TOR VERGATA
UNIVERSITÀ DEGLI STUDI DI ROMA

cnit consorzio nazionale interuniversitario per le telecomunicazioni

Extensible In-band Processing (EIP)

IRTF COIN RG interim meeting
Thursday, February 10, 2022

Stefano Salsano
Università di Roma Tor Vergata / CNIT
stefano.salsano@uniroma2.it

Agenda

- ➢ Background: network programmability
- ➢ Technical introduction to EIP (Extensible In-band Processing)
- ➢ EIP use cases
- ➢ Further technical discussion on EIP
- ➢ EIP prototype for Linux / eBPF
- ➢ Wrap up & feedback request

Network programmability (r)evolution of IP

→ IP Nodes can be more "easily" reprogrammed!

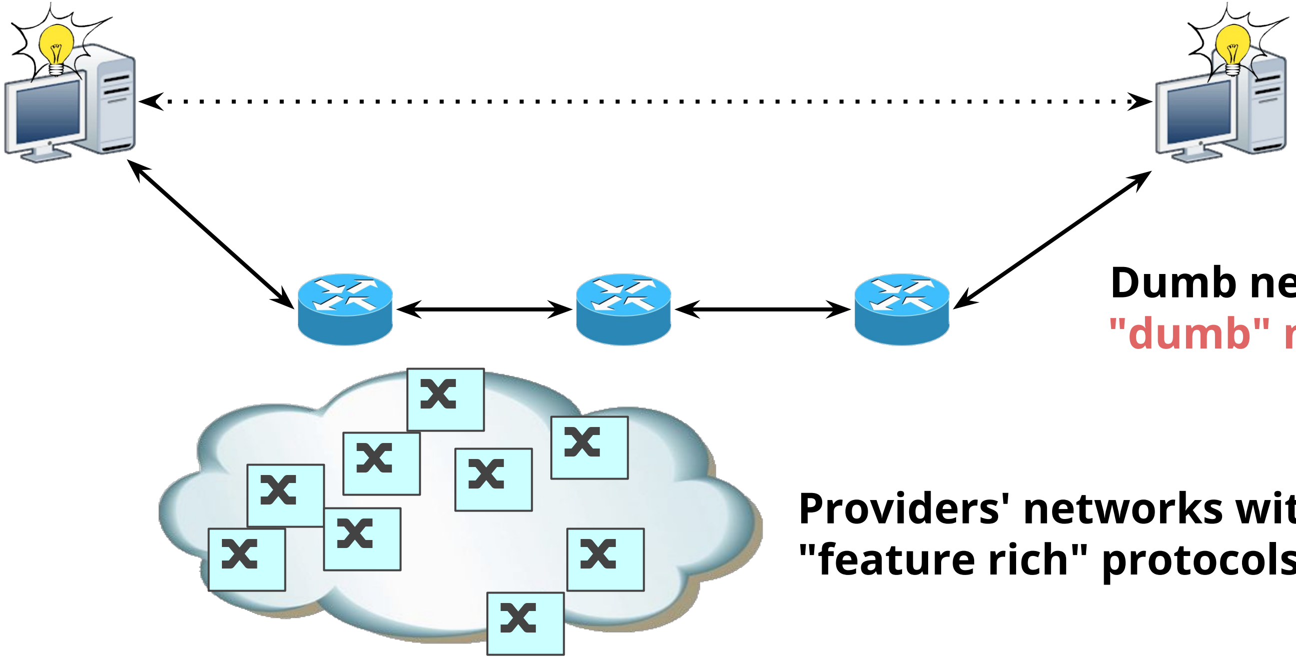
→ A more complex network layer is now feasible, see for example SRv6 Network Programming model

The end-to-end principle made up the Internet... maybe we can go further?

(see my talk @ IETF 111 - https://tinyurl.com/ip-revol-ietf-111)

We can move from here...

Smart hosts

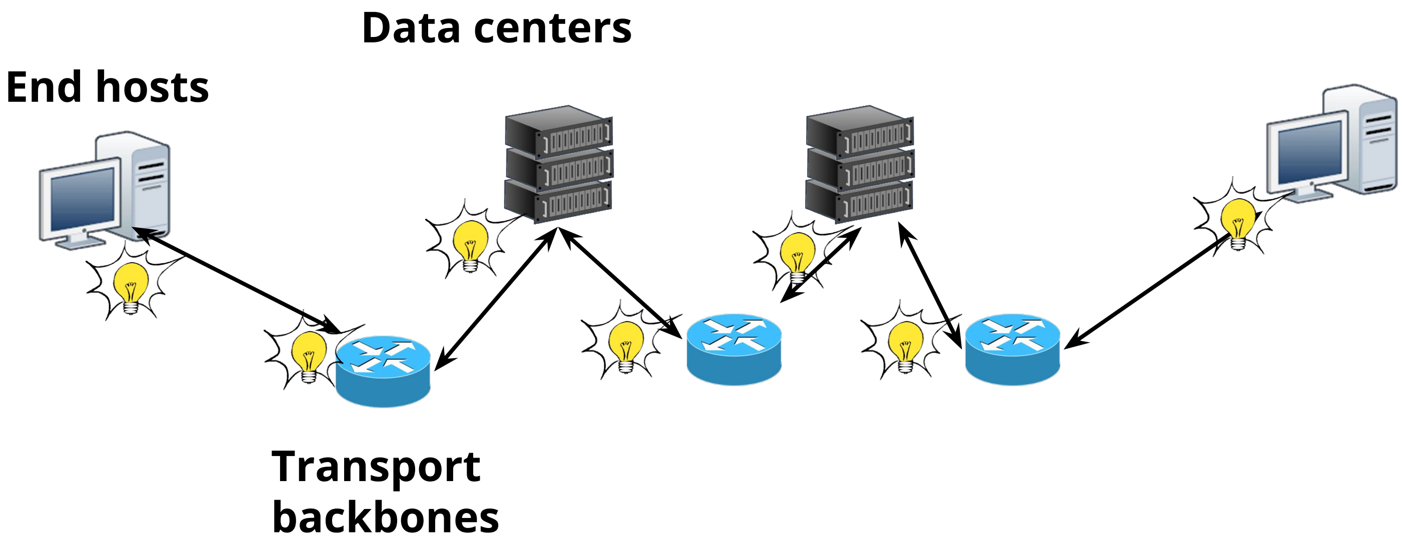


Dumb network nodes, "dumb" network layer

Providers' networks with specific "feature rich" protocols (e.g. ATM, MPLS)

Intelligence

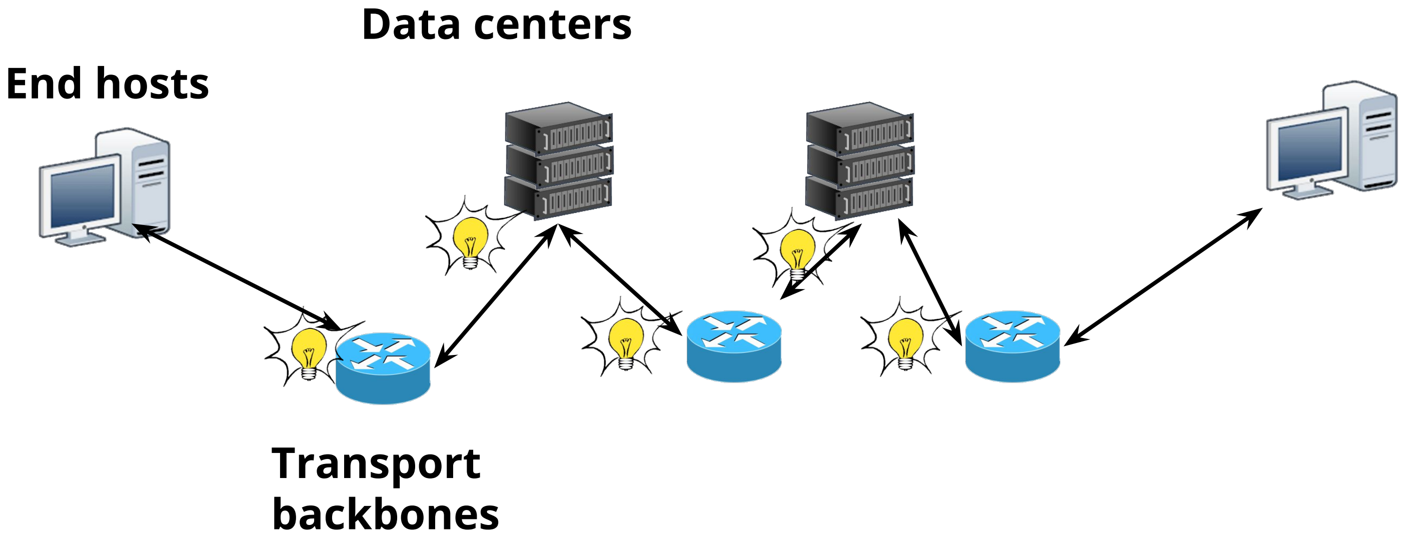
… to here



Unified (host/routers) and feature-rich network layer !

Intelligence

… and to here



In some scenarios… routers can be "smarter" than hosts (at network layer) !

Intelligence

Agenda

TOR VERGATA
UNIVERSITÀ DEGLI STUDI DI ROMA

cnit consorzio nazionale interuniversitario per le telecomunicazioni

- ➢ Background: network programmability
- ➢ Technical introduction to EIP (Extensible In-band Processing)
- ➢ EIP use cases
- ➢ Further technical discussion on EIP
- ➢ EIP prototype for Linux / eBPF
- ➢ Wrap up & feedback request

Extensible In-band Processing (EIP)

Definition of generic and extensible mechanisms to carry extensible information (EIP) in IPv6 packets headers, like

- monitoring information to be collected in transit
- parameters to be used for packet processing

Nodes can **read and write** this "extensible" EIP information.

Nodes can take packet processing decisions depending on the EIP information inserted in the packet by the source and by previous intermediate nodes.

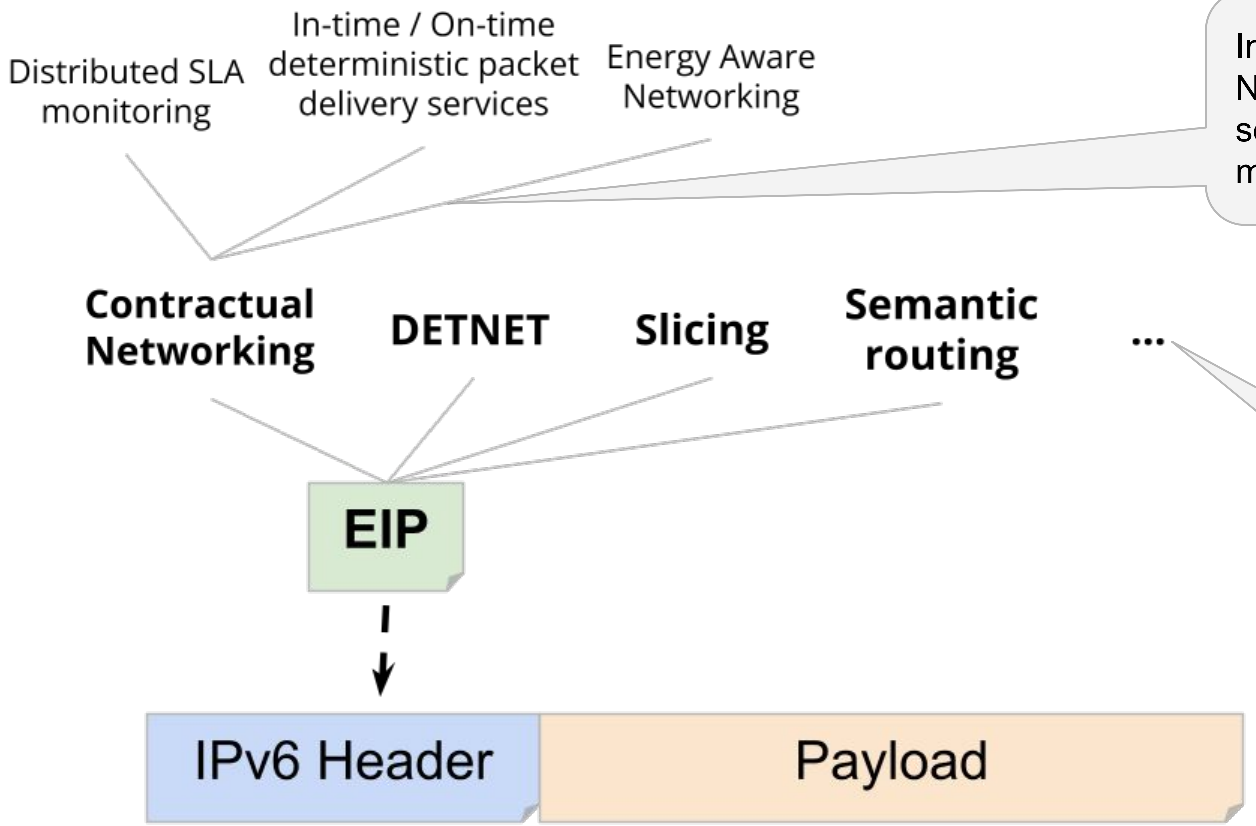
Extensible In-band Processing (EIP)

Note that, IPv6 header is already "extensible" by definition, there are "Extension Headers" and "Options" that can be defined inside some Extension Headers… but there are practical barriers in this extension process: the layer 3 functionality is "ossified".

