



NEA: An SDN Switch Architecture Suitable for Application-Oriented MAC

Diego Nunes Molinos¹(✉), Romerson Deiny Oliveira², Marcelo Silva Freitas³,
Marcelo Barros de Almeida⁴, Pedro Frosi Rosa¹, and Flavio de Oliveira Silva¹

¹ Faculty of Computing, Federal University of Uberlandia (UFU),
Joao Naves de Avila Avenue, 2121, Santa Monica, Uberlandia, Brazil
{diego.molinos,pfrosi,flavio}@ufu.br

² Department of Computer Science, Unimontes, Montes Claros, Brazil
romerson.oliveira@unimontes.br

³ Department of Exact Sciences, Federal University of Jataí (UFJ), Jataí, Brazil
msfreitas@ufg.br

⁴ Electrical Engineering Faculty (FEELT), Federal University of Uberlandia (UFU),
Joao Naves de Avila Avenue, 2121, Santa Monica, Uberlandia, Brazil
marcelo.barros@ufu.br

Abstract. After several decades most of the initial Internet design remains the same. The most important changes just happened on the application and physical layers while network and transport layers remain the same. Internet architecture has been turning into a limiting factor for its own evolution, and currently, several researchers have been working on new approaches that aim at redesigning the architecture. Software-Defined Networking (SDN) has been standing out for delivering more flexibility to the network. Based on the SDN paradigm, the Title Model represents a revolutionary approach to reaching out to the applications' new requirements. The Entity Title Architecture (ETArch) is the materialization of this model, offering improvements concerning the network's addressing and routing aspects. The programmability at the data link level is essential for the efficient use of network resources, and the ETArch is not restricted only to the software elements of the architecture. The network element responsible for the physical interconnection inside the network must have the capacity to guarantee Quality of Service (QoS) requirements driven by the applications, providing support to the communication space.

1 Introduction

Several types of communication have been making massive use of the Internet. The applications are demanding more and more resources from the network. Although those applications require mobility, security, energy efficiency, bandwidth, and Quality of Service (QoS), they usually rely on the network to satisfy their requirements. Software-Defined Networking (SDN) offers better and efficient control over networking devices from a centralized controller to improve network management.

Due to the decoupling of the data plane and control plane, as expected in SDN, a distance from the functions implemented in hardware and software is created, making the hardware's behavior a single granular network fabric [4]. This contributed so that all flexibility and programmability were carried over to the network control level, practically unexplored at the data plane level. The fact is that relying entirely on the software layer to provide flexibility and programmability can result in loss of quality of service (QoS).

The Entity Title Architecture (ETArch) is based on the SDN paradigm, which aims to offer an architecture with features and support to context-oriented algorithms that allow changes and adjustments according to the network requirements [2, 3].

ETArch enables requirements defined by the applications to permeate through the architecture layers down to the data plane level. At this level, the networking device must comprehend those requirements and modify their operation's context to guarantee, for example, mobility [3].

It is important to note that several proposals offer flexibility and programmability at the lower level of the network. Some of them aim to explore the parallel processing feature from reconfigurable platforms like Field Programmable Gate Array (FPGA) and the capacity of dynamic reconfiguration to enable programmability at the Link level [5].

The fact is, there is no easy way to develop a programmable solution using reconfigurability. Reconfigurable platforms that offer dynamic reconfiguration require a great effort from the designer to design hardware blocks, data structures, orchestration, and control of all elements. On the other hand, some proposals aim to explore the network flexibility through the software solutions, for instance, [11] and [10]. It is important to note that these platforms' programmability is directed to their application field, and they tend to be limited.

This work aims to present a new switch architecture for SDN networks with MAC driven by the application called NEA Network Element. This architecture is Linux-based, and it uses some native Linux tools such as Bridges, VLANs, and Ebtables to enable flexibility at the lowest level.

The remainder of the document is structured as follows: Sect. 2 brings fundamental concepts of ETArch, presents the state-of-the-art related to SDN network elements, and the general design of ETArch Network Element. Section 3 presents the details for the architecture. Finally, Sect. 4 offers a discussion about this proposal, some concluding remarks, and suggestions for future work.

2 Background

This section provides a desirable background about the ETArch environment and important concepts, Networks Elements and a brief about ETArch switch.

