NFVRG
Internet-Draft
Intended status: Informational
Expires: November 18, 2017

M. A. Vazquez-Castro
T. Do-Duy
UAB
S. P. Romano
A. M. Tulino
Unina

May 17, 2017

Network Coding Function Virtualization
draft-vazquez-nfvrg-netcod-function-virtualization-01

Abstract

This document describes network coding as a network function. It also describes how a network coding function can be virtualized and integrated with virtual network functions architectures. The network coding function is not a traditionally implemented network function in dedicated hardware as those that have triggered network function virtualization. It refers to a novel network functionality that generalizes classic packet-level end-to-end coding. Classic packet-level end-to-end coding helps in the provision of quality of service by trading off delay and reliability. Network coding goes beyond that by enabling in-network optimized re-encoding, which can provide both throughput gains and diverse network-controlled degrees of reliability. Consequently, a virtualized network coding function can serve as a flow engineering tool over virtualized networks (e.g. over network slices).

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on November 18, 2017.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 2
2. Conventions used in this document 4
3. Network coding as a network function 5
4. Virtual Network Coding Function 6
4.1. Virtualization of flows 6
4.2. Integration with ETSI NFV architecture 7
4.3. Example 8
4.3.1. The SHINE use case 9
5. Conclusions 13
6. Acknowledgements 13
7. IANA Considerations 13
8. Security Considerations 13
9. References 14
9.1. Normative Information References 14
9.2. Conceptual ground basis 14
9.3. Application references 14
Authors' Addresses 16

1. Introduction

Network coding(NC) is a novel technology that can be seen as the generalization of classic point to point coding to coding for network flows. As with classic coding, both information theoretical and algebraic codes literature provide the conceptual solid basis of NC. Such conceptual basis has clarified NC benefits and corresponding tradeoffs, which need to be considered in practical implementations of the technology.

NC does not replace end-to-end (packet-level block) coding, which is a well-established technology for the per-flow provision of quality of service by trading off delay and reliability. Instead, NC

provides coding within and across network flows relying on in-network re-encoding based on service-intent-oriented policy strategies. By means of such policy strategies, the provision of quality of service that NC can offer can be tailored according to desired network service intent.

For its operation, NC relies on having access to network, computation and storage resources throughout the network. Such novel networking, computational and storage ingredients of a coding technology calls for novel practical design approaches to truly exploit the potential capabilities of NC.

On the other hand, Network Function Virtualization (NFV) and NC can be seen as different ways to address different challenges in the design of upcoming network technologies. Moreover, NC is not a traditionally implemented network function in dedicated hardware, which are the network functions that have triggered the design of NFV architectures. However, in this document we show the feasibility and benefits of virtualizing the network coding function.

The objective of this document is not to explain network coding technology. The interested reader should find this information outside this document.

The objective of this document is fundamentally two fold. First, we show that NC can be designed as a (modular) network function. The modularity is convenient for the user and is given as sets of elementary functionalities (toolboxes) that are defined independent of the physical network. Second, we show that the NC function requirements of connectivity, computation and storage resources find a natural practical design solution in the integration of the NC function with available NFV architectural frameworks. Such solution is described here and it combines network protocol-driven and system modular-driven design approaches.

The resulting Virtual Network Coding Function (VNCF) can be useful for upcoming networking needs derived from network virtualization.

In order to provide the readers with a flavor of how the ideas presented in this draft might be applied to real-world communication scenarios, we will describe an interesting use case related to the creation of a hybrid satellite-terrestrial infrastructure for the effective delivery of multimedia contents to end-users. The architecture in question envisages a combination of multicast, simulcast and unicast communication scenarios where satellite links are exploited to support local in-network caching. The satellite acts as the interconnection link between distributed in-network

caches and terrestrial CDN (Cotent Delivery Network) and/or feeds edge-network caches at micro-centre locations.