We will collect requirements from a few use cases, then define EIP as "unified" extension approach, overcoming the "ossification barriers".

Open Source prototypes of EIP will also support the process.

EIP can support several use cases



In turn, the Contractual Networking use case can be seen as a "container" for many specific services

This list of use cases is an example, we are looking for feedback to focus on the most interesting use cases

Applicability

TOR VERGATA
UNIVERSITÀ DEGLI STUDI DI ROMA

cnit consorzio nazionale interuniversitario per le telecomunicazioni

We do not need/want to apply this paradigm everywhere!

New solutions can work in "limited domains" (a.k.a. "controlled domains")

See "Limited Domains and Internet Protocols" - RFC 8799 (Independent submission)

EIP Use case : Contractual networking

Contractual networking is an application scenario for EIP

What is a contract ? A **complex and distributed services** offered by a network.

The complex and distributed service is composed by several components like forwarding/metering/filtering/accounting/ (a composite service can include one or more of these "components")...

Contractual networking (my view)

TOR VERGATA
UNIVERSITÀ DEGLI STUDI DI ROMA

cnit consorzio nazionale interuniversitario per le telecomunicazioni

What is a contract ? A **complex and distributed service** (composed by features like forwarding/metering/filtering/accounting/…) offered by a network provider

Two phases:

1) Contracts are "negotiated" (Control plane)
2) Contracts are "enforced", i.e. packets are processed according to the negotiated contract (Data plane)

Scope of EIP definition

The definition of EIP will focus on:

data plane (= contract enforcement/monitoring phase)

and will NOT focus on:

control plane (= contract definition and negotiation phase)

EIP: packet processing/forwarding

TOR VERGATA
UNIVERSITÀ DEGLI STUDI DI ROMA

cnit consorzio nazionale interuniversitario per le telecomunicazioni

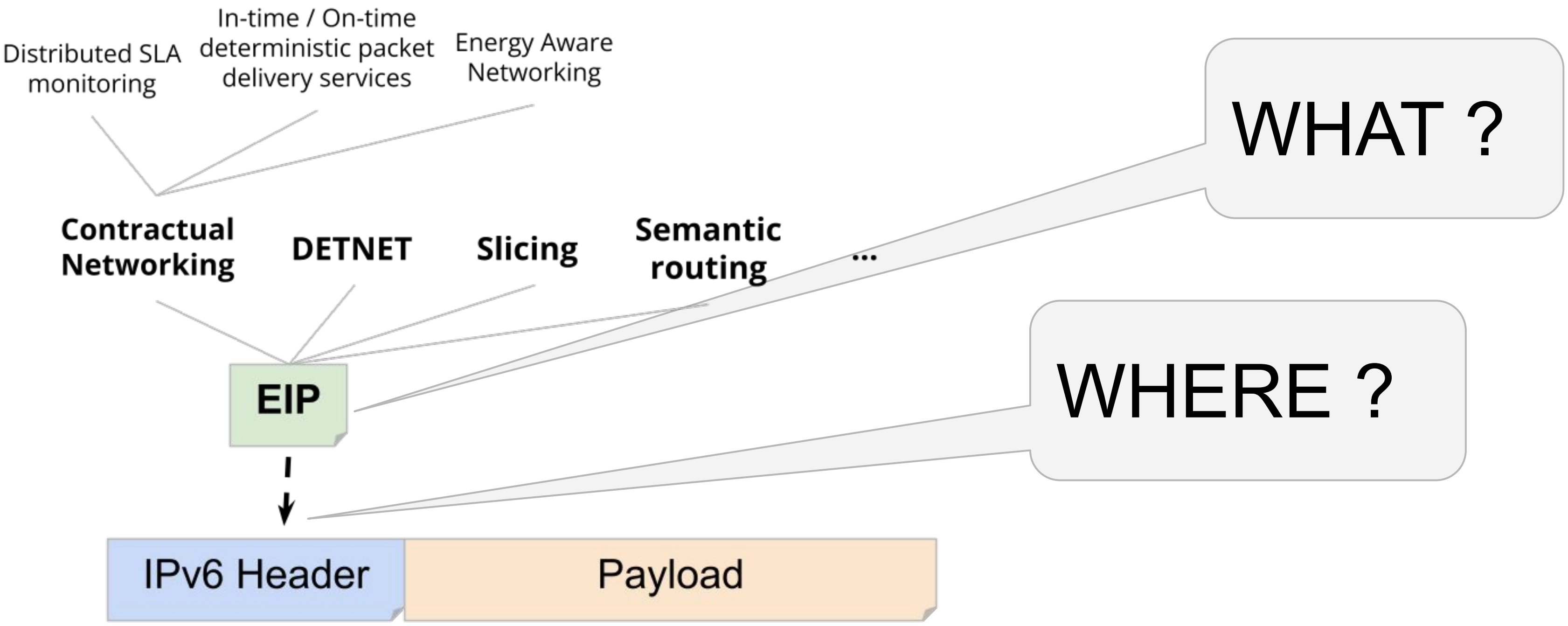
The processing / forwarding of packets in EIP can be based on the combination of the following information:

- information inserted by the source (or edge node), per flow and/or per packet
- information **dynamically** inserted by other layer 3 nodes in the path, based on the state of the node and on the info in the packet, per flow and/or per packet
- "regular" IP layer information (e.g. routing information)

The information inserted by the source and by intermediate nodes can include node programmability instructions such as match-action instructions

=> Network-wide dynamic and stateful packet processing

Definition of EIP



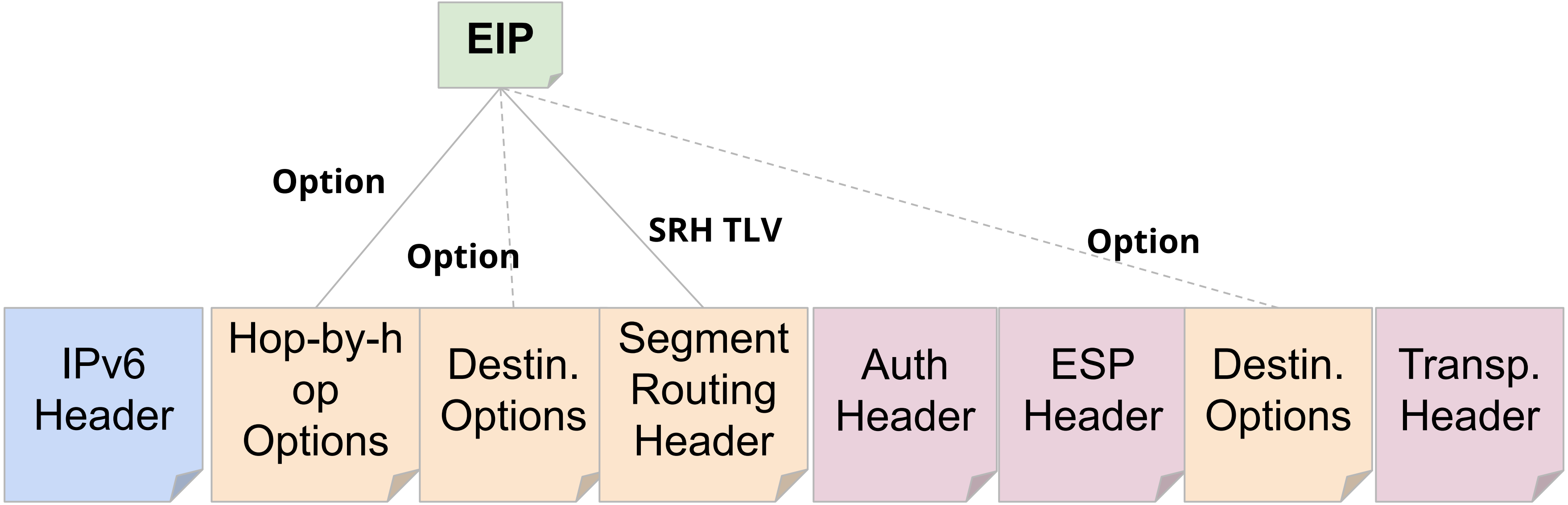
WHAT ? : definition of EIP (e.g. TLVs)

Specific content of EIP information will be defined considering the requirements of the different use cases.

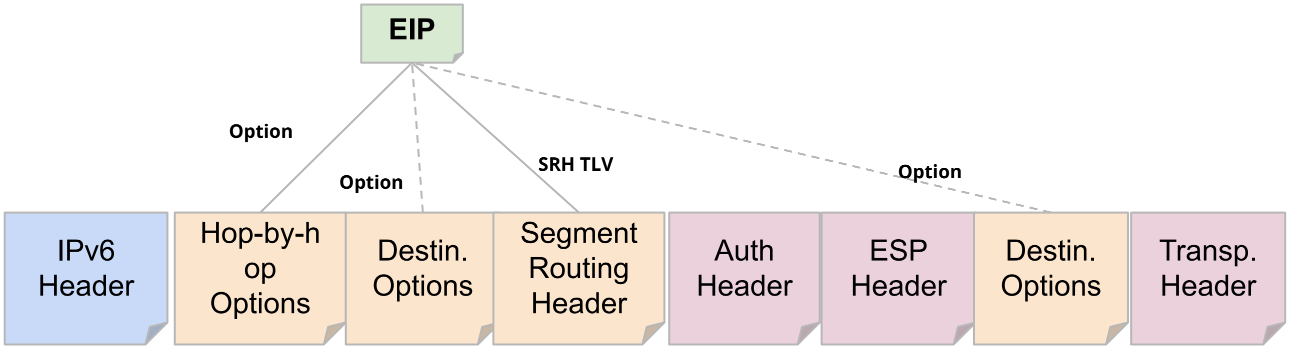
Some requirements will be common to different use cases
e.g. timestamping, authentication (HMAC), "contract" identification

A "library" of protocol components?

Carrying EIP information in IPv6… WHERE?



Carrying EIP information in IPv6… WHERE?



Find some initial brainstorming in this document:

https://eip-home.github.io/eip-headers/draft-eip-headers-definitions.txt

Agenda

- ➢ Background: network programmability
- ➢ Technical introduction to EIP (Extensible In-band Processing)
- ➢ EIP use cases
- ➢ Further technical discussion on EIP
- ➢ EIP prototype for Linux / eBPF
- ➢ Wrap up & feedback request

EIP use case : Deterministic Networking

RFC 8655 [1] and RFC 8938 [2] respectively specify the Architecture and data Plane Framework for Deterministic Networking.

RFC 8939 [3] 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.

[1] https://datatracker.ietf.org/doc/rfc8655/

[2] https://datatracker.ietf.org/doc/rfc8938/

[3] https://datatracker.ietf.org/doc/rfc8939/

EIP use case : Deterministic Networking

There are **known limitations** in the current IP data plane for DetNet (RFC 8939).

Adding explicit DetNet information in IP packets could simplify the implementation of Deterministic Networking in IP routers and hosts and provide additional features.

EIP is very well suited to this purpose!

EIP use case : Slicing

Slicing is discussed in IETF teas WG. The slicing framework is descrited in a WG draft [1]. Slicing needs (at least!) an identifier in the packets, proposals are under discussion in teas WG.

EIP can provide more than a simple slice identifier… it could be used to support some requirements and features identified in [1] and in other documents submitted to teas WG, complementing and extending the currently proposed solutions.