2.1 Entity Title Architecture - ETArch

To approximate the Application layer semantically to the lower layers, allowing the requirements defined by the applications to permeate through the architecture layers, the Entity Title Architecture (ETArch) has been designed.

ETArch does not have a fixed layer structure as in the TCP/IP architecture [2]. In place of transport and network layers, ETArch defines the Communication layer. As shown in Fig. 1, this layer differs from traditional ones by being flexible and shaping according to application-specific communications requirements such as, for example, QoS, low power consumption, and bandwidth.

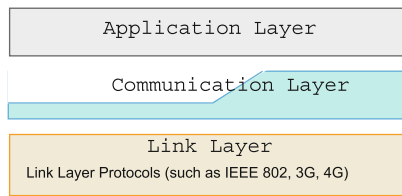


Fig. 1. ETArch layer architecture pattern. Figure adapted from [8]

From the ETArch perspective, it is important to comprehend the concepts of Entity, Workspace, and the control agent to facilitate the understanding of how the architecture works.

2.1.1 Entity and Title

Entities can be defined as any element that wants to communicate in a distributed environment [1]. Entities can be applications, devices, computers, a network element, or even a sensor. Entities are only identified by a title and a set of capabilities.

2.1.2 Workspaces

The Workspaces are logical multi-end communication buses driven by the applications and carries a set of requirements and they don't exist in the initial network configuration. They are created when an entity wants to offer content. When another entity is willing for this content, DTS must register it and attach it to the Workspace that offers that content.

ETArch specifies two types of Workspaces, namely, Data and Control Workspaces [8]. In the Fig. 2 is observed as an example of Data and control Workspace. In this figure, the DTS domain consists of two agents, DTSA1 and DTSA2, each controlling a Network Elements (NEs) subset. To this end, each agent abstracts the topology of the NEs it controls.

2.1.3 Domain Title Service - DTS

A DTS represents a set of ETArch controllers in a distributed system and their goal is to separate the network into manageable subnets within ETArch [2]. A Domain Title Service Agent (DTSA) acts in the control plane, managing the network elements. DTSA is responsible for creating, managing, and dropping Workspaces on network elements. It is important to note that each DTSA exercises control over a limited number of network elements [5]. A typical ETArch environment is illustrated in Fig. 2.

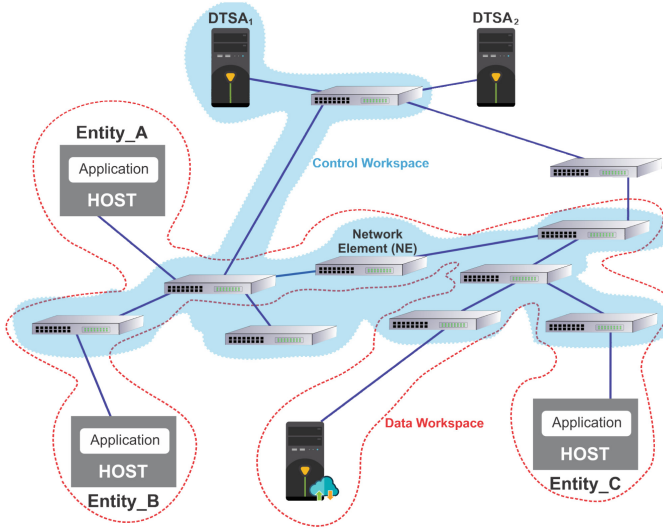


Fig. 2. Typical ETArch environment with data workspace, control workspace, DTSA and entities

It is important to note that each DTSA's act on the network control plane, and all communication between DTSA's and network elements is performed through a Control Workspace. On the other hand, the network elements act on the network data plane level.

2.2 Network Element in ETArch Architecture

Usually, SDN implementations use OpenFlow-based switches or make use of some virtualization technique [6, 7]. The ETArch architecture requires some flexible support to the programmability in the lowest level of the network. The flexibility of SDN networks is coupled to the capacity of programmability present in their layers. It is observed that several implementations assume that there is connectivity at the link level of the network.

In this sense, ETArch needs a flexible network element that supports several Workspaces per physical port. As previously presented, Workspace is a logical

path that connects all entities that are part of the same communication domain in the ETArch architecture. Workspaces by nature are a multicast communication domain segregated. The same entity can be part of several Workspaces, which means that a given Workspace does not have direct contact with another Workspace. In addition to its intrinsic multicast nature, it can also be emphasized that QoS's concept is also intrinsic to its nature, just because DTSA's on network elements deploy a Workspace according to the requirements driven by the application [8]. It is appropriate to think that Workspaces can offer finer-grained resource management.

