is to show how the IP networking layer, and in particular the IPv6 protocol is already evolving following the requirements of services and applications. Our proposed EIP solution is aligned with this ongoing trend, hence it represent a viable approach to be deployed in real networks in the near future.

Section 5 provides some details at the protocol level on the the EIP header and its Information Elements. A proposal for the evolution of the Internet needs to be backed by implementations (“running code”) and testbeds to show the feasibility and receive feedback on the design choices. For these reasons, we are building an open source prototype for the EIP solution. We describe the current status of the implementation and prototype in section 6.

The proposed EIP framework is at an early stage of development. We look for the involvement of a community that can contribute to: i) the definitions of use cases (extending the ones that we have mentioned and proposing new ones); ii) the design of the EIP header with its Information Elements for the various use cases; iii) the open source implementation of the prototypes of EIP framework. We have set up the entry point for this community in the EIP home page [4].

2 EXTENSIBLE IN-BAND PROCESSING - EIP

The design the IP networking layer has been strongly influenced by the so called *end-to-end* concept, which prescribed putting “complex” functions in the IP hosts and “simple” functions in network forwarding devices (IP routers). To overcome this limitation, Network operators needed to use additional layers (e.g. ATM and then MPLS) in addition to the IP layer to put the "complex" functions and features that are needed to run operators networks (backbones and access networks), see Fig. 1-a. We also note that in the real world, the end-to-end concept has been disregarded with the introduction of middleboxes devices like NATs, TCP accelerators, but this has been seen as an unavoidable mistake and perceived as an architectural problem.

Recently we observe a clear trend in extending the functionality of the IP networking layer, going beyond the plain packet

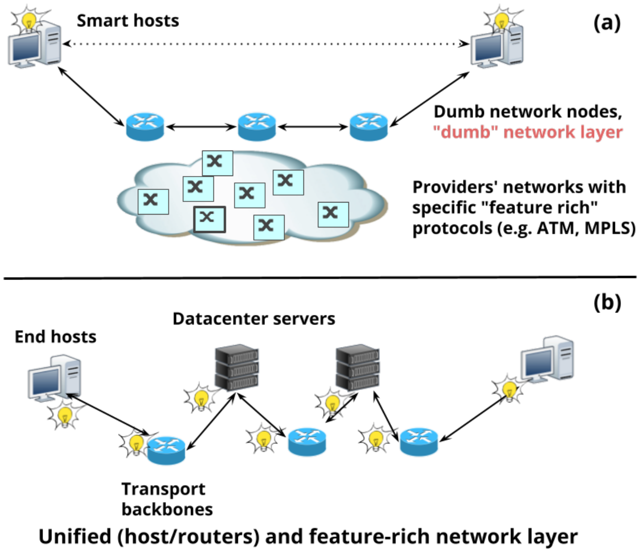

Figure 1: The evolution of IP networking towards a feature-rich and unified (host-router) layer

forwarding. An example of this trend is the rise of the SRv6 *network programming* approach [9]. With SRv6, the routers can implement "complex" functionalities and they can be controlled by a *network program* that is embedded in IPv6 packet headers. The operators can find all the needed functionality in the IPv6/SRv6 dataplane with no need of middleboxes nor of a separate MPLS layer. Another example is the INT (IN band Telemetry) solution for network monitoring [43].

The Extensible In-band Processing (EIP) proposal is aligned with this trend, which will ensure a future proof evolution of networking architectures. We envisage a feature-rich, extensible and programmable IPv6 networking layer, in which end-hosts, routers, virtual functions, servers in datacenters co-operate so that services can be implemented in the smartest and more efficient way, see Fig. 1-b. In the past, a similar vision was proposed in [42].

The EIP solution foresees the introduction of an EIP header in the IPv6 packet header. The proposed EIP header is extensible and it is meant to support a large number of different use cases. In general, both end-hosts and transit routers can read and write the content of this header. Depending of the specific use-case, only specific nodes will be capable and interested in reading/writing the EIP header. The use of the EIP header will be confined to a single domain or to a set of cooperating domains, so there is no need of a global, Internet-wide support of the new header for its introduction. Moreover, there could be usage scenarios in which legacy nodes can simply ignore the EIP header and provide transit to packets containing the EIP header.

An important usage scenario considers the transport over a provider network, in particular the network portion from an ingress edge node to an egress edge node. In this scenario, the ingress edge node can encapsulate the user packet coming from an access network into an outer packet. The outer packet is carried inside the provider network until an egress edge node, which will decapsulate the inner packet and deliver it to the destination access network or to another transit network, depending on the specific topology and service. We assume that the IPv6/SRv6 dataplane is used in the provider network, this means that the ingress edge node will be the source of an outer IPv6 packet in which it is possible to add the EIP header. The outer IPv6 packet (containing the EIP header) will be processed inside the "limited domain" of the provider network, so that the operator can make sure that all the transit routers either are EIP aware or at least they can forward packets containing the EIP header. In this usage scenario, the EIP framework operates "edge-to-edge" and the end-user packets are "tunneled" over the EIP domain.