[1] https://datatracker.ietf.org/doc/draft-ietf-teas-ietf-network-slices/

Agenda

TOR VERGATA
UNIVERSITÀ DEGLI STUDI DI ROMA

cnit consorzio nazionale interuniversitario per le telecomunicazioni

- ➢ Background: network programmability
- ➢ Technical introduction to EIP (Extensible In-band Processing)
- ➢ EIP use cases
- ➢ Further technical discussion on EIP
- ➢ EIP prototype for Linux / eBPF
- ➢ Wrap up & feedback request

Wire speed processing

EIP can be designed to meet the "wire speed" requirement, which can require hardware support

Of course this may constrain the definition of some functionality, but it's possible to "keep it simple": EIP needs to be designed so that it can be "easily" implemented in HW

Security and privacy

As for security… we need to add (optional) authentication in EIP headers (e.g. HMAC as done for SRH in SRv6)

In general, our security concerns are aligned with those of SRv6 Network Programming model.

As for privacy, EIP does not reveal information about the applications that use it

Deployability

EIP capable nodes will be deployed in "limited domains". Such a domain will include network devices that are EIP capable and can include legacy devices that have been tested as being capable to "ignore" the EIP information.

The definition of EIP based services will clarify the requirements for the deployabily of a given EIP based service in the "limited domain" in terms of network device and host capabilities.

Agenda

TOR VERGATA
UNIVERSITÀ DEGLI STUDI DI ROMA

cnit consorzio nazionale interuniversitario per le telecomunicazioni

- ➢ Background: network programmability
- ➢ Technical introduction to EIP (Extensible In-band Processing)
- ➢ EIP use cases
- ➢ Further technical discussion on EIP
- ➢ EIP prototype for Linux / eBPF
- ➢ Wrap up & feedback request

EIP - Develop prototypes / "running code"

TOR VERGATA
UNIVERSITÀ DEGLI STUDI DI ROMA

cnit consorzio nazionale interuniversitario per le telecomunicazioni

We're building an open source prototype for EIP processing in Linux.

The packet processing is based on eBPF/XDP, hence it is very efficient.

The prototype relies on the HIKe / eCLAT framework for eBPF programming (a flexible and powerful framework to write eBPF programs using a high level language, see next slide!)

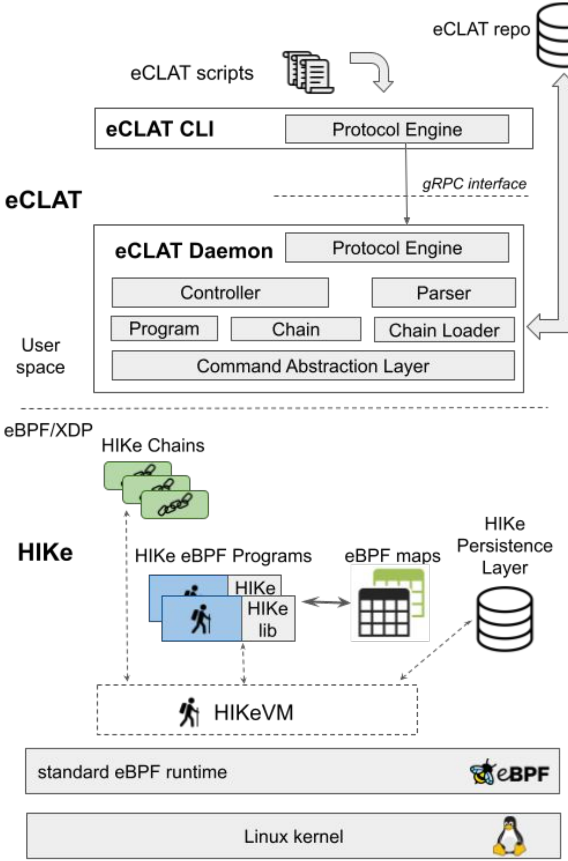
We could also work on a P4 prototype (anyone interested?)

HIKe / eCLAT framework for eBPF programming

eBPF Programming Made Easy with HIKe and eCLAT

https://tinyurl.com/hike-eclat-paper (paper)
https://tinyurl.com/hike-eclat-long (extended)

Technical documentation
https://hike-eclat.readthedocs.io/



Basic EIP prototype

The basic EIP prototype is a docker container which includes:

- the development environment for EIP based on HIKe/eCLAT
- a minimal testbed with a Traffic Generator (TG) node and a "System Under Test" (SUT) node that implements EIP functionality.

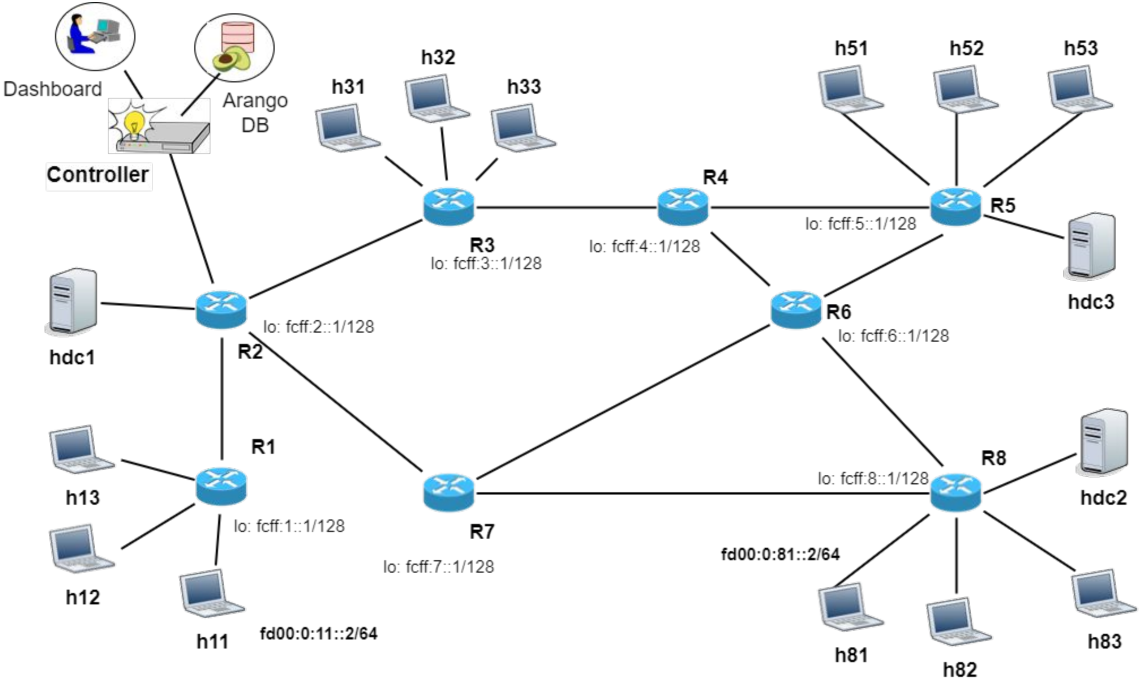
```
+------------------+      +------------------+
|        TG        |      |        SUT       |
|                  |      |                  |
|         enp6s0f0 +------+ enp6s0f0 <--- HIKe VM XDP loader
|                  |      |                  |
|                  |      |                  |
|         enp6s0f1 +------+ enp6s0f1         |
|                  |      |          + cl0  <-|- towards the collector
+------------------+      +----------|-------+
                                     |
                          +----------|------+
                          |          + veth0|
                          |                 |
                          |    COLLECTOR    |
                          +-----------------+
```

EIP advanced testbed

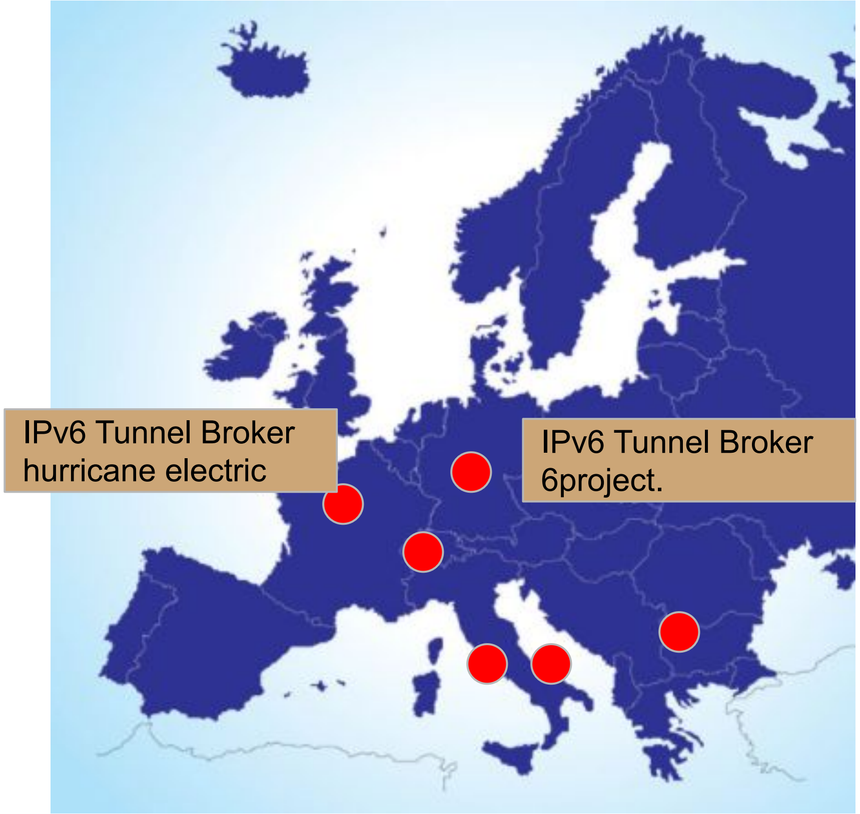
A Virtual Machine for Virtualbox and VMware hypervisors.

In the Virtual Machine it is possible to deploy an emulated network with 8 routers and 15 hosts (with mininet).

The 8 routers emulate an IPv6 backbone with dynamic routing (ISIS protocol).



Distributed field trial over public Internet



A set of IPv6 endpoints over different academic and commercial ISP networks

We can send IPv6+EIP packets in inter-domain real Internet scenarios

Agenda

TOR VERGATA
UNIVERSITÀ DEGLI STUDI DI ROMA

cnit consorzio nazionale interuniversitario per le telecomunicazioni

- ➢ Background: network programmability
- ➢ Technical introduction to EIP (Extensible In-band Processing)
- ➢ EIP use cases
- ➢ Further technical discussion on EIP
- ➢ EIP prototype for Linux / eBPF
- ➢ Wrap up & feedback request

Wrap up

"EIP - Extensible In band Processing" is a proposal for the evolution of network layer (IPv6) capabilities

EIP can support the needs of research initiatives on IP/Internet evolution (helping to turn them into reality)

Next steps

We've set up an informal Interest Group on Extensible In-band Processing, in order build a community (operators, vendors, academia)

Get feedback from key researchers, industry experts, standardization experts

Write scientific papers => Present in workshops and conferences

Develop prototypes / "running code"

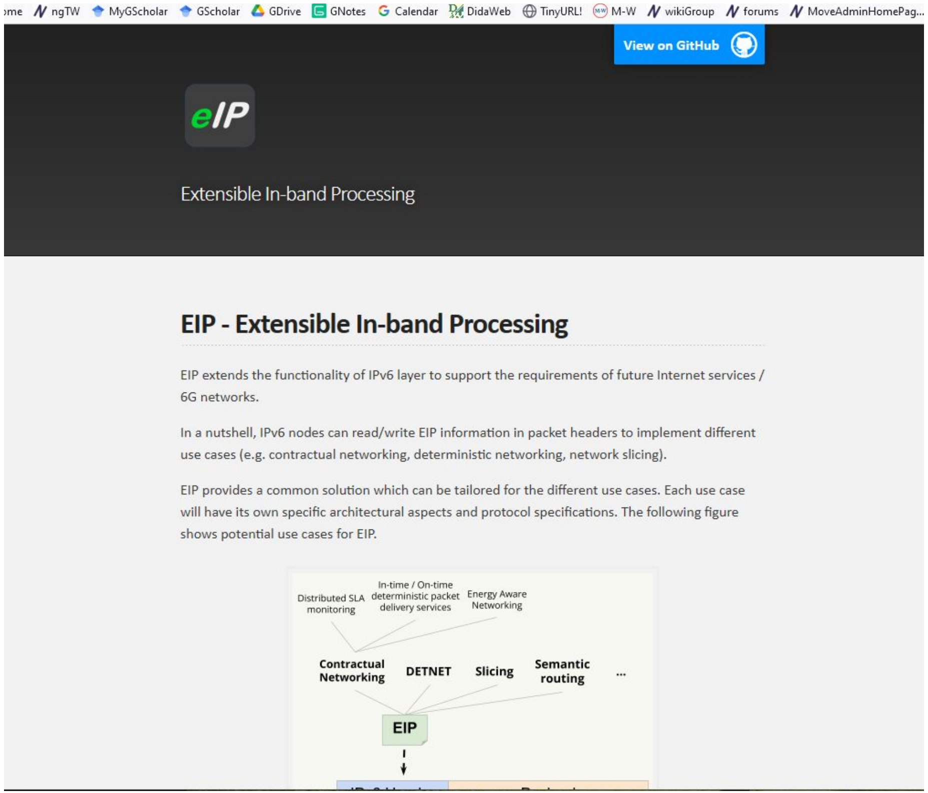
Present and discuss in IETF (?)