3 NEA: Network Element Architecture for SDN Networks

Communication reliability has become a visible challenge for SDN networks. The separation of the control plane from the network data plane allows achieving a certain level of flexibility in the network management. However, it is observed that, regardless of the model or architecture used, every communication has different goals and requirements. Applications such as banking transactions, streaming services, and download/upload services have different communications requirements.

Naturally, communications require an end-to-end data routing infrastructure, but what is needed for a banking application in terms of security is quite different from what is required when downloading/uploading images on social networks.

In the ETArch, the main issue is directed towards multicast communications. Basically, ETArch institutes communications through Workspaces, where, for example, multicast are naturally implemented. The purpose is to have reliable transmissions regardless of the number of connected entities.

Focusing on the network element, the adoption of the Linux-based system to improve network solutions has gained attention in recent years. Projects such as Open vSwitch [9], Rocker Switch [12], and SwitchDev [13] are examples of L2/L3 data plane forwarding that make use of the Linux operating system to compose their solutions for data forwarding.

In this scenario, a dedicated solution offers more performance. However, it suffers from upgrades and the insertion of additional modules. Linux-based solutions offer a lower cost. Updating a new feature or even an upgrade is not a critical task, all of this, not to mention the ease of building a scalable solution. The NEA architecture overview is illustrated in Fig. 3.

The NEA architecture was designed to be modular, making it easy to upgrade or update. The NEA core was based on natives modules of Linux kernel such as Bridges, 802.1P/Q, VLANs, and Ebtables tools to support lowest level forwarding.

3.1 VLAN Filtering

In addition to manipulating the routing table and flow control, Linux Bridges can offer VLAN-oriented flow filtering, which contributes to QoS's improvement.

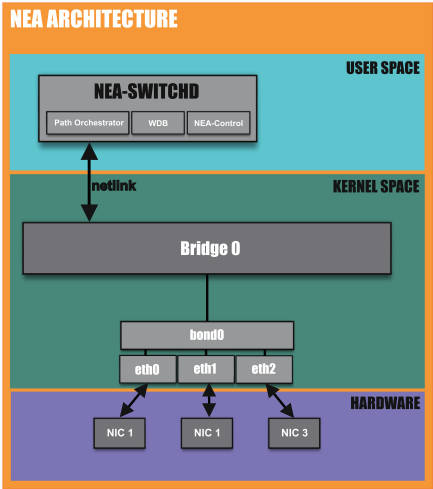


Fig. 3. NEA - Network Element Architecture overview

The protocols used in VLANs provide filtering services that support dynamic definition and establishment of communication groups in a network environment. The frames are filtered and addressed only to a particular group of recipients.

To make it possible to materialize the Workspace concept in this proposal and the fact that Linux Bridge is critical to handle all interface traffic, it is necessary to use IEEE802.1Q tagging mechanisms to identify within Bridge which frames are relative to certain Workspace for orchestration policies. It is important to note that many network interface cards do not have support for tagging frames. Therefore, all the tags will be inserted on the ingress flow and removed after the egress flow. This scenario is illustrated in Fig. 4.

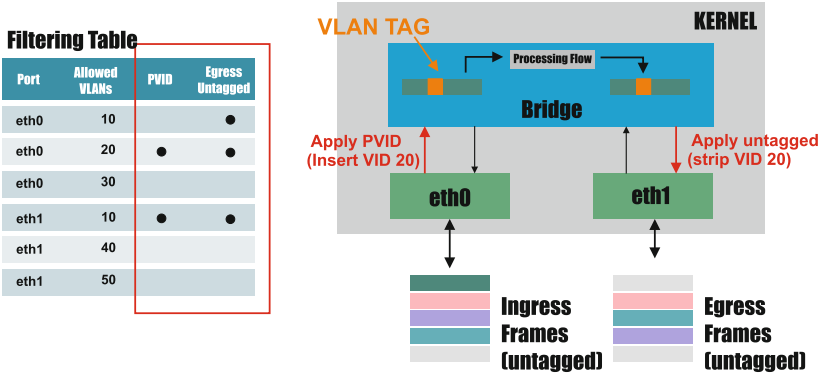


Fig. 4. VLAN-bridge routing architecture - untagged packets - Figure adapted from [14]

NEA offers support to the ETArch architecture through the NEA-SWICTHD. It is a user-space module that enables the ETArch controller interface (DTSA) and acts with the Workspace Orchestrator.

