Control Plane Routing Protocol for the Entity Title Architecture: Design and Specification

Natal Vieira de Souza Neto,
Flávio de Oliveira Silva
and Pedro Frosi Rosa

Faculty of Computing
Federal University of Uberlândia
Uberlândia, MG, Brazil
Email: natal@mestrado.ufu.br, flavio@ufu.br,
pfrosi@ufu.br

João Henrique de Souza Pereira

Innovation, Research and Development
Algar Telecom
Uberlândia, MG, Brazil
Email: joaohs@algartelecom.com.br

***Abstract*—Current and future applications pose new requirements that Internet architecture is not able to satisfy. In this context, new network architectures, focusing on different aspects, are being designed and deployed. The Entity Title Architecture (ETArch) is a clean-slate Software Defined Networking based approach which aims to satisfy different applications requirements such as multicast traffic, mobility and Quality of Service. This work presents the design and specification of the routing protocol used by ETArch, by describing the services, primitives and associated rules. This work contributes with ETArch in a central point of the architecture, the inter-networking. The multi-objective routing mechanism described in this work takes into account, applications requirements such as mobility and security. Moreover, the protocol presented works at the control plane and uses ETArch workspace concept, representing a new class of routing mechanism that differs from classical approaches such as Distance Vector or Link-State.**

Keywords–Software Defined Networking; Routing; Protocol Specification.

I. INTRODUCTION

The advances in software, hardware and communications brought a world where mobile devices with high resolution cameras, different sensors, equipped with multiple wireless interfaces are connected to clouds of servers using broadband access networks. New services and applications emerged and the Internet architecture [1], proposed in the sixties, that also collaborated to this scenario, is not able to satisfy the new applications requirements such as mobility, Quality of Service (QoS) and security [2].

To face these challenges research groups around the world [3][4] are designing and deploying new network architectures, some of them, based on a clean-slate approach in order to bring life the Future Internet.

One of these network architectures, which uses a clean-slate approach, is the Entity Title Architecture (ETArch), based on the Entity Title Model [5]. ETArch uses a naming and addressing scheme based on a topology-independent designation that uniquely identifies an entity, named Title, and on the definition of a logical bus which gathers multiple entities, willing to communicate driven by specific purpose, named Workspace. Workspaces are, dynamically, created/removed according to user's specific needs. Users could be linked (attached) to the workspaces during its life cycle.

The workspace is capable to handling the requirements of users and applications over time. Our research regarding ETArch demonstrated some of these requirements such as naming and addressing [5], QoS [6], multicast [7] and mobility in [8].

A central point in an architecture are the inter-networking mechanisms, which allows the network to scale its coverage in a geographical perspective. This work contributes with ETArch, by presenting the design and specification of the routing protocol. To do so, it describes the services, primitives and rules associated to this protocol. This paper is a conceptual work to define the routing mechanism to ETArch, that runs in the control plane by using a Software Defined Networking (SDN) as its underlying interconnection strategy.

The remaining of the paper is structured as follows: Section II presents an overview of related work about routing on the Internet and new network architectures. Section III introduces ETArch concepts. Section IV describes the routing approach and mechanisms defined to ETArch, and finally, Section V presents some concluding remarks and future work.

II. RELATED WORK

Several algorithms like Chandy-Misra, Merlin-Segall, Toueg or Frederickson were proposed to handle routing [9]. Also, according to [9], none of these approaches is good to be used in large scale networks, such as the Internet. In the Internet architecture, two classic approaches were more used: the link state routing and the distance vector routing. Several other algorithms use both approaches or one inspired approach [10].

The approaches adopted by the Internet architecture are based on a weight on the graph edges. The weight is calculated using the distance through number of hops or queuing time [10]. The main difference between both is based on the knowledge of complete network topology or only the directly connected neighbors, as used by the distance vector.

As long as the Internet scaled, different researchers focused on the routing problem by considering the new scales and requirements. One of the proposed approaches is related with the separation of the routing into two different layers using IP addresses [11] or even by rethinking the network layer routing and forwarding [12].

Another approach presented by the research community is the aggregation, where a router aggregates others, and a hierarchical vision is possible with a router being the parent of others. For the use in scalable problems, the aggregation is a interesting approach. The compact routing was also proposed to resolve problems related to big number of data in the routers, where the lines in the routing table grow fast [13][14].