EIP - Interest Group

We've set up an informal Interest Group on Extensible In-band Processing, in order build a community (operators, vendors, academia)

https://eip-home.github.io/eip/

EIP - Interest Group



Home page:

https://eip-home.github.io/eip/

Wiki:

https://github.com/eip-home/eip/wiki

Mailing list:

eip@postino.cnit.it

EIP - Interest Group

We've set up an informal Interest Group on Extensible In-band Processing, in order build a community (operators, vendors, academia)

https://eip-home.github.io/eip/

Are you interested in participating to the informal Interest Group?

Please join the mailing list http://postino.cnit.it/cgi-bin/mailman/listinfo/eip

EIP - Feedback?

Get feedback from key researchers, industry experts, standardization experts

Are you interested in some of the presented use cases of EIP (contractual networking, deterministic networking, slicing)?
Do you have feedback on these use cases?

Are you interested in other use cases? Can you suggest them?

EIP - Feedback?

Get feedback from key researchers, industry experts, standardization experts

Do you have feedback/suggestions on the technical approach?

Any other general comment or feedback?

Thank you for your attention!
Any question?

stefano.salsano@uniroma2.it

My previous talks

Network programmability (r)evolution of IP

talk @ IETF 111 - https://tinyurl.com/ip-revol-ietf-111

Extensible In-band Processing (EIP)

talk @ IETF 112 - https://tinyurl.com/eip-ietf-112

Bibliography

Extensible Internet (EI) proposal

H. Balakrishnan et al., “Revitalizing the public internet by making it extensible,” SIGCOMM Comput. Commun. Rev., Apr. 2021

There is now a significant and growing functional gap between the public Internet, whose basic architecture has remained unchanged for several decades, and a new generation of more sophisticated private networks. To address this increasing divergence of functionality and overcome the Internet's architectural stagnation, we argue for the creation of an Extensible Internet (EI) that supports in-network services that go beyond best-effort packet delivery. To gain experience with this approach, we hope to soon deploy both an experimental version (for researchers) and a prototype version (for early adopters) of EI. In the longer term, making the Internet extensible will require a community to initiate and oversee the effort; this paper is the first step in creating such a community.

Balakrishnan H, Banerjee S, Cidon I, Culler D, Estrin D, Katz-Bassett E, Krishnamurthy A, McCauley M, McKeown N, Panda A, Ratnasamy S, Rexford J, Schapira M, Shenker S, Stoica I, Tennenhouse D, Vahdat A, Zegura E

Dynamic Packet State

I. Stoica and H. Zhang, "Providing Guaranteed Services Without Per Flow Management", Proceedings of SIGCOMM'99, Boston, MA, September 1999

Existing approaches for providing guaranteed services require routers to manage per flow states and perform per flow operations. Such a stateful network architecture require each router to maintain and manage per-flow state on the control path, and to perform per-flow classification, scheduling, and buffer management on the data path. This makes a stateful architecture less scalable and robust than stateless network architectures like the original IP and the recently proposed Diffserv. On the other hand, services provided with current stateless solutions, Diffserv included, have lower flexibility, utilization, and/or assurance level as compared to the services that can be provided with per flow mechanisms.

In this work, we propose techniques that do not require per flow management (either control or data planes) at core routers, but can implement guaranteed services with levels of flexibility, utilization, and assurance similar to those that can be provided with per flow mechanisms. In this way we can simultaneously achieve high quality of service, high scalability and robustness. The key technique we use is called Dynamic Packet State (DPS), which provides a lightweight and robust mechanism for routers to coordinate actions and implement distributed algorithms.

X. Zhang, H.-C. Hsiao, G. Hasker, H. Chan, A. Perrig, and D. G. Andersen, "SCION: Scalability, Control, and Isolation on Next-Generation Networks," in 2011 IEEE Symposium on Security and Privacy, May 2011, pp. 212–227, doi: 10.1109/SP.2011.45.

We present the first Internet architecture designed to provide route control, failure isolation, and explicit trust information for end-to-end communications. SCION separates ASes into groups of independent routing sub-planes, called trust domains, which then interconnect to form complete routes. Trust domains provide natural isolation of routing failures and human misconfiguration, give endpoints strong control for both inbound and outbound traffic, provide meaningful and enforceable trust, and enable scalable routing updates with high path freshness. As a result, our architecture provides strong resilience and security properties as an intrinsic consequence of good design principles, avoiding piecemeal add-on protocols as security patches. Meanwhile, SCION only assumes that a few top-tier ISPs in the trust domain are trusted for providing reliable end-to-end communications, thus achieving a small Trusted Computing Base. Both our security analysis and evaluation results show that SCION naturally prevents numerous attacks and provides a high level of resilience, scalability, control, and isolation.

A spot market for interdomain connectivity

K. L. Calvert, J. Griffioen, A. Nagurney, and T. Wolf, "A vision for a spot market for interdomain connectivity," in 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), Jul. 2019

In this paper we consider an alternative possibility for routing money in the Internet ecosystem based on a spot market for interconnection service that operates alongside, or in addition to, the traditional contract service model. Recent work by others showed that under certain assumptions, enabling transit providers (i.e., networks that carry packets between networks) to sell their excess capacity on a best-effort basis improves both provider profit and consumer surplus. While prior work only focused on pricing strategies, we explore the technical feasibility of such a market in this paper. We consider what is needed to make such a spot market possible and focus on the interaction between technical and economic considerations. In particular, we describe two approaches that demonstrate how economic software defined exchanges (ESDXs) can be used as trusted intermediaries to tie the forwarding service to the flow of money.

Full version of the presentation

Agenda

TOR VERGATA
UNIVERSITÀ DEGLI STUDI DI ROMA

cnit consorzio nazionale interuniversitario per le telecomunicazioni

- ➢ Background: network programmability
- ➢ Technical introduction to EIP (Extensible In-band Processing)
- ➢ EIP use cases
- ➢ Further technical discussion on EIP
- ➢ EIP prototype for Linux / eBPF
- ➢ Wrap up & feedback request

Network programmability (r)evolution of IP

TOR VERGATA
UNIVERSITÀ DEGLI STUDI DI ROMA

cnit consorzio nazionale interuniversitario per le telecomunicazioni

→ **IP Nodes can be more "easily" reprogrammed!**

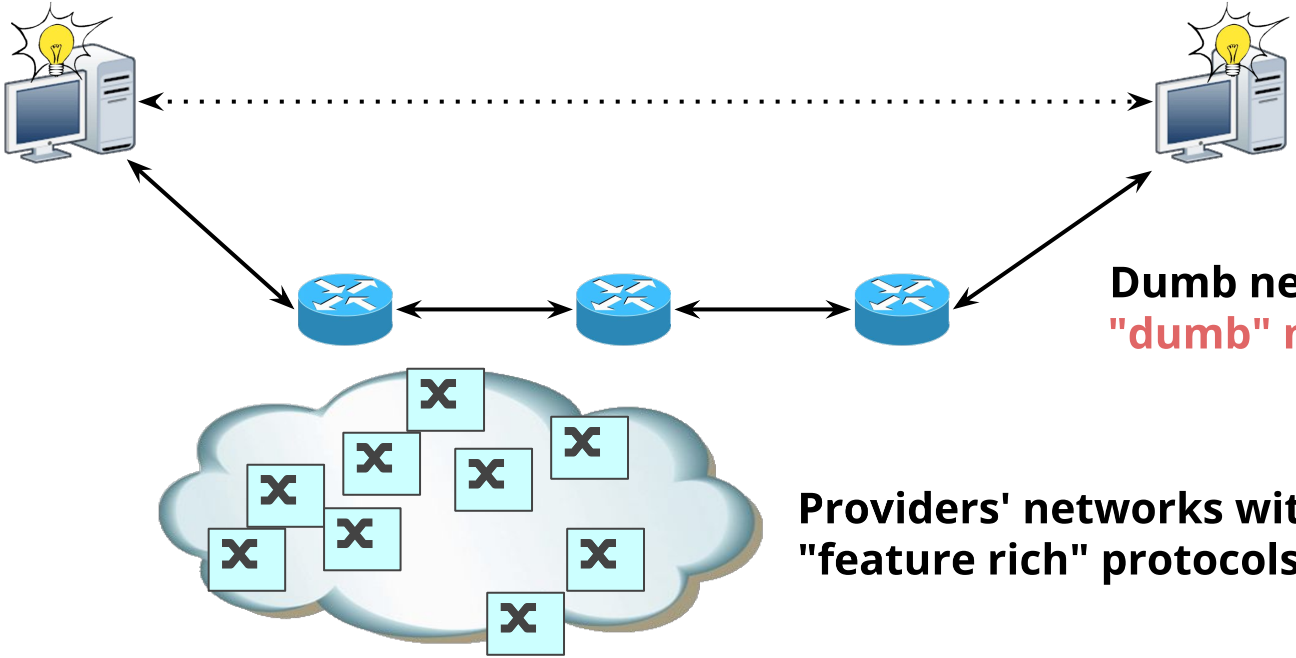
→ **A more complex network layer is now feasible, see for example SRv6 Network Programming model**

The end-to-end principle made up the Internet… maybe we can go further?

(see my talk @ IETF 111 - https://tinyurl.com/ip-revol-ietf-111)

We can move from here...

Smart hosts

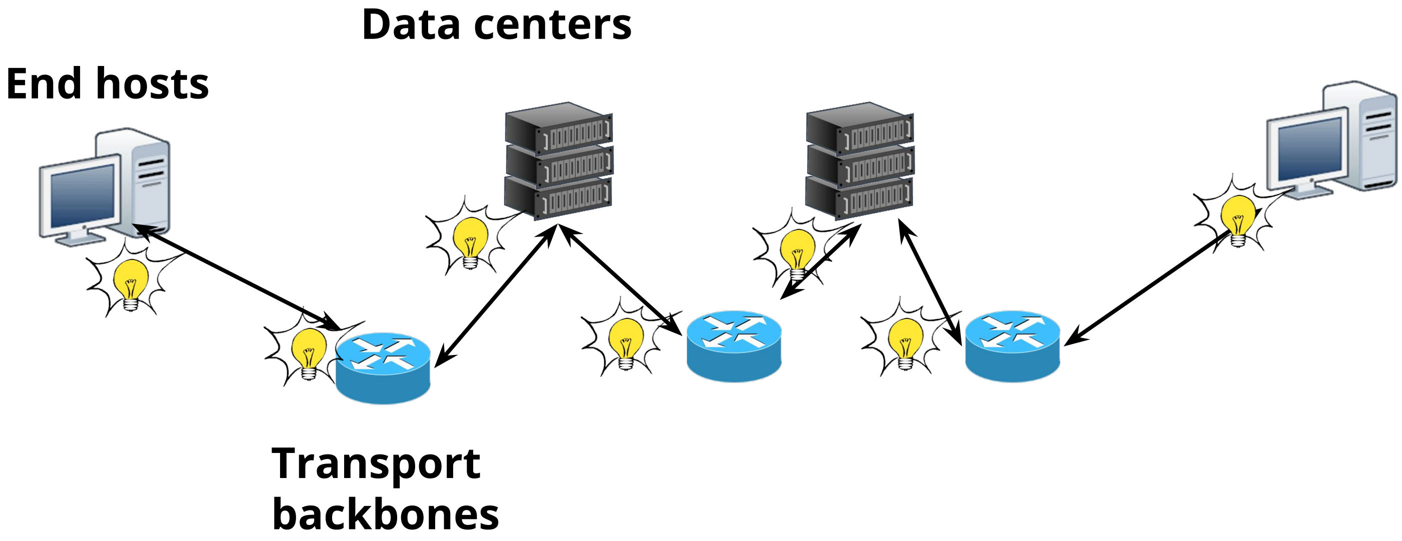


Dumb network nodes,
"dumb" network layer

Providers' networks with specific "feature rich" protocols (e.g. ATM, MPLS)

