On The Analysis of Multicast Traffic Over The Entity Title Architecture

Flávio de Oliveira Silva*†, Maurício Amaral Gonçalves†, João Henrique de Souza Pereira*,
Rafael Pasquini†, Pedro Frosi Rosa†, and Sergio Takeo Kofuji*
*Polytechnic School, University of São Paulo, São Paulo - Brazil
flavio@pad.lsi.usp.br, joaohs@usp.br, kofuji@pad.lsi.usp.br
†Faculty of Computing, Federal University of Uberlândia, Minas Gerais - Brazil
flavio@facom.ufu.br, mauricio@algartelecom.com.br, pasquini@facom.ufu.br, frosi@facom.ufu.br

***Abstract*—New applications bring a new set of requirements that the Internet is not able to satisfy in a proper way. Internet architecture must be reviewed and researchers from all over the world are engaged in the design of a new Internet. Software Defined Networking (SDN), which is materialized in OpenFlow, represents an extraordinary opportunity to rethink computer networks. In this paper, taking advantage of SDN and the concepts of our previous work regarding the Entity Title Model, we present a proof of concept OpenFlow based implementation of the Entity Title Architecture. It is a *clean slate* network architecture for future networks where multicast and mobility are seamlessly provided. By using this implementation, we describe some experiments conducted and present a comparison between a video application implemented first, using the TCP/IP stack and then using our architecture focusing on its multicast capabilities and by consequently reducing bandwidth consumption. The results presented in this paper show that this consumption near the source by using our architecture remains constant while using TCP/IP it increases monotonically.**

I. Introduction

The design of the Internet and its main concepts came to life in the sixties [1] and its core protocols where designed in the seventies [2]. After four decades and a huge success, most of that initial design is still in place. However, applications vastly different from those that initially used the network are now being developed and bringing a new set of requirements, such as mobility, that current Internet is not able to satisfy in a proper way due to its limitations [3].

The Internet architecture must be reviewed and researchers from all over the world are engaged in the design of a new Internet, from the ground up. This so called *clean slate* approach frees the research from the legacy of the current architecture and fosters innovations [4]. At a future time, when results should be deployed, the research will then be focused to the transition from the current Internet to the Future Internet.

Software Defined Networking represents an extraordinary opportunity to rethink networks by using some abstractions that decouples the software that controls the network elements from the hardware, offering a open and well defined interface to control and modify the hardware behavior.

SDN, whose philosophical bases where presented by the 4D architecture [5] and refined by the Ethane architecture[6], allows new networking protocols to be defined and readily experimented in real conditions in production networks.

SDN, currently, is materialized in OpenFlow [7]. Essentially, OpenFlow separates the data plane from the control plane, defining an OpenFlow switch. While OpenFlow works with current networks, it can be used to shape and deploy future network architectures. In this context, SDN and OpenFlow are fostering our research by enabling a viable deploy and experimentation.

In previous works we introduced the Entity Title Model, which is a vision of how entities are enable to semantically [8] specify their requirements and capabilities, in order to communicate with each other by using a specific naming and addressing scheme, based on a topology independent name that unambiguously identifies an entity [9], i.e., its Title. The Domain Title Service (DTS)[10], is a distributed system over the network elements responsible for aiding communicating entities in locating their peers and in negotiating the establishment and maintenance of their conversations, accordingly to the Entity Title Model.

In this paper we present some components of the Entity Title Architecture, a proof of concept network architecture based on the Entity Title Model, implemented by using OpenFlow, which demonstrates the feasibility of our approach by focusing the multicast aspect. We do so by further describing the DTS, a key concept of the architecture and how it relates to OpenFlow in Section II. In Section III we describe some experiments conducted and presents a comparison between a video application, implemented first using TCP/IP stack and then using the presented architecture. In Section IV we present some related works and finally in Section V we make some concluding remarks and presents a future work.

II. The Entity Title Architecture at a Glance

The Entity Title Architecture is a realization of the Entity Title Model [9] and some basic concepts of this architecture are the Domain Title Service (DTS), featuring an Entity, its Title and a *clean slate* naming and addressing scheme, where mobility and multicast are seamlessly provided. The focus of this paper is to present the multicast native capabilities of the architecture.