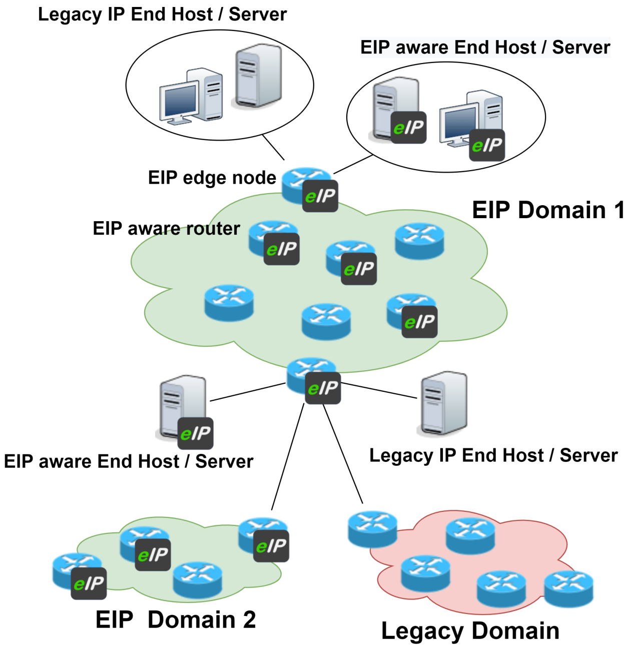
Legacy IP End Host / Server
EIP aware End Host / Server
EIP edge node
EIP aware router
EIP Domain 1
eIP
EIP aware End Host / Server
Legacy IP End Host / Server
EIP Domain 2
Legacy Domain

Figure 2: EIP framework

The architectural framework for EIP is depicted in Fig. 2 and it is discussed in more details in [40]. An EIP domain is made up by EIP aware routers and can also include legacy routers. At the border of the domain, EIP edge nodes are used to interact with legacy End Hosts / Servers and with other domains. It is also possible that an End Host / Server is EIP aware, in this case the EIP framework could operate "edge-to-end" or "end-to-end". The potential interaction with a cooperating EIP aware domain is also shown in Fig. 2.

Fig. 3 shows how the EIP header, carried in the the IPv6 header, can support a number of use cases, some of them will be discussed in section 3.

The EIP header can be carried in two different ways inside the IPv6 Header: 1) as a new Option for the IPv6 Hop-by-Hop (HBH) Extension Header; 2) as a new TLV for the Segment Routing Header (SRH), which is an IPv6 Routing Extension Header. The EIP header will carry different *EIP Information Elements* that are defined to support the different use cases. The EIP protocol details are specified in [41] and will be discussed in section 5.

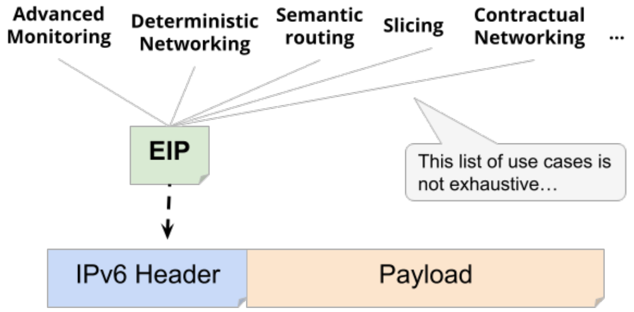

Figure 3: The EIP header, carried inside the IPv6 Header, supports a number of use cases (the figure is not exhaustive)

2.1 Advantages of the EIP header

The IPv6 Header has been designed to be "extensible", thanks to its Extension Headers and to the "Options" that can be defined inside some Extension Headers. Nevertheless, the process to define and standardize such extensions is long and complicated, creating a practical barrier to the innovation in the networking layer. The definition of a common EIP header (i.e. a single Option type for the Hop-by-hop Extension Header and/or a single TLV for the Segment Routing Header) that supports multiple use cases can be beneficial in this respect, for the reasons that are mentioned hereafter.

1. The number of available Option Types in HBH Header is limited, likewise the number of available TLVs in the Segment Routing Header (SRH) is limited. Defining multiple Option Types or SRH TLVs for multiple use case is not scalable and puts pressure on the allocation of such codepoints.

2. The definition and standardization of specific EIP Information Elements for the different use cases will be simplified, compared to the definition of a new Option Type or SRH TLVs.

3. Different use cases may share a subset of common EIP Information Elements.

4. Efficient mechanism for the processing of the EIP header (both in software and in hardware) can be defined when the different EIP Information Elements are carried inside the same EIP header.

3 EIP USE CASES

3.1 Advanced monitoring

Traditional network monitoring solutions are based on the centralized collection of monitoring data collected by network nodes (SNMP/Netflow). These traditional solutions are "slow" and have scalability issues. They are not meant to monitor a large number of single flows in real time, they are meant to monitor large aggregates of flows (e.g. all traffic flowing on a given link). Typical time scale of the reaction time of the management system is in the order of minutes (e.g. 1-5 minutes). The traditional approach separates the data plane and the management plane. The nodes collect counters and statistics, then they communicate with the NMS (Network Management Stations) over the management plane. Usually, "polling" is used by the NMSs to periodically retrieve information from nodes.

Current technologies make it possible to define more complex monitoring operations to be performed by nodes in the data plane. The protocol extensibility offered by EIP is the natural complement to this new advanced monitoring approach. Thanks to information carried in the EIP header, it is possible to use the data plane also to synchronize monitoring operations across different nodes and it is possible to collect monitoring information in real time. Moreover, data plane entities can be used to sample and aggregate monitoring information and to forward the aggregated information to the management layer with no need of polling.

