CTRQ 2016

The Ninth International Conference on Communication Theory, Reliability, and Quality of Service

ISBN: 978-1-61208-455-8

February 21 - 25, 2016

Lisbon, Portugal

CTRQ 2016 Editors

Eugen Borcoci, Politehnica University of Bucharest, Romania

Abdulrahman Yarali, Murray State University, USA

CTRQ 2016

Forward

The Ninth International Conference on Communication Theory, Reliability, and Quality of Service (CTRQ 2016), held between February 21-25, 2016 in Lisbon, Portugal, continued a series of events focusing on the achievements on communication theory with respect to reliability and quality of service. The conference also brought onto the stage the most recent results in theory and practice on improving network and system reliability, as well as new mechanisms related to quality of service tuned to user profiles.

The processing and transmission speed and increasing memory capacity might be a satisfactory solution on the resources needed to deliver ubiquitous services, under guaranteed reliability and satisfying the desired quality of service. Successful deployment of communication mechanisms guarantees a decent network stability and offers a reasonable control on the quality of service expected by the end users. Recent advances on communication speed, hybrid wired/wireless, network resiliency, delay-tolerant networks and protocols, signal processing and so forth asked for revisiting some aspects of the fundamentals in communication theory. Mainly network and system reliability and quality of service are those that affect the maintenance procedures, on one hand, and the user satisfaction on service delivery, on the other hand. Reliability assurance and guaranteed quality of services require particular mechanisms that deal with dynamics of system and network changes, as well as with changes in user profiles. The advent of content distribution, IPTV, video-on-demand and other similar services accelerate the demand for reliability and quality of service.

The conference had the following tracks:

- Communication theory, reliability, and Quality of Service

We take here the opportunity to warmly thank all the members of the CTRQ 2016 technical program committee, as well as all the reviewers. The creation of such a high quality conference program would not have been possible without their involvement. We also kindly thank all the authors that dedicated much of their time and effort to contribute to CTRQ 2016. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

Also, this event could not have been a reality without the support of many individuals, organizations and sponsors. We also gratefully thank the members of the CTRQ 2016 organizing committee for their help in handling the logistics and for their work that made this professional meeting a success.

We hope CTRQ 2016 was a successful international forum for the exchange of ideas and results between academia and industry and to promote further progress in the in the field of communication theory, reliability and quality of service. We also hope that Lisbon, Portugal provided a pleasant environment during the conference and everyone saved some time to enjoy the beauty of the city.

CTRQ 2016 Advisory Committee

Eugen Borcoci, Politehnica University of Bucharest, Romania
Pascal Lorenz, University of Haute Alsace, France
Raj Jain, Washington University in St. Louis, USA

CTRQ 2016

Committee

CTRQ 2016 Advisory Committee

Eugen Borcoci, Politehnica University of Bucharest, Romania
Pascal Lorenz, University of Haute Alsace, France
Raj Jain, Washington University in St. Louis, USA

CTRQ 2016 Technical Program Committee

Mandicou Ba, Université de Reims Champagne-Ardenne, France
Dirk Bade, University of Hamburg, Germany
Jasmina Barakovic Husic, BH Telecom – Sarajevo, Bosnia and Herzegovina
Monique Becker, Telecom SudParis - Institut Mines-Telecom, France
Roberto Beraldi, "La Sapienza" University of Rome, Italy
Eugen Borcoci, Politehnica University of Bucharest, Romania
Christos Bouras, University of Patras - Computer Technology Institute & Press «Diophantus», Greece
Damian Bulira, Wroclaw University of Technology, Poland
Laurent Ciavaglia, Alcatel-lucent, Italy
Minh Ngoc Dao, University of British Columbia Okanagan, Canada / Hanoi National University of Education, Vietnam
Mario De Felice, Sapienza University of Rome, Italy
Javier Del Ser Lorente, TECNALIA Research & Innovation, Spain
Manfred Droste, Universität Leipzig, Germany
Andras Farago, The University of Texas at Dallas - Richardson, USA
Gianluigi Ferrari, University of Parma, Italy
Alexandre Fonte, Polytechnic Institute of Castelo Branco, Portugal & Centre for Informatics and Systems of the University of Coimbra (CISUC) , Portugal
Tulsi Pawan Fowdur, University of Mauritius, Mauritius
Julio César García Alvarez, Universidad Nacional de Colombia Sede Manizales, Colombia
Rita Girão Silva, DEEC-FCTUC / INESC-Coimbra, Portugal
Apostolos Gkamas, University Ecclesiastical Academy of Vella of Ioannina, Greece
Teresa Gomes, University of Coimbra, Portugal
Teodor Lucian Grigorie, University of Craiova, Romania
Antti Hakkala, University of Turku, Finland
Bjarne J. Helvik, NTNU, Norway
Robert Ching-Hsien Hsu, Chung Hua University, Taiwan
Ilias Iliadis, IBM Research - Zurich, Switzerland

Table of Contents

Cloudcasting: A New Architecture for Cloud Centric Networks

Richard Li, Kiran Makhijani, Lin Han
American Research Center
Huawei Technologies
Santa Clara, California, USA
e-mail: {renwei.li, kiran.makhijani, lin.han}@huawei.com

***Abstract*— Network overlays play a key role in the adoption of cloud oriented networks, which are required to scale and grow elastically and dynamically up/down and in/out, be provisioned with agility and allow for mobility. Cloud oriented networks span over multiple sites and interconnect with Virtual Private Network (VPN) like services across multiple domains. In literature, there have been some proposals to implement network overlays such as, Virtual eXtensible Local Area Networks (VXLAN) as the data plane and Border Gateway Protocol/Ethernet VPN (BGP/EVPN) as the control plane. However, none of them meets all the above requirements. This paper presents the new network architecture, called Cloudcasting, along with its reference model and related protocols, both on the control plane and the data plane, which can demonstrably meet all the requirements. The cloudcasting architecture includes four elements: Cloud Rendezvous Point (CRP), Cloud Switching Point (CSP), Cloud Control (CCC) protocol, and Virtual Extensible Network (VXN) Encapsulation Protocol.**

***Keywords-Cloud; Network Overlay; Network Virtualization; Routing, Multi-Tenancy Virtual Data Center; VXLAN; BGP; EVPN*.**

I. INTRODUCTION

The key characteristics of Cloud-oriented data center architectures are resource virtualization, multi-site distribution, scalability, multi-tenancy and workload mobility. These are typically enabled through network virtualization overlay technologies. Initial network virtualization approaches relate to layer-2 multi-path mechanisms such as, Shortest Path Bridging (SPB) [3] and Transparent Interconnection of Lots of Links (TRILL) [5] to address un-utilized links and to limit broadcast domains. Later, much of the focus was put into the data plane aspects of the network virtualization, for example, VXLAN [1], Network Virtualization using Generic Routing Encapsulation (NVGRE) [2], and Generic Network Virtualization Encapsulation(GENEVE) [9]. These tunneling solutions provide the means to carry layer-2 and/or layer-3 packets of tenant networks over a shared IP network infrastructure to create logical networks. Though, due to their lack of corresponding control plane schemes, they require painstaking orchestration of the system for the virtual network setup and maintenance [10][11]. Even more recently, MP-BGP/EVPN [4] has been proposed as a control plane for virtual network distribution, and has foundations of the VPN style provisioning model. This requires additional changes to an already complex and a heavy protocol that was originally designed for the inter-domain routing. The deployment of MP-BGP/EVPN in data center networks also brings in corresponding configurations, for example, defining autonomous systems (AS), that are not really relevant to the data center infrastructure network.

The existing solutions such as, multi-path, custom-orchestrations and Multiprotocol-BGP (MP-BGP) [6][7] are a class of virtual network architectures that consume protocol data structures of substrate networks, therefore, we refer to them as *Embedded Virtual Networks*. The term *substrate network* henceforth will be used to describe a base, underlying, or an infrastructure network upon which user networks are built as virtual network overlays.

In this paper, a new network virtualization approach is proposed, which does not require changing the substrate protocols. It can connect different types of virtual networks through its own routing scheme. Since, such scheme can be organized over any substrate network topology and routing arrangement; it is referred to as *Extended Virtual Networks*.

Even though *Embedded Virtual Network* (the term is inspired from [17]) solutions mentioned above have irrefutable benefits, they also have several limitations. Of which the most significant and relevant to cloud-scale environments is their dependence on the substrate networks. In addition to being scalable and reliable, a cloud scale network must also be elastic, dynamic, agile, infrastructure-independent, and capable of multi-domain support. There has not been a single technology which works as a converged architecture for network virtualization. In this paper, we propose an *Extended Virtual Network* framework that operates on top of substrate network and offers primitives for cloud auto-discovery, dynamic route distribution as needed. Operationally, this new network architecture allows for agile provisioning and allows for the interconnection of hybrid clouds.

The rest of the paper is organized as follows. Section II describes a reference model for cloudcasting, and major functions of its reference points. Section III explains the signaling communication primitives between the cloudcasting reference points and Section IV uses a multi-tenant virtualized data center as a deployment example. While Section V highlights the advantages and strengths of the solutions, Section VI compare our solution the related work. Lastly, Section VII briefly lays out the directions for our future work.

II. CLOUDCASTING MODEL

A converged virtual routing scheme can be described by two primary factors; an infrastructure-independent virtual network framework, and a unified mechanism to build an overlay of various types of user networks with different address schemes. On these basis, a new virtual routing scheme called *Cloudcasting,* is proposed with the following characteristics

(1) *Auto discovery*: A signaling scheme that enables us to add, delete, expand and virtualize a tenant's network with minimum configuration.
(2) *Auto distribution:* A signaling scheme that connects multiple virtual networks with each other or asymmetrically as needed.
(3) *Auto Scale:* The ability to provide and serve high scale of tenants in a location-agnostic manner.

A ***cloudcasting network*** is an IP network, which is shared and used by multiple tenant clouds to route traffic within a single virtual network or between different virtual networks. We use the terminology of tenant cloud to emphasize that a tenant or a user network may reside anywhere on the substrate network with a highly dynamic routing table. The IP address space in one tenant cloud may overlap with that in another cloud and these are not exposed to the shared IP infrastructure network.

The cloudcasting reference model, is shown in Figure 1. Each customer has its own network shown as *Tenant Cloud A, B and C*, a shared substrate IP network that was built independently and can encompass multiple administrative domains. This model describes a centralized conversational scheme, in which tenant clouds or Virtual Extensible Networks (VXNs) announce their presence as well as membership interests to a centralized designated authority, called Cloudcasting Rendezvous Point (CRP), via a cloudcasting network virtualization edge element called Cloudcasting Switching Point (CSP).

To communicate among the network elements, a new signaling protocol, called **C**loud**C**asting **C**ontrol (CCC) protocol is defined with three simple primitives facilitating cloud auto-discovery and cloud route distribution. The protocol primitives are defined as below and are further illustrated in Figure 2.