An Entity has communication requirements and capabilities that can be semantically understood from top to bottom layers. Besides this, it has at least one Title and a location that is

978-1-4673-4523-1/12/$31.00

variable over time. Some examples of entities: a content, a service, a sensor, a smart phone, an application, a process.

The Title is a topology independent designation to ensure an unambiguous identification of an entity. One title designates only one entity, but one entity may have more than one title. The title plays a key role in order to provide the horizontal addressing [11] of the entities.

The Domain Title Service (DTS)[10] is a distributed system over the network elements, responsible for maintaining information regarding all the entities in that domain, such as the associated titles, the communication requirements and capabilities over time, the connection setup and the maintenance of them. The DTS may be divided into in several parts, being the Domain Title Service Agent (DTSA) the system's cornerstone.

The DTS deals with all the control aspects of the network and maintains the knowledge, inside the network, about itself, playing an important role at central aspects of networking like naming and addressing.

The communication between the entities is provided by the workspace, that represents the path where data is transported to all entities connected to such workspace. A workspace is created when an entity needs to communicate with another for a specific purpose, such as video-conferencing or file sharing. In order to create a workspace, the entity must specify the requirements it has and capabilities it may offer in conversing with other entities in the workspace. For example, the entity may require secrecy and delivery guarantees from its peers, while also offering a maximum bandwidth value. If the requirements change during the conversation, the DTS brokers their renegotiation between the entities of that workspace.

A detailed description on how the requirements and capabilities are expressed and stored within the DTS is out of the scope of this paper, but in a few words an OWL (Web Ontology Language) based syntax is used to express them, accordingly to what is specified by the Entity Title Model [9] and this information is stored at the DTSA.

All entities that share a workspace see the same message exchange. That is, any message sent by one entity is multicast to all the other entities in the workspace. Delivery, ordering, or other guarantees are provided only if required, thus making an efficient use of the physical layer. If an entity is interested in a conversation going on in an existing workspace, the DTS provides the tools for discovering such workspaces, so the entity may join the conversation at any time. That is, given that the entity passes any authentication and authorization restrictions associated to the workspace.

A. *Entity Title Architecture Protocol Stack*

Considering the previously presented concepts, the Entity Title Architecture can only be realized with a new protocol stack, especially at the Transport and Network layers. In fact, we consider a new layer, called Communication Layer, which contains functionality that today are related to these layers, as depicted at Fig. 1. This protocol stack is compatible with current application layer protocols as denoted by some protocol names presented at the figure. Is important to notice that the compatibility is not limited to the protocols presented at the figure.

The unusual representation of the Communication Layer highlights that this layer may contain functionalities only as required. Then, at a local network, only a packet ordering can be required and no routing is necessary, thus, the thin portion of layer is used. When handling inter-networking, with secrecy and QoS, for example, the full layer is used instead.

The Communication layer uses current link layer protocols and at this moment, we consider IEEE 802 family of protocols, making possible to deploy the architecture at current devices using wired and wireless access links.

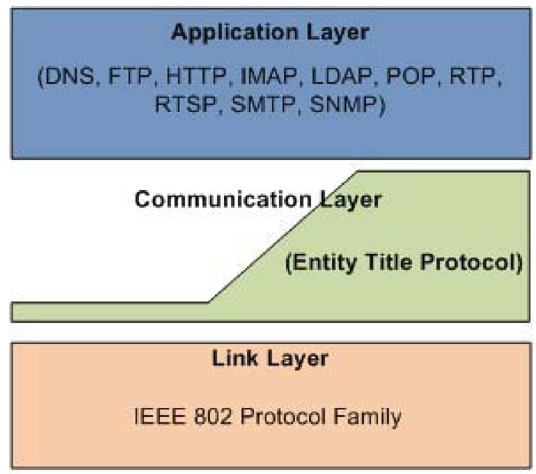


Fig. 1. Entity Title Architecture Protocol Stack.