The example architecture will be orchestrated through an enhanced NFV management framework exposing Network Coding functionality as a Virtual Network Function (VNF). Such a function will in our case implement a novel "combined coding" technique targeting the optimization of multicast-enabled transmissions in the presence of caching. More precisely, it will leverage cutting-edge solutions for decentralized random caching which, combined with an original content distribution technique based on coded multicast, will allow us to abtain "order-optimal" performance.

In a nutshell, the above mentioned technique allows us to somehow multiplex multiple transmission chunks on a single packet, thus enabling us to meet the twofold objective of optimizing the use of the broadcast communication medium while at the same time dramatically increasing the security level of satellite-enabled transmissions, by making them resilient to network attacks like snooping and eavesdropping.

2. Conventions used in this document

The following terms defined in this document can be found in the ETSI NFV [etsi_gs_nfv_002_v1.2.1] and the IETF [I-D.irtf-nwcrg-network-coding-taxonomy].

Coherent Network Coding: Source and destination nodes know network topology and coding operations at intermediate nodes.

Noncoherent Network Coding: Source and destination nodes do not know network topology and intermediate coding operations. In this case, random network coding can be applied.

Flow: A stream of physical packets logically grouped from the network coding perspective. These packets may come from the same application (in that case they are identified by the five-tuple: source and destination IP address, transport protocol ID, and source and destination port of the transport protocol), or come from the same source host (in which case they are identified by the 3-tuple source and destination IP address, Type of Service (TOS) or Diffserv code point(DSCP)). This distinction depends on the use-case where network coding is applied.

Intra-flow coding: Network coding over payloads belonging to the same flow.

Inter-flow coding: Network coding over payloads belonging to multiple flows.

End-to-end coding : Transport stream is coded and decoded at end-points.

Coding node: Node performing coding operations.

Virtualized Infrastructure Manager (VIM): functional block that is responsible for controlling and managing the NFVI compute, storage and network resources, usually within one operator's Infrastructure Domain.

Virtualized Network Function (VNF): implementation of a Network Function that can be deployed on a Network Function Virtualization Infrastructure (NFVI).

Virtualized Network Function Manager (VNFM): functional block that is responsible for the lifecycle management of VNF.

NFV Infrastructure (NFVI): totality of all hardware and software components which build up the environment in which VNFs are deployed.

NFV Orchestrator (NFVO): functional block that manages the Network Service (NS) lifecycle and coordinates the management of NS lifecycle, VNF lifecycle (supported by the VNFM) and NFVI resources (supported by the VIM) to ensure an optimized allocation of the necessary resources and connectivity.

NFV Management and Orchestration (NFV-MANO): functions collectively provided by NFVO, VNFM, and VIM.

3. Network coding as a network function

NC design involves different domains. We can identify at least three domains:

Coding domain - domain for the design of network coding codebooks, coherent or noncoherent encoding/decoding schemes, performance benchmarks, appropriate mathematical-to-logic maps, etc. This is a domain fundamentally designed by coding theorists.

Functional domain - domain for the design of the functional properties of NC to achieve the desired design requirements upon abstractions of networks and systems. This domain jointly requires to consider physical-logical abstraction, identification of network coding application to either inter-flow or intra-flow network coding,

service intent and related networking for the provision of quality of service.

Protocol domain - domain for the design of physical signaling/ transporting of the network coded information flow as one way or interactive protocols.

The functional domain can be designed interpreting NC as a network function. In order to provide the designer with sufficient flexibility, NC elementary functionalities can be grouped as a set of toolboxes that the designer can use. We define the following three toolboxes:

- o Coding/Re-encoding/Decoding Functionalities (CRDF).

- o Flow Engineering Functionalities (FEF) performing optimization of available network resources to optimally perform NC to meet the service design targets depending on the (statistical) status of the networks (congestion, link failures, etc).

- o Physical/Abstraction Functionalities (PAF) performing interaction with available storage and computation physical resources that are abstracted by the other toolboxes.

4. Virtual Network Coding Function

4.1. Virtualization of flows

An important differentiating aspect of NC with respect to traditional networking technologies is the following. A network flow for a NC network function is understood as a stream of physical packets logically grouped from the network coding perspective.