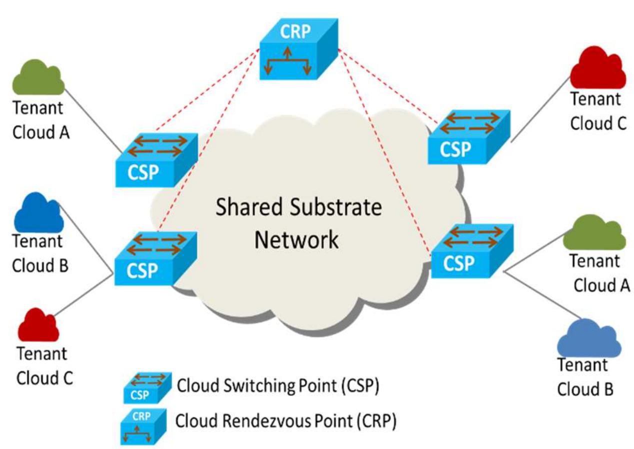


Figure 1. Cloudcasting Reference Model.

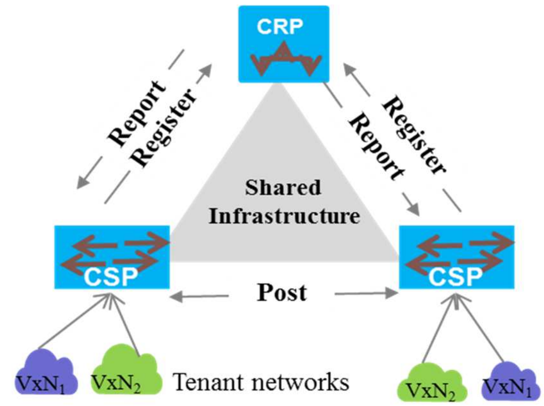


Figure 2. Cloudcasting Framework.

Register message: A virtual network interest and self-identifying announcement primitive from CSP to CRP.

Report message: A response from CRP to all CSPs with similar virtual network interests.

Post message: A CSP to CSP virtual network route distribution primitive.

The details of aforementioned cloudcasting network elements and their properties in cloudcasting framework are discussed as below.

A. Virtual Extensible Network

A Virtual Extensible Network is a tenant cloud or a user network. It is represented by a unique identifier with a global significance in cloudcasting network. Using this construct, it is possible to discover all its instances on the substrate IP fabric via CRP. VXN identifiers are registered with CRP from CSPs to announce their presence. There are various possible formats to define the VXN, for instance, an alphanumeric value, number or any other string format. In the preliminary work we have defined it as a named string which is mapped to a 28-bit integer identifier, thus enabling support for up to 256 million clouds.

B. Cloud Switch Point

A Cloud Switch Point is a network function that connects virtual networks on one side to the substrate IP network on the other side. It can be understood as an edge of a virtual network that is cloudcasting equivalent of a Virtual Tunnel End Point (VTEP) [1] in VXLAN networks or an Ingress/Egress Tunnel Router (xTR) in the LISP domain [15] and may similarly be co-located with either on a service provider's edge (PE) router, on a top of rack (ToR) switch in a data center, or on both.

A CSP participates in both auto-discovery and auto-route distribution. In order to establish a forwarding path between two endpoints of a virtual network or of two different virtual networks, a CSP first registers with the CRP its address and VXN identifiers it intends to connect to. Then the CRP will report to all CSPs that have the same VXN membership interest. Finally, the CSP will communicate with those other CSPs and exchange their routing information. On the data forwarding plane, a CSP builds a virtual Forwarding Information Base (vFIB) table on per VXN basis and route/switch traffic to the destination virtual networks accordingly.

C. Cloud Rendezvous Point

A Cloud Rendezvous Point is a single logical entity that stores, maintains and manages information about Cloud Switching Points and their VXN participation. The CRP maintains the latest VXN to CSP membership database and distributes this information to relevant CSPs so that they can form peer connection and exchange virtual network routes automatically. A report message is always generated whenever there is a change in the virtual network membership database. However, CRP is oblivious to any change in vFIB (described above in CSP).

III. CLOUDCASTING COMMUNICATION PRIMITIVES

Now, we describe cloudcasting communication primitives used among CRP and CSPs. Figure 3 illustrates the layering of the virtual routing over any substrate layer and overlay control messages between CSP and CRP.

The encapsulation message format is shown above in Figure 4. A well-known TCP destination port identifies the cloudcasting protocol and CCC header contains the specification for the register, report and post messages.

A. Cloudcasting Register Message

An auto-discovery of virtual networks involves two messages. The first message is the Cloudcasting Register Message that originates from CSPs to announce their presence and interests with the CRP to learn about the other CSPs with the same interest of VXNs. A Register message specification includes the CSP address and VXN identifier list of its interest. An interest is defined as an intent to participate in a specific virtual network. For example, a vxn_{red} on csp_1 expresses 'interest' to join vxn_{red} on csp_2.

As an example, consider virtual networks vxn_{red} and vxn_{green} are attached to csp_1. Then, the register message contains a tuple as follows

$$Register\ \{sender: csp_1, [vxn_{red}, vxn_{green}]\}$$

After the CRP receives a cloudcasting register message, it scans its CSP membership database to look for the same VXN identifiers.

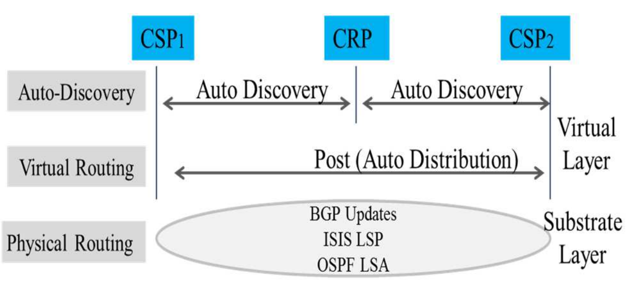


Figure 3. Cloudcasting Protocol Primitives

If it finds one (or more), a cloudcasting report message is generated and sent to all the CSPs with the same interest, otherwise, it simply logs the VXN in its CSP database.

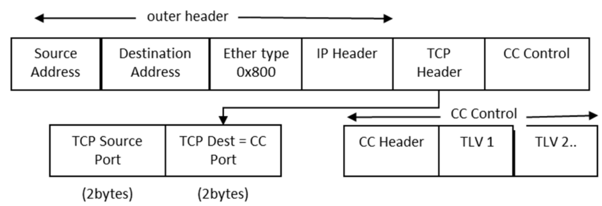


Figure 4. Cloudcasting Control Message Format

B. Cloudcasting Report Message

The CRP generates cloudcasting report messages in response to a cloudcasting register message to inform CSPs of other CSPs' address and their associated VXN identifiers. If the CRP finds other CSP(s) with the same VXN membership (or interested VXNs), then the Report messages are generated for that CSP as well as the other found CSPs. A Report message is sent to each CSP, that contains other CSP addresses for the shared interest VXNs. As an example, consider CRP already has csp_2 with interest vxn_{red}. Upon receiving a cloudcasting register message from csp1 as described earlier, two report messages are generated as below for csp_2 and csp_1, respectively:

$$Report\ (csp_2)\ \{to: csp_1, [interest: vxn_{red}]\}$$

$$Report\ (csp_1)\ \{to: csp_2, [interest: vxn_{red}]\}$$

In this manner, auto-discovery of virtual network locations is accomplished that is based on interest and announcement criteria.

C. Cloudcasting Post Message

The cloudcasting post messages facilitate route distribution as needed. As a cloudcasting report message is received, the CSP will connect with other CSPs to exchange their routing information that includes VXN identifiers, a Generic VXN encapsulation (GVE) tag and the network reachability information within the VXN along with the address family. The list of network reachability information type includes but not restricted to IP prefixes (such as, IPv4, IPv6), VLANs, MAC addresses or any other user defined address scheme.

As an example, when a report as described earlier is received, the following Post will originate from csp_1.

$$Post\ (csp_1, csp_2)\ \{vxn_{red}, gve: i, [AF: IPv4, prefix\ list...]\}$$

In the example above, it is shown that csp_1 sends a post update to csp_2 which states that vxn_{red} will use encapsulation tag *'i'*; and that it has certain ipv4 prefixes in its IP network.

The routing (network reachability) information has the flexibility to support various address families (AF) defined by IANA as well as certain extensions not covered under the IANA namespace.

 ISBN: 978-1-61208-455-8

D. Cloudcasting Transport - Generic VXN Encapsulation

In a cloudcasting network, all network devices will work exactly the same as before on the data plane except the Cloud Switch Points (CSP). A CSP will perform encapsulation and decapsulation by following the VXN vFIB table. A VXN vFIB table includes the routing information for a virtual network on a remote CSP where a packet should be destined to. The route information was learned by exchanging Post messages between CSPs.

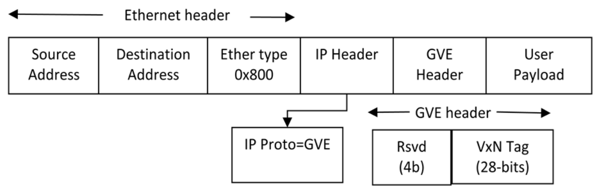


Figure 5. Cloudcasting Data Plane Encapsulation

The format for VXN encapsulation is shown in Figure 5 above in which IP protocol is set to GVE and following IPv4 header is the 32-bit GVE-header. The protocol number for GVE will be assigned by IANA.

IV. USE CASES

Figure 6 shows a cloudcasting-enabled virtualized data center. As discussed earlier in Section I, the CRP is a logically centralized node that is accessible by all the CSPs. A leaf-spine switch architecture is used as a reference to explain cloudcasting deployment. A plausible co-location for CRP could be with the spine node, however, it may be anywhere in the substrate network as long as CSPs can reach it with the infrastructure address space. In Figure 6, several tenant networks are shown as connected to different CSPs and CSP function itself is co-resident with the leaf switches. Each CSP has a virtual FIB table for both encapsulation and decapsulation of traffic along with the tenant network to CSP memberships (dynamically learned through auto-discovery).

The cloudcasting control protocol flow is shown in lighter color lines between CRP and CSPs and among CSPs. At the bottom of the Figure 6. only the logical GVE data path tunnels with dotted lines for tenant 1 on CSP-1, CSP-3 and CSP-4 are shown.

V. EVALUATION AND ANALYSIS

The cloudcasting architecture and primitives have been implemented in our research laboratory. We have successfully used the cloudcasting architecture and control protocol to implement the following use cases:

- Multi-Tenancy Virtual Data Centers
- Multi-Site Interconnection of Data Centers
- Interconnection of Hybrid Clouds
- VPN Accesses to Virtual Data Centers

First and foremost, we emphasize that the cloudcasting architecture represents a paradigm shift. It is a truly converged technology for virtual networks, clouds, and VPNs. No matter what the structure of the underlying substrate network is, any/all types of virtual tenant networks can be constructed in the same way by using cloudcasting.