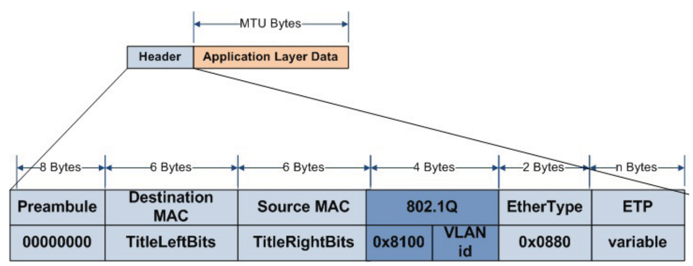


Fig. 2. Communication Layer Protocol Data Unit.

In order to handle the dynamic behavior of the Communication Layer, a protocol with a variable header was defined, as presented in Fig. 2. The frame data is based on 802.1Q, and a novel variable size field, called ETP (Entity Title Protocol) was inserted to support our communication model. To be compatible with current networks, the source and destination MAC addresses contains the leftmost and rightmost bits of Title of the destination entity. Usually, the destination is the DTS in case of control primitives, or a workspace in case of the data plane communication. This approach represents a paradigm change regarding naming and addressing at current networks. After this header, the payload contains the data from the Application Layer. Using this approach, the Entity Title Architecture retains compatibility with current Application

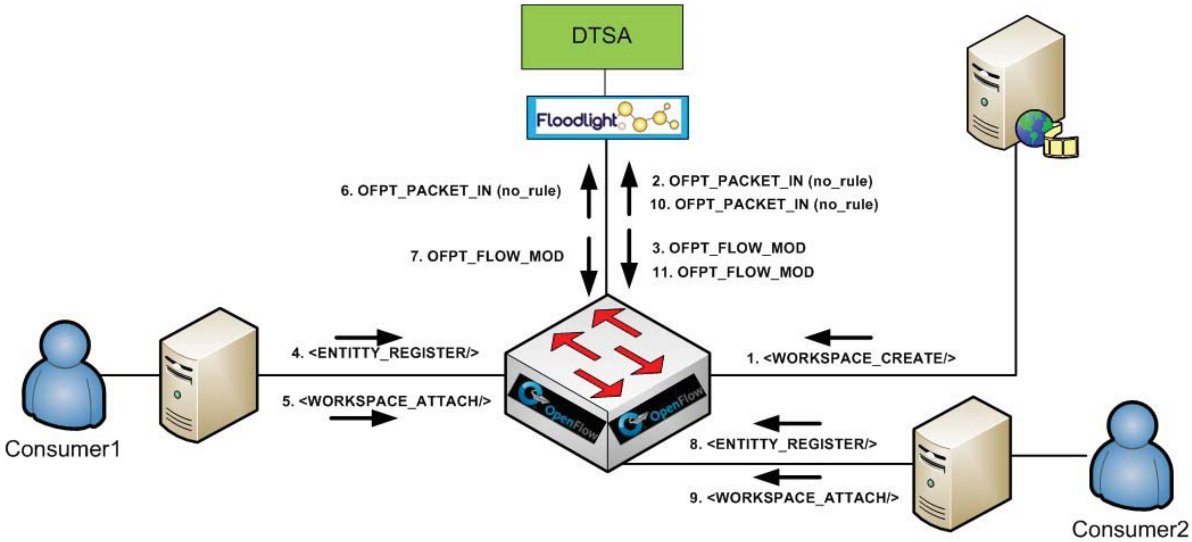


Fig. 3. Control Primitives Exchanged to Create and Adapt the Workspace.

Layer services. Our experiments executed until this moment shows that just some few lines of code need to be changed in order to make an application that uses the network, compatible with the Entity Title Architecture by using our socket library.

B. DTSA as the Controller

The DTSA is responsible for both keeping the entities' and workspaces' metadata, and for coordinating the network elements to implement workspaces. Hence, implementing the DTSA and, therefore, the DTS requires control of network elements, which is not viable at the current networks. However, by using the SDN abstraction, the architecture can come to life and prototyping become possible by using OpenFlow [7].