NC can optimize the NC operation abstracting such physical flow as a mathematical model, which can be subject of computational manipulation. This makes NC to be naturally integrated into a virtualized framework of abstract entities such as virtual network or network slices. This is because in the NC case, not only the network and resources are abstracted, but also the stream of packets is abstracted.

Consequently, when interpreting NC as a functionality provided to the network, NC function virtualization simply consists of integrating the NC functional toolboxes described in the previous section into existing architectural NFV frameworks. The virtualization of the network flow is managed by the NC function (CRDF toolbox), and the virtualization of all the functionalities described in Section 3 has no difference with respect to any other network function.

4.2. Integration with ETSI NFV architecture

Figure 1 shows our proposed virtual NC network function (VNCF). It is integrated with the ETSI NFV architecture given the abstracted underlying physical system/network as part of NFVI.

The integration naturally includes too exchanges between VNCF and NFV-MANO over reference points.

Clearly, the functionalities of the FEF toolbox need to interact with the NFVO, VNFM, and VIM. Note that the NFVO two main responsibilities of orchestration of NFVI resources across VIMs and the life-cycle management of network services, fit perfectly the needs of the FEF and PAF toolboxes. Specifically, the FEF can obtain available network, connectivity and computation resources, geo-statistical status of the networks such as congestion, link failures, etc. With these, NC operation can be optimized to meet the service design targets given the service-specific design constraints. The optimization may result into manipulation of the (non-physical) flows and other flow engineering policies. On the other hand, the FEF can interact with the VIM to obtain the allocation, upgrade, release, etc of NFVI resources.

```
        +-------------------------------------------+  +---------------+
        |   Virtualized Network Functions (VNFs)    |  |               |
        |  -------   -------   -------   -------    |  |               |
        |  |     |   |     |   |     |   |     |    |  |               |
        |  | VNF |   | VNF |   |VNCF |   | VNF |    |  |               |
        |  |     |   |     |   |     |   |     |    |  |               |
        |  -------   -------   -------   -------    |  |               |
        +-------------------------------------------+  |               |

        +-------------------------------------------+  |               |
        |         NFV Infrastructure (NFVI)         |  |      NFV      |
        | -----------    -----------    ----------- |  |  Management   |
        | | Virtual |    | Virtual |    | Virtual | |  |      and      |
        | | Compute |    | Storage |    | Network | |  | Orchestration |
        | -----------    -----------    ----------- |  |               |
        | +---------------------------------------+ |  |               |
        | |         Virtualization Layer          | |  |               |
        | +---------------------------------------+ |  |               |
        | +---------------------------------------+ |  |               |
        | | -----------  -----------  ----------- | |  |               |
        | | | Compute |  | Storage |  | Network | | |  |               |
        | | -----------  -----------  ----------- | |  |               |
        | |          Hardware resources           | |  |               |
        | +---------------------------------------+ |  |               |
        +-------------------------------------------+  +---------------+
```

Figure 1: ETSI NFV framework with one VNCF box as part of the set of available VNFs

4.3. Example

We describe here a high-level example of a general procedure of interaction between the VNCF and the NFV-MANO. The NFV-MANO has repositories that hold different information regarding network services (NSs) and VNFs (VNCF is part of VNFs). There are four types of repositories as follows:

- o VNF catalogue represents the repository of all usable VNF packages, supporting the creation and management of the VNF packages.

- o NS catalogue represents the repository of all usable NSs.

- o NFV instances is the repository that holds details of all VNF instances and NS instances, represented by either a VNF record or a NS record, respectively, during the execution of VNF/NS life-cycle management operations.

- o NFVI resources is the repository that holds information about NFVI resources utilized for the establishment of NS and VNF instances.