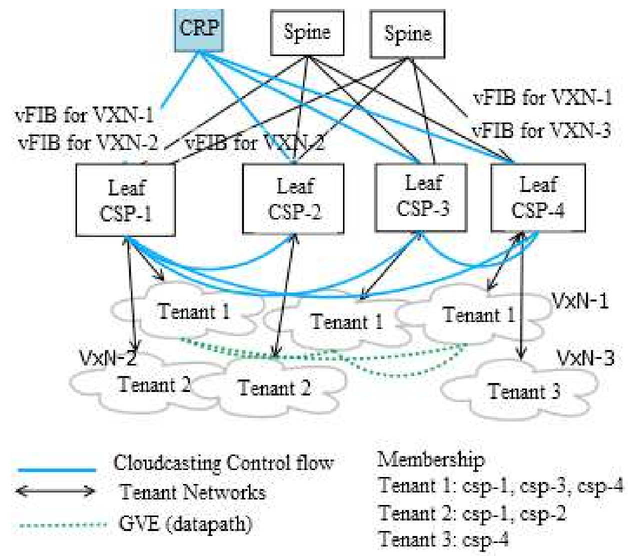


Figure 6. Cloudcasting Enabled Deployment.

The Cloudcasting suitability and applicability can only be verified vis-à-vis characteristics of the cloud-scale environments. Therefore, we have laid importance on the primary characteristics of cloud centric networks that are elasticity, efficiency, agility, and distribution.

The Cloudcasting control plane is *elastic*, because it can grow and shrink independently of (1) the heterogeneous protocols of the substrate network, (2) number of virtual network attachment points, the CSPs, (3) number of domains (autonomous systems), (4) number of routes within a user's virtual network, and (5) mobile nature of the host stations.

The Cloudcasting control plane is *efficient*, because (1) no CSP distributes routes to other CSPs that they are not interested in, (2) thus, no CSP receives and stores routes of virtual networks of non-interest or the ones it is not connected to. In addition, the control plane is fully *distributed* in such a manner that through a single primitive (post-update); change in the tenant networks can be announced immediately, from the spot of change without configuration changes.

The Cloudcasting allows for *agile* networking. Every time when a new CSP is added, it is only required to configure the newly added CSP by using a few lines of commands. Every time when a CSP is deleted, no additional configuration change or for that matter nothing else needs to be done. This is because cloudcasting has a built-in auto-discovery mechanism that has not been seen in the *embedded virtual networks*.

The Cloudcasting data plane *scales* as well. Its default GVE encapsulation protocol allows to support 256 million clouds. In other technology such as, VXLAN, it only up to 16 million clouds are supported.

Due to the limitation of space, we won't discuss and describe other more desirable characteristics.

VI. Related Work

There are several works available that partially solve network virtualization problem; however, they do not provide a complete and consistent solution that sufficiently fulfills all basic requirements discussed earlier in this paper. In what follows, we discuss and compare a few prominent network-overlay approaches.

A. IETF NVO3

The cloudcasting architecture and protocol shares some goals chartered by the IETF working group NVO3 (Network Virtualization Overlays over Layer 3) [16]. The purpose of the NVO3 is to develop a set of protocols and/or protocol extensions that enable network virtualization within a data center environment that assumes an IP-based underlay. Cloudcasting varies from NOV3 in that cloudcasting is not just restricted to the data center, and it doesn't expect a specific structure or protocol conventions in the underlay. The NVO3 architecture may seem to be a reformulation of the BGP architecture, where NVEs (Network Virtualization Edge) and NVA (Network Virtualization Authority) resemble iBGP speakers and Route Reflectors, respectively, and NVO3-VNTP [14] resembles BGP update messages between an iBGP speaker and its Route Reflector. And therefore, NVA (RR) needs to learn and store routes from NVE (iBGP speaker) and then distribute those routes to other NVEs (iBGP speakers). However, this is not the case in Cloudcasting, wherein virtual route information is a function between CSPs, and the CRP is not involved. CRP is used for cloud membership auto-discovery and thus enables agile provisioning. Auto-discovery is missing from NVO3. We should emphasize that CRP has no route database inside. Auto-discovery mechanism in the Cloudcasting has a significant impact on the size of the database in CSP, and is also a common differentiator with other related work as discussed in the following sections.

B. VXLAN and BGP/EVPN

VXLAN is a data plane for network overlay encapsulation and decapsulation, and BGP/EVPN has been proposed as the control plane for VXLAN [4][12][13]. It works by adding new patches to BGP, which was originally designed for inter-domain routing for service providers.

The use of BGP in a data center will require some unnecessary operational actions and design concepts. For example, in order to deploy BGP/EVPN, the network operator must configure something like an AS (autonomous system) in substrate networks, which is not really a data center design concept.

Running BGP in a data center can also lead to serious scalability problems of peering sessions between iBGP speakers (VTEP-BGP). Typically, to address this problem, deployment of Route Reflectors (RR) is suggested which then speaks with every other VTEP-BGP to synchronize their BGP-RIB. As a result, no matter if a VTEP needs routes, all the other VTEPs will always send their routes to the VTEP either directly or indirectly through a Route Reflector, and the VTEP is required to filter out not needed routes through Route Target and other BGP policies. Distributing not needed virtual routes from RR to VTEP-BGP will levy an unnecessary overhead on the substrate network and burn CPU power, processing these BGP messages.

Operating BGP in the data centers not only makes operational cost of data centers as high as that of a service provider's network it also lacks the agility because BGP heavily relies on configurations (it is well known that configuration errors are a major cause of system failures [8]). For example, when a new BGP-VTEP is added/removed the operator has to configure all the BGP peering relationships by stating which BGP neighbors are peering among each other.

Finally, observe that when BGP was first designed, some principles were built-in; for example, iBGP peers should have received and synchronized the same copies of routes. In the case of clouds, many such principles are not applicable anymore.

Compared with BGP/EVPN, our cloudcasting architecture does not suffer from the drawbacks described above. By the means of auto-discovery and route distribution, only specific routes of a virtual network are distributed. Moreover, the role of CRP does not require it to be an intermediate hop between two CSPs to distribute the routes. The detailed comparison and evaluation is in progress and will be published elsewhere.

C. LISP based data center virtualization

Although Locator ID Separation Protocol (LISP) [15] is not an inherent data center virtualization technology, it has a framework to support network overlays. LISP achieves this by distributing encapsulated tenant (customer) routing information and traffic over provider (substrate) network through its control plane based on a mapping system. The LISP architecture includes Ingress/Egress Tunnel Routers (xTRs) and a mapping system (MS/MR) that maintains mappings from LISP Endpoint Identifiers (EIDs) to Routing Locators (RLOCs). LISP requires mapping information to be pulled on-demand and data-driven, xTRs also implement a caching and aging mechanism for local copies of mapping information.

Compared with LISP, Cloudcasting CSPs and LISP xTRs are similar in that they are the virtualization tunnel endpoints performing encapsulation and decapsulation. But the virtual route RIB or mapping databases are different in that (1) LISP's mapping database is a separate protocol element and xTR's local mapping database is built by pulling and kept by caching and aging, while a CSP's virtual network RIB is local and significant only to the CSP itself; (2) An xTR's local database is built on demand after receiving a packet without knowing its mapping information, while CSP's virtual network RIB is signaled through the cloudcasting control protocol; (3) A CSP can auto-discover other CSPs which join the same cloud, while

 ISBN: 978-1-61208-455-8

LISP xTR can only know about another particular xTR after querying the mapping database.

VII. Future Work

In this paper, we have presented a new routing scheme, called cloudcasting, for virtual networks. Some of the areas being further investigated and formulated include:

- Multicast Support: How does multicast work for different tenant networks on a common and shared substrate network?
- Interface between substrate and virtual network: A careful, thorough research on such interfaces is an area that may be covered through a converged policy model for cloudcasting.

While deliberately left out are discussions on various related topics such as, security, mobility, global scalability and inter-domain deployment models, these topics are being actively worked upon and are the key research areas moving forward.

VIII. Conclusion

Cloud-scale networking environments require a technology where virtual networks are first class objects; such that the coarse policies and routing decisions can be defined and applied on the virtual networks. Cloudcasting is a routing system based on converged, unified network virtualization and will evolve better because of lower provisioning costs and enhanced agility through auto discovery when compared with current virtual network schemes where virtual networks will rise only up to the limits of substrate network technologies.

Acknowledgment

The authors are thankful to the members of the Cloudcasting project, particularly those involved in the early design and proof of concept and prototype development for their collaboration and fruitful discussions.

References

[1] M. Mahalingam, et al, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", RFC 7348, 2014.

[2] M. Sridharan, et al, "NVGRE: Network Virtualization using Generic Routing Encapsulation", draft-sridharan-virtualization-nvgre-08 (work in progress), April 13, 2015.

[3] IEEE 802.1aq, "IEEE Standard for Local and metropolitan area networks—Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks—Amendment 20: Shortest Path Bridging", June 2012, doi: 10.1109/IEEESTD.2012.6231597.

[4] A. Sajassi et al., "A Network Virtualization Overlay Solution using EVPN", draft-ietf-bess-evpn-overlay-01.txt, work in progress, February 24, 2015.

[5] J. Touch, R. Perlman, "Transparent Interconnection of Lots of Links (TRILL): Problem and Applicability Statement", RFC 5556, May 2009.

[6] D. Fedyk, P. Ashwood-Smith, Allan, A. Bragg, and P. Unbehagen, "IS-IS Extensions Supporting IEEE 802.1aq Shortest Path Bridging", RFC 6329, April 2012.

[7] D. Eastlake, T. Senevirathne, A. Ghanwani, D. Dutt, and A. Banerjee, "Transparent Interconnection of Lots of Links (TRILL) Use of IS-IS", RFC 7176, May 2014.

[8] T. Xu, Y. Zhou, "Systems Approaches to Tackling Configuration Errors: A Survey", Article No.: 70, ACM Computing Surveys (CSUR) Volume 47 Issue 4, July 2015, pp. 70.1-70.41, doi:10.1145/2791577.

[9] J. Gross, T. Sridhar, et al., "Geneve: Generic Network Virtualization Encapsulation", draft-ietf-nvo3-geneve-00, May 2015.

[10] Cisco Nexus 7000 Series NX-OS VXLAN Configuration Guide. Available from: http://www.cisco.com/c/en/us/td/docs/switches/datacenter/sw/nx-os/vxlan/configuration/guide/b_NX-OS_VXLAN_Configuration_Guide.html.