An Openflow switch contains one or more *flow tables* that are used for packet lookup and forwarding. Each entry at the *flow table* contains a key and associated actions that are executed in the presence of a match. The behavior of the switch is driven by the *flow table* state and the *Controller* is responsible for it's maintenance, providing the ability to change the switch behavior on the fly by modifying the flow table.

The ability to change the switch behavior on the fly by modifying the flow table makes it suitable to the experimentation of new naming and addressing schemes.

The flow table [7] will handle the information to produce the workspace materialization. It's important to notice that this implementation assumed the use of OpenFlow 1.0 based switches, and in this case almost all the header fields used by OpenFlow to perform the match against the flow table are not suitable to be used, because the communication does not rely on the TCP/IP stack. In this case, the implementation use just some of these fields to accommodate the entity's title, particularly Ethernet source and destination address, giving a title the maximum size of 96 bits.

Newer versions of OpenFlow offers the concept of an extensible match and this is a very interesting feature for this work, but an assumption at this moment is that the Entity Title Architecture must be ready to run at most available OpenFlow based substrates, specially those that use hardware switches.

As the DTSA's task of coordinating network elements is closely related to that of managing flows by an OpenFlow controller, we have decided to implement the first on top of the latter. In a nutshell, we extended the FloodLight open-source OpenFlow controller [12] to closely work with the DTSA. To this proof of concept, FloodLight was selected because it combines ease of development and, at the same time, can be deployed in production networks.

The extension to the Floodlight controller consisted in a new module that instantiates the DTSA and handles the exchange of DTS control primitives, by listening the communication with the OpenFlow switch. By default, all primitives that do not match any of the rules in the switch flow table are sent to the DTSA, as illustrated by Fig. 3. When a message is received, the listener is called and checks if the message is a defined primitive, as detailed in table I. If so, the message is delivered to DTSA that processes it and modifies the switches using a *flow_mod*.

Initially an entity willing to provide data requests the *WORKSPACE_CREATE* message. This primitive will be forwarded by the switch to the DTSA, using the OpenFlow *OFPT_PACKET_IN* message. DTSA will receive this indication and will create a Workspace storing it's Title, requirements and capabilities. As well, using the Workspace Title a new flow table entry key will be created and a rule indicating that all packets sent to this workspace should have as output the physical port where the entity that created the workspace is located. By using the OpenFlow *OFPT_FLOW_MOD* message, this rule will be added to the flow table.

A registered entity that wants to receive the data provided by the workspace, should attach to it by using a *WORKSPACE_ATTACH* message. This primitive also will be forwarded to the DTSA and in the same manner, by using

TABLE I
PRIMITIVES AND THEIR SEMANTIC REGARDING OPENFLOW.

Primitive	Meaning
ENTITY_REGISTER	Registers an entity at the DTS. To be registered an entity must present its title, capabilities and communication requirements
WORKSPACE_CREATE	Create the workspace. Using a *flow_mod* message adds a new flow identified by the Entity Title, in this case a bit string up to 96 bits stored at the fields Ethernet Source and Destination address the entry key at the flow table
WORKSPACE_ATTACH	Attaches an entity in a workspace and using a *flow_mod* message, updates the output ports to include this entity
ENTITY_UNREGISTER	Removes an entity from the DTS and updates flow tables
WORKSPACE_DETACH	Removes an entity from a existing workspace and updates flow tables accordingly
WORKSPACE_DELETE	Removes the flow entries regarding a workspace

the OpenFlow *OFPT_PACKET_IN* and *OFPT_FLOW_MOD* messages, it will modify the flow table to include the physical port of the requesting entity into the current workspace. At this moment, for this specific workspace the action at the flow table will be modified an will contain an output to all the physical ports where the local entities of that workspace are connected. Another entity could be attached to the workspace by pursuing the same procedure and becoming part of the sharing entities, as depicted by Fig. 3.

III. EVALUATION

To experiment and evaluate the Entity Title Architecture, and specially the DTSA and the addressing by using the workspace and its multicast capabilities, a prototype DTSA as presented in Section II-B was implemented and some experiments were conducted.