Assume a network abstracted as a set of N coding nodes, each with encoding/re-encoding/decoding and (possibly) multi-link connectivity. A user of the VNCF wants to provide an ultra-reliable service (e.g. mission-critical communications) to the N nodes. The performance objectives are given as a set of N reliability and delay objective performance metrics, which are geo-location dependent. We call this VNCF instantiation as a virtual geo-network coding function (VGNCF), which is activated and its management and orchestration take place.

A detailed interaction with the architectural blocks (some under definition) is as follows.

- o TBD

The next section will briefly introduce a real-world application scenario associated woth the effective deivery of multimedia content in a hybrid satellite-terrestrial network.

4.3.1. The SHINE use case

SHINE stands for "Secure Hybrid In Network caching Environment". It has two main distinctive features, associated with, respectively, the broadcast-enabled satellite core and the edge distribution networks. Within the former part of the network, we rely on network coding in order to define a coded multicast technique allowing us to improve both performance and security of communications. At the edges of the distribution network, which also act as in-network caches, we instead leverage cutting-edge streaming technologies (namely, MPEG-DASH and/or WebRTC) in order to optimize content distribution towards the end users of the network.

A high-level view of the SHINE architecture is reported in Figure 2. The picture highlights the main logical components of the architecture, in terms of macro-blocks and related functionality. Namely, we identify the following elements:

1. a source encoder block, taking on the responsibility of properly encoding the original content in order to allow for the subsequent coded multicast transmission over the satellite network;

2. the core satellite-enabled communication infrastructure, looking after DVB-enabled transmission of coded multicast frames from the content provider to the edge caches, both during the cache

population phase and during the steady-state operation of the CDN;

3. two different "flavors"" of edge access networks: (i) a WebRTC-enabled access network, included in the architecture in order to demonstrate SHINE's operation in the presence of this novel real-time communication infrastructure at the edges of the overall content delivery architecture; (ii) an MPEG-DASH enabled access network, included in the architecture in order to demonstrate SHINE's capability of leveraging such a well-assessed web-based distribution approach.

```
                                                  +-------+ Satellite GW
                                                /-|       | -
                                            /---  |       |  \-
                                        /---      +-------+    \-
                                    /--              /-  |       \ Network
encoder
                                /---               /-    |        \-
+-------+
                            /---                /-      /          \-
|       |
                        /--                  /-       |             \-
|       |
                    /---                    /-         |
\|       |
                /---                      /-           |
|       |
               --                        /-            |
|       |
           +---+                       /-              /
+-------+
           |   |                     /- Coded Multicast|                  -/
           |   |Edge cache         /-  Transmission    |                -/
           |   |                 /-                    |              -/
           |   |               /-                      /
+----------------+
+---------------+            /-                        |        |
|
|          +---+|          /-                          |        |
|
|               |        /-                            |        |Content
Provider|
|               |      /-                              |        |
|
|               |    /-                                /        |
|
|               |   -                                 |         |
|
|               | +---+                               |
+----------------+
+---------------+ |   |                            +-|-+              +---+
                  |   | Edge cache                 |   |              |   |
Origin
                  |   |                  Edge cache|   |              |   |---+
server
                  |   |                            |   |              |   |   |
farm
            +---------------+                      |   |              |   |
+-|-+
```

```
            |     +---+     |           +---------------+             |   | | |
|
            |               |           |          +---+|             +---+ | |
|
            |               |           |               |                 | | |
|
            |               |           |               |                 +---+
|
            |               |           |               |                   |
|
            |               |           |               |
+---+
            +---------------+           |               |
        Edge distribution network       +---------------+
                                   Edge distribution network
```

Figure 2: The SHINE use case

The system components which are of uttermost importance in this document, in view of the observation that they can highly benefit from the effective utilization of Network Coding as a Virtual Network Function are analyzed in further detail in the following.