[11] VMware® NSX for vSphere (NSX-V) Network Virtualization Design Guide. Available from: https://www.vmware.com/files/pdf/products/nsx/vmw-nsx-network-virtualization-design-guide.pdf.

[12] E. Rosen, and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, 2006.

[13] Sami et al., draft-boutros-bess-vxlan-evpn-00.txt, "VXLAN DCI Using EVPN", January 2016.

[14] Z. Gu, Virtual Network Transport Protocol (VNTP), draft-gu-nvo3-vntp-01, October 2015.

[15] D. Farinacci, V. Fuller, D. Meyer, and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, January 2013.

[16] M. Lasserre, et al., "Framework for Data Center (DC) Network Virtualization".

[17] M. Yu, Y. Yi, J. Rexford, and M. Chiang, "Rethinking virtual network embedding Substrate support for path splitting and migration", ACM SIGCOMM Computer Communication Review, Volume 38 Issue 2, April 2008, pp. 17–29.

Performance Bounds for Regular LDPC Codes for Asymmetric Channels

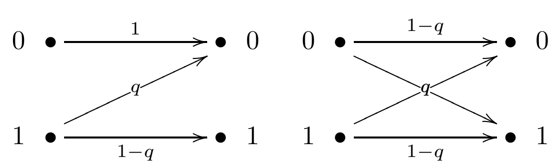
Pål Ellingsen
Department of Computing, Mathematics and Physics
Bergen University College
Bergen, Norway
Email: pal.ellingsen@hib.no

***Abstract*—It is well known that it is not possible to achieve capacity on an asymmetric channel using an even input distribution. In recent literature, complex code constructions has been proposed that gives rise to uneven input distributions to the channel such that capacity in theory can be achieved. However, it is of interest to know how well we can do on these channels with ordinary, linear codes due to the other desirable properties of such codes. In this paper, density evolution for symbol dependent channels is used in combination with a classical theorem by Gallager to bound the performance of regular Low Density Parity Check (LDPC) codes by showing that the check node degree of the graph describing a regular LDPC code, must go to infinity if the code is to achieve capacity on the Z-channel. Based on this, performance bounds for different check node degrees are calculated, and it is also shown that this is only a problem for small error probabilities.**

Keywords–Asymmetric channel; Regular LDPC codes; Gallager's theorem

I. INTRODUCTION

A binary asymmetric channel is a class of channels in which the probability of symbol error depends on the input symbol. In this paper, we will study a particular instance of this class, namely the binary asymmetric channel where the error probabilities of the two input values 0 and 1 are set to 0 and q as shown in Figure 1(a). This channel is also called the the Z-channel.



(a) The Z-channel (b) The binary symmetric channel

Figure 1. Channel models

Since the transition probabilities of the Z-channel are non-equal, the capacity not only depends on the probability of bit errors on the channel, but also on the input bit probability to the channel. Binary asymmetric channels have received considerable attention in classical coding theory, and many important works on this topic were compiled by Kløve in [1]. Since the introduction of iterative decoding, the coding community has succeeded in making codes that approach the Shannon bound on symmetric channels; meanwhile the relative performance of the best known codes for asymmetric channels has fallen behind. The cause for this has probably been the fact that due to their symbol dependent nature, asymmetric channels can not be analyzed using the techniques available for symmetric channels like density evolution in its original form. Further, the optimum input distribution actually depends on the error probability of the channel so it is impossible to achieve capacity on an asymmetric channel using a code with an even input probability. Since all linear codes have an even input distribution to the channel, capacity can not be achieved using a linear code, further complicating the task of approaching capacity on asymmetric channels. In this paper, new performance bounds for such codes are found by using a theorem by Gallager regarding the check node degree distribution of LDPC codes on the binary symmetric channel, and showing a similar result for the Z-channel. A necessary property of regular LDPC codes that are to approach the Shannon bound for all values of the error probability q is also proved.

In the rest of this paper, some background material on LDPC codes is first presented in Section II, then the development of the performance bound is given in Section III. Results are analyzed in Section IV, and finally, conclusions and possible future work are given in Section V and Section VI.

II. BACKGROUND

LDPC codes is a class of codes that uses belief propagation to attain near-capacity decoding. The codes are sparse linear block codes that may be pseudorandom or result of an explicit construction. The code may be represented as a bipartite graph construction called a Tanner graph (see Figure 2) where the parity checks of are represented by $\boxplus$ and the variable nodes are represented by $\bigcirc$. The decoding can be viewed as message passing on the same graph. Initially each variable nodes send messages to its parity check nodes indicating the probability of it being a $+1$ versus a -1. The check nodes returns the probabilities from all its neighbors, except from the node itself. The subsequent iterations proceeds analogously, except for that the information passed form the variable nodes is based on both the channel values and the information received in the previous iteration.

Recently there has been some interest in the use of LDPC codes for asymmetric channels. In [2] the use of LDPC codes on the some types of binary input fading multiple access channels (MAC) without channel state information (CSI) is investigated, while in [3], new code constructions are developed

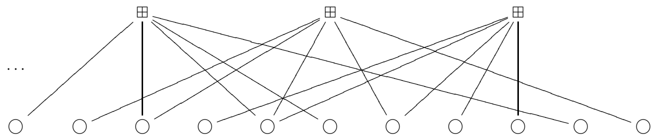


Figure 2. Tanner graph

based on transforming LDPC codes designed for the Binary Symmetric Channel (BSC) to bias the input distribution to the channel. Lately Mondelli, Marco and Urbanke [4] studies three different techniques designing concatenated codes with optimum input distribution for binary asymmetric channels. However, the bounds on performance of ordinary, regular LDPC codes are also of great interest as these codes are well known and widely used.

III. Performance bounds

In his seminal paper “Low-Density Parity-Check Codes”, Gallager [5] proved the following theorem which shows that decoding error probability for the binary symmetric channel is bounded away from 0 for all codes of rates above a threshold that depends on the check node degree d of the graph describing the code. We prove that there exists a similar bound for the Z-channel when considering regular, linear codes, and state as a corollary that as a consequence, the check node degree must go to infinity to achieve linear capacity for all values of q.

Theorem 1 (Gallager)**.** Let a regular parity check code of length n and rate r with check node degree d be used on a BSC with crossover probability q and let the codewords be used with equal probability. Let

$$q_d = \frac{1+(1-2q)^d}{2}. \tag{1}$$

Then

$$r > \frac{h(q_d)-h(q)}{h(q_d)}, \tag{2}$$

where $h(\cdot)$ is the binary entropy function, implies that for a fixed d the probability of decoding error is bounded away from 0 by an amount independent of n.

Proof: See [5]. ■

With certain adaptations, Gallager’s theorem is also true for the Z-channel.

Theorem 2. Let a linear, regular LDPC code of rate r and length n with check node degree d be used on a Z-channel with error probability q and assume the codewords are equiprobable. If

$$q_d = \frac{1+(1-2q)^{d/2}}{2} \tag{3}$$

and

$$r > \frac{2h(q_d)-h(q)}{2h(q_d)}, \tag{4}$$

where $h(\cdot)$ is the binary entropy function, then the probability of decoding error is bounded away from 0 by an amount independent of n.

Proof: We will follow Gallager’s proof of Theorem 1 closely in the following. Let u be a transmitted codeword, and let v be the received sequence. If we consider u and v as instances of the variables U and V, the mutual information between two variables U and V is given by

$$\begin{aligned} I(U;V) = h(U) - h(U|V) = \\ -\sum_u p(u)\log(p(u)) + \sum_{u,v} p(u,v)\log(p(u|v)) \end{aligned} \tag{5}$$

For simplicity, we will write $\sum_u p(u)\log(p(u)) = \overline{\log(p(u))}$ and $\sum_{u,v} p(u,v)\log(p(u|v)) = \overline{\log(p(u|v))}$. Then, the average mutual information per bit in a codeword can be written

$$\frac{1}{n}I(u,v) = -\frac{1}{n}\overline{\log(p(u))} + \frac{1}{n}\overline{\log(p(u|v))} \tag{6}$$

or, by the symmetry of the mutual information function

$$\frac{1}{n}I(u,v) = -\frac{1}{n}\overline{\log(p(v))} + \frac{1}{n}\overline{\log(p(v|u))} \tag{7}$$

The probability of decoding error is bounded away from 0 if there exists an ϵ independent of n for which the conditional probability $p(u|v)$ satisfies

$$\frac{1}{n}\overline{\log(p(u|v))} \geq \epsilon > 0 \tag{8}$$

We will proceed to prove the existence of such an ϵ by expanding the terms of equation (6).

The code has nr message bits, and thus there are 2^{nr} messages in the code, so assuming the codewords are equiprobable

$$\begin{aligned} -\frac{1}{n}\overline{\log(p(u))} &= -\frac{1}{n}\sum_u p(u)\log(p(u)) & (9)\\ &= -\frac{1}{n}\sum_u 2^{-nr}\log(2^{-nr}) & (10)\\ &= -\frac{1}{n}2^{nr}\cdot 2^{-nr}\cdot(-nr) & (11)\\ &= r \end{aligned}$$

For a linear code, the average weight of a codeword is $n/2$, and since each 1-digit in the sequence u has probability q of being different from the corresponding digit in v, the average conditional probability $p(v|u)$ is $q^{qn/2}(1-q)^{(1-q)n/2}$. Then, we can write

$$\begin{aligned} \frac{1}{n}\overline{\log(p(v|u))} &= \frac{1}{n}\log(q^{qn/2}(1-q)^{(1-q)n/2}) & (12)\\ &= \frac{1}{n}\cdot\frac{n}{2}(q\log(q)+(1-q)\log(1-q)) \\ &= -\frac{h(q)}{2} & (13) \end{aligned}$$

If we average over all parity checks, the weight of the nodes involved in each parity check should be $d/2$. Now, the probability that a parity check is satisfied is the probability that an even number of errors have occurred in the $d/2$ 1-nodes that belongs to the parity check. If we sum over all even-number error events we get the probability q_d that a parity sums to 0.

$$\begin{aligned} q_d &= \sum_{i \text{ even}} \binom{d/2}{i} q^i (1-q)^{d/2-i} & (14)\\ &= \frac{1+(1-2q)^{d/2}}{2} & (15) \end{aligned}$$

To verify (14), rewrite the right hand side as

$$\frac{(1-q+q)^{d/2}+(1-q-q)^{d/2}}{2} \tag{16}$$

and expand it as a binomial series.