A simple topology, as depicted at Fig. 4, was defined. At the right side, a server contains a Video Application that produces a stream based on MJPEG. At the left side, at a host, one or more clients where instantiated during the experiments. Between these two hosts there are three OpenFlow switches. Although it is a simple topology, it reflects a common situation where a server and a client are separated by a group of switches. The topology was created using MININET [13], a system for rapidly prototyping OpenFlow based networks.

To perform a comparison of the Entity Title Architecture and the use of the TCP/IP architecture for networking, two different server applications where created. The first one based on the UDP and IP protocols and the second one based on our approach. Essentially, these applications are the same, and the main difference between them is just how the sockets are created and used.

At the application layer, a Real-time Transport Protocol (RTP) [14] based message is created, then, in the first case, Datagram Socket is used to send this message. The second video application, that uses the Workspace, creates a *Finsocket*, which is based in Raw Sockets. The Raw Sockets does not use the TCP/IP stack and directly creates a frame and send it over the physical medium. In fact the *Finsocket* does create a frame based on the Ethernet frame, but it does not contain the traditional information at its headers. Instead, the source address contains the leftmost bits of the workspace title and the destination address field, it's rightmost bits.

Each server application was started and a different number of clients connected to it, requesting data. Considering the UDP/IP server application as the number of clients increases, the bandwidth usage increases as well, since several streams are instantiated. The video server that uses the Entity Title Architecture remains with a constant use of the bandwidth at the source, no matter the number of clients. This happens because the data is sent to workspace and a client connects to workspace, not directly to server. The Fig. 5 shows the bandwidth usage obtained in the comparison.

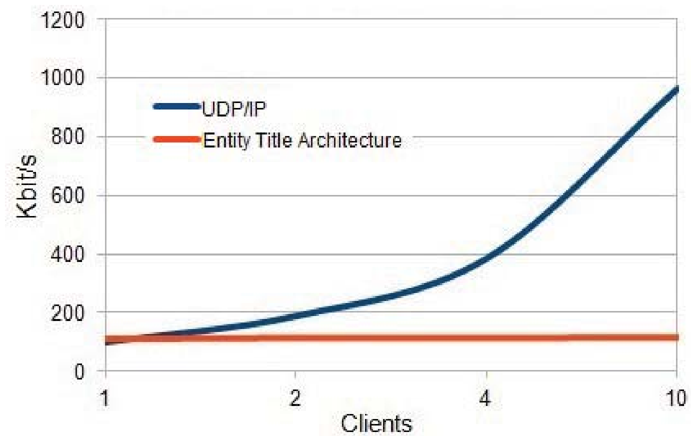


Fig. 5. Bandwidth usage at the source versus the number of clients.

Fig. 6, shows the bandwidth usage in each situation over the time for a fixed number of clients. The UDP/IP based application shows a higher consumption as the number of clients is increased during observation, whereas the workspace based data transmission remains constant, no matter the number of clients.

IV. RELATED WORK

At this moment, several research groups are working towards a Future Internet architecture. At the European Union, almost a hundred different projects are funded within the Seventh Framework Programme (FP7), under the Objective 1.1, the Network of the Future, and from those some are directly related to Future Internet like 4WARD, CHANGE, MEDIEVAL, PURSUIT, SAIL, SENSEI, TRILOGY and UNIVERSELF [15]. A detailed description of these projects is out of the scope of this work, but in general they are based on a *clean slate* approach and addresses different aspects of the desired Future Internet.