Yang proposed the New Internet Routing Architecture (NIRA) [15] which gives users the ability to choose different sequence of providers in order to routing the primitives. So the users can make this choice taking into account the type of applications. The work proposed here also has this approach in common; however, the routing is affected by the applications requirements considering QoS and also Quality of Experience (QoE) parameters. Besides that, the primitives could be routing using different paths simultaneously or by having an alternative path on its header, that could be used in case of a failure of the initial one [16][17].

On the sensor networks, some protocols have been proposed for the secure routing, by forwarding every network node and creating routing tables by using grouping algorithms [18]. Also, the scaling problems of the Internet were treated in some works, and the routing is presented, and the concern relative to the routing in Future Internet [19].

The works cited present different forms to treat the routing problems at the TCP/IP architecture and other architectures proposed in some works. Also, some architectures focused on Future Internet was proposed like Recursive InterNetworking Architecture (RINA) [20] and ETArch [5]. The latter uses a horizontal address and does not have some TCP/IP characteristics, by separating identifier and localizer. With ETArch, it is possible to apply SDN concepts to use other communications approach prepared to new challenges on the future [21].

The workspace in ETArch is described in the next section. The concept of workspace is similar to other concepts introduced by the community. The Open Network Operating System (ONOS) [22][23] project is using the Intent concept. A similarity of workspace and Intent functions can be found.

ETArch is similar to other projects that treat clean-slate Routing, but the main difference of this work is to propose a routing protocol processing only in the control plane. An example of other project is Mobility First, also a clean-slate approach, but it treats routing using the traditional way, with processing on control and data planes [24][25]. So, the ETArch novelty is the clean-slate architecture able to meet the application requirements.

III. ENTITY TITLE ARCHITECTURE

The creation of the Entity Title Architecture was motivated by some requirements found in the current Internet: Energy Efficiency, Mobility, Multicast, Quality of Service (QoS), Scalable, Security, and others. The problem of routing is associated with several requirements. The architecture concepts were presented on [7][8][21][26].

The routing made in the current Internet architecture considers basically the distance between two hosts, and the packets being transferred knows the final destination (identified by the host). This is problematic because the hosts, which is involved in the communication, should be identified using a

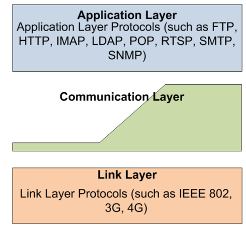

Figure 1. ETArch Layer Architectural Pattern

hierarchical structure. Currently, this identification is made setting an Internet Protocol (IP) address to each host. When a host changes its location, the communication being done is lost, because the identification changed. The idea of set IP therefore is questionable, and no set IP is a problem considering that the current algorithms use IP to transmit the packets from source to destination.

Another remaining problem of the current Internet is the multicast traffic. Today, a communication is made between two hosts, then all packets contain the source and destination IP addresses. A packet sent from the source host does not have a list of destination hosts, but only one IP address relative to one host. It is clear that the routing in new architectures can not to consider the destination host, but what is being located. The separation between identifier and location of a host is present in several Internet protocols by regarding for the Future Internet. A similar concept are the Content Distributed Networks (CDN), where the initial look for a destination is not made looking for a host, but a content, independently of its location.

Energy efficiency and QoS are two requirements to future routing. The new forms of routing to Internet should be able to consider these requirements, and not only best way based on distance between two hosts. It is necessary because one application can order routes that consume less energy or that have different level of QoS.

As mentioned at the Introduction, ETArch does not have a fixed layer like TCP/IP structure. A communication layer was introduced where TCP/IP would be. This logical layer is different of the TCP/IP layer, because as shown in Figure 1, it is flexible depending of the requirements in specific communications. An example is an application that ordering only one requirement (like determined bandwidth, for instance), then the communication layer need to prepare the scenario considering this bandwidth. In another example, an application may request several requirements (like bandwidth, low consumption of energy and QoS, for instance), then the communication layer need to prepare the scenario considering all these requirements.

To create this flexible layer, able to provide communication with different requirements, the ETArch was designed using some concepts, summarized below:

- Entity: Anything that can communicate. So, an Entity can be a host, a user, a process, a link, a mobile phone, and others. Independently if it is responsible to run applications or to control the network.
- Title: it is an unambiguous name to uniquely identify an entity. For example, a workspace has a Title.
- DTS: The Domain Title Service is responsible to control the network. It works like a controller present in SDN, handling the communication and the requirements over time. It is composed by agents named DTS Agent (DTSA) which act as controller.