A received word v can be described by the parities of the $n(1-r)$ parity checks together with the received values in some set of nr linearly independent variable nodes. The entropy of a received word is $h(v) = -\overline{\log(p(v))}$ and the entropy of a parity check is $h(q_d)$. Then, since the entropy of a variable node is at most 1 bit and dependencies only can reduce the overall entropy, we have

$$-\frac{1}{n}\overline{\log(p(v))} \leq (1-r)h(q_d)+r \tag{17}$$

If we substitute (9), (12) and (17) into (6) we get

$$\frac{1}{n}\overline{\log(p(u|v))} \geq \frac{h(q)}{2} - (1-r)h(q_d) \tag{18}$$

By hypothesis, there is an $\epsilon > 0$ that satisfies

$$r = \frac{2h(q_d)-h(q)+\epsilon}{2h(q_d)} \tag{19}$$

Substituting for (19) in (18) we get

$$\frac{1}{n}\overline{\log(p(u|v))} \quad \geq \quad \epsilon \tag{20}$$

■

Based on this theorem, we can formulate the following corollary.

Corollary 1. Let $C_{\frac{1}{2}}(q)$ be the maximum achievable rate for error free decoding of a linear code on a Z-channel with error probability q. Given the prerequisites of Theorem 2, a necessary condition for an LDPC code to achieve $C_{\frac{1}{2}}(q)$ for all values of q is that d must go to ∞.

Proof: The capacity of the Z-channel is

$$\max_p(h(p(1-q)) - ph(q)) \tag{21}$$

where p is the input distribution and q is the error probability. Since we restrict ourselves to using linear codes, the highest achievable rate of the channel is

$$\begin{aligned} C_{\frac{1}{2}}(q) &= h(\frac{1}{2}(1-q)) - \frac{1}{2}h(q) && (22)\\ &= 1-\frac{1}{2}((1+q)\log(1+q) - q\log q) && (23)\end{aligned}$$

For computational convenience we will use the form

$$1-\frac{h(q)}{2h(q_d)} \tag{24}$$

instead of the original form of the bound in (4).

Assume d is finite.

Let q^* be the value of q for which

$$C_{\frac{1}{2}}(q) = 1-\frac{h(q)}{2h(q_d)}. \tag{25}$$

By the monotonicity of the log function there is a unique q^*. When $q=\frac{1}{2}$, $q_d=\frac{1}{2}$ for all values of d so $1-\frac{h(q)}{2h(q_d)}=\frac{1}{2}$, and also $C_{\frac{1}{2}}(\frac{1}{2}) = h(\frac{1}{4})-\frac{1}{2}$. Consequently, when $q=\frac{1}{2}$

$$C_{\frac{1}{2}}(q) < 1-\frac{h(q)}{2h(q_d)} \tag{26}$$

for all values of d.

Further, $h(q_d)=0$ for $q=0$ since $q_d(0)=1$ so $1-\frac{h(q)}{2h(q_d)}$ is not defined for $q=0$. We can, however, find an expression for $1-\frac{h(q)}{2h(q_d)}$ as $q\to 0$ by taking the limit

$$\lim_{q\to 0} 1-\frac{h(q)}{2h(q_d)}, \tag{27}$$

Initially, we can simplify (27) to

$$\lim_{q\to 0} 1-\frac{h(q)}{2h(q_d)} \quad = \quad 1-\frac{1}{2}\lim_{q\to 0}\frac{h(q)}{h(q_d)} \tag{28}$$

Let $d'=\frac{d}{2}$. Then

$$\lim_{q\to 0}\frac{h(q)}{h(q_d)} \quad = \quad \lim_{q\to 0}\frac{q\log q}{\frac{1-(1-2q)^{d'}}{2}\log\frac{1-(1-2q)^{d'}}{2}} \tag{29}$$

Since both numerator and denominator in (29) go to 0 when q goes to 0, we can apply L'Hopital's rule so that

$$\begin{aligned} &\lim_{q\to 0}\frac{q\log q}{\frac{1-(1-2q)^{d'}}{2}\log\frac{1-(1-2q)^{d'}}{2}}\\ &= \lim_{q\to 0}\frac{1+\frac{\log e}{\log q}}{\frac{d'(1-2q)^{d'-1}}{2}\left(2\log\frac{1-(1-2q)^{d'}}{2}+\log e\right)\frac{1}{\log q}} && (30)\end{aligned}$$

We calculate the limit of the denominator in (30) separately by applying L'Hopital's rule repeatedly:

$$\begin{aligned} &\lim_{q\to 0}\left(2\log\frac{1-(1-2q)^{d'}}{2}+\log e\right)\frac{1}{\log q} && (31)\\ &= \lim_{q\to 0}\frac{2\log\frac{1-(1-2q)^{d'}}{2}}{\log q}+\frac{\log e}{\log q} && (32)\\ &= \lim_{q\to 0}\frac{2\log\frac{1-(1-2q)^{d'}}{2}}{\log q}\\ &= \lim_{q\to 0}\frac{2\cdot 2d'(1-2q)^{d'}\frac{\log e}{1-(1-2q)^{d'}}}{\frac{\log e}{q}} && (33)\\ &= \lim_{q\to 0}\frac{4qd'(1-2q)^{d'}}{1-(1-2q)^{d'}}\\ & && (34)\end{aligned}$$

Applying L'Hopital's rule to (34) yields

$$\begin{aligned} &\lim_{q\to 0}\frac{4qd'(1-2q)^{d'}}{1-(1-2q)^{d'}} && (35)\\ &= \lim_{q\to 0}\frac{4d'(1-2q)^{d'}-8q(d'^2-d')(1-2q)^{d'-2}}{2d'(1-2q)^{d'-1}} && (36)\\ &= \lim_{q\to 0}\frac{2d'(1-2q)^{d'}}{d'(1-2q)^{d'-1}} && (37)\\ &= 2 && (38)\end{aligned}$$

Substituting (38) into (30) gives us

$$\lim_{q\to 0} \frac{1+\frac{\log e}{\log q}}{\frac{d'(1-2q)^{d'-1}}{2}\left(2\log\frac{1-(1-2q)^{d'}}{2}+\log e\right)\frac{1}{\log q}}$$
$$= \lim_{q\to 0} \frac{1}{d'(1-2q)^{d'-1}}$$
$$= \frac{1}{d'} \tag{39}$$

Finally, substituting (39) into (28) and setting $d' = \frac{d}{2}$ again we get

$$\lim_{q\to 0} 1-\frac{h(q)}{2h(q_d)} \quad = \quad 1-\frac{1}{d} \tag{40}$$

Now, since we assume d is finite, (26) and (27) imply that $0 < q^* < \frac{1}{2}$ so there exists some interval $\langle 0, q^*\rangle$ for which $1-\frac{h(q)}{2h(q_d)} < C_{\frac{1}{2}}(q)$ making it impossible to achieve $C_{\frac{1}{2}}(q)$ for those values of q.

Assume $d \to \infty$.
Since $(1-2q) \in \langle -1, 1\rangle$ for $q \in \langle 0, 1\rangle$ we see that

$$\lim_{d\to\infty} q_d \quad = \quad \lim_{d\to\infty} \frac{1+(1-2q)^{d/2}}{2} \tag{41}$$
$$= \quad \frac{1}{2} \tag{42}$$

for all $q \in \langle 0, 1\rangle$, and so

$$\forall q \in \langle 0, 1\rangle \lim_{d\to\infty} h(q_d) = 1 \tag{43}$$

Thus, under these conditions, the limit of (24) when d goes to infinity becomes

$$\lim_{d\to\infty} 1-\frac{h(q)}{2h(q_d)} = 1-\frac{h(q)}{2} \tag{44}$$

We want to know when

$$C_{\frac{1}{2}}(q) < 1-\frac{h(q)}{2h(q_d)}. \tag{45}$$

Using (21) this expands to

$$h(\frac{1}{2}(1-q)) - \frac{1}{2}h(q) < 1-\frac{h(q)}{2h(q_d)} \tag{46}$$

When $d \to \infty$, we can substitute (44) into (46), and the inequality becomes

$$h(\frac{1}{2}(1-q)) - \frac{h(q)}{2} < 1-\frac{h(q)}{2} \tag{47}$$

This reduces to

$$h(\frac{1}{2}(1-q)) < 1 \tag{48}$$

From the properties of the entropy function of a binary variable we can deduce that this is true for all values of $q \in \langle 0, 1\rangle$. Further, from (40) we get

$$\lim_{d\to\infty}\lim_{q\to 0} 1-\frac{h(q)}{2h(q_d)} = 1 \tag{49}$$

Thus $1-\frac{h(q)}{2h(q_d)}$ goes to 1 when q goes to 0. Further, since the entropy function $h(q)$ is symmetric about the line $q = \frac{1}{2}$, the function $1-\frac{h(q)}{2h(q_d)}$ is also symmetric about $q = \frac{1}{2}$ and therefore $1-\frac{h(q)}{2h(q_d)}$ also goes to 1 when q goes to 1.

We can conclude that $C_{\frac{1}{2}} \leq 1-\frac{h(q)}{2h(q_d)}$ for all $q \in [0, 1]$ when $d \to \infty$. Ergo, for a code to achieve capacity for all q, d must go to ∞. ∎

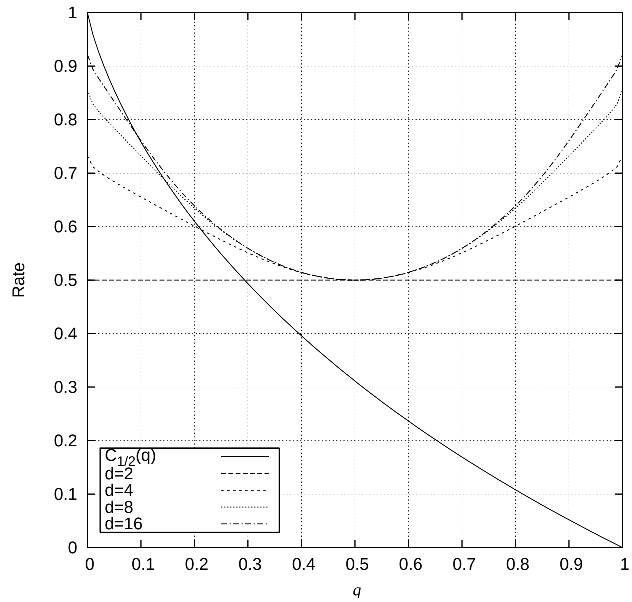


Figure 3. Comparison between $C_{\frac{1}{2}}(q)$ and the upper bound from (4) for different values of d.