The 4WARD *Netinf* [16] concept, based on an information-centric paradigm, is related to the Domain Title Service (DTS)

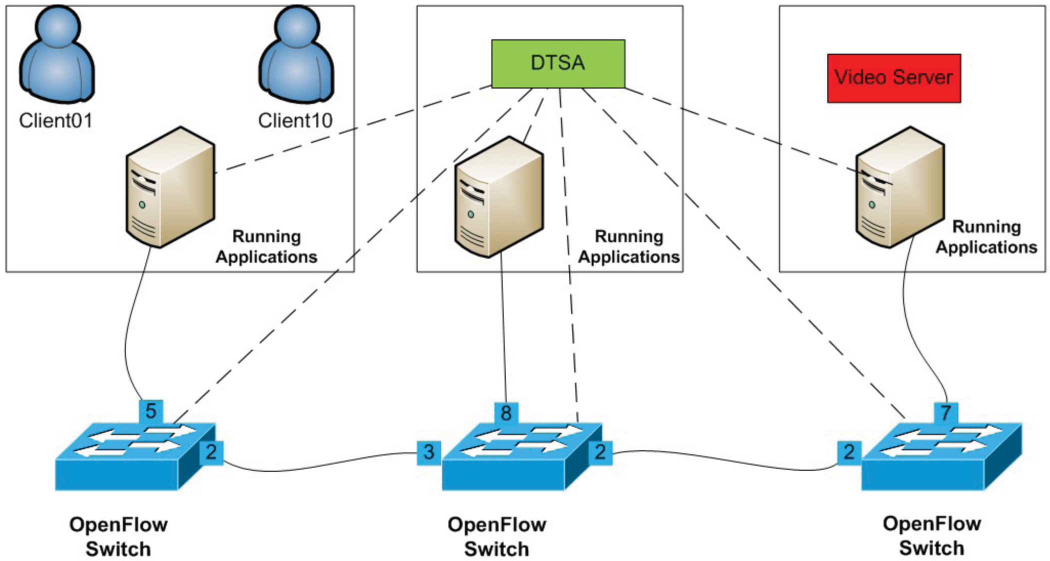
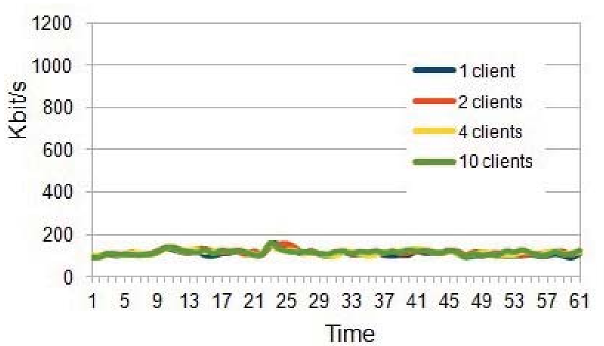


Fig. 4. Scenario used for experimentation.

(a) TCP/IP architecture approach.



(b) Entity Title architecture approach.

Fig. 6. Bandwidth usage versus the number of clients.

and its naming based on the Title can leverage *Netinf* concept. The DTS can deal with the information and with the context of the consumers taking into account their communication needs at each context, supporting their change over time.

The Entity Title Architecture be used at the communications layer to the real world architecture envisaged by SENSEI [17] project, and besides that, the concept of addressing by the use of a Title is suitable for real world Internet and its sensor networks.

At the United States, the Future Internet Architecture (FIA) [18], which represents a consolidation from a previous program, contains at this moment four projects that are dealing with aspects of the network such as: content-centric networks, mobility, cloud Computing and security. The MobilityFirst [19] network architecture has focus in mobility and proposes a new protocol stack that considers a new naming scheme based on a Globally Unique Identifier (GUID) which can provide mobility and multicast. The Title is related to the GUID, but the workspace concept provides an out-of-band control of the packets delivery while in MobilitFirst architecture the control happens in-band. This MobilityFirst architecture is planning to work with experimental design based on SDN by using NetFPGA [20] based switches.

In this scenario, with different projects, the Entity Title Architecture represents an additional proposal that might contribute to this research area. The panorama presented sustains the main ideas regarding this work, that are: a new protocol stack for the Internet that replaces TCP/IP stack, a new naming and addressing scheme, an experimental approach using SDN and the vision of collaboration between the research community.

V. Concluding Remarks and Future Work

Considering the new set of requirements, Internet architecture must be reviewed. This process of revision using a *clean slate* can free researchers of current shortcomings, providing a rich environment for experimentations.

In this paper we present a SDN based implementation of the Entity Title Architecture. This work focused in the presentation of the main concepts of the architecture and how they related with OpenFlow. Others aspects of the architecture like naming, discovery, routing, security, among others, where not covered and will be presented in future works.