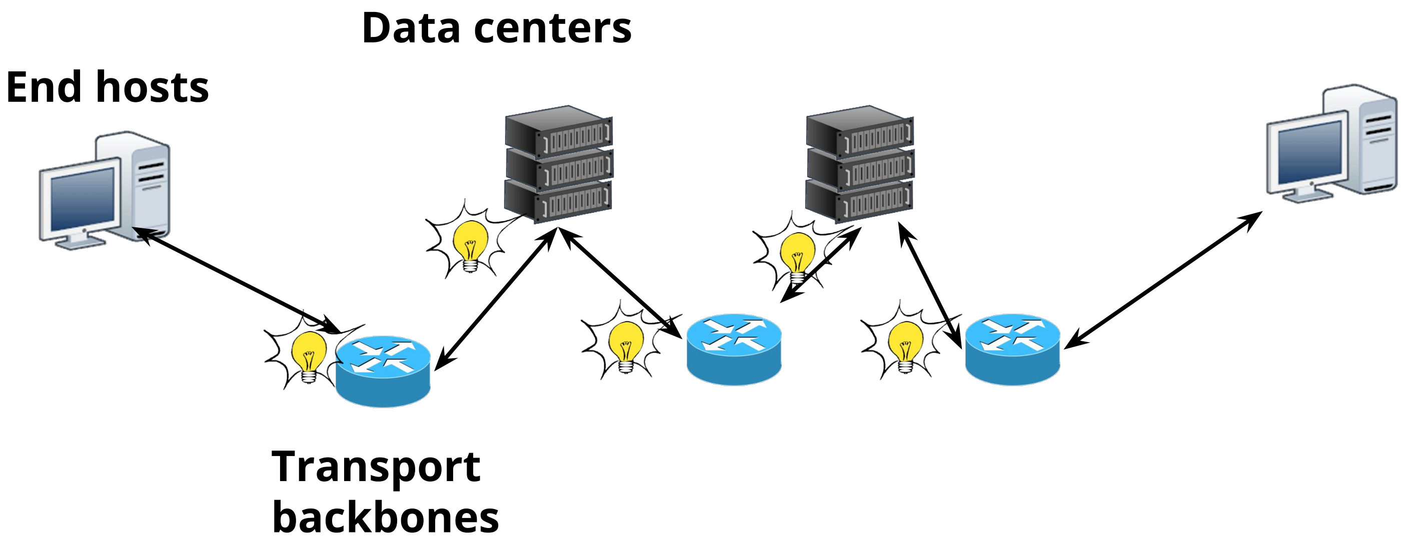
 Intelligence

… to here



Unified (host/routers) and feature-rich network layer !

… and to here



<u>In some scenarios</u>… routers can be "smarter" than hosts (at network layer) !

Agenda

TOR VERGATA
UNIVERSITÀ DEGLI STUDI DI ROMA

cnit consorzio nazionale interuniversitario per le telecomunicazioni

- ➢ Background: network programmability
- ➢ Technical introduction to EIP (Extensible In-band Processing)
- ➢ EIP use cases
- ➢ Further technical discussion on EIP
- ➢ EIP prototype for Linux / eBPF
- ➢ Wrap up & feedback request

Extensible In-band Processing (EIP)

Definition of generic and extensible mechanisms to carry extensible information (EIP) in IPv6 packets headers, like

- monitoring information to be collected in transit
- parameters to be used for packet processing

Nodes can **read and write** this "extensible" EIP information.

Nodes can take packet processing decisions depending on the EIP information inserted in the packet by the source and by previous intermediate nodes.

Extensible In-band Processing (EIP)

TOR VERGATA
UNIVERSITÀ DEGLI STUDI DI ROMA

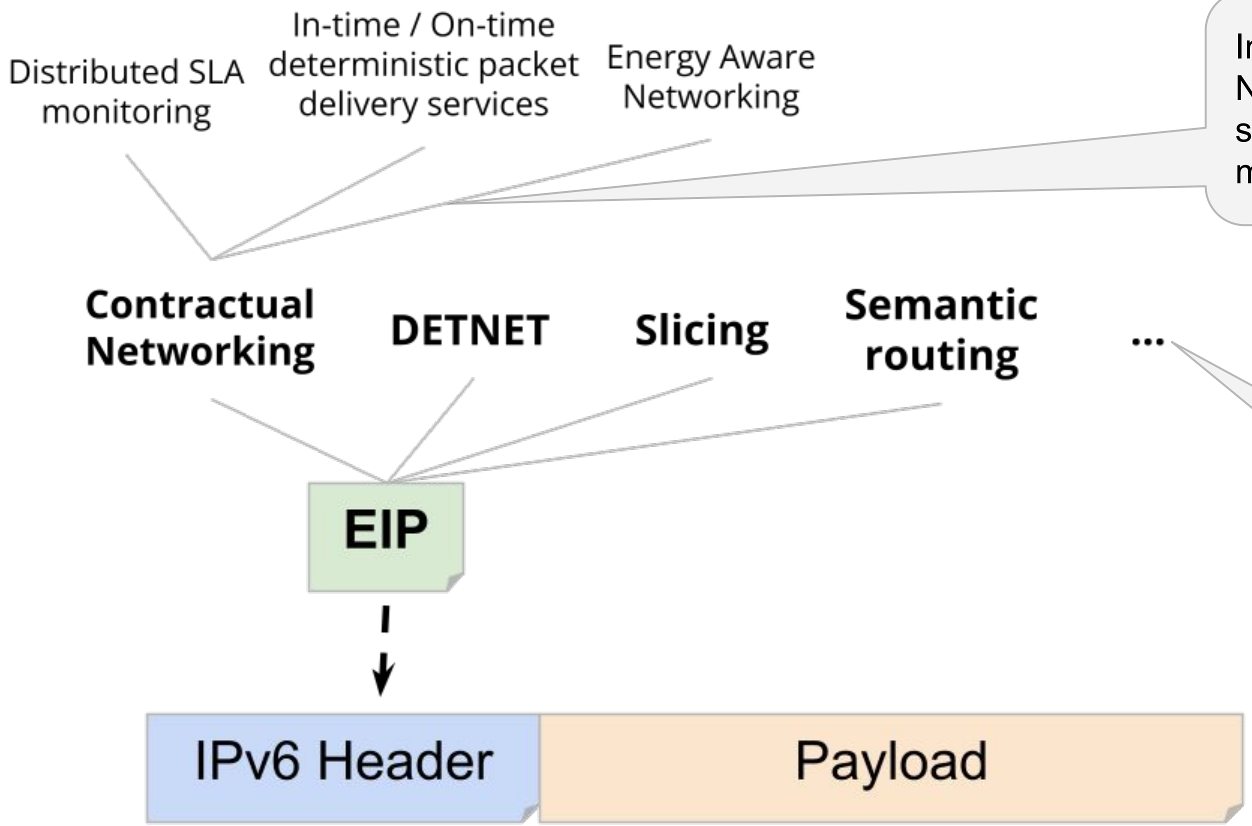
consorzio nazionale
interuniversitario
per le telecomunicazioni

Note that, IPv6 header is already "extensible" by definition, there are "Extension Headers" and "Options" that can be defined inside some Extension Headers… but there are practical barriers in this extension process: the layer 3 functionality is "ossified".

We will collect requirements from a few use cases, then define EIP as "unified" extension approach, overcoming the "ossification barriers".

Open Source prototypes of EIP will also support the process.

EIP can support several use cases



In turn, the Contractual Networking use case can be seen as a "container" for many specific services

This list of use cases is an example, we are looking for feedback to focus on the most interesting use cases

Applicability

We do not need/want to apply this paradigm everywhere!

New solutions can work in "limited domains" (a.k.a. "controlled domains")

See "Limited Domains and Internet Protocols" - RFC 8799 (Independent submission)

Clarification on "network programmability"

We are NOT proposing a new revolutionary approach like what "active networks" tried to do years ago.

Our interpretation of "Network programmability" is compatible with the already standardized concepts of RFC 8986 "Segment Routing over IPv6 (SRv6) **Network Programming**": SRv6 nodes implement rather complex **behaviors** at Layer 3, these SRv6 behaviors can be combined (now) and new behaviors can be defined and standardized (in the future).

Therefore, SRv6 technology represents a concrete example of how complex functionality can be defined at Layer 3 (in limited domains, by the way).

EIP Use case : Contractual networking

TOR VERGATA
UNIVERSITÀ DEGLI STUDI DI ROMA

cnit consorzio nazionale interuniversitario per le telecomunicazioni

Contractual networking is an application scenario for EIP

What is a contract ? A **complex and distributed services** offered by a network.

The complex and distributed service is composed by several components like forwarding/metering/filtering/accounting/ (a composite service can include one or more of these "components")...

Contractual networking (my view)

TOR VERGATA
UNIVERSITÀ DEGLI STUDI DI ROMA

cnit consorzio nazionale interuniversitario per le telecomunicazioni

What is a contract ? A **complex and distributed service** (composed by features like forwarding/metering/filtering/accounting/...) offered by a network provider

Examples of services in contractual networking:

- Distributed SLA monitoring
- In-time / On-time deterministic packet delivery services
- Energy aware forwarding - e.g. real time monitoring of energy consumption to ensure that it does not exceed a negotiated limit

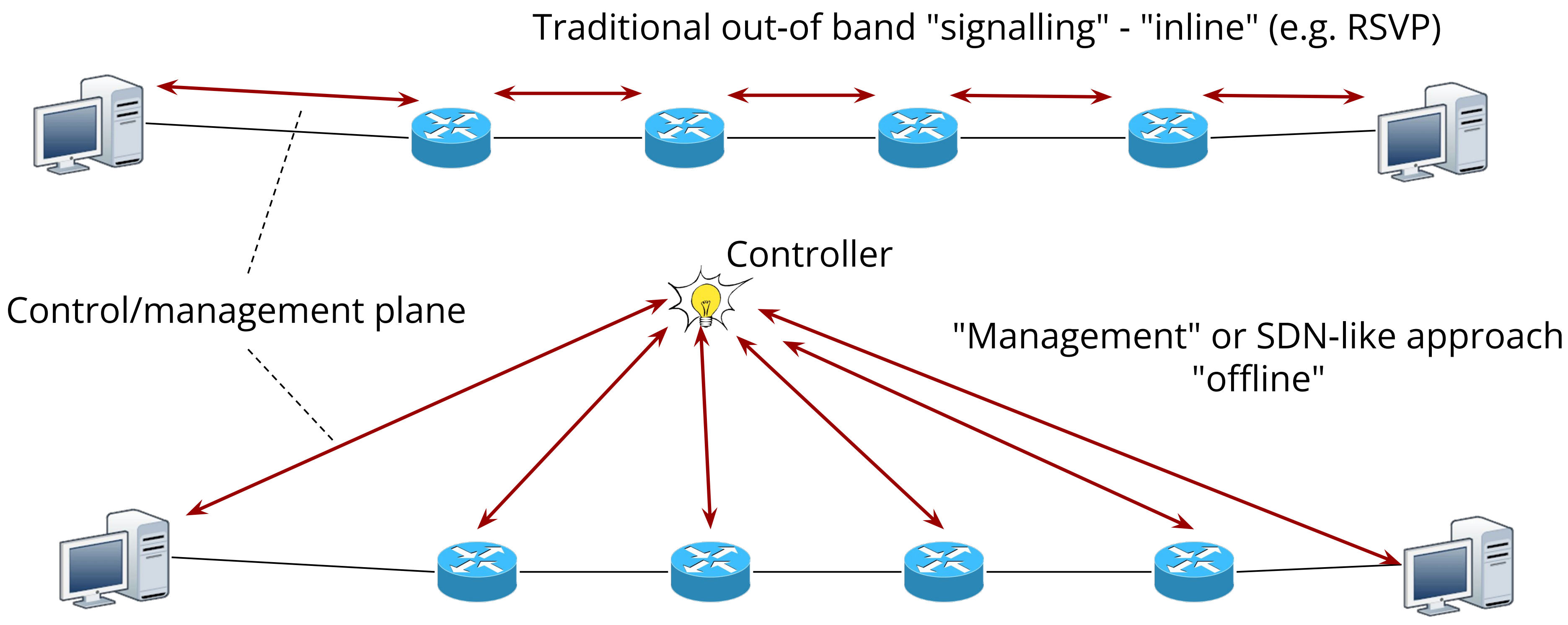
Contractual networking (my view)