The *advanced monitoring* is an umbrella term for a set of specific use cases that can be defined. In particular, so far we have considered: i) the "in-band" Path tracing functionality described in [19], used to collect the egress interface identifier, a timestamp and the interface load state from all routers that are crossed by a packet using a limited overhead in the packet; ii) an end-to-end delay measurement functionality corresponding to the one provided by the Simple Two-way Active Measurement Protocol (STAMP) [21], but realized in-band instead of using additional UDP packets for delay measurement.

3.2 Semantic routing

The traditional IP addressing architecture [37] uses IP addresses to identify an interface on a network device. Traditional routing establishes paths through networks toward destination IP prefixes. Then, the routers forward the IP packets based on the decisions taken by the routing protocols. Over time, routing has been enhanced to support decisions based on additional information contained in the packets and policy configured in the routers. This is known as Policy Based Routing [46]. This approach is further extended by the so-called "Semantic Routing" [16] [29]. Semantic Routing is a new routing paradigm that supports enhanced forwarding and routing to select different paths (different from the shortest path) for different flows. These decisions are based on the additional semantics.

Today, several challenges must be addressed by the routing protocols to face the increasing traffic demand. These challenges span different areas, such as mobility, scalability, and security. According to [16], Semantic Routing extends the traditional routing by adding semantic information to IP packets. It can be implemented in different ways, such as placing additional information into existing fields, adding semantics to the IP addresses, or adding new fields to the packets. Some solutions may require defining new routing protocols or extending the existing ones to consider these additional semantics when taking the route selection decisions. Other approaches do not require any changes at all. A router can behave in three different ways: the additional information can be opaque (i.e., not visible to the routers), transparent (i.e., the router is able to see the information, but the processing is not influenced by it or can be active (i.e., the router process the information).

Forwarding behavior is changed to consider other information present in the packet and policy installed (configured or programmed) in the routers. The packet forwarding engine of routers forwards the IP packets according to the fields of the packet. Packet fields considered can be the fields of the IP header, with or without additional semantics, packet payload, or new fields. These fields are matched against forwarding instructions. These forwarding instructions can be installed by routing protocols or configured through a management interface or installed by a Software-Defined Networking (SDN) controller or based on the network congestion, network conditions, and traffic loads. EIP comes into play when it is useful to define new fields in IP packets, and it can easily support the requirements of Semantic Routing.

[29] presents several use cases of Semantic Routing, such as *Content-based routing* (CBR) or *Geotagging*. In this work, we focus on the Geotagging use case. With Geotagging, it is possible to add a tag to the IP packet. The tag carries location information, for example the location of the source or the location of the destination node. [29] suggests encoding the geographic location information directly in the addresses. On the other hand, in our proposal we extended the EIP framework by adding a new EIP information element that carries the location (geotag) information.

3.3 Deterministic Networking

[RFC8655] and [RFC8938] respectively specify the Architecture and data Plane Framework for Deterministic Networking. [RFC8939] specifies the IP data plane for DetNet, with this design choice: existing IP-layer and higher-layer protocol header information is used to support flow identification and DetNet service delivery. There are known limitations in the current IP data plane for DetNet, as discussed in [RFC8939]. In particular, the IP data plane for DetNet, uses 6-tuple-based flow identification, where "6-tuple" is destination address, source address, IP protocol, source port, destination port, and DSCP (optional matching on the IPv6 Flow Label field). Flow aggregation may be enabled via the use of wildcards, masks, lists, prefixes, and ranges. IP tunnels may also be used to support flow aggregation. The result of this design is:

* Operational complexity: from a practical standpoint, this means that all nodes along the end-to-end path of DetNet flows need to agree on what fields are used for flow identification. Possible consequences of not having such an agreement include some flows interfering with other flows, and the traffic treatment expected for a service not being provided.

* Lack of unified end-to-end sequencing information: service protection (if enabled) cannot be provided end to end, only within sub-networks.

EIP is used to add explicit DetNet information in IP packets (e.g. flow identification and sequencing). This simplifies the implementation of Deterministic Networking in IP routers and hosts and provides additional features with respect to current support of Detnet in IP networks.

4 NETWORKING ARCHITECTURE EVOLUTION

In this section we provide a review of the three different categories in which we have broadly classified the approaches for the evolution of networking architectures: i) solutions above the networking layer; ii) clean slate solutions; iii) evolutionary solutions. For the first two categories we only mention few representative examples, while for the third category, we provide a more detailed analysis. Based on this analysis, we can say that the "evolutionary" approach represents an ongoing trend which has already gained momentum: many solutions are under development and standardization. This trend will continue in the coming years and our EIP proposal is well aligned with it.

4.1 Solutions above the layer 3

The recent "Extensible Internet" proposal [23] starts from the consideration that in-network services that go beyond best-effort packet delivery are already deployed within *private* networks of over-the-top providers (called CCPs: Cloud and/or Content Providers). This has been done without changing the architecture and protocols of the networking layer (at least from the perspective of the end user. A set of in-network services have been deployed in the CCPs Point of Presences (PoP), like caching, flow termination, load balancing. According to [23]: i) the goal of the "Extensible Internet" is to allow the public Internet "*to offer similar capabilities in an open and extensible manner*"; ii) these services should be layered *on top* of packet delivery, hence no changes in the networking layer are needed.

It is also worth mentioning that an important evolution of the networking architectures is happening above the transport layer with the QUIC protocol [1], which has recently been standardized as RFC 9000 [28].