Although OpenFlow can be used to implement new naming, routing and addressing schemes, the literature regarding this subject does not contain detailed descriptions about how this can be accomplished and this work intends to contribute regarding this subject, as well. Thus, besides demonstrating experimentally the Entity Tile architecture, this works also shows how an IP centered OpenFlow switch, compliant with OpenFlow 1.0 specification, may be used in networks that completely drop the TCP/IP stack from the data plane by using a new semantic for the flow table.

The evaluation of the implemented architecture, showed that the bandwidth used at the source remains constant no matter the number of clients connected to it. The impact of this fact is that actual links can be used to supply services like ultra high definition videos with an efficient energy consumption.

This was an expected result, because the Entity Title architecture is based on a new naming and addressing scheme, where the destination address is the workspace and while the packet is sent to a workspace, all entities that are part of it receives this packet bringing to the architecture a seamless multicast capability. The workspace also brings mobility, so, while in the same DTSA , it can move between ports and in the presence of this event, the flow table will be automatically updated. The mobility between DTSAs is also being deployed and experimentally tested.

In order to scale the experiments, we have started experimenting this prototype using the OFELIA test bed [21] and reporting on these experiments is the subject of future work.

The results show that we are facing a viable approach to bring richer and efficient services to the network, collaborating with the research that aims to define, design and deploy next generation computer network architectures.

REFERENCES

[1] P. Baran, "On distributed communications networks," *IEEE Transactions on Communications Systems*, vol. 12, no. 1, pp. 1–9, Mar. 1964.

[2] V. Cerf and R. Kahn, "A protocol for packet network intercommunication," *Communications, IEEE Transactions on*, vol. 22, no. 5, p. 637–648, 1974.

[3] T. Zahariadis, D. Papadimitriou, H. Tschofenig, S. Haller, P. Daras, G. D. Stamoulis, and M. Hauswirth, "Towards a future internet architecture," in *The Future Internet. Future Internet Assembly 2011: Achievements and Technological Promises*, ser. LNCS, J. Domingue, A. Galis, A. Gavras, T. Zahariadis, and D. Lambert, Eds. Berlin, Heidelberg: Springer-Verlag, 2011, vol. 6656, p. 7–18. [Online]. Available: http://www.springerlink.com/content/978-3-642-20897-3#section=881237&page=15&locus=86

[4] J. Roberts, "The clean-slate approach to future internet design: a survey of research initiatives," *annals of telecommunications - annales des télécommunications*, vol. 64, no. 5-6, pp. 271–276, May 2009. [Online]. Available: http://www.springerlink.com/content/e240776641607136/

[5] A. Greenberg, G. Hjalmtysson, D. A. Maltz, A. Myers, J. Rexford, G. Xie, H. Yan, J. Zhan, and H. Zhang, "A clean slate 4D approach to network control and management," *SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 5, p. 41–54, Oct. 2005. [Online]. Available: http://doi.acm.org/10.1145/1096536.1096541

[6] M. Casado, M. J. Freedman, J. Pettit, J. Luo, N. McKeown, and S. Shenker, "Ethane: taking control of the enterprise," in *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*, ser. SIGCOMM '07. New York, NY, USA: ACM, 2007, p. 1–12. [Online]. Available: http://doi.acm.org/10.1145/1282380.1282382

[7] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: enabling innovation in campus networks," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, p. 69–74, Mar. 2008, ACM ID: 1355746.