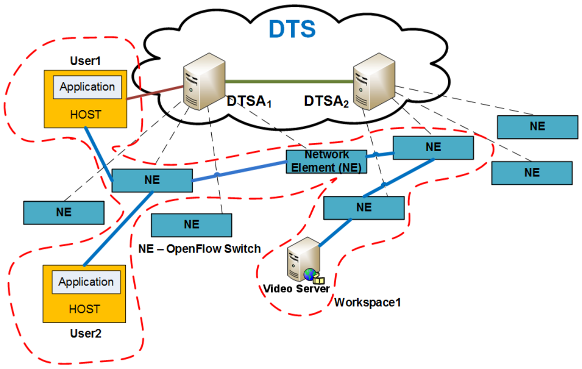

Figure 2. ETArch Workspace Vision

- Workspace: it is a logical bus shared by different entities, where an entity sends and receives information to/from other entities.

The communication in ETArch is made by using the workspace. Workspaces do not exist in initial network setup. If an entity wishes to transmit/provide something, it must create a workspace, which is registered by the DTS through their agents (DTSA). Whether another entity wishes to share something provided by one previously created workspace, it must be linked (attached) with this workspace. This way, all entities which wish to participate to the transmission must be attached on the workspace. The DTS is responsible to control the various workspaces that entities can create and attach. Capabilities are assigned to the workspace such as bandwidth, and so on.

The DTSA has information about the topology of the Network Elements (NE) in its domain. Current works over ETArch use OpenFlow switches as NE, and each NE is controlled by a DTSA. All workspaces created by the entities are stored by the DTS workspace database, then, when an Entity wants to attach in a workspace, the DTSA has enough information about the specified workspace.

The first problem to be treated in ETArch is when a given DTSA does not have information about a workspace, which has been registered in another DTSA. Thus, the routing protocol proposed in this paper starts by the problem of finding a workspace which is being seeked by a DTSA.

Figure 2 shows an example of an existing workspace. In the figure, two DTS Agents ($DTSA_1$ and $DTSA_2$) are controlling some NE. Note that each DTSA has topology information about all NE in its domain. A Video Server is connected to a NE controlled by $DSTA_2$. The Video Server is an entity and it is streaming a video using $Workspace_1$. When the entity named $User_1$ decides to join (attach) in $Workspace_1$, and when it occurs all information sent by the Video Server will be forwarded to $User_1$. In implementations on the ETArch, when the information about the attaching of $User_1$ with $Workspace_1$ arrives on the DTS (control plane), the two DTSAs configure rules on NE to forwarding all data of $Worskapce_1$ to specific ports, so that data goes to the $User_1$. At this moment, the data plane has only one consumer entity and it seems an unicast connection. If the entity $User_2$ decides to join (attach) to $Workspace_1$ the same behavior is made, and the DTSAs configure rules to forwarding the data to User1 and User2. Look that the traffic is split in the NE around the consumer entities.

Thus, at ETArch, the problem on routing has an additional problem and it is divided in two parts: i) how to find a workspace, as it is not fixed as hosts in the Internet; and, ii) how to include the Entity requirements in the best route choice.

Current works on ETArch and their implementations use the distance to define de best route. It works, however, to continue evolving the architecture is necessary a routing protocol prepared to use the requirements required by the applications. An Entity can order low energy consumption, a minimal bandwidth or specific QoS rule. So, the routing protocol proposed on the next section is prepared to work with requirements in different levels. A level can be low, medium or high, ie, the Entity can order a requirement like low distance and high energy consumption: it means that the best route to this Entity should be a low energy consumption (because is a high level for the Entity), and the distance need not be the lowest considering the topology. Between low, medium and high levels, the DTSA decides the best path to the workspace be expanded to the Entity.

A workspace allows the DTS knows the requirements of the applications and adjust network to satisfy these requirements. In current Internet, the application layer is responsible to treat requirements, and take it into the network is the main contribution when using workspace concept.

ETArch currently does not provide communication between different DTSAs, and the routing protocol proposed on the next section is prepared to provide an Entity joins in a workspace in other DTSA. The problem with two DSTA is how the DTSA searchs a workspace that is not in its local database.