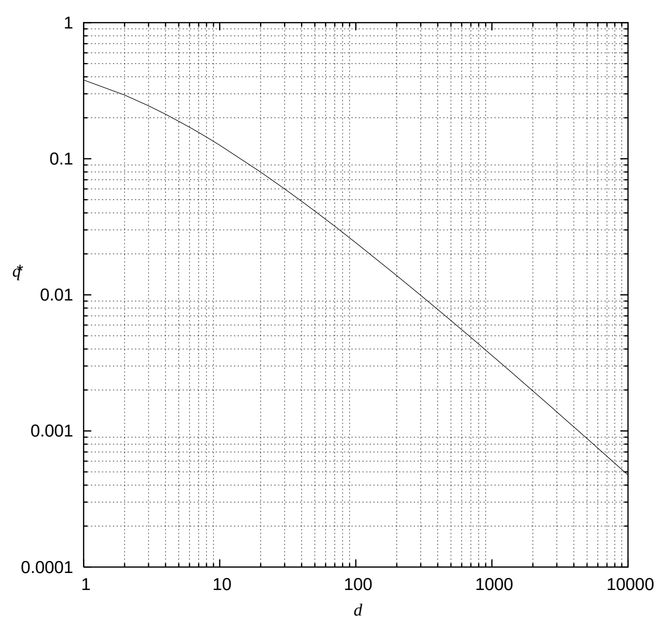


Figure 4. The intersection point q^* as a function of d

IV. Results and Discussion

As stated in Corollary 1: for finite d there will be some values of q for which $1 - h(q)/2h(q_d) > C$ so capacity cannot be achieved for all q. In a practical implementation, d will necessarily have to be finite, so the interesting question is how fast the intersection point q^* between $C_{\frac{1}{2}}(q)$ and $1-h(q)2h(q_d)$, illustrated in Figure 3 for different values of d, approaches 0 as d grows. It appears to be a hard problem to find an exact solution of the equation $C_{\frac{1}{2}}(q) = 1 - h(q)/2h(q_d)$, but numerical evaluation can give an indication of the rate at which q^* approaches 0. Results for d from 1 up to 10000 are given in Figure 4, and it is apparent that q^* decreases relatively slowly with increasing d.

V. Conclusion

Hence, building on Gallager's theorem, we have proved that the check node degree of the Tanner graph of an LDPC code imposes a lower bound on error free decoding also for the Z-channel. We have shown that this implies that capacity can not be achieved for all values of q for finite d, and that as d goes to infinity, the interval of q where error probability is bounded away from zero goes to 0. However, we see that this interval decreases only slowly, so a very high check node degree is a necessary condition to achieve $C_{\frac{1}{2}}$ for small values of q. Thus, these results tell us that the performance of regular LDPC codes on the Z-channel is bounded away from zero for small values of q, and that optimum performance is hard to achieve as q goes to 0. However, the theorem provides a bound on the performance of a regluar LDPC code for a given d.

VI. Future work

The results in this paper have been proved for regular LDPC codes. However, the class of irregular LDPC codes has so far proved to have the best performance, at least for symmetric channels. A natural step would therefore be to extend the above results to irregular codes as well.

References

[1] T. Kløve, Error Correcting Codes for the Asymmetric Channel. HiB, N-5020 Bergen, Norway: Department of Informatics, University of Bergen, 1995.

[2] N. Marina, "Ldpc codes for binary asymmetric channels," in Telecommunications, 2008. ICT 2008. International Conference on. IEEE, 2008, pp. 1–7.

[3] R. Gabrys and L. Dolecek, "Coding for the binary asymmetric channel," in Computing, Networking and Communications (ICNC), 2012 International Conference on. IEEE, 2012, pp. 461–465.

[4] M. Mondelli, R. Urbanke, and S. H. Hassani, "How to achieve the capacity of asymmetric channels," in Communication, Control, and Computing (Allerton), 2014 52nd Annual Allerton Conference on. IEEE, 2014, pp. 789–796.

[5] R. G. Gallager, "Low density parity check codes," IRE Trans. Information Theory, vol. 8, pp. 21–28, 1962.

[6] P. Ellingsen, "Iterative coding for the asymmetric channel," University of Bergen, Tech. Rep. 295, April 2005.

[7] C.-C. Wang, S. R. Kulkarni, and H. V. Poor, "Density evolution for asymmetric memoryless channels," in Proc. 3rd International Symposium on Turbo Codes, 2003.

 ISBN: 978-1-61208-455-8

Over the Top Content Streaming Adaptive System- Implementation and Validation

Serban Georgica Obreja, Radu Iorga, Eugen Borcoci,
Cristian Cernat, Marius Vochin
University POLITEHNICA of Bucharest
Bucharest, Romania
Emails: serban@radio.pub.ro,
radu.iorga@elcom.pub.ro
eugen.borcoci@elcom.pub.ro
cristian.cernat@elcom.pub.ro
marius.vochin@elcom.pub.ro

Jordi Mongay Batalla
National Institute of Telecommunications
Warsaw, Poland
email: jordim@interfree.it

Daniel Negru, Joachim Bruneau-Queyreix
LaBRI Lab, University of Bordeaux
Bordeaux, France
Emails: daniel.negru@labri.fr
jbruneauqueyreix@labri.fr

***Abstract* — Adaptive content streaming is an efficient and cheap solution to achieve a good Quality of Service and Experience for media streaming, having an Over- the –Top light architecture, i.e., working on top of the current IP technologies. The adaptation in the system considered here consists in content server initial optimized selection based on multi-criteria algorithm and then in-session media adaptation (using dynamic adaptive streaming) and/or server switching. In this paper, a first set of results on the implementation and functional validation of the system are presented. It focuses on presenting some of the functional validation scenarios and the tests results of the server selection component.**

Keywords — Content delivery; Dynamic Adaptive Streaming over HTTP; Quality of Service; Quality of Experience; Monitoring, Server and Path selection.

I. INTRODUCTION

The light Over-the-Top (OTT) architectures are recently developed for media/content delivery over the current public Internet. These are more simple and cheaper in comparison to complex solutions involving the network resources management and control architecture, like Content Oriented Networking [1].

This paper considers an OTT-style content streaming system, proposed, designed and implemented by the European *DISEDAN* Chist-Era project [2], (*service and user-based **DI**stributed **SE**lection of content streaming source and **D**ual **A**daptatio**N**,* 2014-2015). The business actors involved are: *Service Provider (SP)* delivering the content services to the users and possibly owing and managing the transportation network; *End Users (EU)* that consumes the content; a *Content Provider (CP)* could exist, owning some *Content Servers (CS)*. In DISEDAN light architecture, it is assumed that CSs are also owned by the SP.

The DISEDAN solution novelty [2][9]-[12], consists in: (1) *two-step server selection mechanism* (at SP and at EU) using algorithms that consider context- and content-awareness and (2) *dual adaptation mechanism* consisting of *media adaptation* and *content source adaptation* (by *streaming server switching*) when the quality observed by the user suffers degradation during the media session.

For in-session media adaptation, the *Dynamic Adaptive Streaming over Hypertext Transfer Protocol - HTTP (DASH)* technology has been selected. The DASH is a recent multimedia streaming standard, to deliver high quality multimedia content over the Internet, by using conventional HTTP Web servers [3-6]. It minimizes server processing power and is video codec agnostic. A DASH client continuously selects the highest possible video representation quality that ensures smooth play-out, in the current downloading conditions. This selection is performed on-the-fly, during video play-out, from a pre-defined discrete set of available video rates and with a pre-defined granularity (according to video segmentation).

This presents the DISEDAN testbed where the system components and samples of validation scenarios have been implemented. This paper study is focused on phase 1 of functioning, i.e., initial server selection. The specific DASH topics are treated in other studies [12].

The paper structure is the following. Section II is a short overview of related work. Section III outlines the overall DISEDAN architecture and main design decisions. Section IV contains the paper main contributions, focused on DISEDAN system testing and validation performed on a real life testbed. Section V contains conclusions and future work outline.

II. RELATED WORK

Assuring a "good" Quality of Experience (QoE) based on QoS control at server and transport levels is an important feature of the real time media related services. Apart from resource provisioning method, which supposes a complex management infrastructure at network level, adaptive methods are recently considered as good solutions [3][4].

Media flow adaptation is used in recent standards [5][6], as a significant technique to improve the QoE. *Content server switching* during session (assuming that replica servers are available) can additionally be decided if the media adaptation no longer produces good results. To these two, the DISEDAN solution adds an initial server selection phase (based on cooperation between SP and EU) integrating all three in a single solution. The initial server selection can be based on optimization algorithms like *Multi-Criteria Decision Algorithms (MCDA).* In [8][9], several scenarios are proposed, analyzed and evaluated, considering the availability of different static and/or dynamic input parameters. Therefore several control plane design decisions are possible [10], different in complexity/performance. The dynamic capabilities for the initial CS selection and then for adaptation decisions depends essentially on the power of the DISEDAN monitoring system [10]. The system combines in a novel solution the DASH functionalities with additional monitoring in order to finally realize the dual adaptation.

III. SUMMARY ON DISEDAN SYSTEM ARCHITECTURE AND DESIGN DECISIONS

This section provides a short description of the DISEDAN architecture and main design decisions. More complete description is given in [10][11][12].

The connectivity between CSs and EU Terminals (EUT) is assured by traditional *Internet Services Providers (ISP) / Network Providers (NP*) - operators. The ISP/NPs do not enter explicitly in the business relationships set considered by DISEDAN, neither in the management architecture. Service Level Agreements (SLAs) might be agreed between SP and ISPs/NPs, related to connectivity services offered by the latter to SP; however such SLAs are not directly visible at DISEDAN system level.

The system is flexible; it can work either in OTT style, or over a managed connectivity service offered by the network. An implicit assumption is that network environment is the traditional TCP/IP. No reservation for connectivity resources, neither connectivity services differentiation at network level are supposed (but they are not forbidden). The SP does not commit to offer strong QoS guarantees for the streaming services provided to EUs, therefore no SLA relationships between EUs and SPs management entities is mandatory. However, it is assumed that a Media Description Server exists, managed by SP, to which EUT will directly interact.

The media streaming actions are transport-independent. The EUT works as a standalone application; no mandatory modifications applied to the conventional SP; however, SP should provide some basic information to EUT, to help it in making initial server selection (and optionally to help in-session CS switching). The decision about dual adaptation (media flow adaptation and/or CS switching) is taken mainly locally at EUT, thus assuring user independency and avoiding complex SP-EUT signaling during the session. Several CSs exist, known by SP; so, the SP and/or EUs can operate servers' selection and/or switching. DISEDAN does not treat how to solve failures, except attempts to do media flow DASH adaptation or CS switching.

The work [11] has defined all requirements coming for EU, SP and derived the general and specific system requirements and some assumptions and constrains. The architecture (Figure 1) has been determined by such requirements. Details on that are included in [11]. The functional blocks correspond respectively to SP, EUT and CS. Note that only relevant to DISEDAN blocks are shown in the picture.

The Service Provider (SP) includes in its Control Plane:

MPD File generator – dynamically generates Media Presentation Description (MPD) XML file, containing media segments information (video resolution, bit rates, etc.), ranked list of recommended CSs and, optionally - current CSs state information and network state (if applicable).

