Management System Scalability Evaluation in Multi-domain Content Aware Networks

Eugen Borcoci, Mihai Constantinescu, and Marius-Constantin Vochin
University POLITEHNICA of Bucharest
Bucharest, Romania
emails: eugen.borcoci@elcom.pub.ro
mihai.constantinescu@elcom.pub.ro, marius.vochin@elcom.pub.ro

Abstract — **This paper studies and evaluates the scalability properties of a resource management subsystem as a component of a networked media eco-system. The overall system aims to offer multimedia delivery with configurable guarantees of Quality of Services, over multi-domain networks, to large communities of users. The transport infrastructure is based on creation of data plane logical slices named Virtual Content Aware Networks. These slices are realized under control of a management plane, comprising centralized per-network domain "controllers", cooperating to construct parallel data planes, which span multiple IP network domains independently managed. In this multi-domain context, the management system scalability properties are important, while the problem is similar to the multi-controller inter-communication in emerging multi-controller Software Defined Networking technologies. The management system architecture considered in this paper has been previously defined. This work additionally provides a simulation model and results, concerning the scalability of the multi-controller communications subsystem. It is shown that the proposed control approach is conveniently feasible in a multi-domain network environment.**

Keywords — Content-Aware Networking; Software Defined Networking, Multi-domain; Management; Resource provisioning; Future Internet.

I. INTRODUCTION

Many novel architectural solutions are proposed for the Future Internet and these are still open issues for research. Their major target is to solve some of the recognized fundamental architectural limitations of the traditional Internet, and reduce its ossification, while answering better to the current challenges related to new services needs and also offering a better support for the Internet global extension [1][2][3].

A significant trend recognized in the current and also estimated for the future Internet is a strong information/ content-centric orientation, including media content distribution. Consequently, changes in high level services and networking have been recently proposed, including modifications of the basic architectural principles. Some "revolutionary" approaches are often referred to, as *Information Centric Networking (ICN)*, or equivalent, *Content Oriented/Centric Networking (CON/CCN)* [4][5].

In parallel, evolutionary (or incremental) solution emerged, introducing Content-Awareness at Network layer (CAN) and Network-Awareness at Applications layers (NAA). This approach increases the amount of information on the flows transported at network level (compared to the content-agnostic IP) and provides summary information about transport characteristics to the application/services layers. Thus, a powerful cross-layer optimisation loop can be created between the transport on one side and applications and services on the other side.

An "orthogonal" new trend, targeting to achieve more flexibility and programmability in networking is the *Software Defined Networking* (SDN) architecture and its associated OpenFlow protocol [6][7][8]. In SDN the control plane and data planes are decoupled and the network intelligence is more centralized in so-called SDN controllers, thus offering possibilities of enhanced management and also flexible/programmable control of the resources. However, applying the SDN centralized control in a wide area network (WAN) context does not scale [9]. Several controllers are required, each one controlling a limited network region. In order to get an overall logic view upon the whole network, it is necessary an inter-controller communication subsystem. The scalability of this is an open research issue, and is also studied in this work.

The European FP7 ICT research project, *"Media Ecosystem Deployment Through Ubiquitous Content-Aware Network Environments"*, ALICANTE [10], adopted the NAA/CAN approach, to define, design, and implement a Media Delivery Ecosystem, spanning multiple network domains.

This work considers as a basis the ALICANTE management architecture [10][11], which is, conceptually, partially similar to a multi-controller SDN architecture, with respect to the distribution of the main management and control functions among several controllers. Communication between controllers is necessary in order to accomplish multi-domain tasks.

This paper is *a continuation and extension of the work presented in* [1]. It has extended the scenarios to study the scalability aspects of the ALICANTE multi-controller signaling subsystem (where each controller independently allocates transport resources in its managed domain). The simulation approach has been based on the *Extended Finite State Machines* (EFSM) model [12]. Note that the main task of the studied subsystem is to provide resources for a multi-domain Virtual Content Aware Network (VCAN) constructed at request of a Service Provider (SP) entity, which will be the actual VCAN "user".

We recall that scalability of a system (in case that this system can be abstracted to a graph) can be seen from two points of view: horizontal and vertical scaling. Scaling *horizontally* (or *scaling out*) means that the system allows addition of more “nodes” to it (e.g., adding a new computer to a distributed software application), while still preserving an approximate linear increase in cost and complexity. Scaling *vertically* (or *scaling up*) means that the system allows addition of resources to a single node in a system (e.g., addition of CPUs or memory to a single node). *In this study, the target is the horizontal scalability properties of the inter-controller communication subsystem.*

Section II presents samples of related work. Section III shortly describes the multiple-domain resource management architecture considered in this study. Section IV defines the inter-controller communication subsystem. Section V introduces the simulation model to study the signaling scalability and Section VI presents samples of the simulation results. Conclusion, open issues, and future work are shortly outlined in Section VII.

II. RELATED WORK

The ICN/CCN/CON approaches are very promising for the future Internet [2][3][4][5]. However, they raise some research and especially deployment challenges, given novel paradigms proposed, concerning naming and addressing, content-based routing and forwarding, management and control framework, in-network caching, etc. Other issues are related to scalability, especially related to the amount of processing and resources in the ICN routers and also security. ICN/CCN/CON technologies require significant changes of the current Internet deployments and protocols.

That is why, some evolutionary (incremental) approaches have been proposed such as CAN/NAA, built as overlays upon existing Internet infrastructures. These evolutionary solutions are (hopefully) able to support in a better way the seamless development of the networked media systems and also the market orientation towards content, while paving the road to full ICN/CCN.

Content Delivery Networks (CDN) [13], using large physical and logical infrastructures, provide improved content related services based on content replication. They offer fast and reliable applications and services by distributing content to cache or edge servers located close to users. A CDN is a collection of network elements arranged for more effective delivery of content to end-users. The typical functionality of a CDN includes [13]: content outsourcing and distribution services to replicate and/or to cache content to distributed surrogate servers, on behalf of the origin server; request redirection and content delivery services to direct a request to the closest suitable surrogate server; content negotiation services to meet individual user needs; management services for network components, accounting, and to monitor and report on content usage. Different from ICN/CCN, the CDNs can use the current Internet support; they are largely developed in the real world. However the CDNs need a large number of powerful servers and should apply sophisticated procedures related to content distribution and caching policies.

In SDN [6][7][8][14], the main network intelligence is centralized in SDN controllers. Such an approach offers a better and flexible control of the resources, quality of services, etc., due to the possibility to have an integrated/overall knowledge of the network capabilities in the control plane and by allowing programmability of the network resources (forwarding plane). To this aim, standardized south-bound (between controller and forwarding devices – e.g., like OpenFlow) and north-bound interfaces (API for applications) are currently developed for the SDN “layer”. Therefore, operators and service providers will get more freedom and gain speed in developing their services, without waiting long time for new releases of vendor’s networking equipment.

Although SDN technology seems to be very attractive, e.g., for data centers but also for core wide area networks, it exposes also many research challenges and open issues, both from architectural and from deployment point of view. The degree of centralization and relationship with scalability and reliability are examples. An extension of the SDN concepts is proposed in so-called *Software Defined (Internet) Architecture* [14], where the idea is to decouple the architecture from infrastructure, aiming to lower the barriers to architectural evolution. The SDIA approach tries to exploit SDN concepts but also traditional technologies (e.g., Multi Protocol Label Switching MPLS, software forwarding-performed on edge routers, etc.) in order to obtain evolvable architectures. SDN and SDIA are still evolutionary in contrast with those technologies called “clean slate” - like ICN/CCN, which are architecturally disruptive.