IV. Routing Algorithm Specification

As can be seen in Equation 1, the processing time (Tp) of a router can be considered the sum of routing time (Tr) and the sum of switching time (Ts), to all n packets. It means that each data in a router has the processing to determine the route and the switch times of the packet.

$$Tp = \sum_{i=1}^{n} Tr_i + \sum_{i=1}^{n} Ts_i \qquad (1)$$

The idea of the new approach proposed here is to the Tp in the data plane is only the sum of Ts. In this way, the sum of Tr must be zero. Since the entire route can be defined on control plane, before the communication starts, ETArch permits to use the SDN concepts to the data plane be only responsible for switching.

In ETArch, the workspace concept permits routing considering several requirements, like number of hops, bandwidth, energy consumption and others. In the architecture, the Entity calls the DTSA asking by a given workspace, and the DTSA looks for the workspace and extends it, through several NE to reach the Entity.

The idea proposed in this paper is to use the workspace concept to reach the destination. When an entity wants receive some data it must look for a workspace. Then, it sends to DTS (the controller) the request for the workspace attach. The DTS attaches the entity on it, and the transmissions over the workspace are sent to the entity.

Previous works [5][26] explain how these rules are configured in the NE to extend a workspace through the network.

Here is proposed a protocol design to find a workspace by considering DTSA topology and the link of several DTSA.

The routing protocol runs over control plane. It was built considering two situations: when a workspace is in the same DTSA as the Entity that wishes attach; and when a workspace is not in the same DTSA. The first situation is named intra-DTSA routing and the second is named inter-DTSA, similar to inter-networking or inter-domain used in current networks and topologies.

A. Intra-DTSA Routing

The intra-domain routing happens when an Entity requests an attach with a workspace and the DTSA which controls the Entity has enough information about the specified workspace. This information can be: the workspace was created by an Entity plugged in that DTSA; or some Entity in that DTSA is already attached with the workspace. Then the intra-DTSA needs only to extend the workspace, i.e., to inform all NE on the way the new rules to forwarding data to the Entity.

The DTSA must be prepared to recognize Entity requirements and to decide the best path. It can be stated that the intra-DTSA routing is similar to the link state mechanism.

B. Inter-DTSA Routing

The inter-DTSA routing happens when an Entity requests an attach with a given workspace and the DTSA, which controls the Entity, has no information about the specified workspace. In this case, the first step is to look for the workspace.

For this propose, the first service to be specified is WORKSPACE_LOOKUP. This message is sent from a DTSA to its Master. A Master DTSA is the resolver of several DTSAs and has information about all workspaces created or extended by any Entity.

The Figure 3 shows some DTSAs linked with a Master-DTSA (D_1). The control informations sent by the DTSA, as WORKSPACE_LOOKUP, for instance, are sent through a control workspace. This is a private control workspace that creates a bus between some DTSAs and their Master. When a DTSA receives the attach information from an Entity and it does not have information about the workspace, it sends a WORKSAPACE_LOOKUP to the Master-DTSA.

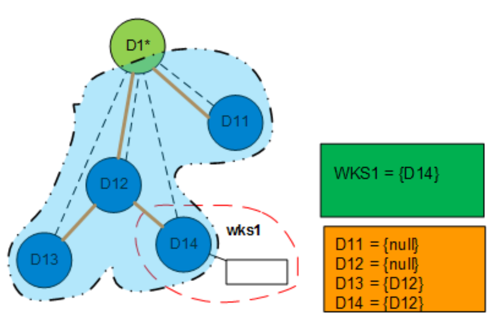

Figure 3. Private Control Workspace

The protocol has an algorithm to decide the best route. It starts when a WORKSPACE_ATTACH message arrives in the DTSA. The Master-DTSA has information about all workspaces extended or created by any NE in its domain. When a workspace is requested and the Master does not have knowledge about the workspace, it should forward the message to other Master DTSAs. A Master DTSA has a control workspace with other Master DTSAs as shown in the Figure 4, where the Masters D_1 and C_1 are connected by a control workspace. This control workspace is named Public and two or more Master-DTSAs are linked by this workspace, and the level of this linking can be worldwide.

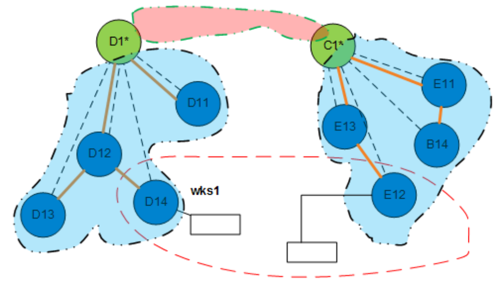

Figure 4. Public Control Workspace

It is important to note that there are two timeouts when WORKSPACE_LOOKUP is sent. The first is the max time to the DTSA to receive the first path. The second is the max time that the DTSA waits more paths when it already received the first path. It is because a request can result zero, one or more responses.