The source encoder is a software module implementing the main logic behind the proposed coded multicast technique. It is in charge of

transforming the original content and applying the required transformations in order to arrive at a representation format that is suitable for the subsequent coded multicast transmission. The component in question has indeed to look after both the cache population phase and the actual content delivery phase. The cache population phase envisages that the edge caches pre-fetch some content, based on appropriate functions of the content library, as well as on information about estimated future users' demand for content. During the delivery phase, on the other hand, the source forms a multicast "codeword" to be transmitted over the shared link in order to meet the actual users' content demands. As already stated, we envisage that the cache population phase is carried out through transmission (over the satellite core network connecting source node with edge caches) of content chunks. As to the content delivery phase, it takes place through DVB-encapsulated transmission, over the satellite network, of coded multicast frames.

Satellite Core Network is the network segment that basically interconnects the Source Encoder, which produces and processes multimedia contents, and several Edge Networks, where the in-network caches represent the boundary network elements. The satellite network trunk leverages standard DVB-S or DVB-S2 broadcast

The delivery phase hence occurs after the placement phase, when traffic is high and network resources are scarce and expensive (e.g., in the evening). At the beginning of this phase, each user reveals its request for one of the m files. The server is informed of these K requested files. In response, the server sends RF bits (or the equivalent of R files) over the shared link. The number R is called the rate of the server transmission or equivalently load of the satellite link. From the server transmission and its local cache content, each user needs to be able to recover their requested files. As already anticipated, SHINE looks after both the content placement and delivery phases. The objective is to minimize the rate R with which every possible set of user demands can be satisfied. The constraints are the storage limit during content placement and the recovery requirement during content delivery. Both phases are generic for both coded and uncoded schemes, but naively performed in the uncoded case. In fact, when relying on uncoded or naive multicasting during the delivery phase, it is well known that the optimal caching strategy is to cache the top M most popular files at each user cache. Though, this is in general far from optimal when coding can be used in the delivery phase. Thanks to the adoption of the dynamically provided Virtual Network Coding Function, SHINE discloses the potential of caching-aided code design and illustrates its major advantages compared to the optimal caching policy under uncoded (naive) multicasting. In a nutshell, the designed architecture shows how the combined use of edge caching and coded

multicasting represents a promising approach to simultaneously serve multiple unicast demands via coded multicast transmissions, leading to order-of-magnitude bandwidth efficiency gains.

5. Conclusions

This memo presents a preliminary version of proposal for the design of NC as a network function. It is also discussed that it can be virtualized and integrated into a NFV architecture.

6. Acknowledgements

The authors want to thank Dr. Harald Skinnemoen for useful comments and discussions. The first author wants to thank Dr. Carlos J. Bernardos and Luis M. Contreras for useful discussions.

The authors also want to acknowledge the following ongoing projects.

1. GEO-VISION - GNSS driven EO and Verifiable Image and Sensor Integration for mission-critical Operational Networks. EU funded project under the call H2020-GALILEO-2014-1 by the European Global Navigation Satellite Systems Agency (project reference 641451).

2. SatNetCode - Satellite Network-Coding for high performance, semantic-aware mission-critical visual communications. This project is funded by the European Space Agency, under contract No. 4000115046/15/NL/US.

3. HENCSAT - Highly Efficient Network Coding for Satellite Applications Test-bed. This project is funded by the European Space Agency, under contract No. 4000118143/16/NL/EM.

4. SHINE - Secure Hybrid In Network caching Environment. This project is funded by the European Space Agency, under Contract No. 4000118273/16/NL/CLP.

7. IANA Considerations

This memo includes no request to IANA.

8. Security Considerations

This memo includes no Network Coding Function Virtualization - specific security definitions yet.

9. References

9.1. Normative Information References

[etsi_gs_nfv_002_v1.2.1]
"Network Function Virtualisation (NFV); Architectural Framework", 2014.

[etsi_nvf_whitepaper]
"Network Functions Virtualisation (NFV). White Paper 2", 2014.

[I-D.irtf-nwcrg-network-coding-taxonomy]
Firoiu, V., Adamson, B., Roca, V., Adjih, C., Bilbao, J., Fitzek, F., Masucci, A., and M. Montpetit, "Network Coding Taxonomy", draft-irtf-nwcrg-network-coding-taxonomy-01 (work in progress), October 2016.

