Information Exchange Management as a Service for Network Function Virtualization Environments

Article *in* IEEE Transactions on Network and Service Management · September 2016

DOI: 10.1109/TNSM.2016.2587664

3 authors:


Lefteris Mamatas
University of Macedonia


Stuart Clayman
University College London


Alex Galis
University College London

Information Exchange Management as a Service for Network Function Virtualization Environments

Lefteris Mamatas, *Member, IEEE,* Stuart Clayman, *Member, IEEE,* and Alex Galis, *Member, IEEE.*

***Abstract*—The Internet landscape is gradually adopting new communication paradigms characterized by flexibility and adaptability to the resource constraints and service requirements, including Network Function Virtualization (NFV), Software-Defined Networks (SDNs), and various virtualization and network slicing technologies. These approaches need to be realized from multiple management and network entities exchanging information between each other. We propose a novel *Information Exchange Management as a Service* facility as an extension to ETSI's NFV Management and Orchestration (MANO) framework, namely the Virtual Infrastructure Information Service (VIS). VIS is characterized by the following properties: (i) it exhibits the dynamic characteristics of such network paradigms; (ii) it supports information flow establishment, operation, and optimization; and (iii) it provides a logically-centralized control of the established information flows with respect to the diverse demands of the entities exchanging information elements. Our proposal addresses the information exchange management requirements of NFV environments and is information-model agnostic. The paper includes an experimental analysis of its main functional and non-functional characteristics.**

***Index Terms*—Network Function Virtualization, NFV Management and Orchestration, Information Exchange Management as a Service**

I. Introduction

There is a major shift in the Internet towards using *virtualized and programmable network functions* offering efficient resource utilization, optimized service function availability, dynamic resource scaling (both up & down for elasticity), network function flexibility, as well as adaptability benefits. The Network Function Virtualization (NFV) [5] concept implements network functions in software (such as middleboxes) by running them on commodity hardware like servers and switches, thereby reducing both the specialized infrastructure and the operational costs. Furthermore, the Virtualized Network Functions (VNFs) and the proposed equivalent NFV architectures [5], [6] bring significant efficiency and flexibility benefits. Considering that the number of middleboxes deployed in the Internet is comparable to the number of routers, NFV will be beneficial.

The above aspects are associated with a number of management and orchestration challenges which need to be addressed. The challenges include: (i) how to exploit this dynamism and flexibility, (ii) how to ensure that the required functions are being deployed and operating in a coherent and on-demand basis, and (iii) how to confirm that the solution remains manageable [7]. In this context, the European Telecommunications Standards Institute (ETSI), which leads the relevant NFV activities, proposed the Management and Orchestration (MANO) framework. MANO focuses on the provisioning of VNFs and the relevant operations, including orchestration and lifecycle management capabilities of the associated physical and virtual resources supporting the VNFs [6]. Most of the NFV platforms in research collaborative and industrial projects are influenced by MANO [7].

An important aspect here is to design the right resource management abstractions which enable efficient orchestration of such flexible functions, while hiding the heterogeneity of the multi-vendor equipment. We argue that these capabilities should be enabled by distributed NFV Entities, (which include NFV management components, VNFs, together with legacy management features for Network Functions), all having the necessary information to perform dynamic configuration changes [7] and/or to consume the information based on service necessity. According to [7], a facility supporting lightweight coordination among distributed decision makers with an aim to optimize both the usage of resources and the performance of services, is a key research issue.

Along these lines, the ETSI NFV ISG introduced reference points exchanging information elements and control messages [6], i.e. the interconnection points between the MANO functional blocks and the external management entities. A number of ETSI documents ([8], [9], [10], [11]) elaborate the definitions of the interfaces and the relevant information entity specifications. Although particular information exchange requirements are identified throughout the documents, the details of such operations and the protocols are left for future work or considered implementation issues.

In this paper, we architect and implement an *Information Exchange Management as a Service* solution. This realizes *Information Exchange Orchestration* which we define as an augmentation of information exchange management and its relevant processes with capabilities for logically centralized information flow establishment, optimization, coordination, and synchronization. Since the flows communicate *management/control information elements* or *VNF state* are different from other monitoring flows or data flows, they are referred to as information flows or state flows in this paper.

For effective management, it is important to maintain both global and local views of the network environment in a resource efficient way, but according to the diverse requirements

L. Mamatas is with the Department of Applied Informatics, University of Macedonia, 156 Egnatia Street, GR-546 36 Thessaloniki, Greece. S. Clayman and A. Galis are with the Department of Electronic and Electrical Engineering, University College London, Torrington Place, London WC1E 7JE, U.K.

This work was partially supported by the European Union DOLFIN [1] and UniverSELF [2] projects of the 7th Framework Program, and the 5GEX [3] and SONATA [4] projects of the EU H2020.

of entities producing or consuming particular information. We suggest that the MANO architecture should be extended with our proposal, thus improving MANO's service provisioning and network resources orchestration capabilities, through supporting adaptable information exchange features. Using off-the-shelf monitoring software does not suffice, since it does not match the dynamic and flexibility characteristics of NFVs.