4.2 Clean slate and revolutionary solutions

The Information-centric Networking [39] approach considers changing the Internet paradigm from interconnecting hosts to accessing "named" content. A number of specific architectures and solutions follow this approach, for example Named Data Networking (NDN) [30] and Content Centric Networking (CCN) [44]. These solutions consider a "clean slate" approach which is meant to replace the IP layer with a new networking layer.

The ITU-T Focus Group "Technologies for Network 2030" [27] has recently promoted a set of activities regarding the evolution of Internet. A set of use cases and network requirements has been collected in [18]. The "New IP" solution [31, 32] has been proposed as an answer to these requirements. This solution considers substantial changes to the current IP networking layer, like the replacement of IP addresses with variable length addresses.

4.3 Evolutionary solutions

In the last few years, we have witnessed important innovations in IP networking, centered around the emergence of Segment Routing for IPv6 (SRv6) [8] and of the SRv6 "Network Programming model" [9]. With SRv6 it is possible to insert a *Network program*, i.e. a sequence of instructions (called *segments*) in a header of the IPv6 protocol, called Segment Routing Header (SRH). Each instruction (or segment) can simply represent a topological waypoint to be crossed in the path toward the final destination address, or it can also identify a specific operation (called *behavior*) to be executed by the networking layer of the node reached by the packet. The list of instructions can be called "Segment List", "SRv6 Policy" or "Network Program" associated with the packet. The RFC 8986 [9] identifies a first standard set of behaviors that are part of the Network Programming Model and can be referenced in Segment Lists, for example encapsulation/decapsulation operations, forwarding over specific interfaces, processing in a specific "Virtual Routing Function" associated to a tenant. The interesting aspect is that a segment identifier, represented with an IPv6 address can at the same time identify a specific physical location of a node, the behavior (operation) to be executed by the node, and even some parameter associated with the operation to be performed on that particular packet. The SRv6 Network Programming model can be used to support services and features like VPNs, Traffic Engineering, Fault Protection/Restoration in IP backbones and Data Center networking scenarios. It is of particular interest the capability of SRv6 to be integrated with host based networking in cloud computing and data center networking scenarios. From the point of view of a transport IP backbone, with SRv6 it is possible to include in a single IPv6 based layer the functionality that is currently split between IP and MPLS layers. This results in a great simplification of the network architecture, with relevant savings in operational costs (as well as in energy costs). With the foundational documents already promoted to RFCs (e.g. [8] and [9]), the SRv6 technology is still in hectic evolution with more than 10 internet drafts endorsed by the IETF SPRING working group and around 100 active individual internet drafts. A very important enhancement that is under development is the "segment compression" feature, allowing a single IPv6 address to carry a number of "compressed" Segment Identifiers. This feature overcomes the problem of carrying a relatively large overhead in packet headers when using the default approach of representing each segment with a different IPv6 address (16 bytes long). The internet draft [33] provides an updated report of more than 10 large scale deployments of SRv6 in production operators' networks. The recent edition of "MPLS, SD & AI Net World Congress" held in Paris in April 2022 has been focused on "The SRv6 Momentum" [11]. Rakuten (an operator in Japan with more than 4 millions subscribers) [38][13] and Bell Canada presented amazing results from their SRv6 deployments. Note that both deployments are already based on the concept of SRv6 segment compression that we have just introduced. Rakuten claims that "transitioning to SRv6 will increase network resiliency and support a wider range of SLAs that are foundational for upcoming 5G and IoT services" and "reduce power consumption by nearly 30 percent". The presentation by Bell Canada [5] shows "how SRv6 and a P4 programmable switch fabric can reduce carrier network services costs by up to 90%, [...], and power consumption by as much as 66% and thus facilitate a cost effective migration of latency sensitive applications to the network edge". A growing community of technology providers and operators is proposing SRv6 as a solution for the transport networks of 5G/6G networks. This is called "SRv6 Mobile User Plane" [34]. A technology demo was showcased by Intel during the last Mobile World Congress [14]. All in all, we believe that SRv6 will become the de-facto standard for the networking layer in transport backbones in the years to come. For this reason we build EIP on top of the current status and foreseeable evolution of the SRv6 dataplane. We note that the standardization process of SRv6 has been rather difficult, and it has come after a long opposition coming from the advocates of a "pure" end-to-end model for IPv6 networks. An important assumption in this respect, that helped to overcome the oppositions, is that the deployment of the SRv6 technology and features does not need to happen ubiquitously in the Internet. Instead, SRv6 services are meant to work inside "limited domains" (a.k.a. "controlled environments") [10]. Hence, a network operator can introduce advanced features in the IPv6 networking layer (also going beyond the status of the current standardization) with no impact on the adjacent domains in a multi domain end-to-end path. The lesson we have learned from SRv6 design, standardization and deployment (including the mentioned recent deployments by Rakuten and Bell Canada) is that there is room for enhancements of the networking layer that can be put in large scale production even at an early stage of standardization. Open source ecosystems have also played and are playing an important role in this process.