Selection algorithm – runs Step 1 of server selection process. It exploits *MCDA* [7][8] to rank the CSs and media representations.

Monitoring module – collects information from CSs and processes it to estimate the current state of each CS.

The End User Terminal (EUT) includes the modules:

Data Plane: DASH (access and application) – parses the MD file received from SP and handles the download of media segments from CS; *Media Player* – playbacks the downloaded media segments. The standard ISO/IEC 23009-1, "Information technology -- Dynamic adaptive streaming over HTTP (DASH)" [5], defines the DASH-Metrics client reference model, composed of *DASH access client (DAC)*, followed by the *DASH-enabled application (DAE)* and *Media Output* (MO) module. The DAC issues HTTP requests (for DASH data structures), and receives HTTP request responses.

Control Plane: Content Source Selection and Adaptation engine – implements the dual adaptation mechanism; *Selection algorithm* – performs the Step 2 of server selection process. It can also exploit MCDA, or other algorithms to select the best CS from those recommended by SP; *Monitoring module* – monitors changing (local) network and server conditions.

The CS entity includes the modules:

Data Plane: Streaming module – sends media segments requested by End Users; *Monitoring module* – monitors CS performance metrics (CPU utilization, network interfaces utilization, etc.).

Figure 1 shows the main functional steps: (1) EUT issues to SP a media file request. (2) SP analyzes the status of the CSs and runs the CS selection algorithm. (3) SP returns to EUT an ordered list of candidates CS (SP proposal, embedded in a MD- xml) file. (4) The EUT finally selects the CS, by running its own algorithm. (5) EUT starts asking segments from the selected CS. During media session, the EUT makes quality and context measurements. Continuous media flow DASH adaptation is applied, or, (6) CS switching is decided. From the EU point of view, the steps 1-2-3 composed the so-called Phase1 and steps 4-5-6 composed the so-called the Phase 2.

During the receipt of consecutive chunks, the user's application can automatically change the rate of the content stream (based on DASH measurements, which are out of scope in this paper) and/or also can switch to another CS.

The design decisions are taken for the Control Plane, details on this being given in [10].

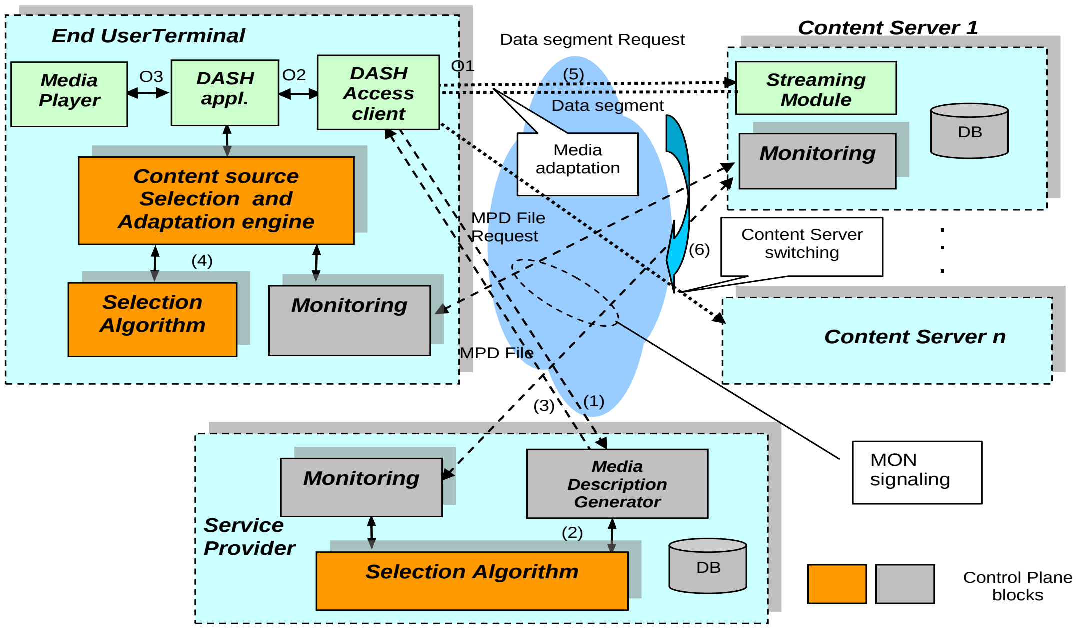


Figure 1. DISEDAN general architecture

IV. DISEDAN SYSTEM TESTING AND VALIDATION

This section is dedicated to present the test configuration and validation scenarios. The overall goal is to validate the content server selection phase in different server and network load conditions. Two sample scenarios, out of the complete set performed, are given as examples in this paper.

A. *Testbed configuration*

An experimental testbed has been built for DISEDAN functionalities validation (Figure 2). The system comprises three independent IP network domains, equipped with several core and edge/border routers (Linux based). No QoS technologies are active in these networks. Several DISEDAN entities are connected to this network: SP, EU, and CS through some access networks. Note that the presence of the access networks in the overall system is not essential, given the OTT-style of work for DISEDAN.

B. *Basic signaling test and validation*

The sequence of steps for functional validation scenario is illustrated in Figure 2 as a set of actions:

1. The EUT requests a streaming service from SP. The request contains the ID of the media service requested.

2. The SP gets from its local database the identity of the servers hosting the requested content. Using its monitoring module, the SP interrogates the monitoring agents located on the CSs about the CS state. The main monitoring parameters collected by SP are: CS processor load, CS free memory (normalized at the total memory), number of streaming processes on CS, total bandwidth on the network interface.

3. After receiving the monitoring parameters, the SP selects, using the MCDA, the best servers from the list of servers hosting the requested resource.

4. An ordered list of selected servers is returned to EUT. The best server from SP's point of view is on the first position on the list.

5-6. The EUT performs a second selection step. The *Round Trip Time (RTT)* and the distance in terms of *hop count* are measured from EUT to the CSs in the list recommended by SP. The communication on fifth step is performed only with the SP's selected servers not with all CSs. Based on this metric the best server is selected by EUT.

Final step: the EUT requests the service from the selected CS; the latter will start streaming packets towards the EUT.

All the functional steps described above have been validated by capturing and analyzing the messages exchanged between EUT, CS, SP.

C. *Validation of the server selection algorithm in different servers load conditions*

This test illustrates how the CS server's selection is influenced by the load of the content servers. The same setup as the one illustrated in Figure 2 was used.

In the first phase, the terminal requests from Service Provider a new service. The request follows the steps of the selection procedures, as described in the previous section, till it is accepted by the SP and the terminal gets the movie stream from the selected server. The selection of the best CS servers is done by the MCDA algorithm running on SP. It selects the best server based on the list with the network distance between the terminal and the content servers and the lists with the CPU load on content servers, the number of sessions on CSs, the load on the serving network interface on CSs, and the occupied memory on CSs. The values of these parameters are first normalized and then passed to the MCDA algorithm organized into a matrix, MCDA matrix, as it is described in [8][9].

 ISBN: 978-1-61208-455-8

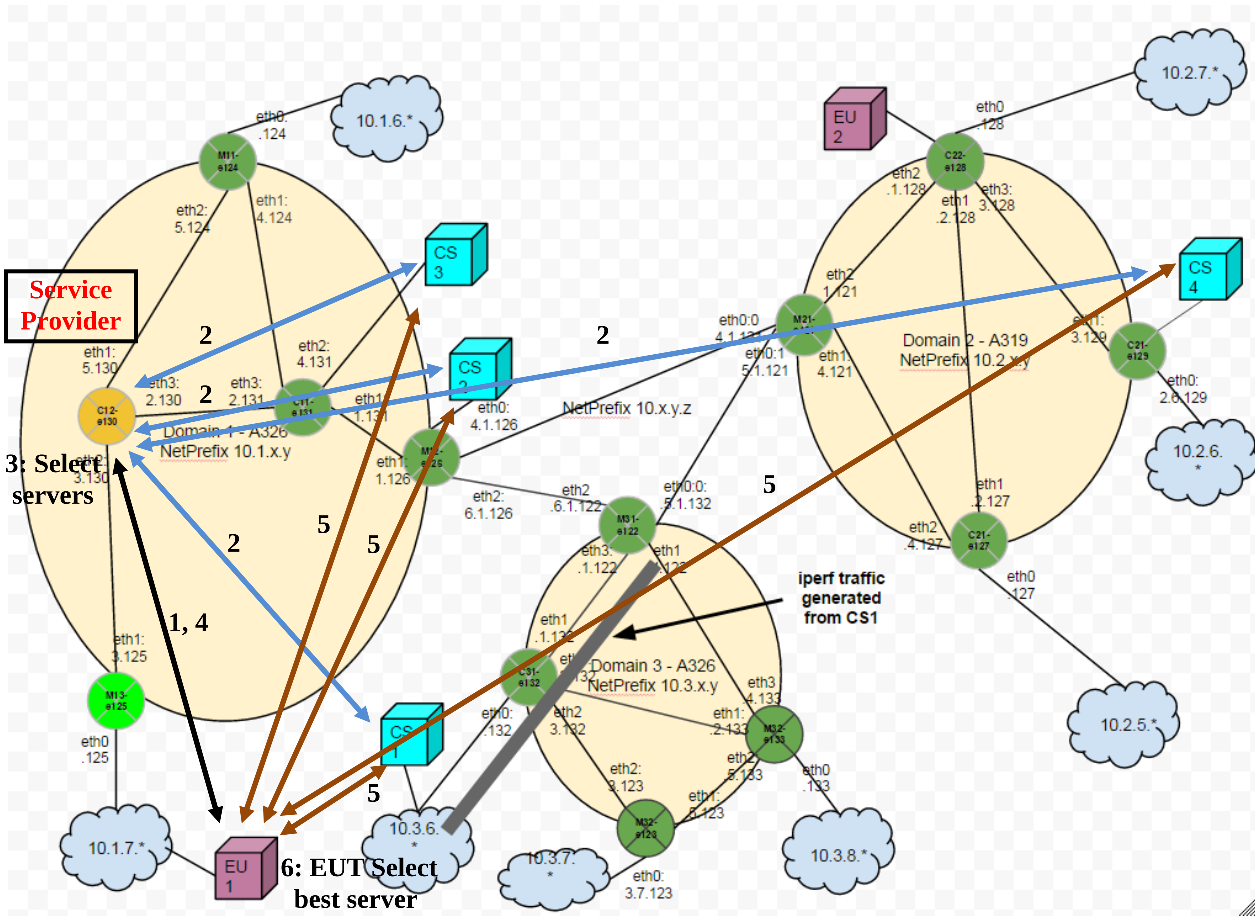


Figure 2. DISEDAN System testbed; interactions between system's modules

The normalization is done such that a small value indicates a busy server, while a big value indicates a free server. The MCDA algorithm uses a **min-max** approach. First, it selects for each server the minimum value among the values of the normalized monitoring parameters for that server. So, it has the worst score for each server. Secondly, it selects the best server as the one that have the greatest score among the scores determined in the first phase.