Other approaches to information handling focus on fixed and static networks, such as the TMF Information Framework related works [12] [13]. Although ETSI is working on MANO information modeling aspects (e.g. the working document [8]), there is no consensus from the different stakeholders on the various information model proposals, as these models have not yet fully evolved for the highly dynamic NFV environments and they can only be considered as starting points [7]. Our facility provides information exchange facilities and complements the information modeling work at the level of information exchange orchestration. For these reasons, we created the VIS facility to be information model agnostic. This allows for wider applicability, as it can support particular information models in the future, and it also applies to the multi-segmentation / slicing of a network, where each slice may have its own separated information model.

This paper presents an abstracted and logically centralized information exchange management service, as an architectural feature of ETSI MANO [6], namely the Virtual Infrastructure Information Service (VIS). The VIS orchestrates information flows between the NFV Entities, which are configurable and can be information producers and information consumers (or sources and sinks). The VIS processes involve:

(a) The registration of information producers and consumers with their corresponding requirements and constraints (e.g. information model to use, maximum data rate, "freshness" of information etc);
(b) The negotiation activity between the entities and the VIS that matches producers with consumers, and defines the configuration of the required information flows; and
(c) The information flows establishing and monitoring through using efficient data paths based on the global view of the network and the expressed entity requirements and constraints.

Each information flow establishment considers both the registration information of the participating entities (e.g. NFV Entities) and the global performance goals in the system (coming from relevant orchestration or higher-level management functions and expressing the business strategies of the service and infrastructure providers). At any point, the VIS may trigger a re-negotiation and flow re-establishment for some or all of the information flows, in case of a different high-level performance goal decision or an unexpected event appearance, such as a failure. The VIS also supports the following:

(d) The collection, aggregation/processing, dissemination, storage, and indexing of information;
(e) Various communication methods between the management entities, including the Push/Pull, Publish/Subscribe, and Direct Communication method;
(f) Interfaces for exchanging information and for configuring the information flows;
(g) Alignment to both physical and virtual network space (i.e. for management facilities and VNFs, respectively); and
(h) An extensible architecture, allowing improvements to its behaviour when a relevant demand arises.

Our complementary work [14] presents a fully detailed description of the VIS software components, the sub-components, the interfaces, and the associated data flows, interactions and operations between these components. It includes a VIS functional validation analysis in a Software-Defined Infrastructure context. VIS is available as an open-source solution at [15].

Here we include experiments validating how VIS behaves in terms of the following non-functional key characteristics: (i) its *adaptability* to various numbers of applications, topology sizes, and requested communication methods; (ii) its *flexibility* to support global and local tuning of specified performance trade-offs; and (iii) its *scalability and stability* in cases of resource exhaustion.

Section II contrasts the proposed platform with the related works. Section III motivates our proposal, discusses its information model agnostic operation and presents example use-cases. Section IV highlights the VIS architecture along with its design and implementation details. Section V describes our experimental methodology and validates experimentally the behavior of the proposed platform, in terms of adaptability, scalability, flexibility and stability. Finally, section VI concludes the paper.

II. Related Work

Network Function Virtualization brings IT closer to the communication technologies through the softwarization of network functions. This strategy enables flexibility in service deployment and reduces the operational and infrastructure costs significantly. In practice, it requires a distributed operation of multiple NFV Entities, including MANO functions and VNFs. These distributed decision-making entities operate based on a global, per domain, view or on a local view of the network environment. Such a capability can be supported by an infrastructure that collects, processes, and disseminates information characterizing the system.

We argue that different NFV Entities have their own particular needs in terms of information characteristics and network constraints. For example, a network function that handles a failure is associated with real-time constraints (namely, to fix the error as soon as possible and avoid escalating the problem), but others may work efficiently in the background, exploiting unused resources.

Information manipulation should be abstracted away in a dedicated MANO function, while supporting logically centralized intelligence, and be both adaptable and programmable. In the past, such capabilities were mainly tightly-coupled within software components (being in the same NFV entity that consumes or produces the information). Another option is the use of off-the-shelf monitoring facilities as complementary tools or plugins. However, they are general purpose systems that are not aligned with or adapted to the dynamic requirements of the NFV environments.

Most relevant NFV proposals focus on VNFs or on network state management. Among them, solutions like the [16], [17], [18] handle the state separately, whilst others provide coordinated state management, e.g. [19], [20], [21], [22]. *OpenNF* [19] is a control plane architecture coordinating both internal Network Function (NF) and network forwarding state. It provides a communication path between the NFs and the controller. A protocol for communication between the VNFs and the controllers have been proposed in [23]. In [20], the authors introduce a logically centralized state management solution for middleboxes based on *OpenNF*. It aims to minimize the control-plane interactions through removing the OpenFlow / OpenNF controller from the critical path during state and traffic transfer. In their proposal, the state and packets are transferred between the VNFs in a peer-to-peer fashion.

Other proposals focus on the specific problems of VNF migration or VNF elasticity. In [21], the authors proposed a solution called *Pico Replication (PR)* focusing on the replication of flow-specific state using techniques from Virtual Machine replication systems. *FreeFlow* [22] splits flow-specific state among replicas and dynamically re-balances both existing and new flows across them, enabling elasticity (i.e. scaling up or down) of network services.

In contrast to the above VNF state handling proposals, VIS is an extension to the MANO architecture providing abstracted information management facilities to different types of NFV entities, such as NFV management entities and VNFs. Additionally, VIS supports the exchange of state and management information between the MANO functions and the VNFs. The complex problems of VNF inter-communication, including state synchronization due to VNF migration, or information support for SFC orchestration aspects are left for future work.