Let us review other recent activities on the extension of the networking layer to support more complex functions, which have already reached tangible results. By far, the most prominent topic concerns the extensions related to network monitoring. The concept of INT "In-band Network Telemetry" has been proposed since 2015 [36] in the context of the definition of use cases for P4 based data plane programmability. The latest version of INT specifications dates November 2020 [43]. INT specifies the format of headers that carry monitoring instructions and monitoring information along with data plane packets. The specific location for INT Headers is intentionally not specified: an INT Header can be inserted as an option or payload of any encapsulation type. Within the IETF community, the In-band Telemetry concept has been adopted by the IPPM working group, renaming it "In-situ Operations, Administration, and Maintenance" (IOAM). The internet draft [7] is about to become an IETF RFC. Note that IOAM is focused on "limited domains" as defined in [10]. The in-situ OAM data fields can be encapsulated in a variety of protocols, including IPv6. The specification details for carrying IOAM data inside IPv6 headers are provided in draft [6], which is also close to becoming an RFC . In particular, IOAM data fields can be encapsulated in IPv6 using either Hop-by-Hop Options header or Destination options header. Another prominent example of extensions to IPv6 for network monitoring is specified in [15], which defines an IPv6 Destination Options header called Performance and Diagnostic Metrics (PDM). The PDM optional header provides sequence numbers and timing information as a basis for measurements. A further extension example is the "Alternate Marking Method", a recently proposed performance measurement approach. The draft [20] (also close to becoming an RFC) defines a new Hop-by-Hop Option to support this approach. Finally, Path Tracing [19] proposes an efficient solution for recording (with timestamps) the route taken by a packet and also defines a new Hop-by-Hop Option. The support of Hop-by-Hop headers in IPv6 routers has some known issues that the IETF community is trying to address [26]. These issues are critical when Hop-by-Hop headers are meant to be used on the Internet at large, while they can be addressed and mitigated in the scope of limited domains.

5 DEFINITION OF EIP HEADER AND INFORMATION ELEMENTS

The current state of the definition of the EIP header is contained in [41]. The EIP header can be carried as an Option in the IPv6 Hop By Hop (HBH) Extension Header, or as a TLV in the Segment Routing Header (SRH), as depicted in Fig. 4. Hereafter, we will only consider the first case (EIP header as a HBH Option), shown in Fig. 5. In general, the EIP header contains one or more EIP Information Elements (IEs), the one shown in Fig. 5 carries one IE.

The first byte in Fig. 5 represents the Option Type. The first three bits (001) of the Option Type mean that the option is to be ignored

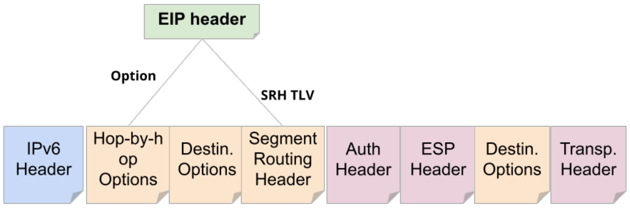

Figure 4: Carrying the EIP header in the IPv6 header

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
                                +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                |0|0|1|   EIP   |Opt Data Len=xx|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|1 0|  Data Len |     EIP extended LTV Code     |  LTV content  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            LTV content (optional, variable length)            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5: EIP Option with one Information Element.

when not implemented (packet is not dropped) and its content can change at every hop, as defined in [12]. As for the remaining 5 bits, they have to be assigned by IANA following an IETF standardization process. In our current testbed implementations we have used the experimental type $0x3e$, as defined in [17].

The second byte carries the length of the Option content. After the first two bytes there is the content of the EIP header, i.e. a sequence of EIP Information Elements (only one in the figure). IEs are structured in a Length Type Value (LTV) fashion. The first two bits are used to define the length of the EIP Type (or code) in bytes, which can be 1, 2 or 3 bytes. Short codes of one byte should only be used for use cases where high performance and low overhead are crucial, as there are only a maximum of 256 available. The codes of two bytes are referred to as "extended codes", there are 65536 such codes available, the codes of three bytes are called "double extended codes" and there are 2^{24} such codes available. The example in the figure shows an extended code (2 bytes). The length of the IE is specified in the next six bits, which represent the total length in four octects units, not including the first four octects, so that the length can be up to 256 bytes.

The design of the IEs starts from the analysis of the use cases, this is an ongoing process as only a part of the use cases discussed in section 3 have been considered so far. The detailed definition of the format and content of the defined Information Elements is available in [41], hereafter we provide a high level description.

Concerning **Semantic Routing**, we have defined *Geotagging* IE. It supports the introduction of Localization information in the EIP header to be used for routing purposes. The IE can be used to represent either the position of the destination or of the source nodes, depending on the specific use case. The destination geotag can be directly used in the forwarding of the packet based on the geographical position. When the Geotagging IE is used to carry the source position, it is used to communicate this information to other nodes "in-band". Geotagging can be very useful in routing scenarios for Mobile Ad-hoc NETworks (MANETs). In the Geotagging IE, latitude and longitude are encoded using Geohash [45]. With this representation, we can achieve a precision in the order of few hundreds meters using 32 bits, or a precision in the order of 1 cm using 64 bits (more details in [41]).

For the **Advanced Monitoring** in-band Path Tracing use case, we have defined a *Compact Path Tracing* IE which is a porting of the solution proposed in [19]. We recall that the goal is to record for each hop an interface identifier, the load of the interface and a timestamp, using a limited amount of bytes. From the interface identifiers it is possible to reconstruct the nodes that have been crossed by the packet. As defined in [19], the information to be collected at every hop is stored in a structure called Midpoints Compressed Data (MCD), and a stack of MCDs is stored in the IE.