3.2 NEA-Switchd

As shown in Fig. 5, it can be seen that the NEA-Switchd module is composed of the modules: Path Orchestrator, NEA-Control, and WorkspaceDataBase (WDB).

The main sub-module is the Path Orchestrator, which is responsible for the orchestration of Workspaces. The Path Orchestrator sub-module through the Netlink interface will manipulate the Bridge routing structures. This includes the creation, removal, update, definition of flow priority levels. As already explained in the previous topic, the structure of a Workspace is associated with a VLAN.

The NEA-Control allows primitives from other network elements through the ETArch protocols and messages from the network controller (DTSA).

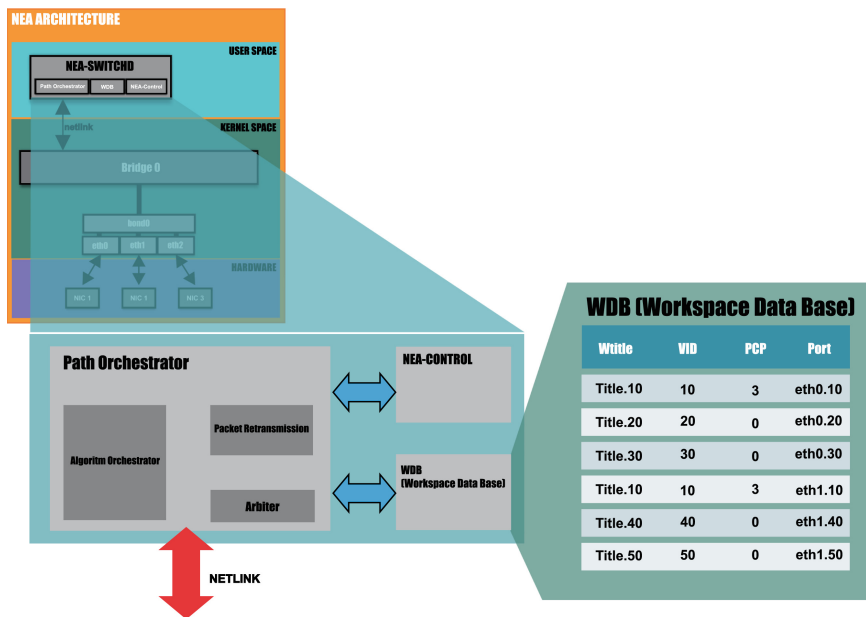


Fig. 5. Network Element Architecture modules

3.2.1 Arbiter

The Arbiter module is responsible for receiving data traffic from Bridge and making the forwarding decision. Possible decisions are: (i) Forward the ETArch

primitives to the NEA-Control module, these primitives are for communication between network elements and constitute what the Network Control Workspace, (ii) Forward the primitives from the network controller for NEA-Control, and (iii) Provide inputs for the Algorithm Orchestrator module to perform the orchestration.

3.2.2 Workspace Data Base (WDB)

Besides the entire data structure for L2 forwarding is already inserted at the Kernel level and managed by Bridge, including the FDB (Forwarding Data Base), it is necessary to establish a database to store information from the Workspaces, the WDB table is formed by the following fields: (i) Workspace Title, it is the identification of the Workspace as provided for in the title model already explained earlier in this document, (ii) VID, the field used to relate a given Workspace to a VLAN framework, (iii) PCP, used to define the priority level of that VLAN and (iv) Port, to identify the Workspace-linked port VLAN.

4 Concluding Remarks

Exploring flexibility and programmability at all network architecture layers has become the key to improving applications' QoS. One problem with this is that all implementations assume there is connectivity at the network link level, and they end up, in turn, dedicating efforts to build solutions that explore the highest levels of network architecture.

This work presents a network element for SDN networks with MAC support driven by applications with a focus on the ETArch architecture. The flexibility is achieved by using VLAN techniques associated with the ability to program high-level functions for managing and materializing the Workspace concept.