There are a number of *Information as a Service* proposals, in the context of clouds, that are mainly focussed on data analytics or SOAs and business-aligned services, such as [24], [25], [26]. The VIS *Information Exchange Management as a Service* proposal focuses on the information management aspects rather than on the information itself.

Some solutions, like OpenDaylight [27], use *netconf* (or its RESTful equivalent *restconf*) that supports communication of configuration/operational data, RPCs and notifications. The netconf protocol is tightly coupled with the YANG information model [28], [29], and is used for the installation, manipulation and deletion of network devices configuration, while the YANG model represents both configuration and state data of network elements. Netconf is standardized and supports transaction-safe configuration of devices. Compared to VIS, netconf is a protocol for device configuration rather than an abstracted information exchange service for a wide range of NFV entities, including MANO functions and VNFs. VIS does not exclude communication with the network devices using a similar protocol as one of the options for the deployed information flows.

In our work, we consider *Information Exchange Management* as a cornerstone feature of the MANO architecture whereby Information manipulation is provided by a logically centralized service, in a way that is consistent with the general performance goals of the system. Thereby, a clear network view is maintained: at a system; at a domain level; or at a local level using logically centralized intelligence, techniques for programmability, and an abstracted design. To our knowledge, this is the first work proposing a functionally rich *Information Exchange Management as a Service* facility that is aligned to NFV environments.

III. Abstracting Information Exchange

In this section, we motivate the use of VIS as a facility for handling information exchange in NFV environments, and elaborate its information-agnostic operation and discuss representative use-cases.

A. An Information Service as a MANO Extension

We show the Network Functions Virtualisation Management and Orchestration (MANO) framework [6] and its relation with our VIS platform in figure 1. MANO presents the management and orchestration aspects for the provisioning of VNFs with their related operations, such as the functions for configuration and the infrastructure that hosts them, where the latter is called Network Function Virtualization Infrastructure (NFVI). MANO consists of three main functional blocks: (i) the Virtualized Infrastructure Manager (VIM) which is responsible for controlling and managing the NFVI compute, storage and network resources; (ii) the VNF Manager (VNFM) that performs the VNFs lifecycle management; and (iii) the NFV Orchestrator (NFVO) performing resource orchestration (via the VIM and NFVI) and the lifecycle management of network services. The MANO framework includes a number of data repositories and reference points (as functional descriptions of interfaces) and other external functional blocks interacting with MANO, including the Element Management System (EMS), the VNFs, the Operation System Support (OSS) / Business System Support functions (BSS) and the NFVI.

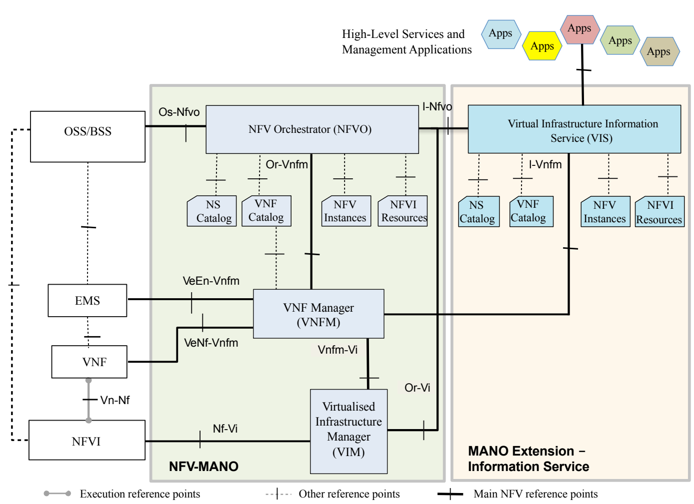


Fig. 1. The VIS as an NFV MANO Extension

A number of ETSI documents define the specific MANO interfaces and their information exchange primitives ([8], [9],

[10], [11]), but the connectivity service details for the relevant information flows are either not described or considered implementation issues. In these documents, a number of information interoperability aspects are identified. They advocate the adoption of an information producer-consumer paradigm using loosely-coupled interfaces and allowing different entities to consume the information based on service necessity, e.g. the services, applications and associated business and operational processes. Beyond that, they advise the use of either a pub-sub mechanism for notifying the context information changes, that may support information filtering, or using a relevant polling process. An example interaction is presented between the NFVO and the VIM, in order for the former to follow the resource allocation updates. Our VIS system implements such features, including the dynamic matching of information producers with consumers, the definition of the granularity level of information, pub/sub, together with polling mechanisms and information filtering.

We argue that the MANO information exchange capabilities should be abstracted away within a logically-centralized information service, realizing the above features, while being scalable, adaptable, and flexible to the diverse orchestration and service requirements. Such a strategy brings the following advantages: (i) the information flows are adaptable to the orchestration requirements and the dynamic network conditions; (ii) crucial NFV entities overcoming a systematic problem could be prioritized; (iii) information elements may be communicated to various information consumers and represented in compatible formats, and (iv) the co-existing information exchange processes can be optimized in a collective manner.

We show the proposed VIS augmenting NFV MANO with abstracted information exchange capabilities in figure 1. In this view VIS is considered as an external functionality and we present the high-level VIS architecture and its basic interactions with the three main MANO functions, namely the NFV Orchestrator, the VNF Manager, and the Virtualized Infrastructure Manager. The NFVO and the high-level services and management applications can influence the general behavior and performance of VIS (e.g. by defining global performance goals). There is a new reference point in the figure – *I-Nfvo* – which connects the VIS functionality to the NFV Orchestrator.