For the **Advanced Monitoring** in-band end-to-end Delay Monitoring use case, we have defined the *Stamp* IE to collect timestamps at the source and destination (based on [21] and the *Timestamps* IE that can collect timestamps at each hop.

In order to support **security**, we have defined an HMAC information element, mirroring the HMAC SRH TLV defined in [8], to be used when authentication and data integrity are required. We have also proposed an authenticated version of the path tracing IE, where a single HMAC field is enough to guarantee the integrity of the received data.

Finally, we note that a portion of the code space for the EIP Information Elements (for each of the three sizes of the codes) has been reserved to "user defined" IEs which are not subject to standardization. This availability of user defined code will also simplify experiments in deployed networks.

6 PROTOTYPE AND TESTBED

We developed a prototype implementation of the EIP framework. The implementation currently supports some of the Information Elements defined for the Semantic Routing and Advanced Monitoring use cases. We are working to extend the supported use cases and to add the support of further uses cases, by defining and implementing further Information Elements. Our prototype consists of a packet generator/dissector and a Linux based EIP aware router.

The EIP packet generator/dissector is used to generate (and analyze) IPv6 packets containing the EIP header and the supported EIP Information Elements. This has been implemented as a Python application using the Scapy library. It can generate packets with the EIP header containing a set of Information Elements and can customize their content. It can also be used to dissect the received packets. The IEs mentioned in the previous section are supported along with several other IEs defined in [41]. The EIP packet generator/dissector is available at [3], which also provides the updated list of supported IEs.

The Linux based EIP aware router prototype is based on a set of eBPF programs that are capable of processing a subset of the EIP Information Elements defined in [41]. The EIP aware router prototype currently implements the advanced monitoring use case, in particular it supports the path tracing and the in-band end-to-end delay measurement.

Figure 6 shows the topology used for the testbed. It is realized using four Linux namespaces to separate each node.

The router r1 generates a packet containing the EIP header and transmits the packet towards r4. The EIP packet can be generated by

```
+---------------+    +---------------+    +---------------+    +---------------+
|      r1       |    |      r2       |    |      r3       |    |      r4       |
|               |    |               |    |               |    |               |
|           i12 +----+ i21       i23 +----+ i32       i34 +----+ i43           |
|               |    |               |    |               |    |               |
|               |    |               |    |               |    |               |
+---------------+    +---------------+    +---------------+    +---------------+
```

Figure 6: Testbed topology for eBPF implementation.

our Scapy based packet generator or using tcpreply that reproduces a .pcap file. The eBPF programs that process the EIP header can be attached to an XDP hook of the incoming interfaces of the intermediate nodes r2 and r3 and of the destination node r4. We can use tcpdump or the Scapy based packet dissector to capture the packet on all interfaces of every node.

For the demonstrator of the Path Tracing use case, we attach the eBPF program to interface i21 on r2 and to interface i32 of r3, as r2 and r3 act as EIP aware routers. They will populate the Path Tracing information element with data relative to the ingress interface (id and load) and with the timestamp. For the demonstrator of the in-band end-to-end delay monitoring, we attach the eBPF program to the interface of r4, which will populate the Stamp IE in the EIP header.

The eBPF development has been performed using the HIKe/eCLAT framework [2, 24, 35]. The source code of the prototype can be found in [22], the instructions to replicate the experiments are in [25].

7 CONCLUSION

We have described a solution for the evolution of the IPv6 networking architecture called Extensible In-band Processing (EIP). The solution is based on an header to be carried in IPv6 packets. This header that can be read and written by end hosts and routers. We have presented the definition of the EIP header and of a set of EIP Information Elements and an open source prototype implementation of the solution. We have argued that EIP can support the needs of future Internet services, as it can be extended considering the requirements of multiple use cases. In the paper we have considered as use cases: advanced monitoring, semantic routing and deterministic networking. For the deterministic networking we have only a textual description of the use case. For advanced monitoring we have considered two specific sub cases (Path Tracing and in-band end-to-end delay monitoring) and we have completed the definition of the relevant EIP Information Elements. Moreover, we have implemented a tool for the generation of EIP header and a prototype of the dataplane processing operations based on eBPF. For semantic routing, we have presented the definition of EIP Information Element and the generation and dissection of packets of EIP header. We are continuing our work to complete the eBPF implementation of dataplane processing operations, as well as the definition of EIP Information Elements for all the use cases. The EIP solution is open and we have started to build a community around it [4]. We hope that this work can further stimulate interest about EIP.

REFERENCES

[1] A. Langley, et al. 2017. The QUIC Transport Protocol: Design and Internet-Scale Deployment. In *SIGCOMM '17: Proceedings of the Conference of the ACM Special Interest Group on Data Communication*. Association for Computing Machinery, New York, NY, USA, 183–196. https://doi.org/10.1145/3098822.3098842

[2] A. Mayer, et al. 2022. eBPF Programming Made Easy with HIKe and eCLAT. https://raw.githubusercontent.com/hike-eclat/docs/master/tech-docs/hike_eclat.pdf. Accessed: 2022-05-25.

[3] Anon. 2022. EIP Packet Generator/Dissector based on Scapy. https://github.com/eip-home/eip/wiki/eip-scapy. Accessed: 2022-05-24.

[4] Anon. 2022. Extensible In-band Processing (EIP) Home Page. https://eip-home.github.io/eip/. Accessed: 2022-05-23.