Currently, there exist concerns about SDN’s performance, scalability, and resiliency [7][9][15], the main source for these problems being the centralization concept. It is clear that a central controller will have a limited processing capacity and the solution will not scale as the network grows (increased number of switches, flows, high bandwidth needs, etc.). The controller’s performance can be increased, but a second solution is to define a SDN multi-controller architecture. However, SDN still has as objective to get a consistent centralized logical view upon the network; this creates a need for controllers *to cooperate and synchronize their data bases,* in order to provide together a consistent view at network level. Work in progress is developed at IETF towards defining an inter-controller communication system [16]. While, with respect of the vertical protocols between Control and Data Plane we have seen significant progress in specifying Open Flow versions [8], and implementing several types of controllers [7][15], the inter-controller cooperation and scalability issues are still under research.

In order to identify the degree of interest for the problems to be studied in this work, we shortly describe below the relationship of the approach considered here, to the above solutions (ICN/CDN/SDN). The full architecture has been defined in ALICANTE project [10], being a mid-way solution among ICN/CDN/SDN, given that:

- it is an architecture CAN/NAA oriented, overlapping with ICN/CCN in the sense that some degree of content awareness exists at the network element level; due to

this, different levels of QoS guarantees can be offered to the users;
- it has CDN similarities, given that it is oriented to content delivery with QoS guarantees; however it is cheaper, given that it does not need necessarily to replicate content in many caching servers;
- it adopted (for the multi-domain management) the SDN partial centralization concepts where several controllers exist, each managing a region and cooperating with others. In such a way, in our system, an overall image of the multi-domain resources always exists at management level, similar as in SDN.

In particular, the work of this paper is focused on the scalability of the inter-controllers communication protocols, when the number of the involved network domains (to support a requested VCAN) is variable.

III. Multi-domain Management System Architecture

Additional details of the ALICANTE architecture aside of the multi-domain resource managers (controllers) can be found in [10], [11], [17].

In this architecture, several cooperating *environments* are defined, containing business entities/actors:
- User Environment (UE), containing the End-Users;
- Service Environment (SE), containing Service Providers (SP) and Content Providers (CP);
- Network Environment (NE), where we find a novel business entity called CAN Provider and also the traditional Network Providers (NP) managing the network elements, in the traditional way at IP level.

The “environment” is defined as a generic grouping of functions, working for a common goal and, which, possibly, might vertically span one or more several architectural (sub-) layers.

In the Data Plane the logically isolated VCANs are realized as parallel logical data planes, constructed by optimising inter and intra-domain mapping of VCANs, onto several domain network resources. The content awareness is realized by special edge routers called Media Aware Network Elements (MANE) while inside the network domain regular core routers are used. An early design decision has been adopted: VCANs are limited to core networks; they do not span the access and local networks. The reason is related to high variability in edge networks technologies and large variation of the methods to control the connectivity resources in the first and last mile segments of an end to end (E2E) chain.

In the following text of this paper only the management and control plane issues will be discussed as they are the main objectives of this study.

Dynamic (negotiation-based) Service Level Agreements (SLA) can be established between actors. Using SLAs, several Service Providers can independently ask, to a CAN Provider to perform the required resources provisioning for customizable Virtual Content Aware Networks, and then SPs may use them for media flow transport. Network Providers can cooperate to VCAN construction but they still preserve their independency in terms of their own resource allocation for each VCAN requested and constructed. Flexible connectivity services have been achieved in terms of QoS, offering: Fully/partially/un- managed services [17].

The architecture supports both vertical and horizontal integration in terms of SLAs, to offer (edge-to-edge) several levels of guarantees. A partially distributed Management and Control (M&C) plane exists – where each domain has its own Intra-domain Network Resource Manager (Intra-NRM) and an associated CAN Manager). This structure supports all actions for large scale provisioning.

The VCANs are flexible in the sense that they can support (simultaneously) several communication modes: unicast, multicast, broadcast, P2P and combinations with different levels of QoS/QoE, availability, etc. [18]. End Users and some residential gateways called here Home-Boxes can ultimately benefit from CAN/NAA features by using VCANs.

Services Providers are not burdened with tasks to construct the VCANs; they simply ask VCANs (with some characteristics - topology, capacities of traffic trunks, QoS classes, etc.) to be provisioned by a CAN Provider. In case of success of the SLA negotiation between SP and CANP, the VCAN configurations are installed in the routers and then the Service Provider can use the customized connectivity services. The architecture assures QoS and Quality of Experience (QoE) optimization based on: CAN/NAA interaction; cooperation between resource provisioning (based on SLA agreement) and media flow adaptation; hierarchical monitoring at CAN and network layers cooperating with the upper layers. Apart from its focus on media flow delivery with guaranteed QoS, the ALICANTE architecture is enough general as to transport also non-characterized flows of traffic, in “best effort” style. In this way the architecture provides an answer to “network neutrality” requirements.

The simplified ALICANTE VCAN management architecture is presented in Figure 1. The picture only presents the Service Environment (instantiated here by the Service Provider) and Network Environment (the assembly of the bottom blocks). The User Environment is not represented, given that it is not directly involved in VCAN management.

The management and control architecture is conceptually similar to SDN, although not following full SDN specifications (actually the ALICANTE architecture does not use an OpenFlow like protocol). Both architectures are evolutionary and can be seamlessly developed. The Control Plane and Data Plane are separated. Note that Control Plane in SDN terminology is here actually Management and Control Plane. The functionalities like QoS constrained routing, resource allocation, admission control and VCAN mapping are included in the CAN Manager.

The “virtualization” of the network is performed by Intra-domain Network Resources Managers (Intra-NRM), which hides the characteristics of MPLS technology by