The MANO data repositories (i.e. the NS Catalog, VNF Catalog, NFV Instances, and NFVI resources) can potentially be integrated with the VIS *Information Storage and Indexing* function. We did not remove these data repositories from the main MANO architecture, in order to highlight the mapping of the elements and also to allow incremental adoption of VIS. Until their integration, these repositories communicate with the VIS through the reference point *I-Vnfm*. Such integration does not exclude the direct communication of MANO functions with the repositories, but delegates a relevant information exchange decision to the VIS (to use the VIS direct communication method).

B. Information-Model Agnostic Operation

Information elements are exchanged in the operation of the ETSI NFV facilities. Such information may describe a network service, a VNF, a Physical Network Function (PNF), a Virtual Link (VL), the Resource Allocation of the NFVI, aspects of a Service Function Chain like a VNF Forwarding Graph (VNFG), etc. The information elements may be either static, residing in descriptors (e.g. deployment templates for VNFs or network services), or dynamic, residing in records (i.e. runtime representations of VNF or network service instances).

Many information models, which can be used in NFV environments, are being devised and are progressing in parallel. These include: ATIS NFV [30], CIM [31], ETSI Information Model [8], ITU-T Information Model [32], MEF Common Information Model [33], IETF YANG [34], TMF SID [13], or OASIS TOSCA [35]. The proposed models each have particular advantages. The SID Service Model is suitable for OSS/BSS systems, i.e. to represent the information primitives between the MANO, the OSS/BSS and the EMS. YANG is already used as a candidate for modeling the ETSI NFV information elements, where as TOSCA can describe service components and their relationships using a service topology.

A full network service can be realized by chaining VNFs with PNFs, but there is not yet a unified information model that can cross the physical and virtual space from the service to the network resources level. A way forward could be a federated information model using common and consensually defined and inter-operable terms, concepts, objects (e.g. common data types and vocabularies, specifications of fault codes across multiple resources etc). This is defined as a key challenge in [6]. Another option could be to support translation between different information models at different parts of the SFCs. This issue is much more challenging in the context of end-to-end services over a combination of NFV functions, infrastructure, and legacy interconnected network systems, in the highly dynamic NFV environments and their evolutions (e.g. multi-domain services, services over multiple network slices and mobile network extensions).

The VIS provides Information Exchange Management as a Service capabilities that are information-model agnostic, i.e. by focusing on the information exchange aspects rather than the information itself. The VIS information flow establishment features support the negotiation between producers and consumers of meta-data regarding the information model to use. In this way, different information models may be used in different parts of a deployed network service using appropriate model translators. Such aspects are important and deserve their own independent study. We closely follow the evolution of the information models and we plan to integrate the relevant capabilities in the future.

C. Example Use-Cases

Here, we discuss example use-cases inspired by the ETSI NFV works that demonstrate the advantages of using VIS as a MANO extension:

- *Securing resources for several tenants* [36] – MANO is designed to enable resource sharing between different tenants, i.e. a number of co-existing tenants can secure and allocate resources, avoiding resource management race conditions and service degradation. In telco environments that have stringent SLAs, specific reliability and performance requirements may be in place. NFVO is the central point of orchestration of resource consumption by VNFs and network services but the resources are being reserved by the VIM. The problem is becoming more challenging in cases of sport events or natural disasters, where network services should scale up to accommodate the extra traffic. Such information exchange between NFVO, the VIM and the VNFs is crucial to adapt to such challenging network conditions and strict QoS requirements, being able to prioritize tenants or entities managing resources facing performance issues.

- *Reliable operation of NFV environment* [37] – MANO collects reliability parameters and event monitoring or failure events for available VNFs, physical components, or other external functions. After some time, statistical data about network service element failures can be used to handle systematic failures. VIS can handle the collection of such performance and fault information, through interacting with the NFVO, VNFM and VIM, by collecting real-time information from their attached elements. For example, OpenStack instance fault detection and associated event delivery mechanisms can be slow to support a fast failure recovery. Ideally, faults should be analysed and resolved as soon as possible at the functional block that has sufficient and in-time information to perform the root-cause analysis and correlation, and then to determine the necessary corrective action. VIS is responsible for performing this challenging task.

We observe that different operations have alternative requirements for information exchange. For example, the Network Service fault management operation, described here [6], [36], requires real-time and guaranteed delivery of pub-sub type of notifications. However, the VNF Software Image management [6], [36] does have such strict delay requirements, but may be resource-expensive in terms of bandwidth. Such software images may be handled by the VIS (instead of the VIM repositories). The VNF fault management operation [6] assumes the involvement of both NFVO and VNFM, since there is no direct communication between NFVO and VNFs. VIS can be used for the detection of performance issues or faults (e.g. implementing VNF health-checking) and decouple the involvement of the MANO functions at the level of performance issues or fault detection.

The next section has further architectural details of our proposal.

IV. The Virtual Infrastructure Information Service

The Virtual Infrastructure Information Service (VIS) is an information management facility that offers abstracted and logically centralized information manipulation (including information collection, information aggregation / processing, information storage & indexing, and information distribution) across NFV Entities, such as MANO functions and VNFs. The VIS uses two separate interfaces as part of *I-Nfvo* for communication with the NFV Entities and the three core primitives. The interfaces are:

The *Information Management Interface* which is used for information manipulation configuration, including the NFV Entities registration to the VIS, the management of internal VIS functions and the establishment, operation and optimization of information flows and

The *Information Exchange Interface* that offers the actual management information orchestration capability between the VIS and the NFV Entities.