[5] D. Bernier and J. Eriksson. 2022. MPLS WC 2022 - Bell Canada: SRv6 Deployment – Service Programming Reduces Cost. https://www.segment-routing.net/conferences/MPLS-WC-2022-Daniel-Bernier-Jesper-Eriksson/. Accessed: 2022-05-21.

[6] S. Bhandari and F. Brockners. 2022. *In-situ OAM IPv6 Options*. Internet-Draft draft-ietf-ippm-ioam-ipv6-options-07. Internet Engineering Task Force. https://datatracker.ietf.org/doc/html/draft-ietf-ippm-ioam-ipv6-options-07 Work in Progress.

[7] F. Brockners, S. Bhandari, and T. Mizrahi. 2021. *Data Fields for In-situ OAM*. Internet-Draft draft-ietf-ippm-ioam-data-17. Internet Engineering Task Force. https://datatracker.ietf.org/doc/html/draft-ietf-ippm-ioam-data-17 Work in Progress.

[8] C. Filsfils, D. Dukes (ed.) et al. 2020. IPv6 Segment Routing Header (SRH). RFC 8754. https://doi.org/10.17487/RFC8754

[9] C. Filsfils, P. Camarillo (ed.) et al. 2021. Segment Routing over IPv6 (SRv6) Network Programming. RFC 8986. https://doi.org/10.17487/RFC8986

[10] B. E. Carpenter and B. Liu. 2020. Limited Domains and Internet Protocols. RFC 8799. https://doi.org/10.17487/RFC8799

[11] Upperside Conferences. 2022. 23rd Edition of the MPLS, SD & AI Net World Congress. https://www.uppersideconferences.com/mpls-sdn-nfv/mplswc_2022_agenda_day_1.html. Accessed: 2022-05-21.

[12] S. E. Deering and B. Hinden. 2017. Internet Protocol, Version 6 (IPv6) Specification. RFC 8200. https://doi.org/10.17487/RFC8200

[13] A. Dhamija. 2022. MPLS WC 2022 - Rakuten: SRv6 uSID in Deployment to Deliver End-to-end 5G Network Slicing. https://www.segment-routing.net/conferences/MPLS-WC-2022-Amit-Dhamija/. Accessed: 2022-05-21.

[14] E. Doe. 2022. Intel MWC Barcelona Blog: SRv6 Mobile User Plane (SRv6 MUP) Technology Demo. https://medium.com/intel-tech/intel-mwc-barcelona-blog-srv6-mobile-user-plane-srv6-mup-technology-demo-1e5e5ea9ae22. Accessed: 2022-05-21.

[15] N. Elkins, R. Hamilton, and mackermann@bcbsm.com. 2017. IPv6 Performance and Diagnostic Metrics (PDM) Destination Option. RFC 8250. https://doi.org/10.17487/RFC8250

[16] A. Farrel and D. King. 2022. *An Introduction to Semantic Routing*. Internet-Draft draft-farrel-irtf-introduction-to-semantic-routing-04. Internet Engineering Task Force. https://datatracker.ietf.org/doc/html/draft-farrel-irtf-introduction-to-semantic-routing-04 Work in Progress.

[17] B. Fenner. 2006. Experimental Values In IPv4, IPv6, ICMPv4, ICMPv6, UDP, and TCP Headers. RFC 4727. https://doi.org/10.17487/RFC4727

[18] FG-NET2030 – Focus Group on Technologies for Network 2030. 2020. Representative use cases and key network requirements for Network 2030. http://handle.itu.int/11.1002/pub/815125f5-en. Accessed: 2022-05-24.

[19] C. Filsfils, A. Abdelsalam, P. Camarillo, M. Yufit, T. Graf, Y. Su, and S. Matsushima. 2022. *Path Tracing in SRv6 networks*. Internet-Draft draft-filsfils-spring-path-tracing-00. Internet Engineering Task Force. https://datatracker.ietf.org/doc/html/draft-filsfils-spring-path-tracing-00 Work in Progress.

[20] G. Fioccola, T. Zhou, M. Cociglio, F. Qin, and R. Pang. 2022. *IPv6 Application of the Alternate Marking Method*. Internet-Draft draft-ietf-6man-ipv6-alt-mark-14. Internet Engineering Task Force. https://datatracker.ietf.org/doc/html/draft-ietf-6man-ipv6-alt-mark-14 Work in Progress.

[21] G. Mirsky, H. Nydell, et al. 2020. Simple Two-Way Active Measurement Protocol. RFC 8762. https://doi.org/10.17487/RFC8762

[22] G. Sidoretti and S. Salsano. 2022. eCLAT eBPF EIP implementation. https://github.com/netgroup/hikepkg-eip. Accessed: 2022-05-25.

[23] H. Balakrishnan, S. Banerjee et al. 2021. Revitalizing the Public Internet by Making It Extensible. *SIGCOMM Comput. Commun. Rev.* 51, 2 (may 2021), 18–24. https://doi.org/10.1145/3464994.3464998

[24] HIKe - eCLAT Team. 2022. HIKe - eCLAT documentation Home Page. https://hike-eclat.readthedocs.io/en/latest/index.html. Accessed: 2022-05-25.

[25] HIKe - eCLAT Team. 2022. HIKe - eCLAT eip package documentation. https://hike-eclat.readthedocs.io/en/latest/hike_programs.html#eip-package. Accessed: 2022-05-25.