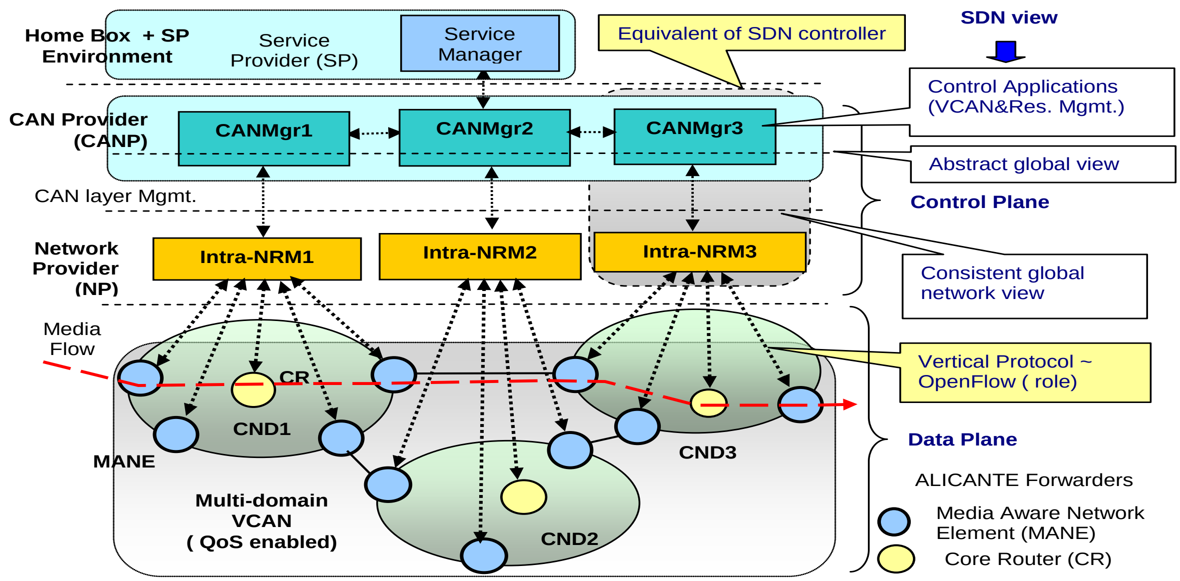

Figure 1. ALICANTE partially centralized management architecture and equivalence with SDN
Notations: Service Environment: SP – Service Provider; Network Environment: CANP - CAN Provider; NP – Network Provider; CND - Core Network Domain; CANMgr - Content Aware Network Manager; Intra-NRM – Intra-domain Network Resource Manager; MANE – Media Aware Network Element

delivering to the CAN Managers an image of abstract matrix of connectivity logical pipes.

In our case the [*CAN Manager + Intra-domain Network Resource Manager*] play together the role of an SDN controller for a network domain, controlling the Media Aware Network Elements (MANE) edge routers and interior regular core routers. Actually, we have a multi-domain logical network governed by several "SDN controllers", – which cooperate for resource management and routing. However, the degree of centralization is configurable in ALICANTE by defining any placement of CAN Managers, the network regions and sets of routers to be controlled.

In both SDN and this architecture the Control Plane software is executed on general purpose hardware. The decoupling of the control with respect to specific networking hardware is realized in the sense that MANE, core routers and network links are viewed by the upper CAN layer in abstract way.

The Data Plane is programmable: all configurations for MANE and Core routers are determined at CAN level by Management and Control (M&C) and then downloaded in the routers. ALICANTE architecture defines the control for a whole network (and not for single network devices): at CAN Manager level there exists an overall image on the static and dynamic characteristics of all VCANs; at Intra-NRM level there is a full control on the network domain associated with that Intra-NRM.

In SDN and our case also, the network appears to the applications and policy engines as a single logical switch. In our case, the network appears at higher layers as a *set of parallel planes VCANs*. This simplified network abstraction can be efficiently programmed, given that the VCANs are seen at abstract way; they can be planned and provisioned independently of the network technology.

IV. Inter-domain Management Communications

One CAN Manager (belonging to CAN Provider) is the initiator of VCAN construction, at request of a Service Provider. The VCANs asked should be mapped onto real multi-domain network topology, while respecting some QoS constraints. This provisioning is done through negotiations [11], performed between the initiator CAN Manager and CAN Managers associated to each network domain. In other words, if necessary, the initiator communicates with other CAN Managers, to finally agree: first, transport resources reservation and then a real allocation (i.e., installation in the network routers) of network resources necessary for a VCAN.

A CAN planning entity inside each CANMgr runs a combined algorithm, performing QoS constrained routing, VCAN mapping and logical resource reservation. In this set of actions, it is supposed that the initiator CANMgr knows the inter-domain topology at an overlay level and also a summary of each network domain topology, in terms of abstract trunks (e.g., *{ingress, egress, bandwidth, QoS class, …}*). This knowledge is delivered by an additional discovery service, whose description is out of scope of this paper. Previous paper [17] has proposed and presented the development and implementation of the combined VCAN mapping algorithm.

The overall system flexibility and scalability essentially depends on its Management and Control. For VCAN planning, provisioning and exploitation, it was adopted per-domain partially centralized solution; this avoids full-centralized VCAN management (non-scalable), but allowing a coherent per-domain management. However, the initiator CAN Manager (like in SDN approach), has the overall consistent image of a multi-domain VCAN.

There is *no per-flow signaling* between CAN Managers. The VCAN related negotiation between SP and CANP (concluded by a SLA) is performed per each VCAN, described in terms of topology and aggregated traffic trunks. The SP negotiates its VCAN(s) with a single CAN Manager irrespective, if it wants a single or a multi-domain spanned VCAN.

An hierarchical overlay solution is applied for inter-domain peering and routing [18], where each CAN Manager knows its inter-domain connections. The CAN Manager initiating a multi-domain VCAN is the coordinator of this hierarchy. It knows the inter-domain topology but does not have to know details on each network domain resources to be allocated to that VCAN. This is a realistic assumption, given the autonomy of each network domain. That is why a negotiation is necessary between the initiator CAN Manager and other CAN Managers involved – to check if the local resources are sufficient. The monitoring at CAN layer and network layer is performed at an aggregated level.

Figure 2 shows an example of inter CAN Managers signaling (i.e., inter-controllers in SDN terminology).

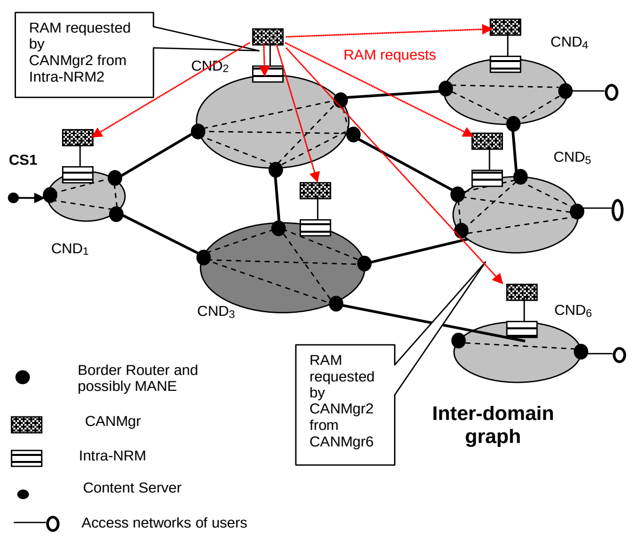

Figure 2. CAN Manager 2 issues random access memory (RAM) requests from each CAN Manager involved in a VCAN
Notations: CS- Content Server ; CND - Core Network Domain; CANMgr - Content Aware Network Manager; Intra-NRM – Intra-domain Network Resource Manager; RAM – (Domain) Resource Availability Matrix; MANE – Media Aware Network Element; CANMgr2- VCAN initiator

Note that in the initial VCAN request from the SP (to the initiator CAN Manager), the ingress and egress points of the VCAN are specified; additionally the initiator CAN Manager knows the inter-domain graph, so it can determine (in a first phase), which domains are involved in this VCAN. This is an internal algorithm running inside the initiator CAN Manager, whose details are out of this paper scope. The result of this is that the initiator CANMgr knows which domains are candidates to support this VCAN, so it knows to whom it should communicate and ask for resources information.

As an example, in Figure 2 the initiator CAN Manager 2, attached to Core Network Domain 2 asked (in hub style) the other involved CAN Managers (1, 3, 4, 5, 6) to deliver to it their network Resource Availability Matrices. Based on the received information, the initiator performs the VCAN mapping.