When the DTSA receives responses, containing a list of paths to extend the workspace, it must decide the best path considering the Entity requirements. Thus, a list of paths works as a routing arguments, being that each member of the list contains paths and information about capabilities. Upon deciding the best route, the DTSA must send a WORKSPACE_CONFIGURATION to each DTSA in the path. This service contains information to each DTSA in the path to extend the workspace until the Entity. When a DTSA receives a WORKSPACE_CONFIGURATION it must: analyzes its topology and decides a route in its NE to extend the workspace. After this, it must send to each NE on the path, rules containing workspace name and new port of out. After this the workspace is extended by that DTSA and the DTSA inserts this information in its local database for next requests does not need new WORKSPACE_LOOKUP.

The execution order of the protocol algorithm in the DTSA is presented in Figure 5. The functions invoked in pseudo code are executed by the DTSA itself, and they can modify the object wksInfo, which maintain informations about the workspace and routes. Also, in the procedure onReceiveWorkspaceAttach, some functions need of specific service messages on the routing protocol, like notifyMaster and requestConfiguration.

The notifyMaster function exists because all workspace extended in some DTSA must be notified to the Master. Thereby, the Master-DTSA can insert in its local database and, when a lookup arrives in Master, it has enough information about the new workspace extended in its topology. Note that Master-DTSA is a server and maintain a lot of information about workspaces in its topology. These information can be stored by using common or distributed databases. Some procedures called in Figure 5 are not presented here and future works can implement the procedure using different techniques. However, two procedures are important to present: lookup and requestConfiguration, because both need specific messages on routing protocol.

Figure 6 shows lookup procedure and 7 shows requestCon-

```
procedure onReceiveWorkspaceAttach(title)
    attached ← true
    wksInfo ← queryInLocalDB(title)
    if wksInfo ≠ null then
        if checkRequirements(wksInfo) then
            attached ←calculateRoute(wksInfo)
            attached ←updateFlowTables(wksInfo)
            attached ←insertIntoLocalDB(wksInfo)
            attached ←notifyMaster(wksInfo)
            attached ←calculateRoute(wksInfo)
                        ▷ wksInfo can be modified in each call
        else
            attached ← false
        end if
    else
        pathList ← lookup(title)
        if pathList ≠ null then
            path ← checkBestPath(pathList)
            attached ← requestConfiguration(path)
            attached ←calculateRoute(wksInfo)
            attached ←updateFlowTables(wksInfo)
            attached ←insertIntoLocalDB(wksInfo)
            attached ←notifyMaster(wksInfo)
            attached ←calculateRoute(wksInfo)
        else
            attached ← false
        end if
    end if
    return attached
end procedure
```

Figure 5. Routing Algorithm Procedure

figuration functions. Both are executed in Master-DTSA.

In the procedure shown in Figure 6, the checkTopology is invoked for the Master-DTSA verifies if its own topology supports the requirements specified by the Entity. The call for lookup is necessary if the Master does not contain informations about the required workspace. The addOwnPath routine is responsible to verify if the DTSA path is related only to that Master.

```
procedure onWorkspaceLookup(title)
    pathList ← queryInLocalDB(title)
    if pathList ≠ null then
        finalList ← pathList
    else
        supported ← false
        supported ← checkTopology()
        if supported then
            pathList ← lookup(title)
            if pathList ≠ null then
                finalList ← addOwnPath(pathList)
            end if
        end if
    end if
    return finalList
end procedure
```

Figure 6. WORKSPACE_LOOKUP Procedure

The procedure requestConfiguration shown in Figure 7 can run in a DTSA or in a Master DTSA. When it runs in Master-DTSA, this will notify all DTSAs in its topology that they are

```
procedure requestConfiguration(wks, entity)
    routeInformation ← defineRoute()
    updateFlowTables(routeInformation)
    insertIntoLocalDB(routeInformation, wks)
    notifyMaster(wks)
end procedure
```

Figure 7. WORKSPACE_CONFIGURATION Procedure

on the chosen path, and notify the public control workspace to other Master DTSAs notify their DTSAs that they are on the path. The requestConfiguration can also arrive in a DTSA, and in this case the procedure shows the behavior as shown in the Figure 7.