[26] B. Hinden and G. Fairhurst. 2022. *IPv6 Hop-by-Hop Options Processing Procedures*. Internet-Draft draft-ietf-6man-hbh-processing-00. Internet Engineering Task Force. https://datatracker.ietf.org/doc/html/draft-ietf-6man-hbh-processing-00

Work in Progress.

[27] ITU-T FG NET-2030. 2020. Focus Group on Technologies for Network 2030. https://www.itu.int/en/ITU-T/focusgroups/net2030/Pages/default.aspx. Accessed: 2022-05-24.

[28] J. Iyengar and M. Thomson (eds.). 2021. QUIC: A UDP-Based Multiplexed and Secure Transport. RFC 9000. https://doi.org/10.17487/RFC9000

[29] D. King and A. Farrel. 2021. *A Survey of Semantic Internet Routing Techniques*. Internet-Draft draft-king-irtf-semantic-routing-survey-03. Internet Engineering Task Force. https://datatracker.ietf.org/doc/html/draft-king-irtf-semantic-routing-survey-03 Work in Progress.

[30] L. Zhang, et al. 2014. Named Data Networking. *ACM SIGCOMM Computer Communication Review* 44, 3 (2014), 66–73.

[31] R. Li, L. Dong, C. Westphal, and K. Makhijani. 2021. Qualitative Communication for Emerging Network Applications with New IP. In *2021 17th International Conference on Mobility, Sensing and Networking (MSN)*. IEEE, New York, NY, USA, 628–637. https://doi.org/10.1109/MSN53354.2021.00096

[32] R. Li, K. Makhijani, and L. Dong. 2020. New IP: A Data Packet Framework to Evolve the Internet : Invited Paper. In *2020 IEEE 21st International Conference on High Performance Switching and Routing (HPSR)*. IEEE, New York, NY, USA, 1–8. https://doi.org/10.1109/HPSR48589.2020.9098996

[33] S. Matsushima, C. Filsfils, Z. Ali, Z. Li, K. Rajaraman, and A. Dhamija. 2022. *SRv6 Implementation and Deployment Status*. Internet-Draft draft-matsushima-spring-srv6-deployment-status-15. Internet Engineering Task Force. https://datatracker.ietf.org/doc/html/draft-matsushima-spring-srv6-deployment-status-15 Work in Progress.

[34] S. Matsushima, C. Filsfils, M. Kohno, P. Camarillo, D. Voyer, and C. E. Perkins. 2022. *Segment Routing IPv6 for Mobile User Plane*. Internet-Draft draft-ietf-dmm-srv6-mobile-uplane-21. Internet Engineering Task Force. https://datatracker.ietf.org/doc/html/draft-ietf-dmm-srv6-mobile-uplane-21 Work in Progress.

[35] A. Mayer, P. Loreti, L. Bracciale, P. Lungaroni, S. Salsano, and C. Filsfils. 2021. Performance Monitoring with H^2: Hybrid Kernel/eBPF data plane for SRv6 based Hybrid SDN. *Computer Networks* 185 (2021), 107705. https://doi.org/10.1016/j.comnet.2020.107705

[36] P4.org. 2015. Improving Network Monitoring and Management with Programmable Data Planes. https://opennetworking.org/news-and-events/blog/improving-network-monitoring-and-management-with-programmable-data-planes/. Accessed: 2022-05-21.

[37] R. Hinden and S. Deering. 2006. IP Version 6 Addressing Architecture. RFC 4291. https://doi.org/10.17487/RFC4291

[38] Rakuten Mobile Advances Its Network for 5G and IoT Services with Cisco 2021. Rakuten Mobile Advances Its Network for 5G and IoT Services with Cisco. https://newsroom.cisco.com/c/r/newsroom/en/us/a/y2021/m06/rakuten-mobile-advances-its-network-for-5g-and-iot-services-with-cisco.html. Accessed: 2022-05-21.

[39] IRTF ICN RG. 2014. Information-Centric Networking Research Group Home Page. https://irtf.org/icnrg. Accessed: 2022-05-24.

[40] S. Salsano et al. 2022. Extensible In-band Processing (EIP) Architecture and Framework. draft-eip-arch. https://eip-home.github.io/eip-arch/draft-eip-arch.html Work in Progress.

[41] S. Salsano et al. 2022. Extensible In-band Processing (EIP) Headers Definitions. draft-eip-headers. https://eip-home.github.io/eip-headers/draft-eip-headers-definitions.html Work in Progress.

[42] Ion Stoica and Hui Zhang. 1999. Providing Guaranteed Services without per Flow Management. *SIGCOMM Comput. Commun. Rev.* 29, 4 (aug 1999), 81–94. https://doi.org/10.1145/316194.316208

[43] The P4.org Applications Working Group. 2022. In-band Network Telemetry (INT) Dataplane Specification, version 2.1. https://p4.org/p4-spec/docs/INT_v2_1.pdf. Accessed: 2022-05-21.

[44] V. Jacobson, et al. 2009. Networking Named Content. In *CONEXT 2009, 5th international conference on Emerging networking experiments and technologies*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/1658939.1658941

[45] Wikipedia. 2008. Geohash. https://en.wikipedia.org/wiki/Geohash. Accessed: 2022-05-24.

[46] Wikipedia. 2022. Policy-based routing. https://en.wikipedia.org/wiki/Policy-based_routing. Accessed: 2022-05-25.