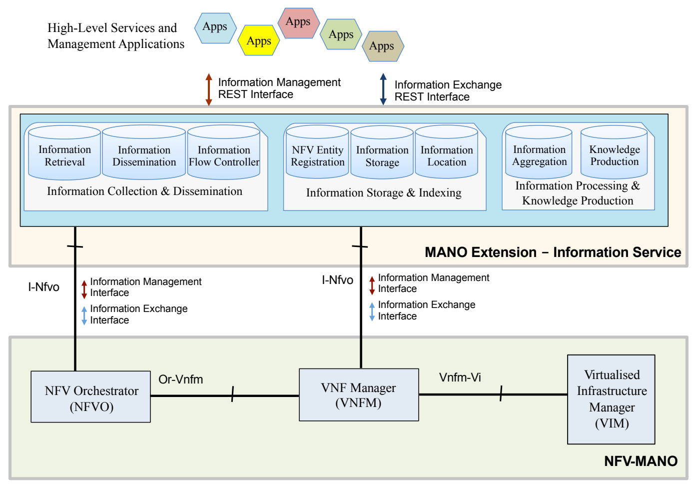


Fig. 2. The VIS Architecture and Basic Interactions

In figure 2, we show the above two interfaces and present the three VIS core functions which are described in detail here:

Information Collection and Dissemination (ICD): The ICD is responsible for organizing communication of management information or VNF state, including optimization of the relevant information flows. It offers facilities for: *Information Collection* – communication of information from the entities to the VIS; *Information Dissemination* – dissemination of information from the VIS to the entities; and an *Information Flow Controller*. The *Information Flow Controller* oversees such functions, including controlling the information flows establishment, operation and relevant optimization aspects. For example, it supports negotiation of information requirements and constraints, matches information sources with sinks, etc. The following communication methods are supported for information: (i) *Pull from Entity* in which VIS pulls requested information from the source on behalf of the sink; (ii) *Pull from Storage* where the sink retrieves information from the VIS storage; (iii) *Publish/Subscribe* method where the VIS keeps the local NFV Entity storages updated with subscribed information; and (iv) *Direct Communication* that implements direct source to sink communication by-passing the VIS. The communication method established by VIS is part of an information flow negotiation decision and can be revoked by VIS when a local or global requirement appears.

Information Storage and Indexing (ISI): The ISI provides storage and indexing functionalities for the VIS. The *NFV Entity Registration* module allows the NFV Entities to express their information manipulation requirements and capabilities. The ISI function maintains an Entity registry, storing specifications for the available information to be collected, retrieved, or disseminated. The *Information Storage* module offers alternative storage options to the information, according to its requirements and characteristics, specified during an entity registration phase beforehand. The *Information Location* module provides information location capabilities to the VIS. These Locaters are pointers to the original data, rather than containing the actual values. Information locaters can be collected as part of an information processing operation or used in the establishment of direct communication flows between NFV Entities. This feature supports the reference elements introduced here [6], which carry references to another information elements and are represented by URIs.

Information Processing and Knowledge Production (IPKP): The IPKP augments VIS with information processing, information aggregation and global picture information production capabilities. The *Information Aggregation* module applies aggregation functions (e.g. MAX, MIN, AVERAGE, ...) to the collected data before they are stored or disseminated. The data may be filtered at the aggregation level for optimization purposes. This component can be flexible enough to be given different aggregation specifications in order to process the data in a varying way. The *Knowledge Production* module generates global picture information through processing / aggregating information. Reasoning and inference mechanisms are best suited for this process, with the requirement that the necessary input information should be immediately available in storage or can be produced in real-time, using an information collection operation – an aspect left for future work.

Overall, the VIS acts as a workflow controller for the information flows that help maintain a global picture of the system, whilst considering the information exchange between NFV Entities and signaling changes in the information flows whenever it is needed. For example, if there is a performance problem or a change in requirements, VIS locates and enforces the most appropriate data paths for the information flows each time. This configuration change of information flows takes place dynamically at any point in time, and is either triggered at a high level (e.g. from NFVO) or at a low level (due to a change in requirements or constraints of an involved NFV Entity). This information flow negotiation facility is related to the information exchange orchestration features (namely control and optimization), and is elaborated below.

A. Controlling and Optimizing Information Flows

To elaborate the realization of the VIS information flow control and optimization aspects we use a representative example. Consider a situation with two NFV Entities: (1) A Virtual Network Management (VNM) NFV Entity that provides management & control facilities for virtual infrastructures, including support of traffic monitoring; and (2) An Entity Placement Optimization (EPO) NFV Entity that optimizes the data flow over a virtual network through adapting the positioning of communicating nodes (e.g. data servers) in response to the dynamic network conditions.