In this scenario, initially, all the servers' monitoring parameters are similar (because the servers are not loaded), except the network distances which are different. In the first phase the selection, is decided by the network distance. The closest server is chosen as the best one.

In the second phase, the terminal triggers another request for the same service. Prior to this action, the CPU on the server which was selected in the first phase is loaded using the ***stress*** Linux application. The CPU is loaded at around 55%. Consequently, the value of the parameter associated with the CPU load in the MCDA matrix will decrease, forcing the MCDA algorithm to select another best server from the list of servers hosting the requested resource.

D. Validation of the server selection algorithm in different network load conditions

This scenario is similar with the previous one; the difference consists in the parameter used to influence the server selection algorithm. The load on the content server network interface will be used to influence the selection. In this case, the MCDA will be forced to select the servers CS1 and CS4 in the first phase, by decreasing the network distance between EUT and CS1/CS4. The CS1 will be the recommended server due to the lowest network distance. Then, the CS1 sending interface will be loaded with traffic. *Iperf* application will be used to load CS1's interface, by sending traffic from CS1 to the router *e122* (Figure 2).

After *iperf* application is started, a second request for the same resource will be initiated by EUT1. Due to the *iperf* traffic, the monitoring parameter *network interface load* will increase significantly on server CS1, which determines the associate MDCA parameter to decrease accordingly. As a consequence, the MCDA algorithm will select CS4 server, which was the second on the list, as the best serving server to provide the requested resource.

 ISBN: 978-1-61208-455-8

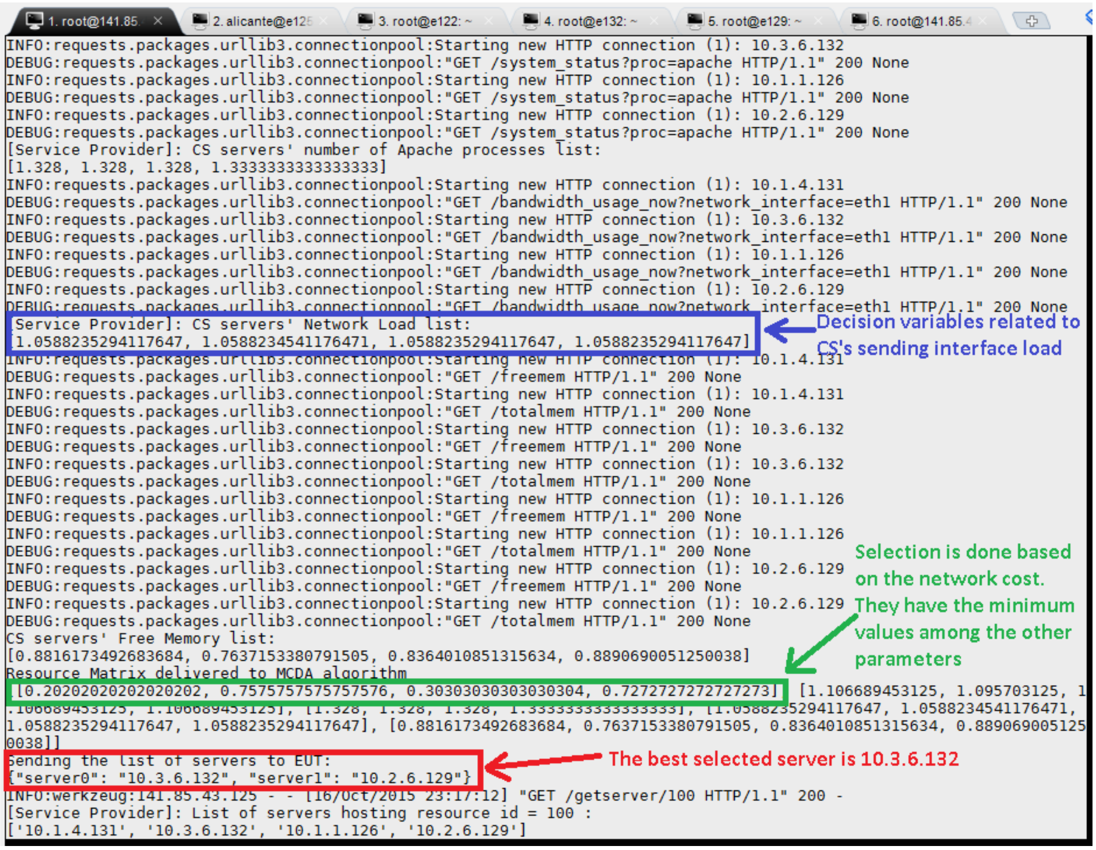


Figure 3. CS selection at SP when the servers are not loaded

In the first phase of this scenario, the content servers are not loaded, which means that decision variables will have similar values. In this case, the selection is done based on the values for the variable representing the network distance between EUT and CSs. The minimum network distance is allocated for CS1 (10.3.6.132), second on the list, followed by CS4 (10.2.6.129), fourth on the list. In Figure 3, the values for the decision variables related to the traffic load on the sending interface for each server hosting the requested resource are shown. Because there is no load on any of the interfaces, the values are similar.

With green there are emphasized the values for the variables related to network cost. They have the minimum values among the values of the parameters in the resource matrix used by MCDA, which means that the selection is done based on these values. As it can be seen, the maximum value among them is for the server 10.3.6.132, which is recommended as the best server, as is emphasized with blue on Figure 3.

In the second phase, the sending interface of the CS1 server is loaded with background traffic, which is generated with ***iperf*** application and is sent to the e122 machine, as can be seen in Figure 2. In this case, the value of the decision variable related to the load on the CS1's sending interface will decrease, determining the MCDA to select as the best server the CS4 machine, with the IP 10.2.6.129. This is illustrated by the SP's messages captured Figure 4. As it is emphasized with blue, the sending interface load decision variable for server CS1 (10.3.6.132) decreased (the second position in the array - its value is around 0.674), determining that this variable will be selected to represent CS1. It also determines the selection of the CS4 server as the best one, due to the ***min-max*** approach in server selection algorithm. So, in the second phase, the selected server is CS4 (10.2.6.129), because it has all the decision variables bigger than 0.674, which is the value of the decision variable selected to represent CS1 - as can be seen in the resource matrix delivered to MCDA in Figure 4. CS4 decision variables are on fourth position in each vector, the servers' order being shown in the last row of messages captured in Figure 3.

V. CONCLUSIONS AND FUTURE WORK

In this paper, a functional validation for the multimedia delivering system proposed in the framework of the DISEDAN project, is presented. A media streaming system, working in OTT style has been considered. To improve the QoS/QoE, the system combines an initial (Phase1) optimized content server selection (based on multi-criteria decision algorithms- MCDA) with in-session (Phase 2) media flow adaptation or content server switching.

The results presented in this paper illustrate the basic functionalities of the DISEDAN system's modules and the operating mode of the MCDA algorithm in the first phase of the content server's selection process, in different content servers and network load conditions.

Figure 4. CS selection at SP when the sending interface on the best CS is loaded with traffic.

The experimental results obtained on the real life testbed validated the capability of the system to properly select the "best server", thus efficiently contributing to prepare the media session phase. A more complete set of results regarding the first phase of the server selection were obtained on a simulation model developed for the DISEDAN system and are presented in [15].

More elaborate tests to find some quantitative values for the QoE improvements are under investigation and will be developed in order to take advantage of the real life prototype available. Also, investigation on the second selection phase and the adaptation profess will be performed.

ACKNOWLEDGMENT

This work has been partially supported by the Research Project DISEDAN, No.3-CHIST-ERA C3N, 2014- 2015.

REFERENCES

[1] 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, pp. 121-127.

[2] *** http://wp2.tele.pw.edu.pl/disedan/ [retrieved:May, 2015]

[3] T. Dreier, "Netflix sees cost savings in MPEG DASH adoption," 15 December 2011. [Online]. Available: http://www.streamingmedia.com/Articles/ReadArticle.aspx?ArticleID=79409. [retrieved: October, 2014].

[4] I. Sodagar, "The MPEG-DASH Standard for Multimedia Streaming Over the Internet," MultiMedia, IEEE, vol. 18, no. 4, 2011, pp. 62 - 67.

[5] ISO/IEC 23009-1, "Information technology -- Dynamic adaptive streaming over HTTP (DASH) -- Part 1: Media presentation description and segment formats," ISO/IEC, Geneve, second edition, 2014.

[6] ETSI TS 126 247 V11.7.0 (2014-07) (UMTS); LTE; Transparent end-to-end Packet-switched Streaming Service (PSS); Progressive Download and Dynamic Adaptive Streaming over HTTP (3GP-DASH), (3GPP TS 26.247 version 11.7.0 Release 11, 2014).

[7] A. P. Wierzbicki, "The use of reference objectives in multiobjective optimization". Lecture Notes in Economics and Mathematical Systems, vol. 177. Springer-Verlag, pp. 468–486.

[8] A. Beben, J. M. Batalla, W. Chai, and J. Sliwinski, "Multi-criteria decision algorithms for efficient content delivery in content networks", Annals of Telecommunications - annales des telecommunications, Springer, vol. 68, Issue 3, 2013, pp. 153-165.

[9] E. Borcoci, M. Vochin, M. Constantinescu, J. M. Batalla, and D. Negru, "On Server and Path Selection Algorithms and Policies in a light Content-Aware Networking Architecture", ICSNC 2014 Conference, http://www.elcom.pub.ro/disedan/docs/ICSNC%202014%20Conf.pdf.

[10] E. Borcoci, C. Cernat, and R. Iorga, "Control Plane Design for a Content Streaming System with Dual Adaptation", AICT 2015 Conference, https://www.thinkmind.org/index.php?view=article&articleid=aict_2015_6_40_10154

[11] E. Borcoci, ed., et al., "D2.1 System requirements and comparative analysis of existing solutions for media content server selection and media adaptation", DISEDAN Project, July 2014, http://wp2.tele.pw.edu.pl/disedan.

[12] J. Mongay Batalla, ed. et.al., , "D2.3 Specification of the Dual adaptation mechanism," DISEDAN Project, 2014.

[13] ***, http://www.postgresql.org, [retrieved: March, 2015].

[14] ***, http://nodejs.org, [retrieved: March, 2015].

[15] O. Catrina, E. Borcoci, and P. Krawiec, "Two-Phase Multi-criteria Server Selection for Lightweight Video Distribution Systems," 27th IFIP TC7 Conference 2015 on System Modelling and Optimization Integration of Optimization, Modeling and Data Analysis for Solving Real World Problems.