[RFC6363] Watson, M., Begen, A., and V. Roca, "Forward Error Correction (FEC) Framework", RFC 6363, 2011.

9.2. Conceptual ground basis

[AHL00] Ahlswede, R., Cai, N., Y. R. Li, S., and R. W. Yeung, "Network information flow", in IEEE Trans. Inform. Theory, vol. 46, pp. 1204-1216, July 2000.

[KOE03] Koetter, R. and M. Medard, "An algebraic approach to network coding", in IEEE/ACM Trans. on Networking, vol. 11, n. 5., pp. 782-795, October 2003.

[LI03] Y.R.Li, S., W. Yeung, R., and N. Cai, "Linear network coding", in IEEE Trans. Inform. Theory, vol. 49, n. 2., pp. 371-381, February 2003.

9.3. Application references

[ALE13] Alegre-Godoy, R. and M. A. Vazquez-Castro, "Spatial Diversity with Network Coding for ON/OFF Satellite Channels", in IEEE Communications Letters, vol. 17, No. 8, pp. 1612-1615, August 2013.

[ALE15] Alegre-Godoy, R. and M. A. Vazquez-Castro, "Network Coded Multicast over Multi-beam Satellite Systems", in Mathematical Problems in Engineering, vol. 2015, Article ID 364234, May 2015.

[DO16.1] Do-Duy, T. and M. A. Vazquez-Castro, "Design of Virtualized Network Coding Functionality foR Reliability Control of Communication Services over Satellite", submitted to Special Issue on Network Coding. International Journal of Satellite Communications and Networking, 2016.

[Do16.2] Do-Duy, T. and M. A. Vazquez-Castro, "Network coding function virtualization", in IEEE 17th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), September 2016, INVIED PAPER.

[HAN15] Hansen, J., E. Lucani, D., Krigslund, J., Medard, M., and F. H. P. Fitzek, "Network coded software defined networking: enabling 5G transmission and storage networks", in IEEE Communications Magazine, 2015.

[HEI09] Heide, J., V. Pedersen, M., H. P. Fitzek, F., and T. Larsen, "Network Coding for Mobile Devices - Systematic Binary Random Rateless Codes", in ICC Workshops, 2009.

[SAX15] Saxena, P. and M. A. Vazquez-Castro, "DARE: DoF-Aided Random Encoding for Network Coding over Lossy Line Networks", in IEEE Communications Letters, vol. 19, No. 8, pp. 1374-1377, August 2015.

[SZA15] Szabo, D., Nemeth, F., Sonkoly, B., Gulyas, A., and F. H. P. Fitzek, "Towards the 5G revolution: A software defined network architecture exploiting network coding as a service", in SIGCOMM Comput. Commun, 2015.

[VAZ15.1] A. Vazquez-Castro, M., "A Geometric Approach to Dynamic Network Coding", in Information Theory Workshop, Jeju, Korea, October 2015.

[VAZ15.2] A. Vazquez-Castro, M., "Subspace coding over Fq-linear erasure satellite channels", in 12th International Symposium on Wireless Communication Systems, pp. 216-220, August 2015.

[VAZ15.3] A. Vazquez-Castro, M. and P. Saxena, "Network Coding over Satellite: From Theory to Design and Performance", in Volume 154 of the series Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, pp. 315-327, September 2015, INVITED PAPER.

Authors' Addresses

M.A. Vazquez-Castro
Autonomus University of Barcelona
Campus de Bellaterra
Barcelona, 08391
Spain

Email: angeles.vazquez@uab.es

Tan Do-Duy
Autonomus University of Barcelona
Campus de Bellaterra
Barcelona, 08391
Spain

Email: tan.doduy@uab.es

Simon Pietro Romano
University of Napoli Federico II
Via Claudio 21
Napoli, 80125
Italy

Email: spromano@unina.it

Antonia Maria Tulino
University of Napoli Federico II
Via Claudio 21
Napoli, 80125
Italy

Email: antoniamaria.tulino@unina.it