V. SIMULATION MODEL

The objective of this paper is the evaluation of the Management and Control signaling overhead, related to the negotiation activities between the actors: SP, CAN Managers, Intra-NRMs, when the number of network domains and CAN Managers is a variable parameter.

Given the complexity of the M&C subsystem, a simulation study has been developed. *Real Time Developer Studio* is a *Specification and Description Language* (SDL) simulator, developed by PRAGMADEV. It comes in two versions: SDL and SDL Real Time (SDL-RT) [19].

SDL-RT is based on ITU standard SDL (using *Extended Finite State Machine Model* - EFSM), extended with real time concepts. It is object oriented, has associated a graphical language and allows modeling real-time features. It combines static, dynamic and representations, supporting classical real time concepts, extended to distributed systems, based on standard languages. It retains the graphical abstraction brought by SDL while keeping the precision of traditional techniques in real-time and embedded software development. In SDL-RT, the C language is used to define and manipulate data. Therefore, it allows re-using legacy code written in C language.

The ALICANTE management simulation model consists in: *one Service Provider; N x CAN Managers, N x Intra_NRMs*, where the N variable is the number of network domains (e.g., 1 ...N_Max). Note that in practice, given the tiered structure of the Internet, the total number of domains (they might be autonomous systems) involved in an average length E2E communication is rather low.

The specific target is to evaluate the time spent from the instant when a Service Provider issues a VCAN request to an initiator CAN Manager, until the final confirmation of the VCAN installation is obtained by the Service Provider. The Service Provider can choose for its request any CANMgr as VCAN initiator, based on their proximity and involvement in the requested VCAN (or, other policy criteria).

The summary description of the management actions follows. The chosen CAN Manager, named afterwards Initiator CANMgr, will interrogate an inter-domain database containing information about inter-domain topology and network capabilities of the others domains, then run the *inter-domain mapping* algorithm. After this, it will communicate with each CANMgr identified by the inter-domain mapping algorithm as being involved in the requested VCAN, in order to find out its resources.

Note that the simulation model assumes parallelism in communication process from the initiator CAN Manager to the others (in "Hub" style). This is an important feature and design decision, assuring the scalability of the negotiation process.

Figure 3 describes the system processes model based on Extended Finite State Machine (EFSM) [20] represented in SDL-RT tool notation. It contains the global variables, the instance of each class and the other blocks involved.

The system SDL model consists of an *interDB* block, (used in simulation only, corresponding to an inter-domain database that contains inter-domain network topology), a SP_cloud block, associated with the SP/CP requestors, and ND (Number of Domains) CANMgr(s).

Figure 4 presents (just as an example sample) a part of the behavior of the CAN Manager EFSM. The typical SDL graphical notations can be seen, defining the interfaces and messages between blocks.

The works [11], [20] fully describe details of the Message *Sequence Charts* for the signaling process evolution between an initiator CAN Manager, and other CAN Managers involved in constructing a multiple domain VCAN. Figure 5 only presents, as an example, an actual sample of this signaling trace, extracted from the simulation tool results when running the system. It emphasizes a main phase of the VCAN construction: Service Level Specification negotiation.

The initiating CANMgr sends a *VCAN_neg_req* to each of the corresponding CANMgr and enter into *"negotiating"* state.

Each corresponding CANMgr checks its own capabilities (by running an intra-domain mapping algorithm), responds to initiating CANMgr with a *VCAN_neg_rsp* message, and transits to *"waiting_for_acceptance_ext"* state. The initiating CANMgr waits for all corresponding CAN_Mgrs to respond, then integrates the responses. Then it returns an integrated response (*return_result_SLS*) message to SP_cloud and waits for a decision. The above response message indicates to SP that all the requested resources are available and can be provisioned.

The Service Provider analyzes the response and sends a provision request to the initiating CANMgr, using the message *accept_SLS*. Then the initiating CANMgr sends a provision request message (*VCAN_prov_req*), to each of the corresponding CANMgr and waits for their confirmation response.