Note that, in the requestConfiguration procedure, presented in Figure 7, the wks and entity are objects containing informations respectively about the workspace and the Entity that required the attach. The function defineRoute is responsible to define the best path considering all of the NE controlled by the DTSA. With this information, it is possible run updateFlowTables, a function where the DTSA sends to each NE in the route rule information to be inserted in flow tables of the NEs.

Considering the procedures, and the functions in each procedure, Table I presents the new primitives that are related with routing at ETArch.

TABLE I. ETArch ROUTING RELATED PRIMITIVES.

Message	*Description*
WORKSPACE_LOOKUP	Used when a DTSA does not have information about a workspace sought.
DTS_NOTIFY	Send from DTSA to Master. It is responsible to inform that a new workspace was created or extended by a DTSA in that Master topology.
WORKSPACE_CONFIGURATION	A DTSA sends this message to the private control workspace informing new extension of a workspace. Each DTSA that receives this message should configure the NE in its own topology necessary for this extension.

The procedures of routing protocol send the messages specified in Table I to the public or private control workspace. When a message, with one of these headers, arrives to a NE, it is forwarded by the control workspaces to a DTSA or Master-DTSA.

With the procedures and messages described in this section, it is possible to ETArch makes communication with two or more DTSAs, i.e., several controllers, defining the route on control plane, before the data starts to be forwarded in data plane. It is important to mention that the parameter attached found in Figure 5 means: the workspace is extended until the requesting Entity, i.e., the Entity is now attached with the workspace.

All procedures presented are subject to a rollback whether one of the functions fails.

V. Concluding Remarks and Future Work

The control plane of the ETArch architecture is provided by the DTS, which is implemented through one or more agents named DTSAs. Each DTSA is responsible by a set of network elements whose topology is driven by local communication needs. In this work, by regarding routing, ETArch procedures and messages were described in order to provide communications using different DTSAs. It is important to remark that, up to this moment, all the communications were made using only one DTSA, i.e., there was no structure to support a workspace over two or more DTSAs.

This paper presented the specification and design of the ETArch routing control plane procedures e rules. The proposed protocol will be incorporate with ETArch modules already deployed. For future works, the requirements as QoS, bandwidth, secure and energy consumption will be incorporated to provide scenarios where it will be possible to take tests and experiments to run the specified algorithms presented here.

Despite of the current work be made for ETArch, it can be applied to the research in SDN as a possible scenario for the routing by using the control plane, by improving the processing time spent in the routing over the current Internet. For the ETArch, this work provides the capability of using different DTSAs (controllers) in the workspace communications. This is an important feature, because, for example, it is not interesting for a carrier providing information about its topology and network elements.

The work presented here follows the SDN concepts, separating the control and data plane, and the routing proposed here is able to define the complete path before the data starts being forwarding. The routing time is spent during the path establishment and the communications phase spend only switching time, differently of today's Internet.

ACKNOWLEDGMENT

This work has been partially funded by the Brazilian agencies: CAPES; CNPq; FAPEMIG; and PROPP/UFU - and by ALGAR Telecom.

REFERENCES

[1] V. G. Cerf and E. Cain, "The DoD internet architecture model," Computer Networks (1976), vol. 7, no. 5, Oct. 1983, pp. 307–318.

[2] T. Zahariadis et al., "Towards a future internet architecture," in The Future Internet. Future Internet Assembly 2011: Achievements and Technological Promises, ser. Lecture Notes in Computer Science, J. Domingue, A. Galis, A. Gavras, T. Zahariadis, and D. Lambert, Eds. Berlin, Heidelberg: Springer-Verlag, 2011, vol. 6656, pp. 7–18.

[3] N. S. Foundation. NSF future internet architecture project. [Online]. Available: http://www.nets-fia.net/ [retrieved: Mar., 2015]

[4] Eurescom. Future internet assembly - european future internet portal. [Online]. Available: http://www.future-internet.eu/home/future-internet-assembly.html [retrieved: Mar., 2015]

[5] J. de Souza Pereira, F. de Oliveira Silva, E. Filho, S. Kofuji, and P. Rosa, "Title model ontology for future internet networks," in The Future Internet, ser. Lecture Notes in Computer Science, J. Domingue et al., Eds. Springer Berlin Heidelberg, 2011, vol. 6656, pp. 103–114.