In this example, shown in figure 3, the VNM (on the left) provides traffic monitoring information from a particular virtual network to the EPO (on the right). The EPO takes optimization decisions for the network based on this information,

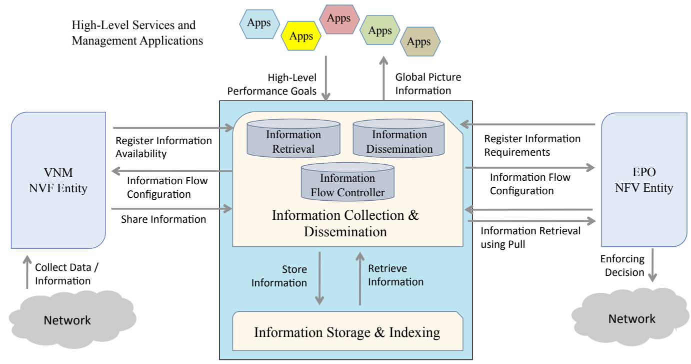


Fig. 3. VIS – Controlling and Optimizing Information Flows

and repositions the communicating nodes in order to optimize the network communication.

The information flow negotiation and optimization processes include three basic phases, elaborated below:

Phase 1 - Entities Registration: In this first phase, the entities, as part of their registration processes, communicate specific information to the VIS using the *Information Management Interface*. This includes: (i) information they can offer instantly or after an information collection process; (ii) information they can offer after a further processing that involves the *IPKP function*; (iii) information they require; (iv) particular constraints in the information source - such as maximum granularity of information collection or minimum network delay; (v) specific requirements for the requested information, such as information accuracy objectives and QoS requirements for the involved information flows; and (vi) supported or requested information model representations to use. Each time a new entity is registered or a configuration update takes place, this triggers one or more information flow negotiation processes (which could be cascading, due to VNF SFC inter-dependencies).

In the example, the VNM registers the information it can offer (including the information type – the topology in this case, and also measurements on the various link loads) and also registers its relevant QoS constraints (for example, it monitors links once per 10 secs). This information can be offered instantly (as it does not require an information collection process to start, since it monitors the network continuously). The EPO registers with the information type, (again topology), as its required information and its QoS requirements (in its case, it requires link load measurements once per 30 secs).

Phase 2 - Information Flow Negotiation: In the second phase VIS, through its *Information Flow Controller* module of the *ICD*, oversees the information flow negotiation processes between the entities providing information and those entities requiring information. An information flow is established between two entities either directly or by involving the VIS, in case the requested information is available in the VIS storage.

This phase is composed of the following steps: (i) selecting a number of potential information flow ends based on the information type, (ii) matching the information sources with information sinks based on the respective information flow requirements and constraints, (iii) determining the information flow configuration with global-level and flow-level optimization considerations. In case of an unsuccessful negotiation (i.e. when the requirements do not match any of the constraints for any combination), the sources and sinks may update their registration information through relaxing their requirements, which then triggers new negotiations.

In the example in figure 3, the VIS matches the VNM with the EPO and decides the information flow parameters, based on the expressed information flow requirements and constraints, the existing network conditions and the potential global performance goals in the system. The information flow decision includes a rule to use the Push/Pull communication method. With this method, the VNM pushes periodically information to the VIS and the EPO pulls the latest information from the VIS less frequently. The VIS stores that information through the ISI function.

Phase 3 - Information Flow Establishment: In this third phase, the VIS establishes the information flow through the *Information Management Interface*. The latter takes as input the information flow configuration decision and enforces it to: (i) the network through the respective entities, and (ii) the VIS functions they are associated with. As the appropriate context environment for the new information flow has been prepared, a suitable path between the participating nodes is then established. This process considers the locations of the entities producing and requiring information and the required VIS nodes (e.g. aggregation points, storage points, etc) as well as the potential traffic characteristics. After that, the *Information Exchange Interface* can be accessed anytime from the information sink entities in order to receive the required information.

In our example, a new information flow configuration is decided on and communicated to the two NFV entities and stored in the VIS. The information flow is established and the EPO can retrieve the required information from the VNM or the VIS using the decided information flow communication method – the Push/Pull method. The EPO NFV Entity can then take network optimization decisions using that information.

There is also a global optimization process in the VIS that is triggered periodically or when a global performance objective change is requested from NFVO or a high-level management application. This process takes optimization decisions using the aggregated information from the configuration and performance of all established information flows and is related with a restructuring of the VIS functions themselves.

The global-optimization algorithms may discard or update information flow configurations already in place for established information flows. This process takes as an input the global picture of all the established information flows, including their performance measurements, and provides as an output different information flow configurations better aligned to the new updated demands for a new global objective. The process may initiate a number of re-negotiations, and we study such a scenario in the experimental results section. As an example, the distributed VIS nodes may be increased, decreased, or repositioned in order to better accommodate all of the established information flows and the global optimization goal. These processes are part of the quality enforcement functionality of the VIS and all the corresponding decisions are being taken within the *Information Flow Controller* module of the ICD function.

In practice, the information flow performance should consider the potential overhead of the negotiations, especially in case of a dynamic environment, or flow inter-dependencies which can result in cascading negotiations. Along these lines, both local and global performance objectives are defined with a priority level (e.g. high, medium or low). This allows VIS to control the responsiveness of information flow configuration to both the dynamically changing network conditions and the requirements at the different network viewpoints. As we show in experimental results (subsection V-B2), VIS allows us to consider the impact of a change in the flow-level configuration to the global performance and vice-versa. The priority level of the global and local performance goals can be defined in ways to satisfy particular demands, e.g. to have fixed information flows in case the impact of negotiation is high.