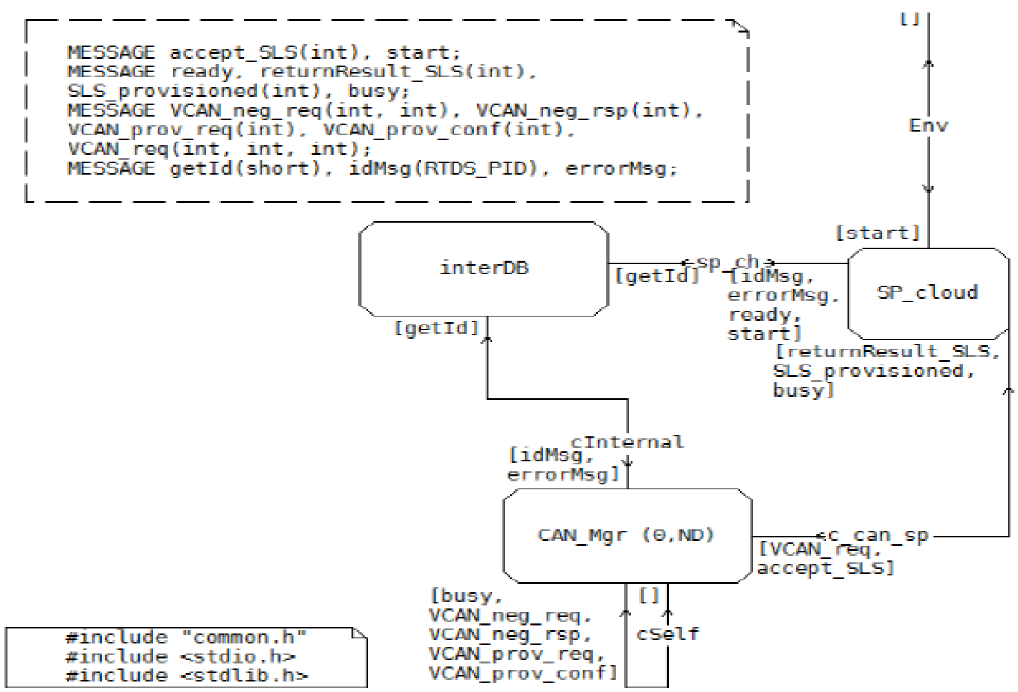

Figure 3. The system model used in RTDS simulations

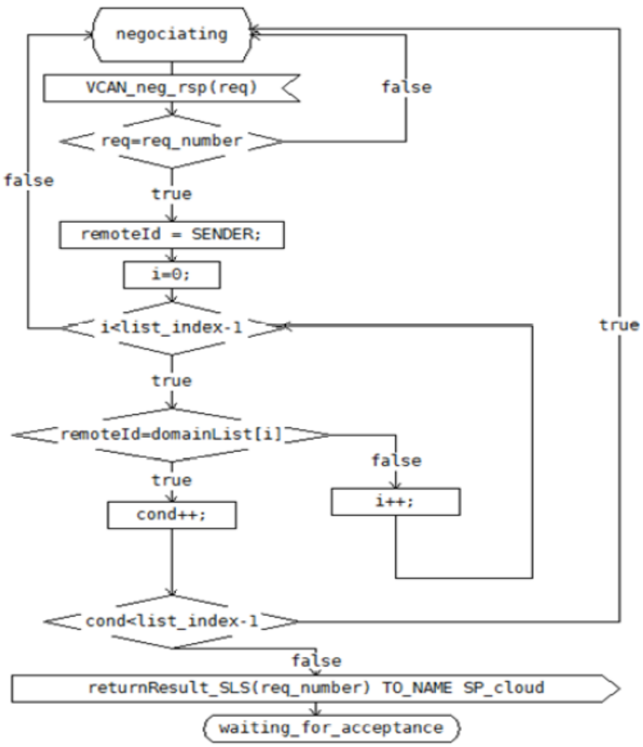

Figure 4. Sample example of the CAN Manager EFSM behavior

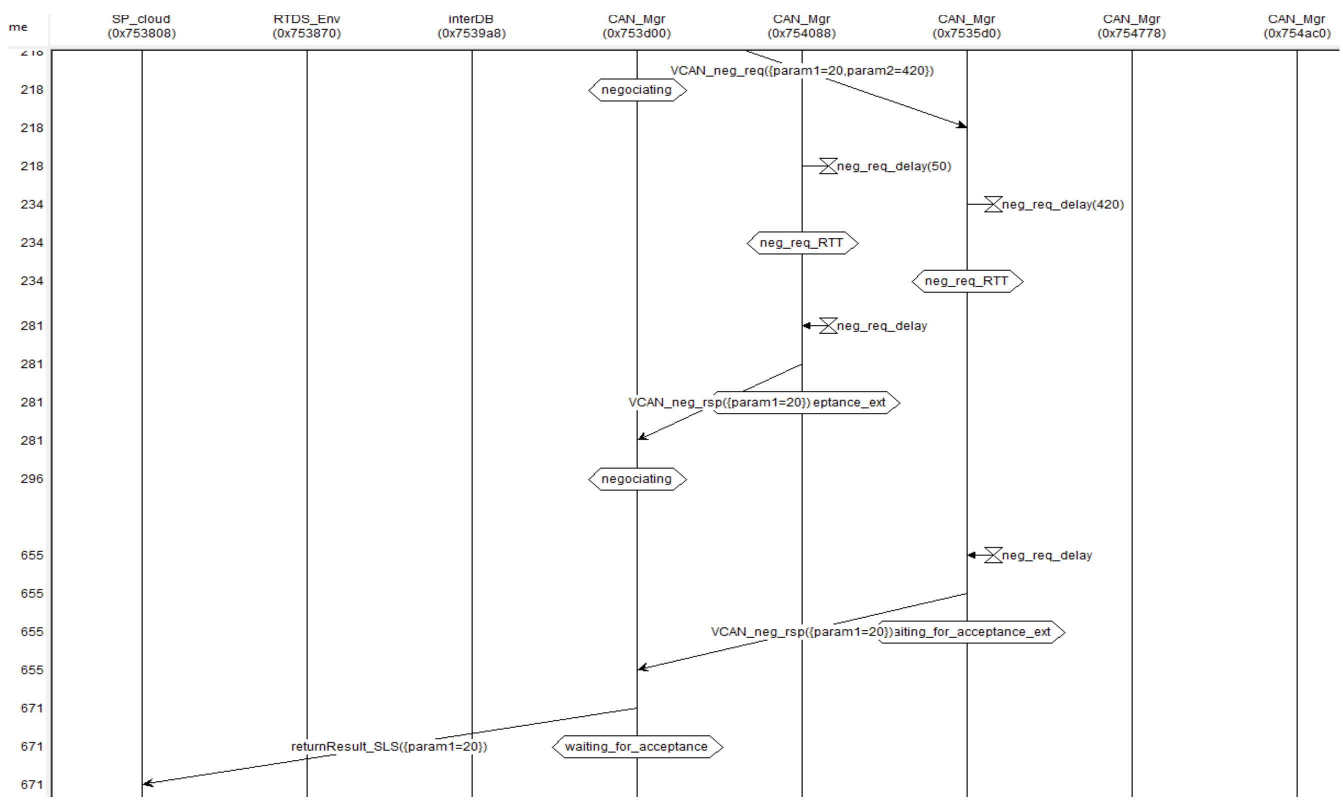

Figure 5. Signaling Message Sequence Chart -sample

VI. SIMULATION RESULTS

The simulations are focused on identifying the system behavior, and to determine a quantitative and qualitative estimation of the signaling time.

Being a real time simulator, the Real Time Developer Studio (RTDS) SDL-RT tool [18], uses the internal PC clock to estimate the time for each task/process from the system. Therefore, the results are defined in "ticks", which are relative time units.

The simulation model just simulates the time consumed by the inter-domain and intra-domain mapping algorithms, but it does not actually compute those algorithms (these computations are described elsewhere [17][18]). However, the result of the mapping algorithm, the chosen CAN Manager and the Round Trip Time (RTT) delay between two communicating CAN Managers are introduced in simulator using configuration files.

The simulation results presented here are extensions and can be considered as complementary to the ones presented in the initial paper [1].

While the simulation model uses an *abstract time clock*, in the experiments done, we can evaluate a *time unit* comparable to 1ms. However, in this study the trends and relative behavior instances (when different parameters are varying) are of more interest than the absolute values. The reason consists in the fact that a real implementation will involve different physical machines (belonging to the Network Provider) running the management software for CAN Managers; also the geographical placement of the domains have its importance w.r.t. quantitative results. WE are mainly interested in evaluating the scalability of the system. That is why we will use the time units (TU) when presenting the results.

Two sets of simulation runs have been performed:

a. considering a fixed delay (constant) RTT value for each corresponding CAN Manager involved

b. considering the same average value as the constant RTT, but with random jitter (uniform distribution). The case b. emulates a real situation where the CAN Managers are placed in different network domains and communicates via Internet. Table I presents some samples where the average delay for a CANMgr to respond to the initiator is 300 ms while a jitter of this time is +/- 100ms. The simulations have changed the number of domains involved, between 2 and 24, covering the use cases from small (w.r.t the number of domains involved) VCANs up to large ones.

The computing performance differences between the two machines (Personal Computers - PCs) are just qualitative criteria on evaluating the performance of a real CAN Manager machine when computing VCAN requests in ALICANTE environment.

Actually several runs have been performed with delay having values among: 100, 200, 300, 1000, 3000, 5000 ms.

This range of values has been selected so as to put into evidence two qualitative cases, concerning the impact upon the total signaling delay:

- processing time inside the various CAN Managers is dominant w.r.t. RTT;
- the RTT is dominant (e.g., 3000, 5000 ms) versus processing time.

Table I. Sample table showing different instants of terminating different partial actions, during a simulation with average delay (RTT) = 300 ms. The maximum number of domains has been 24.

Delay 300 TU (~ms)			Jitter +/-100 TU	N=24 domains	
# of Domains	Start time	VCAN_ req	VCAN_ neg_req	Return Result_ SLA	Stop time
2	344	390	484	921	1404
4	375	421	609	1045	1545
6	343	390	624	1123	1685
8	390	437	780	1388	2075
10	374	436	858	1482	2152
12	375	437	952	1716	2543
14	359	421	999	1810	2699
16	374	436	1107	2043	3104
18	375	437	1170	2200	3292
20	374	437	1279	2434	3635
22	375	437	1357	2621	3947
24	343	437	1420	2793	4212

In order to evaluate the processing power influence upon the total signaling time, the simulations are performed on two different machines, (i.e., named "Processor_1" - low power, and "Processor_2" - high power (Table II).