It is important to note that the concept of Workspace can assume different interpretations based on the reference element, that is, having the network element as a reference, Workspace is materialized through the construction and maintenance of routing tables, using a buffer space for the temporary storage of primitives to guarantee the aggregation of multicast traffic. From the network controller's point of view, it is a logical bus, free of topology and geographic dimension.

Through state-of-the-art, there is a need to offer support to flexible network elements with fine granularity. The flexibility offered by techniques such as OpenFlow offers limited flexibility and coarse granularity compared to this proposal.

The future work will address issues such as the complete switch engine solution proposed in this work and a scenario to compare the NEA performance concerning Openflow.

Acknowledgments. This project has been built-up with the support of MEHAR team. We want to thank all who contributed to our research. Brazilian agency CAPES has partially funded this work.

References

1. de Souza Pereira, J.H., Kofuji, S.T., Rosa, P.F.: Horizontal address ontology in internet architecture. In: 2009 3rd International Conference on New Technologies, Mobility and Security, pp. 1–6. IEEE (2009)
2. de Oliveira Silva, F., Gonçalves, M.A., de Souza Pereira, J.H., Pasquini, R., Rosa, P.F., Kofuji, S.T.: On the analysis of multicast traffic over the entity title architecture. In: 2012 18th IEEE International Conference on Networks (ICON), pp. 30–35. IEEE (2012)
3. Silva, F., Corujo, D., Guimaraes, C., Pereira, J., Rosa, P., Kofuji, S., Neto, A., Aguiar, R.: Enabling network mobility by using IEEE 802.21 integrated with the entity title architecture. In: Proceedings of the IV WPEIF: SBRC 2013 Workshops, pp. 29–34 (2013)
4. de Oliveira Silva, F., de Souza Pereira, J.H., Rosa, P.F., Kofuji, S.T.: Enabling future internet architecture research and experimentation by using software defined networking. In: 2012 European Workshop on Software Defined Networking, pp. 73–78. IEEE (2012)
5. Kalyaev, A., Melnik, E.: FPGA-based approach for organization of SDN switch. In: 2015 9th International Conference on Application of Information and Communication Technologies (AICT), pp. 363–366. IEEE (2015)
6. McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Turner, J.: OpenFlow: enabling innovation in campus networks. *ACM SIGCOMM Comput. Commun. Rev.* **38**(2), 69–74 (2008)
7. Bifulco, R., Rétvári, G.: A survey on the programmable data plane: abstractions, architectures, and open problems. In: 2018 IEEE 19th International Conference on High Performance Switching and Routing (HPSR), pp. 1–7. IEEE (2018)
8. Oliveira, R.D., Molinos, D.N., Freitas, M.S., Rosa, P.F., de Oliveira Silva, F.: Workspace-based virtual networks: a clean slate approach to slicing cloud networks. In: CLOSER, pp. 464–470 (2019)
9. Pfaff, B., Pettit, J., Koponen, T., Jackson, E., Zhou, A., Rajahalme, J., Amidon, K.: The design and implementation of open vswitch. In: 12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 2015), pp. 117–130 (2015)
10. Bosshart, P., Daly, D., Gibb, G., Izzard, M., McKeown, N., Rexford, J., Schlesinger, C., Talayco, D., Vahdat, A., Varghese, G., Walker, D.: P4: programming protocol-independent packet processors. *ACM SIGCOMM Comput. Commun. Rev.* **44**(3), 87–95 (2014). <https://doi.org/10.1145/2656877.2656890>
11. OF-CONFIG 1.2. OpenFlow Management and Configuration Protocol. Open Networking Foundation. Available via DIALOG (2014). <https://opennetworking.org/wp-content/uploads/2013/02/of-config-1.2.pdf>
12. Feldman, S.: Rocker: switchdev prototyping vehicle. In: Proceedings of Netdev 0.1. Available via DIALOG (2015). <https://people.netfilter.org/pablo/netdev0.1/papers/Rocker-switchdev-prototyping-vehicle.pdf>
13. Pirkko, J., Feldman, S.: Ethernet switch device driver model (switchdev). Available via DIALOG (2014). <https://www.kernel.org/doc/Documentation/networking/switchdev.txt>
14. Makita, T.: Virtual switching technologies and Linux bridge. Available via DIALOG (2014). <https://studylib.net/doc/18879676/virtual-switching-technologies-and-linux-bridge>