The VIS handles information flows between NFV entities which have relatively stable requirements, and any extra overhead introduced mainly takes place with application bootstrapping. In our experience, the number of information flows is significantly lower than the number of the co-existing data flows in the network. Clearly, there is a trade off between some overhead (e.g. latency and computation processing) and the flexibility to control the information flows. Another aspect is that this negotiation does not happen with every data flow, but whenever an entity demands change. So, mice data flows can be associated with a fixed information flow configuration and avoid the extra negotiation overhead. We believe this aspect is complicated enough to require its own independent study.

B. The VIS Implementation Details

We now discuss the VIS implementation details, following on from the design specifications presented above. The VIS architecture was carefully designed to support a number of technologies, while providing facilities to select and configure the most appropriate ones each time. We have implemented a number of features that can effectively demonstrate the main VIS capabilities, although a full VIS can support a significantly wider range of technologies. A summary of the associated features and artefacts in the main VIS components is given in Table IV-B.

VIS supports a number of communication methods: Publish / Subscribe, Push / Pull, and Direct Communication (bypassing the VIS). We have implemented two variations of the Push / Pull: (i) the *Pull From Entity* method in which VIS retrieves the requested information from the source on behalf of the sink, and (ii) the *Pull From Storage* method in which the sink retrieves the information directly from the VIS storage.

All the VIS interfaces are REST based and use JSON representations for exchanged information. Each information element is represented by a unique URI, and URI scoping can be used with wildcards. We collect information from the network devices and get performance measurements (i.e. flow and global level) using the Lattice monitoring framework [39]. The VIS supports filtering at both the information collection

VIS Component	Implementation Details and Artefacts
Information Collection and Dissemination	REST based Communication, Entity Registration / Configuration Update, Filtering / Accuracy Objectives [38], JSON Representation of Requirements / Constraints - including a lightweight version, Push/Pull - Pub/Sub - Direct Communication Methods, Integration with Lattice Monitoring System [39], Alternative Placement VIS Nodes and Path Optimization Algorithms [40], Alternative Protocol Stacks for Virtual and Physical Network Space.
Information Storage and Indexing	Redis Key-Value Database [41], Timeindexing Storage [42], URI Representation of Information, URI Scoping Support, Historical Storage Capabilities.
Information Processing and Knowledge Production	Information Aggregation, Aggregation Points' Placement Optimization Algorithms [40], Support for New Aggregation Functions [38], Knowledge Production Triggering, Information Collection for Knowledge Production, Placeholder for Knowledge Production Algorithms.
Information Flow Establishment and Optimization	Information Flows Registry, Information Flows Negotiation Heuristic Supporting Flow Interdependencies and Prioritization Levels in Optimization Processes, Flow-level and Global-level Performance Monitoring, Measurements Visualization, Logically centralized Traffic Engineering for Information Flows.
Information Management and Information Exchange Interfaces	REST based Interfaces, Open APIs for Applications Deployed at both Virtual and Physical Entities, Support for All ICD Features, Lightweight Messaging Option.

TABLE I
SUMMARY OF VIS IMPLEMENTATION DETAILS

and information aggregation levels using appropriate accuracy objectives, which are expressed in the information flow configuration. The VIS supports a number of database technologies for storing data. In our case, we use the *redis* NoSQL database [41] for all information types except those using timestamps, where it is more efficient to use the Timeindexing database [42]. Historical storage capabilities are also supported.

The information flow negotiation facility uses a custom negotiation heuristic and rule parser, having as input the information flow requirements and resource constraints, represented in a JSON format, and producing information flow configurations based on the expressed rules and the specified rule priority levels. The information flow configurations use the same representation, and are communicated from the VIS to the respective NFV Entities using the *Information Management Interface* and are stored in the VIS storage.

The same component considers flow inter-dependencies and may trigger new negotiations when a crucial parameter changes. For example, this can happen when an entity shares information being retrieved from another entity and one of the flows requires changes in its configuration. This aspect is very useful in a Service Function Chaining context, where an adaptation of the service chain can trigger changes in one or more VNFs (e.g. due to updates in the VNF network connectivity topology graph). This is an important aspect that deserves a separate study and is considered as a future work. The output of the negotiation includes determining the most appropriate data paths for the information flows by using the dynamic node selection algorithms presented in [40] and by having the global network view as an input. We use the same algorithms for the optimal placement of all distributed VIS components (e.g. the VIS nodes and the information aggregation points).

V. VIS Platform Experimental Evaluation

This section provides an evaluation and validation of the VIS platform. First, we detail our experimental setup, relevant methodological issues, the performance metrics we used, plus our experimental scenarios. Then we present the experimental results from these scenarios, showing data from runs with 30, 100, and 500 virtual routers.

For our experimental evaluation, we combined and interoperated the VIS with our own experimental Software-Defined Infrastructure platform, called the Very Lightweight Software-Driven Network and Services Platform (VLSP), in order to provide a full working environment. A description of VLSP can be found in [43], where we show the relation of VLSP to other relevant architectural approaches. We used the VLSP as a test facility realizing features from the MANO VNF Manager (VNFM) and the Virtualized Infrastructure Manager (VIM) substrates, e.g. lightweight VNF manipulation, resource allocation and optimization etc. The working proof-of-concept system comprising of the VIS integrated with the VLSP has been deployed on a distributed testbed. Main VIS features have been design and demonstrated in the context of the UniverSELF project [2], [44], [45].