Table II. PC machine main hardware characteristics

PC configuration	Windows Experience Index	
	Processor_1	Processor_2
Processor	6	7.5
Memory(RAM)	5.9	7.8
Primary hard disk	5.9	7.9

Most of simulations are performed on a powerful machine, named "Processor_2". However, some of simulations are performed also on a slower machine, named "Procesor_1". These latter simulations serve as validation of results obtained from “Processor_2". As expected, the results from "Procesor_1" have bigger relative time values compared with the values from "Processor_2", given that the processing time is shorter on powerful "Processor_2" machine (see Table II).

A simplifying assumption, valid in all simulations, is that all messages issued by an entity arrives correctly at their destination (reliability of the communication has not been an objective of this specific study).

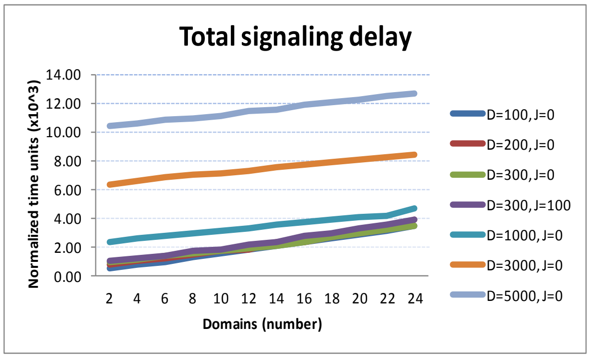

Figure 6. Total signaling delay, versus number of domains, for different values of the delay D (RTT); Processor_2.

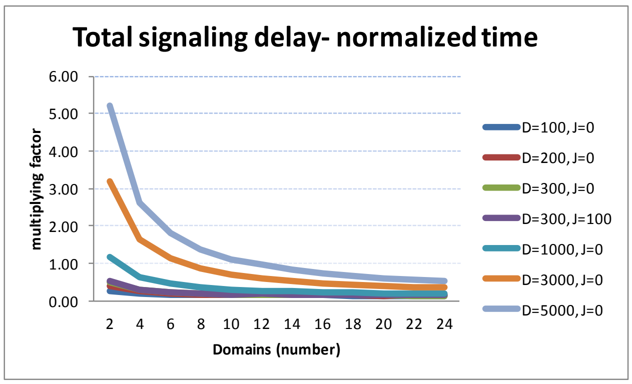

Figure 7. Normalized values (D/N): total signaling delay, Processor_2, versus number of domains, for different values of the delay D (RTT)

Figures 6 and 7 evaluate the performance of the overall system. They present the total signaling time (from the instant when the Service Provider issues a VCAN request to an initiating CAN Manager until the provisioning of the VCAN is finished and finally confirmed at SP). Figure 6 considers absolute values for delays while Figure 7 represents the behavior while using normalized values Different values for the delay (i.e., Round Trip Time – RTT) have been taken in the set of experiments.

Figure 7 shows that the normalized values of the signaling time are converging when the number of domains is increasing. If we consider absolute values of the total signaling time, one can see that the system can construct an average number of 10^2 – 10^3 VCANs/hour, which is quite sufficient in practice.

Three important conclusions can be extracted from the above results:

- The system is scalable w.r.t. number of domains involved, proved by the fact that all diagrams have approximately a linear behavior. This is the major result, showing that the management system can control large VCANs without significant signaling overhead;
- When the total transfer time (processing time plus propagation time through the internet) is dominant, the influence of the number of domains upon the total signaling time is lower (see the graphics for D = 300 ...5000).
- On the contrary, in cases of small average transfer time (e.g., D = 100, 200) the total signaling time increases linearly – with a higher slope when the number of domains increases.

Sets of results like the above may have utility for SP in establishing certain policies, when it makes the preliminary VCAN planning.

The following Figures (8 to 13) evaluate the relative time intervals consumed by different processing components of the overall signaling system. The objective is to identify those components, which produce more time consumption, in order to provide input data for system optimizations.

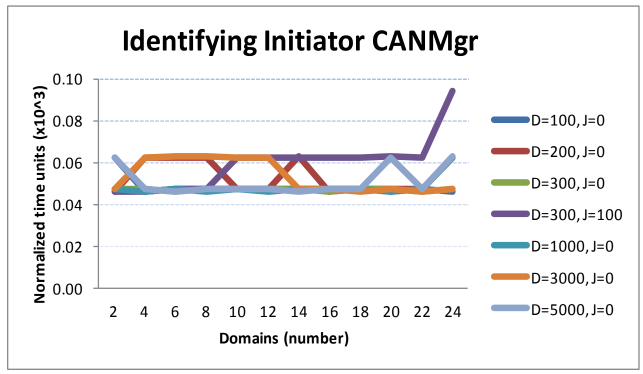

Figure 8. Processing time to identify the Initiator CAN Manager; Processor_2

Figure 8 presents the processing time consumed to identify the Initiator CAN Manager. These results validate the assumption that the number of domains do not influence this value (which is relatively low). The small variations shown in the figure are only statistical ones.

Figures 9 and 10 present the processing time consumed to determine the topology and identifiers of the CAN Managers involved in the required VCAN. One can see a linear increase of this time with the number of network domains; however the network communication transfer time is not involved, so the value is relatively low in the total budget of time.

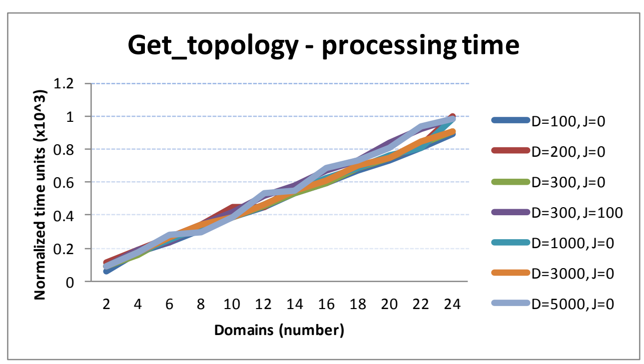

Figure 9. Processing time consumed to acquire the inter-domain topology; Processor_2

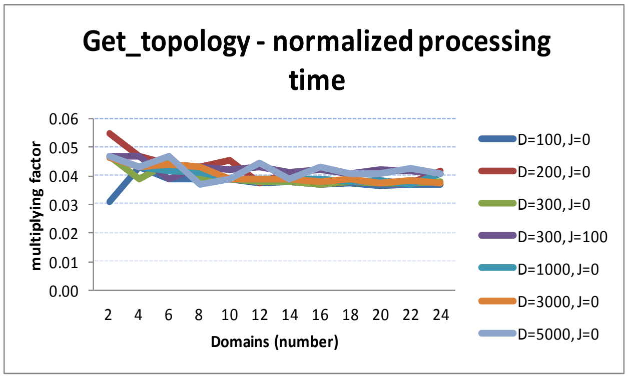

Figure 10. Normalized values: Processing time consumed to acquire the inter-domain topology; Processor_2

Figures 11 and 12 present the processing time consumed to perform SLS negotiations between the Initiator CAN Manager and its partners, i.e., other CAN Managers involved in the required VCAN. As expected, this set of actions has a major contribution to the total signaling time, because, additionally to the processing time, transfer through the network is involved. This is confirmed by the similarities between the diagrams of Figures 6-7 and those of the Figures 11-12.