[6] F. Silva et al., "Entity title architecture extensions towards advanced quality-oriented mobility control capabilities," in Computers and Communication (ISCC), 2014 IEEE Symposium on, June 2014, pp. 1–6.

[7] M. A. Goncalves, P. F. Rosa, and de Oliveira Silva., "Multicast traffic aggregation through entity title model," in The Tenth Advanced International Conference on Telecommunications (AICT), ThinkMind, Ed. IARIA, 2014, pp. 170–180.

[8] C. Guimaraes, D. Corujo, F. Silva, P. Frosi, A. Neto, and R. Aguiar, "Ieee 802.21-enabled entity title architecture for handover optimization," in Wireless Communications and Networking Conference (WCNC), 2014 IEEE, April 2014, pp. 2671–2676.

[9] W. Fokkink, Distributed Algorithms: An Intuitive Approach. Cambridge, Massachusetts: The MIT Press, Dec. 2013.

[10] D. J. W. A. S. Tanenbaum, Computer Networks, international ed of 5th revised ed edition ed. Harlow, Essex: Pearson Education Limited, Jul. 2013.

[11] D. Massey, L. Wang, B. Zhang, and L. Zhang, "A scalable routing system design for future internet," in Proc. of ACM SIGCOMM Workshop on IPv6, 2007.

[12] K. Calvert, J. Griffioen, and L. Poutievski, "Separating routing and forwarding: A clean-slate network layer design," in Fourth International Conference on Broadband Communications, Networks and Systems, 2007. BROADNETS 2007, Sep. 2007, pp. 261–270.

[13] F. Le, G. G. Xie, and H. Zhang, "On route aggregation," in CoNEXT, K. Cho and M. Crovella, Eds. ACM, 2011, p. 6.

[14] S. Strowes, Compact Routing for the Future Internet. University of Glasgow, 2012.

[15] X. Yang, D. Clark, and A. W. Berger, "Nira: A new inter-domain routing architecture," IEEE/ACM TRANSACTIONS ON NETWORKING, 2007.

[16] I. A. Ganichev, "Interdomain multipath routing," Ph.D. dissertation, EECS Department, University of California, Berkeley, Dec 2011.

[17] G. T. Nguyen, R. Agarwal, J. Liu, M. Caesar, P. B. Godfrey, and S. Shenker, "Slick packets," SIGMETRICS Perform. Eval. Rev., vol. 39, no. 1, Jun. 2011, pp. 205–216.

[18] B. Parno, M. Luk, E. Gaustad, and A. Perrig, "Secure sensor network routing: A clean-slate approach," in Conference on Future Networking Technologies (CoNEXT), December 2006.

[19] K. L. Calvert, J. Griffioen, and L. Poutievski, "Separating Routing and Forwarding: A Clean-Slate Network Layer Design," in In proceedings of the Broadnets 2007 Conference, September 2007.

[20] Y. Wang, F. Esposito, I. Matta, and J. Day, "Recursive InterNetworking Architecture (RINA) Boston University Prototype Programming Manual (version 1.0)," CS Department, Boston University, Tech. Rep. BUCS-TR-2013-013, November 11 2013.

[21] F. de Oliveira Silva, M. Goncalves, J. de Souza Pereira, R. Pasquini, P. Rosa, and S. Kofuji, "On the analysis of multicast traffic over the entity title architecture," in 2012 18th IEEE International Conference on Networks (ICON), 2012, pp. 30–35.

[22] ON.LAB. ONOS - open network operating system. [Online]. Available: http://tools.onlab.us/onos.html [retrieved: May, 2014]

[23] ——. ONOS at ONS 2014. [Online]. Available: http://www.slideshare.net/ON_LAB/onos-at-ons-2014 [retrieved: Mar., 2014]

[24] 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. 13.

[25] S. C. Nelson, G. Bhanage, and D. Raychaudhuri, "GSTAR: generalized Storage-Aware routing for mobilityfirst in the future mobile internet," in Proceedings of the sixth international workshop on MobiArch, ser. MobiArch '11. New York, NY, USA: ACM, 2011, p. 1924.

[26] J. C. Lema et al., "Evolving future internet clean-slate entity title architecture with quality-oriented control plane extensions," in The Tenth Advanced International Conference on Telecommunications (AICT), ThinkMind, Ed. IARIA, 2014, pp. 161–167.