What is a contract ? A **complex and distributed service** (composed by features like forwarding/metering/filtering/accounting/...) offered by a network provider

Two phases:

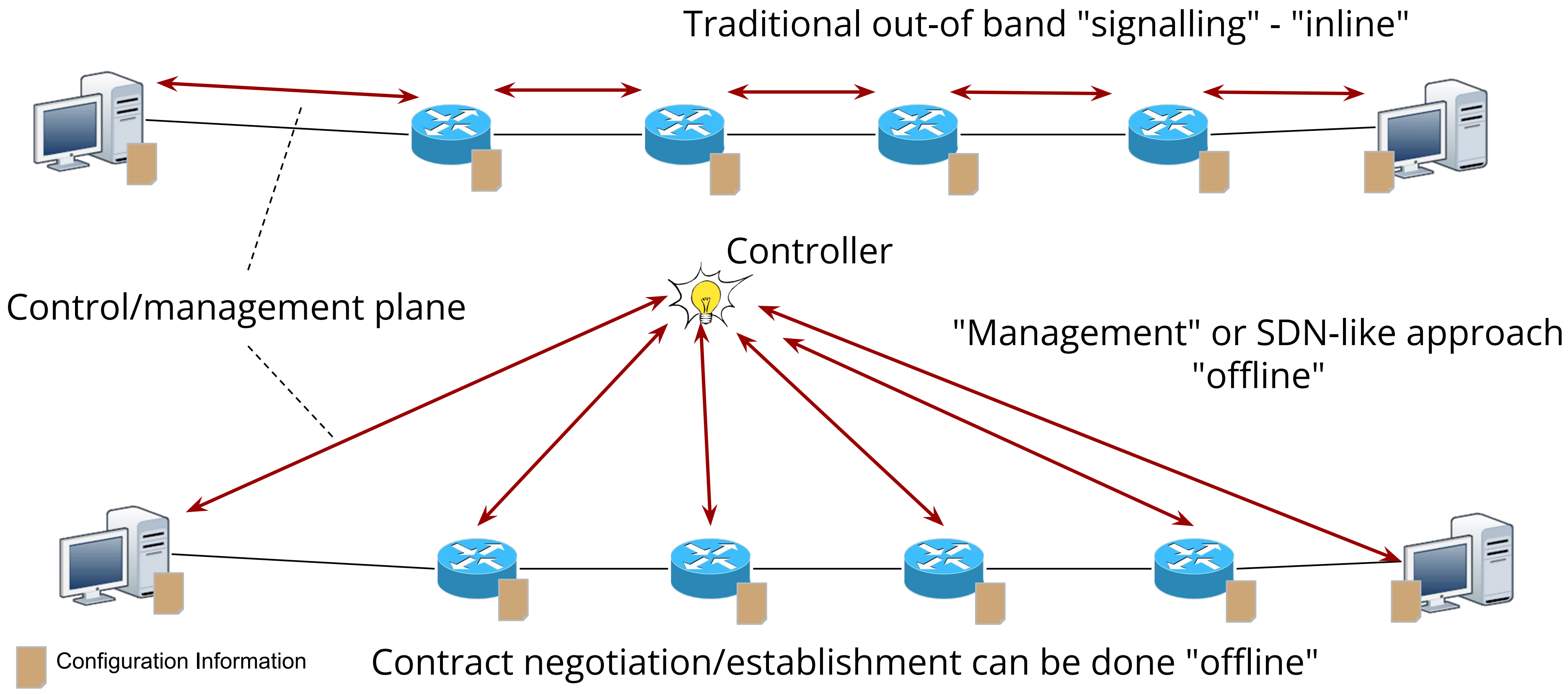
1) Contracts are "negotiated" (Control plane)
2) Contracts are "enforced", i.e. packets are processed according to the negotiated contract (Data plane)

1. Contract negotiation / establishment



Contract negotiation/establishment can be done "offline"

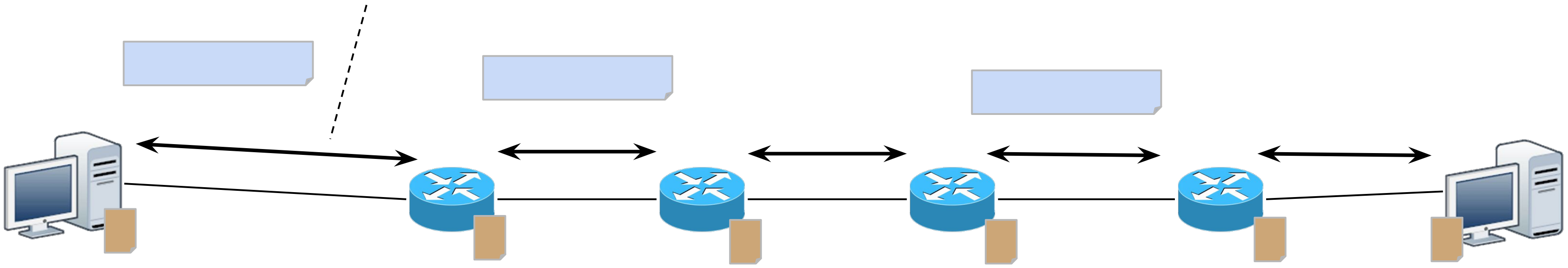
1. Contract negotiation / establishment



Contract negotiation/establishment can be done "offline"

2. Contract enforcement (data plane)

Data plane (IPv6 packets)

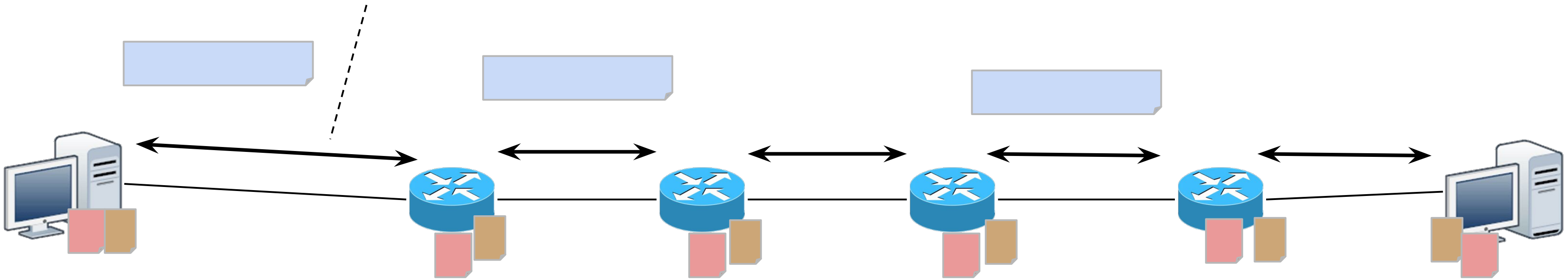


Packet processing in the nodes is based on:

- Configuration information (at contract establishment)

2. Contract enforcement (data plane)

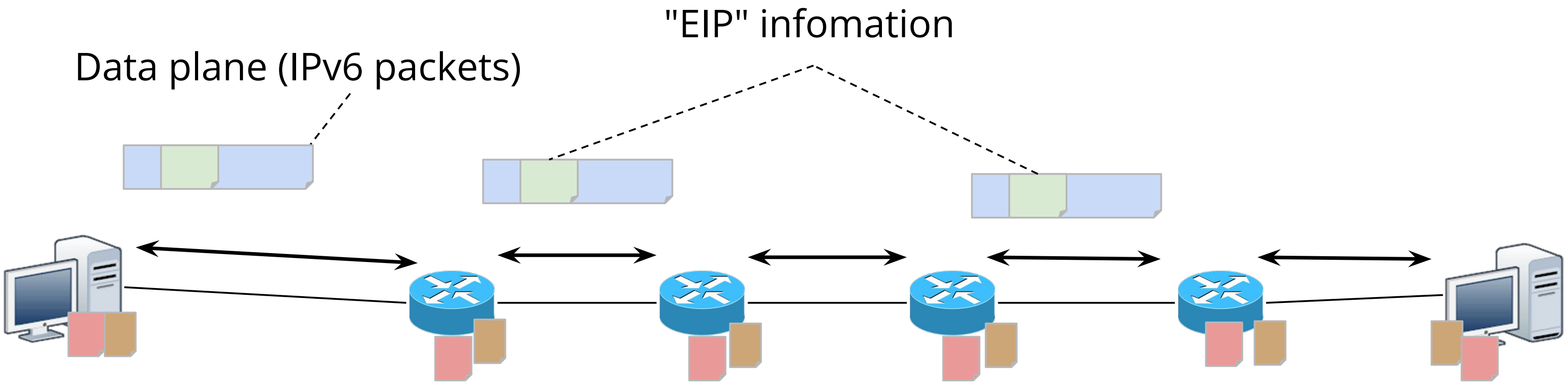
Data plane (IPv6 packets)



Packet processing in the nodes is based on:

- Configuration information (at contract establishment)
- Dynamic state information in nodes (e.g. counters, meters)

2. Contract enforcement (data plane)



Packet processing in the nodes is based on:

- Configuration information (at contract establishment)
- Dynamic state information in nodes (e.g. counters, meters)
- EIP Information in the packet (read/write)

Scope of EIP definition

The definition of EIP will focus on:

data plane (= contract enforcement/monitoring phase)

and will NOT focus on:

control plane (= contract definition and negotiation phase)

EIP: packet processing/forwarding

The processing / forwarding of packets in EIP can be based on the combination of the following information:

- information inserted by the source (or edge node), per flow and/or per packet
- information **dynamically** inserted by other layer 3 nodes in the path, based on the state of the node and on the info in the packet, per flow and/or per packet
- "regular" IP layer information (e.g. routing information)

The information inserted by the source and by intermediate nodes can include node programmability instructions such as match-action instructions

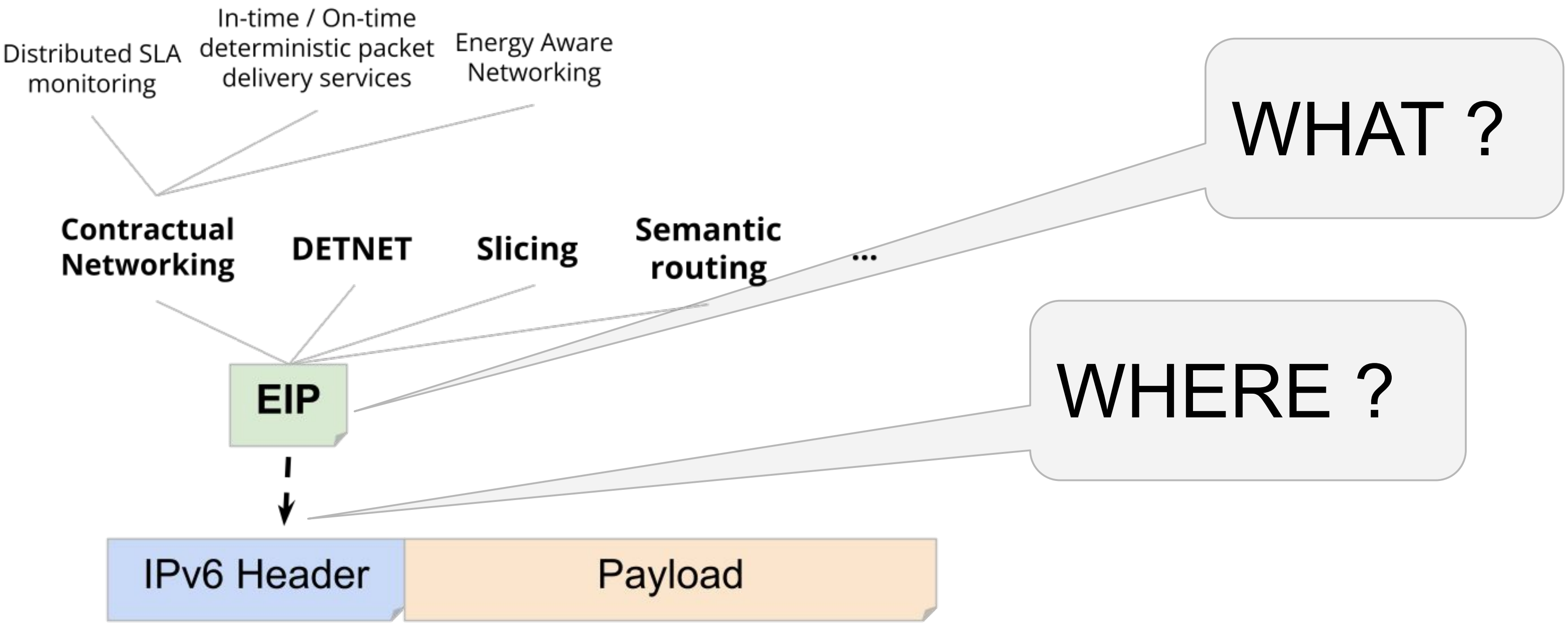
=> Network-wide dynamic and stateful packet processing

Extensible In-band Processing (EIP)

Carrying state information in packets can simplify packet processing in network nodes, reducing the requirements for processing power and memory (RAM and/or TCAM) in the nodes.