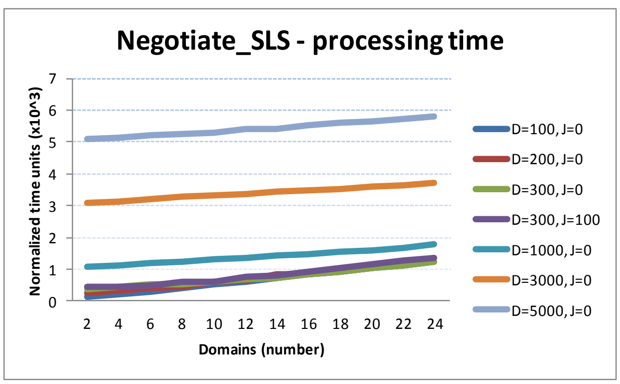

Figure 11. Processing time consumed to negotiate the Service Level Specification (SLS) contracts; Processor_2

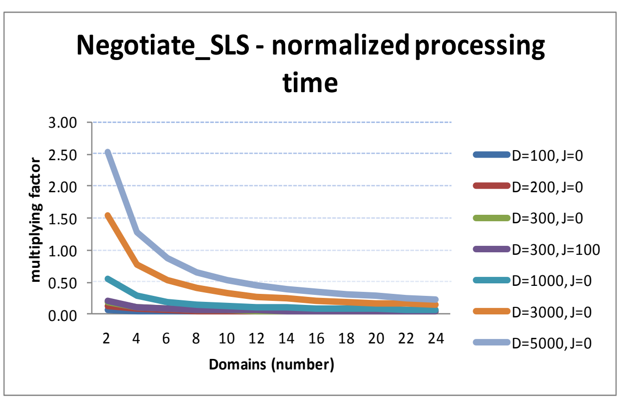

Figure 12. Normalized values: Processing time consumed to negotiate the Service Level Specification contracts (SLS); Processor_2

Note that additional variation for the values presented in Figures 11 and 12 could exist in practice, depending on how frequently a given CAN Manager is updating its Resource Availability Matrix (result of the dialogue CAN Manager and its associated Intra-domain Network Resource Manager). Here, one may have independent policies like push/pull and synchronous or asynchronous communication type.

Figures 13 and 14 present the processing time consumed to provision (install all required configurations in the MANE and core routers) the VCANs in the network. This set of actions has a major contribution to the total signaling time, because the time is consumed by each CAN Manager to trigger installation of the VCAN configurations in the routers (via Intra_NRM) and also the transfer through the inter-domain network is involved for the signaling messages. This is confirmed by the similarities between these diagrams and those of the Figures 11-12. Optimizations in this area can contribute significantly to the overall system performance.

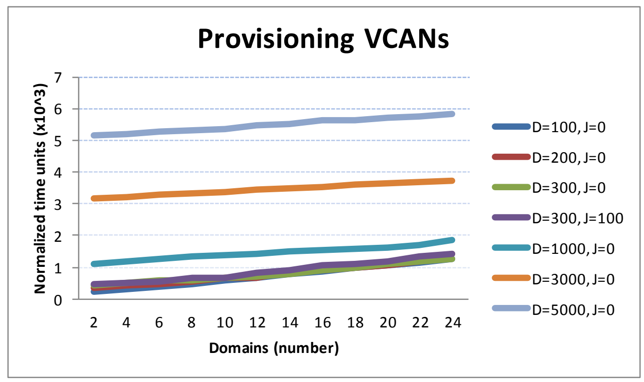

Figure 13. Processing time consumed to provision VCANs in the network; Processor_2

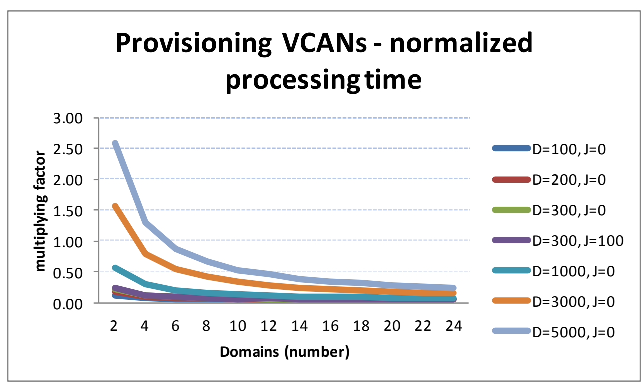

Figure 14. Normalized values: Processing time consumed to provision VCANs in the network; Processor_2

Figure 15 shows a comparison between simulations on "Processor_1" and "Processor_2". The behavior of the systems is similar (but having different slopes in the diagram), while in the case of a slower Processor_1, only the convergence value is different (4500 for "Processor_1" and 2500 for "Processor_2"). Again, the convergence is present, and the difference on convergence value is the result of different computing time inside CANMgr.

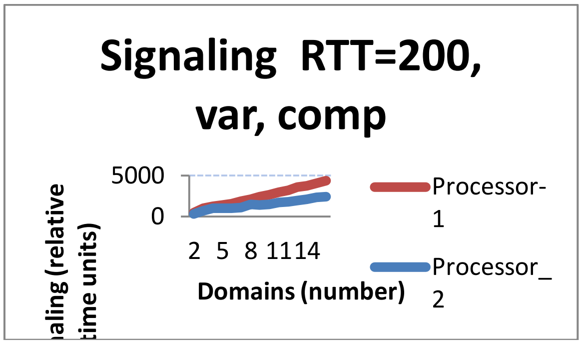

Figure 15. Comparison of the total signaling times; average D=RTT=200; Processor_1 versus Processor_2.

In order to have an additional validation from statistical point of view, a set of simulations was performed on "Processor_2", using different seeds. The simulation results are shown in Figure 16. The same overall behavior is obtained; the small difference of the convergence value occurs due to the seed influence on simulator internal algorithm, shown on the relative time units obtained on each simulation.

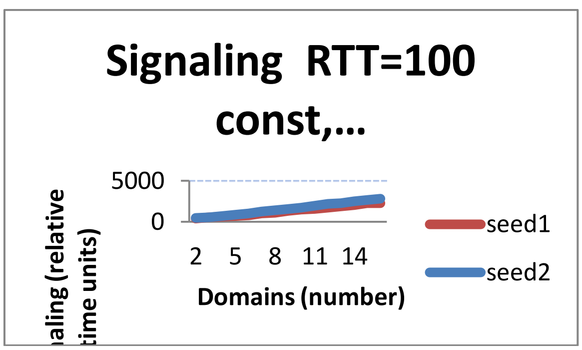

Figure 16. Total signaling time for RTT=100 constant, different seed, Processor_2

For scalability extended evaluation purpose only, a set of simulations was performed, considering 500 domains (CANMgr). As shown in Figure 17 the linear behavior is also exposed up to a high number of domains (N=500).

Other simulations have shown that the components of the total signaling time have roughly the same relative weights for large number of domains experiments, similar to those presented in Figures 9 – 14.

The overall set of simulations (including the extended range ones) provides a confirmation of the initial assumptions about the signaling system performing the multi-domain VCAN construction: the system is scalable w.r.t. number of domains involved. The proof is provided by the fact that all diagrams have approximately a linear behavior, so the management system can control VCANs spanning multiple domains with low signaling overhead.