[8] F. de Oliveira Silva, A. Dias, C. Ferreira, E. De Souza Santos, F. Pereira, I. de Andrade, J. de Souza Pereira, L. Camargos, L. Theodoro, M. Gonçalves, R. Pasquini, A. Venâncio Neto, P. Rosa, and S. Kofuji, "Semantically enriched services to understand the need of entities," in *The Future Internet*, ser. Lecture Notes in Computer Science, F. Álvarez, F. Cleary, P. Daras, J. Domingue, A. Galis, A. Garcia, A. Gavras, S. Karnourskos, S. Krco, M. Li, V. Lotz, H. Müller, E. Salvadori, A. Sassen, H. Schaffers, B. Stiller, G. Tselentis, P. Turkama, and T. Zahariadis, Eds. Springer Berlin / Heidelberg, 2012, vol. 7281, pp. 142–153. [Online]. Available: http://www.springerlink.com/content/1222874ul734676k/abstract/

[9] J. H. d. S. Pereira, F. d. O. Silva, E. Lopes Filho, S. T. Kofuji, and P. F. Rosa, "Title model ontology for future internet networks," in *Future Internet Assembly 2011: Achievements and Technological Promises*, ser. LNCS. Future Internet: Achievements and Promising Technology: Springer-Verlag, 2011, vol. 6656, p. 465.

[10] F. d. O. Silva, J. H. d. S. Pereira, S. T. Kofuji, and P. F. Rosa, "Domain title service for future internet networks," in *Anais do II Workshop de Pesquisa Experimental na Internet do Futuro (WPEIF)*. Campo Grande: SBC, 2011.

[11] J. H. de Souza Pereira, S. T. Kofuji, and P. F. Rosa, "Horizontal addressing by title in a next generation internet," in *2010 Sixth International Conference on Networking and Services (ICNS)*. IEEE, Mar. 2010, pp. 7–11.

[12] Big Switch, "Developing floodlight modules. floodlight OpenFlow controller," http://floodlight.openflowhub.org/developing-floodlight/, Jan. 2012. [Online]. Available: http://floodlight.openflowhub.org/developing-floodlight/

[13] B. Lantz, B. Heller, and N. McKeown, "A network in a laptop: rapid prototyping for software-defined networks," *Proceedings of the Ninth ACM SIGCOMM Workshop on Hot Topics in Networks*, p. 19:1–19:6, 2010, ACM ID: 1868466.

[14] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, *RTP: A Transport Protocol for Real-Time Applications*, ser. Request for Comments. IETF, Jul. 2003, no. 3550, published: RFC 3550 (Standard) Updated by RFCs 5506, 5761, 6051. [Online]. Available: http://www.ietf.org/rfc/rfc3550.txt

[15] E. Commission, "The network of the future - projects," http://cordis.europa.eu/fp7/ict/future-networks/projects_en.html, 2012. [Online]. Available: http://cordis.europa.eu/fp7/ict/future-networks/projects_en.html

[16] M. D'Ambrosio, M. Marchisio, V. Vercellone, B. Ahlgren, and C. Dannewitz, "4WARD. second NetInf architecture description," 2010. [Online]. Available: http://www.4ward-project.eu/index.php?s=file_download&id=70

[17] V. Tsiatsis, A. Gluhak, T. Bauge, F. Montagut, J. Bernat, M. Bauer, C. Villalonga, P. Barnaghi, and S. Krco, "The SENSEI architecture-Enabling the real world internet," in *Towards the Future Internet*. Amsterdam: IOS Press, 2010, pp. 247 – 256.

[18] N. S. Foundation, "NSF future internet architecture project," http://www.nets-fia.net/, 2011. [Online]. Available: http://www.nets-fia.net/

[19] I. Seskar, K. Nagaraja, S. Nelson, and D. Raychaudhuri, "MobilityFirst future internet architecture project," in *Proceedings of the 7th Asian Internet Engineering Conference*, ser. AINTEC '11. New York, NY, USA: ACM, 2011, p. 1–3. [Online]. Available: http://doi.acm.org/10.1145/2089016.2089017

[20] J. Naous, D. Erickson, G. A. Covington, G. Appenzeller, and N. McKeown, "Implementing an OpenFlow switch on the NetFPGA platform," in *Proceedings of the 4th ACM/IEEE Symposium on Architectures for Networking and Communications Systems*, ser. ANCS '08. San Jose, California: ACM, 2008, p. 1–9, ACM ID: 1477944.

[21] OFELIA, "OpenFlow in europe - linking infrastructure and applications," 2011. [Online]. Available: http://www.fp7-ofelia.eu/about-ofelia/