Carrying state information in packets is **more scalable** than keeping state information in network nodes!

Definition of EIP



WHAT ? : definition of EIP (e.g. TLVs)

Specific content of EIP information will be defined considering the requirements of the different use cases.

Some requirements will be common to different use cases
e.g. timestamping, authentication (HMAC), "contract" identification

A "library" of protocol components?

WHAT ? : definition of EIP (e.g. TLVs)

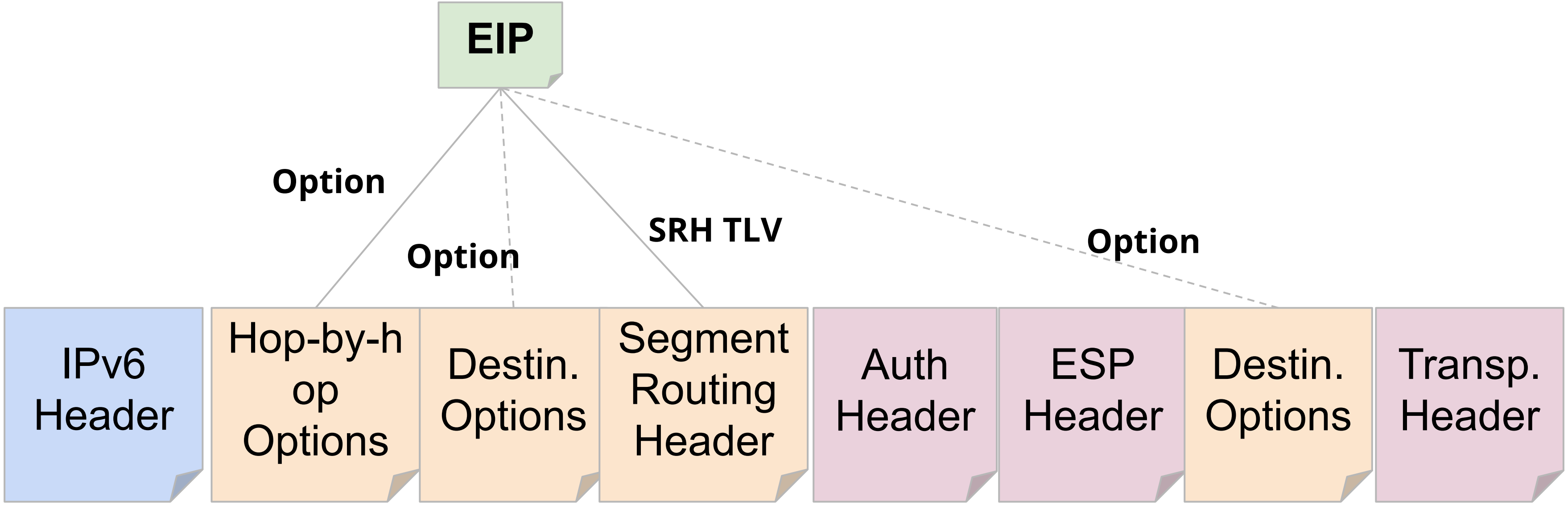
For example, consider the three sub-cases of contractual networking

Distributed SLAs monitoring (packet classification is simplified, can be based on EIP information).

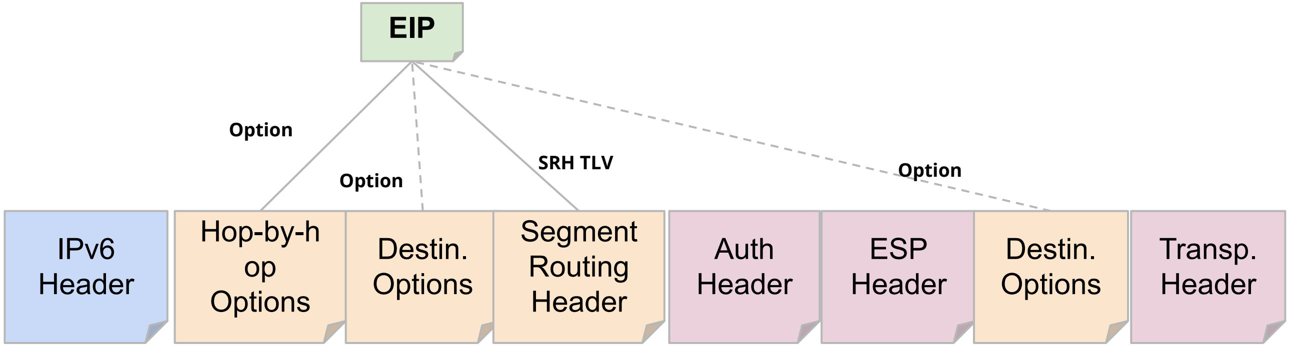
Accounting for energy consumption while forwarding the packets (energy-aware forwarding, decisions can be based on EIP information)

Timing information for in-time / on-time packet delivery

Carrying EIP information in IPv6… WHERE?



Carrying EIP information in IPv6… WHERE?



Find some initial brainstorming in this document:

https://eip-home.github.io/eip-headers/draft-eip-headers-definitions.txt

Agenda

TOR VERGATA
UNIVERSITÀ DEGLI STUDI DI ROMA

cnit consorzio nazionale interuniversitario per le telecomunicazioni

- ➢ Background: network programmability
- ➢ Technical introduction to EIP (Extensible In-band Processing)
- ➢ EIP use cases
- ➢ Further technical discussion on EIP
- ➢ EIP prototype for Linux / eBPF
- ➢ Wrap up & feedback request

EIP use case : Deterministic Networking

RFC 8655 [1] and RFC 8938 [2] respectively specify the Architecture and data Plane Framework for Deterministic Networking.

RFC 8939 [3] 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.

[1] https://datatracker.ietf.org/doc/rfc8655/

[2] https://datatracker.ietf.org/doc/rfc8938/

[3] https://datatracker.ietf.org/doc/rfc8939/

EIP use case : Deterministic Networking

There are **known limitations** in the current IP data plane for DetNet (RFC 8939).

The IP data plane for DetNet as specified in RFC 8939, 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.

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 use case : Deterministic Networking

There are **known limitations** in the current IP data plane for DetNet (RFC 8939).

Adding explicit DetNet information in IP packets could simplify the implementation of Deterministic Networking in IP routers and hosts and provide additional features.

EIP is very well suited to this purpose!

EIP use case : Slicing

Slicing is discussed in IETF teas WG. The slicing framework is descrited in a WG draft [1]. Slicing needs (at least!) an identifier in the packets, proposals are under discussion in teas WG.

EIP can provide more than a simple slice identifier… it could be used to support some requirements and features identified in [1] and in other documents submitted to teas WG, complementing and extending the currently proposed solutions.

[1] https://datatracker.ietf.org/doc/draft-ietf-teas-ietf-network-slices/

Agenda

TOR VERGATA
UNIVERSITÀ DEGLI STUDI DI ROMA

cnit consorzio nazionale interuniversitario per le telecomunicazioni

- ➢ Background: network programmability
- ➢ Technical introduction to EIP (Extensible In-band Processing)
- ➢ EIP use cases
- ➢ Further technical discussion on EIP
- ➢ EIP prototype for Linux / eBPF
- ➢ Wrap up & feedback request

EIP vs. INT

TOR VERGATA
UNIVERSITÀ DEGLI STUDI DI ROMA

cnit consorzio nazionale interuniversitario per le telecomunicazioni

In-band Network Telemetry (i.e. INT with P4 by ONF [1], In-situ OAM by IETF IPPM [2], IFIT In-situ Flow Information Telemetry [3])

- In-band telemetry is focused on collecting telemetry information, EIP is more general. EIP can support In-band Telemetry

[1] https://p4.org/p4-spec/docs/INT_v2_1.pdf

[2] https://datatracker.ietf.org/doc/html/draft-ietf-ippm-ioam-data-17

[3] https://datatracker.ietf.org/doc/html/draft-song-opsawg-ifit-framework-16

Wire speed processing

EIP can be designed to meet the "wire speed" requirement, which can require hardware support

Of course this may constrain the definition of some functionality, but it's possible to "keep it simple": EIP needs to be designed so that it can be "easily" implemented in HW

Security and privacy

As for security… we need to add (optional) authentication in EIP headers (e.g. HMAC as done for SRH in SRv6)

In general, our security concerns are aligned with those of SRv6 Network Programming model.

As for privacy, EIP does not reveal information about the applications that use it

Deployability

EIP capable nodes will be deployed in "limited domains". Such a domain will include network devices that are EIP capable and can include legacy devices that have been tested as being capable to "ignore" the EIP information.

The definition of EIP based services will clarify the requirements for the deployabily of a given EIP based service in the "limited domain" in terms of network device and host capabilities.

Per-flow vs. per-packet

TOR VERGATA
UNIVERSITÀ DEGLI STUDI DI ROMA

cnit consorzio nazionale interuniversitario per le telecomunicazioni

Some services "simply" require the identification of a flow (or of a flow aggregate): all packets of the flow (or flow aggregate) will carry the same information. Often in these cases, one can choose :

1) to explicitly carry a "flow identifier" in the packet (EIP can help)
2) to "reclassify" the packet when needed by putting some classification state in a node.

Other services require information that is different on a packet-by-packet base and sometimes needs to be modified on the fly. The simplest example is to collect monitoring information along the path. Hence, per-packet information is needed to support all potential services.

EIP info in the packet vs. state info in nodes

Carrying explicit EIP information in the packet reduces the state information in the nodes and can simplify the classification operations that need to be performed by forwarding nodes.

This does not exclude the need to keep state information in the node, a mix of the two approaches is needed based on the usage scenario.

Agenda

TOR VERGATA
UNIVERSITÀ DEGLI STUDI DI ROMA

cnit consorzio nazionale interuniversitario per le telecomunicazioni

- ➢ Background: network programmability
- ➢ Technical introduction to EIP (Extensible In-band Processing)
- ➢ EIP use cases
- ➢ Further technical discussion on EIP
- ➢ EIP prototype for Linux / eBPF
- ➢ Wrap up & feedback request

EIP - Develop prototypes / "running code"

We're building an open source prototype for EIP processing in Linux.

The packet processing is based on eBPF/XDP, hence it is very efficient.

The prototype relies on the HIKe / eCLAT framework for eBPF programming (a flexible and powerful framework to write eBPF programs using a high level language, see next slide!)

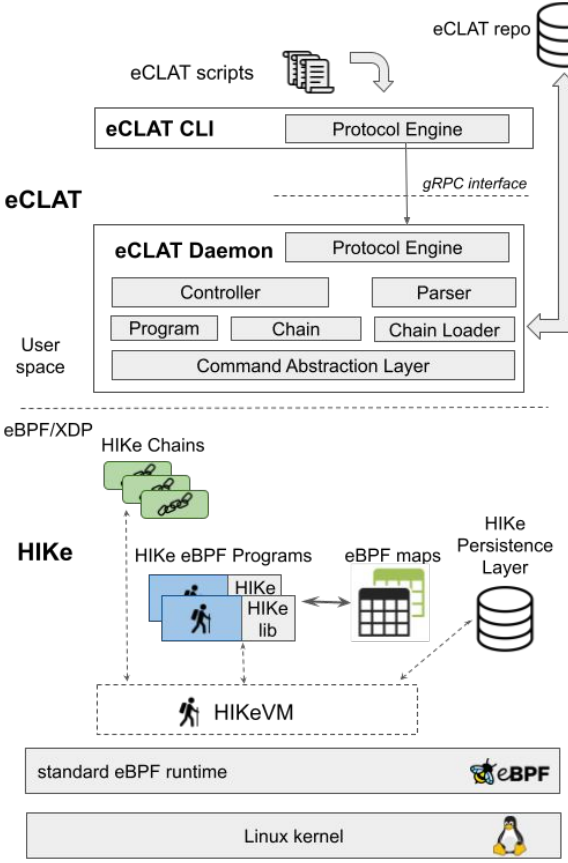
We could also work on a P4 prototype (anyone interested?)