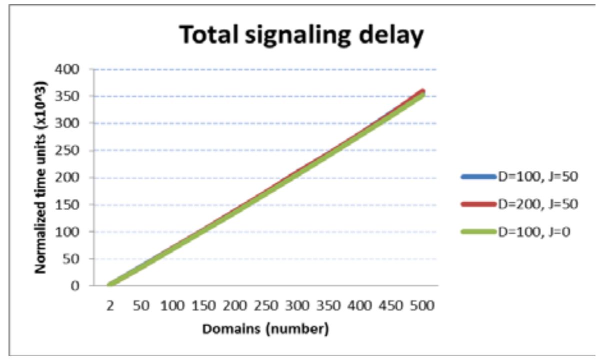

Figure 17. Total signaling delay, versus number of domains, for different values of the delay D (RTT) ; Processor_2, N=500 domains

The major contribution to the total time spent for a VCAN construction is brought by the VCAN provisioning phase, given that three components intervene: a. transfer time (in both directions) through the inter-domain networks, needed for communication between the Initiator CAN Manager and the other CAN Managers involved; b. processing/computation time to determine the resources to be provisioned in each CAN Manager involved in a multiple-domain VCAN; c. transfer time needed by each Intra-NRM to inject the VCAN configurations in all its MANE and core routers. In practice, optimization techniques can reduce the time spent by these components.

VII. CONCLUSIONS

This paper extended the results presented initially in ICNS 2014 paper [1], providing a simulation model and results concerning the scalability of the multi-controller communication subsystem as a functional management component of a media delivery ecosystem. It is shown that the proposed control approach is conveniently feasible in a multi-domain network environment.

First, the management architecture of a multi-domain media delivery system (in particular ALICANTE [10][11]) system has been outlined, showing its partial similarity with the SDN multi-controller architecture.

The architectural equivalence has been analyzed between an SDN regional controller and the pair {CAN Manager and Intra-domain Network Resource Manager} considered in this study. Horizontal scalability problems appear in both SDN and ALICANTE multi-controller environments.

A simulation model based on Extended Finite State Machines approach has been constructed for the ALICANTE management architecture, aiming to evaluate mainly the total signaling time for Virtual Content Aware Networks negotiation and provisioning over a multi-domain environment. Extensive sets of simulation runs have been performed for various network domains, network transfer conditions and processing times.

The main finding of this article is that signaling system is horizontally scalable, versus the number of network domains involved. This has been proved by the approximate linear behavior of the total signaling time. The components of the total signaling time have been also identified. Quantitative metrics have been determined in different use cases, emphasizing the dominance of the processing time or the network transfer times.

Further work should evaluate the capacity of the ALICANTE architecture to get closer to the SDN approach, and also methods to integrate the VCAN construction particular problem into more general SDN controller framework. Another direction is to investigate how one controller can command a given number of network elements (routers) by using a vertical protocol (similar to OpenFlow).

ACKNOWLEDGMENT

This work has been partially supported by the FP7 European research project ALICANTE (FP7-ICT-248652) and partially by the projects POSDRU/107/1.5/S/76909.

REFERENCES

[1] E. Borcoci, M. Constantinescu, and M. Vochin, “Multi-controller Scalability in Multi-domain Content Aware Networks Management,” ICNS 2014 Conference, http://www.iaria.org/conferences2014/ProgramICNS14.html, pp. 62-68.

[2] J. Schönwälder, M. Fouquet, G. D. Rodosek, and I. Hochstatter, “Future Internet = Content + Services + Management,” IEEE Communications Magazine, vol. 47, no. 7, pp. 27-33, July 2009.

[3] T. Anderson, L. Peterson, S. Shenker, and J. Turner, “Overcoming the Internet Impasse through Virtualization,” Computer, vol. 38, no. 4, pp. 34–41, Apr. 2005.

[4] J. Choi, J. Han, E. Cho, T. Kwon, and Y. Choi, “A Survey on Content-Oriented Networking for Efficient Content Delivery,” IEEE Communications Magazine, March 2011.

[5] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, “Networking Named Content,” CoNEXT ’09, New York, NY, pp. 1–12, 2009.

[6] N. McKeown et. al., “OpenFlow: Enabling Innovation în Campus Networks,” March 2008, -http://www.openflow.org/documents/openflow-wp-latest.pdf.

[7] B. A. Nunes, M. Mendonca, X. N. Nguyen, K. Obraczka, and Thierry Turletti, “A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks,” January 2014, http://hal.inria.fr/hal-00825087

[8] OpenFlow Switch Specification, V 1.3.0 (Wire Protocol 0x04), June 25, 2012, https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.3.0.pdf

[9] S. H. Yeganeh, A. Tootoonchian, and Y. Ganjali, “On Scalability of Software-Defined Networking”, IEEE Communications Magazine, pp. 136-141, February 2013.

[10] FP7 ICT project, “MediA Ecosystem Deployment Through Ubiquitous Content-Aware Network Environments,” ALICANTE, No248652, “D2.1: ALICANTE Overall System and Components Definition and Specifications”, http://www.ict-alicante.eu/.

[11] C. Cernat, E. Borcoci, and V. Poenaru, “SLA Framework Development for Content Aware Networks Resource Provisioning,” AICT 2013 Conference, http://www.thinkmind.org/index.php?view=article&articleid=aict_2013_8_30_10084

[12] K. T. Cheng and A. S. Krishnakumar, "Automatic Functional Test Generation Using The Extended Finite State Machine Model," International Design Automation Conference (DAC), ACM, 1993, pp. 86–91.

[13] A. Pathan and R Buyya, “A Taxonomy and Survey of Content Delivery Networks,” http://cloudbus.org/reports/CDN-Taxonomy.pdf. 2008.

[14] B. Raghavan, T. Koponen, A. Ghodsi, M. Casado, S. Ratnasamy, and S. Shenker, “Software-Defined Internet Architecture: Decoupling Architecture from Infrastructure,” HotNets-XI Proceedings of the 11th ACM Workshop on Hot Topics in Networks, 2012, pp. 43-48, doi: 10.1145/2390231.2390239.

[15] A. Tavakkoli, M. Casado, and S. Shenker, “Applying NOX to the Datacenter,” Proc. ACM HotNets-VIII Wksp., 2009.

[16] H. Yin, H. Xie, T. Tsou, D. Lopez, P. Aranda, and R. Sidi, draft-yin-sdn-sdni-00.txt, ”SDNi: A Message Exchange Protocol for Software Defined Networks (SDNS) across Multiple Domains,” June 2012.

[17] E. Borcoci, R. Miruta, and S. Obreja, “Multi-domain Virtual Content-Aware Networks Mapping on Network Resources,” EUSIPCO Conference, 27-31 August 2012, Bucharest, http://www.eusipco2012.org/home.php

[18] R. Iorga, E. Borcoci, A. Pinto et. al., “Management driven hybrid multicast framework for content aware networks,” IEEE Communications Magazine, Vol. 52, January 2014, pp. 158-165.

[19] http://www.sdl-rt.org/, 12.11.2014

[20] ALICANTE Deliverable D8.3, “Trials and Validation,” http://www.ict-alicante.eu/validation/delivrables/, 14.10.2014