HIKe / eCLAT framework for eBPF programming

eBPF Programming Made Easy with HIKe and eCLAT

https://tinyurl.com/hike-eclat-paper (paper)
https://tinyurl.com/hike-eclat-long (extended)

Technical documentation
https://hike-eclat.readthedocs.io/



Basic EIP prototype

The basic EIP prototype is a docker container which includes:

- the development environment for EIP based on HIKe/eCLAT
- a minimal testbed with a Traffic Generator (TG) node and a "System Under Test" (SUT) node that implements EIP functionality.

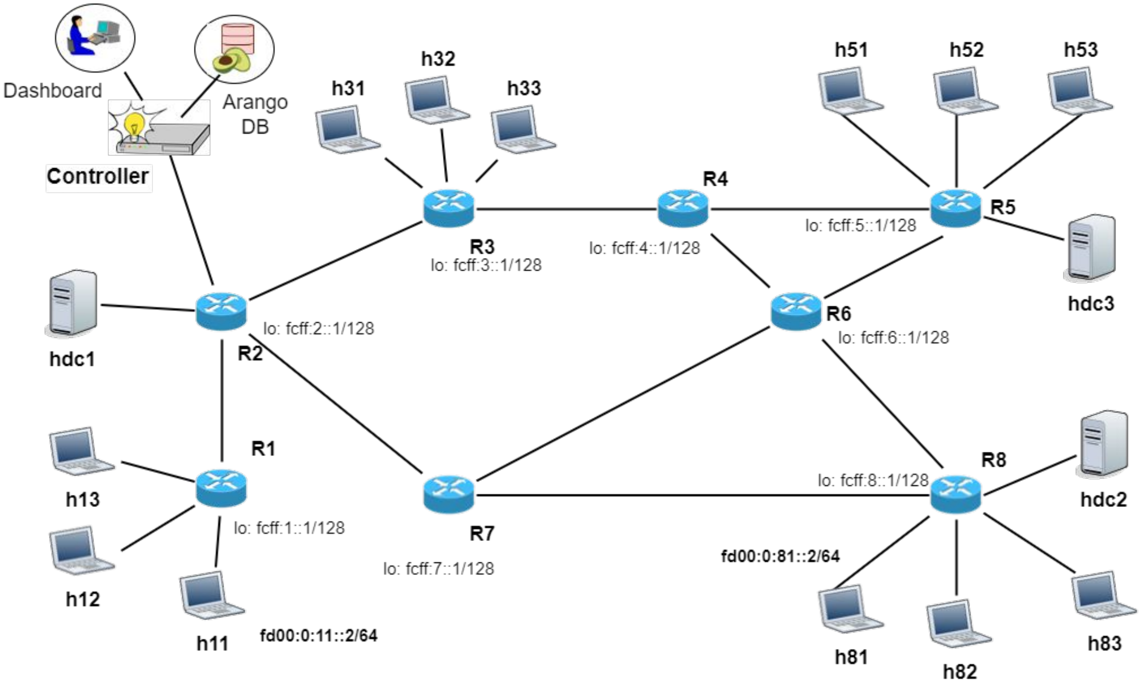
```
+------------------+       +------------------+
|        TG        |       |       SUT        |
|                  |       |                  |
|         enp6s0f0 +-------+ enp6s0f0 <--- HIKe VM XDP loader
|                  |       |                  |
|                  |       |                  |
|         enp6s0f1 +-------+ enp6s0f1         |
|                  |       |         + cl0  <-|- towards the collector
+------------------+       +---------|--------+
                                     |
                           +---------|------+
                           |         + veth0|
                           |                |
                           |   COLLECTOR    |
                           +----------------+
```

EIP advanced testbed

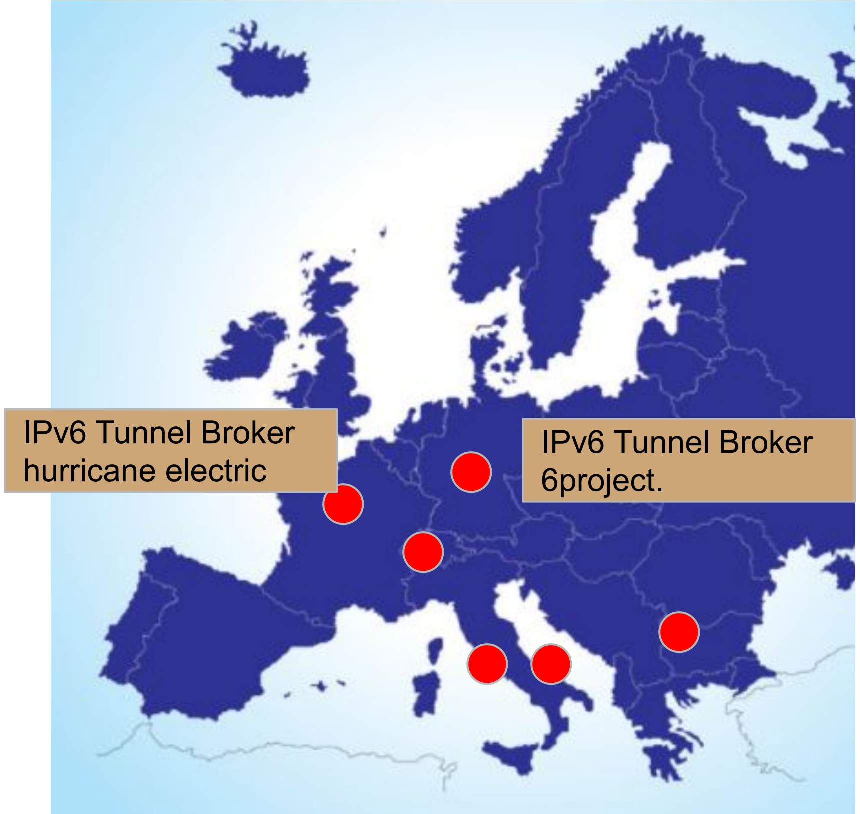
A Virtual Machine for Virtualbox and VMware hypervisors.

In the Virtual Machine it is possible to deploy an emulated network with 8 routers and 15 hosts (with mininet).

The 8 routers emulate an IPv6 backbone with dynamic routing (ISIS protocol).



Distributed field trial over public Internet



A set of IPv6 endpoints over different academic and commercial ISP networks

We can send IPv6+EIP packets in inter-domain real Internet scenarios

Agenda

TOR VERGATA
UNIVERSITÀ DEGLI STUDI DI ROMA

cnit consorzio nazionale interuniversitario per le telecomunicazioni

- Background: network programmability
- Technical introduction to EIP (Extensible In-band Processing)
- EIP use cases
- Further technical discussion on EIP
- EIP prototype for Linux / eBPF
- Wrap up & feedback request

Wrap up

"EIP - Extensible In band Processing" is a proposal for the evolution of network layer (IPv6) capabilities

EIP can support the needs of research initiatives on IP/Internet evolution (helping to turn them into reality)

Wrap up

These initiatives and groups are somehow related to EIP :

- ITU-T Focus Group Network 2030
- Extensible Internet (EI) proposal
- SCION
- IRFT Path Aware Networking RG (panrg)
- IRFT Computing in the Network Research Group (coinrg)

Next steps

We've set up an informal Interest Group on Extensible In-band Processing, in order build a community (operators, vendors, academia)

Get feedback from key researchers, industry experts, standardization experts

Write scientific papers => Present in workshops and conferences

Develop prototypes / "running code"

Present and discuss in IETF (?)

EIP - Interest Group

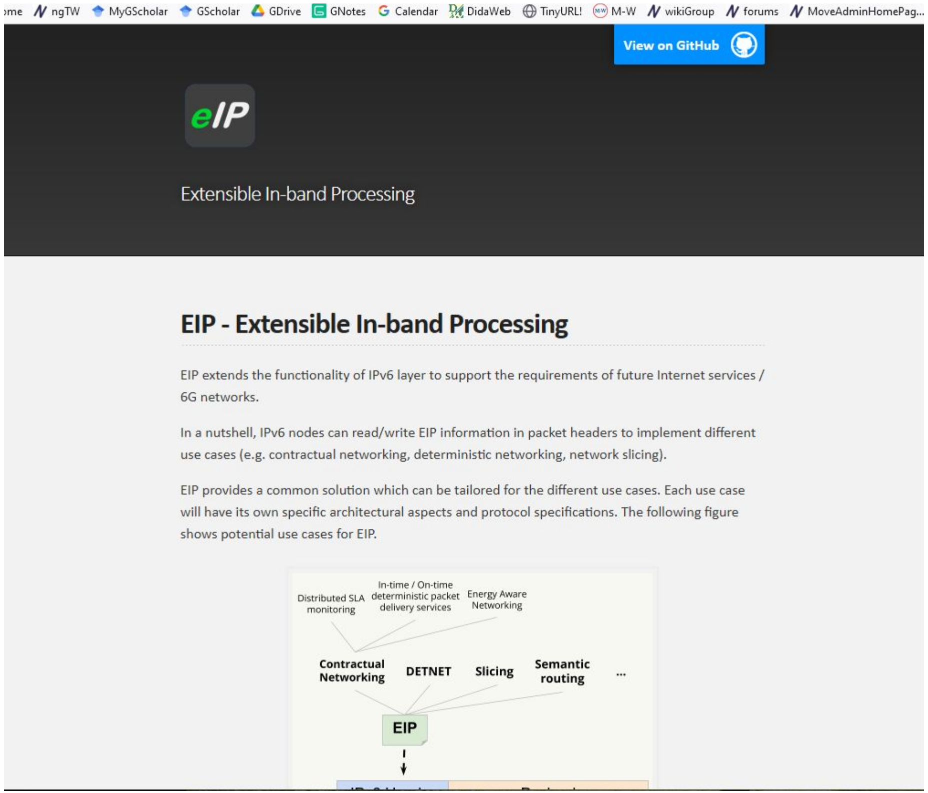
We've set up an informal Interest Group on Extensible In-band Processing, in order build a community (operators, vendors, academia)

https://eip-home.github.io/eip/

EIP - Interest Group

TOR VERGATA
UNIVERSITÀ DEGLI STUDI DI ROMA

cnit consorzio nazionale interuniversitario per le telecomunicazioni


ome ngTW MyGScholar GScholar GDrive GNotes Calendar DidaWeb TinyURL! M-W wikiGroup forums MoveAdminHomePag...

View on GitHub

eIP

Extensible In-band Processing

EIP - Extensible In-band Processing

EIP extends the functionality of IPv6 layer to support the requirements of future Internet services / 6G networks.

In a nutshell, IPv6 nodes can read/write EIP information in packet headers to implement different use cases (e.g. contractual networking, deterministic networking, network slicing).

EIP provides a common solution which can be tailored for the different use cases. Each use case will have its own specific architectural aspects and protocol specifications. The following figure shows potential use cases for EIP.

Home page:
https://eip-home.github.io/eip/

Wiki:
https://github.com/eip-home/eip/wiki

Mailing list:
eip@postino.cnit.it

EIP - Interest Group

TOR VERGATA
UNIVERSITÀ DEGLI STUDI DI ROMA

cnit consorzio nazionale interuniversitario per le telecomunicazioni

We've set up an informal Interest Group on Extensible In-band Processing, in order build a community (operators, vendors, academia)

https://eip-home.github.io/eip/

Are you interested in participating to the informal Interest Group?

Please join the mailing list http://postino.cnit.it/cgi-bin/mailman/listinfo/eip

EIP - Feedback?

Get feedback from key researchers, industry experts, standardization experts

Are you interested in some of the presented use cases of EIP (contractual networking, deterministic networking, slicing)?
Do you have feedback on these use cases?

Are you interested in other use cases? Can you suggest them?

EIP - Feedback?

Get feedback from key researchers, industry experts, standardization experts

Do you have feedback/suggestions on the technical approach?

Any other general comment or feedback?