In our experiments, there is a distributed VIS deployment over a distributed virtual infrastructure. The number of VIS nodes increases as the topology size increases, and we place the VIS nodes according to the topology size, using the PressureTime placement algorithm [40]. At this point of implementation, the VIS capabilities are shared between the distributed VIS nodes deployed onto the virtual infrastructure and the one instance of the VIS software at a physical host that is connected to a centralized database.

Each new virtual router is dynamically assigned to the physical machine with the least processing load, by using a configurable *Placement Engine* built in to VLSP. We plan to experiment further with alternative resource allocation and placement algorithms. A survey of this very important subject of Virtual Network Embedding is presented in paper [46].

A. Experimental Details and Methodology

In our experimental runs, we used the following hardware: (i) 2 servers with 2 Intel Quad (4 cores) 2.5GHz CPUs and 8GB of physical memory, (ii) 4 servers with 8 AMD Opteron Quad-core (4 cores) 2.347GHz CPUs and 32GB of physical memory, and (iii) 5 servers with 16 Intel Xeon (4 cores) 2.27GhZ CPUs and 32GB of physical memory.

Each experimental run started with creation of a new virtual network topology being deployed on all 11 physical servers.

The topology consists of the number of Virtual Routers (VR), specified in the each run configuration, and a number of virtual links being created randomly. The link details are picked from a distribution (i.e. a discrete distribution with a minimum of one, to maintain connectivity). The routers to be connected are chosen at random by using the well-known Barabasi-Albert (BA) preferential attachment model [47]. We use this model as it captures some features of the real Internet topology. We ensure that network disconnection events keep the network connected at all times.

To stress test the VIS, we have created our own example NFV Entities with diverse requirements in terms of information handling, including applications collecting information from the virtual routers and applications requesting information from the VIS. All entities support four communication methods (i.e. the *Pull from Entities*, the *Pull from Storage*, the *Publish/Subscribe* and the *Direct Communication*). The information sources and information sinks have been randomly deployed. As a next step, the VLSP assigned all of the entities to the most appropriate VIS node, where the chosen strategy was to choose the VIS node being closest to them. The entities can specify and update their own requirements at any point of the communication (e.g. by changing their requested communication method, their local performance goal, or their minimum / maximum data rates etc). This triggers appropriate information flow negotiations.

The entities periodically transmit performance measurements to the VIS over the negotiated information flows. We performed tests with entities deployed at the virtual routers or as standalone physical applications, resembling both types of NFV entities (i.e. management components and VNFs). Then, after a warm-up period, the communication began.

All of the experiments have a stochastic nature, with random network topologies and random placements of entities. The test runs have been executed several times to ensure replicability of our observations, where ten replications was deemed appropriate for safe analysis as that produced a very low standard deviation of the values. For each run, data is sampled either from all information flows or from a group of them having similar characteristics, in order to gather the following metrics:

- *Average Response Time:* The average time taken from the request of a piece of information from a sink, to the point that it is received. For the case of Publish/Subscribe, the request is resolved locally (i.e. from the local NFV Entity storage keeping up-to-date information).
- *Information Freshness:* The time taken from the production of the new information to the point it reaches the requesting NFV Entity. This is one way to quantify the *quality of information*.
- *Average CPU Load:* The average CPU load value associated with the VIS software. This allows us to monitor the VIS behaviour, in terms of processing requirements.
- *Total Memory Storage Used:* The total memory storage used in the VIS. The data for this metric comes directly from the internal data structures and the chosen database technology (Redis [41] in our case).

The average values of all the above metrics is calculated every 10 seconds in a separate metric collection aggregator.

B. Experimental Results

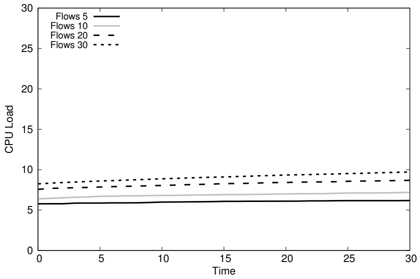
We carried out experiments highlighting aspects such as the adaptability, the flexibility, and the stability / scalability behaviour of the VIS based on the following scenarios:

- *Scenario 1 - Adaptability:* To demonstrate how the VIS adapts to different conditions in terms of NFV Entity requirements and information flows number. Adaptability refers to the ability to change VIS to fit to occurring changes in the information flows.
- *Scenario 2 - Flexibility:* To highlight how the VIS supports concurrent diverse needs, while serving a global performance goal. In other words, showing how the local optimization with the global optimization aspects are being balanced.
- *Scenario 3 - Scalability / Stability:* To show how resource exhaustion can be tackled by enforcing a global performance optimization goal. The limits of the system are explored using an experiment with a large number of virtual routers and many information flows. Scalability refers to the ability of the VIS to handle growing networks elements and usage in a graceful manner and its ability to be enlarged to accommodate that growth. Stability refers to the degree to which VIS must work/operate in a changing environment.

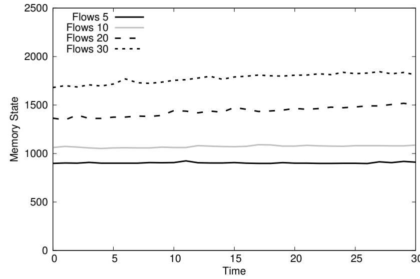
Each of these scenarios are discussed in more detail in the following sections presenting: the *Adaptability of VIS* in section V-B1; the *Flexibility of VIS* in section V-B2; and the *Scalability and Stability of VIS* in section V-B3.

1) Adaptability of VIS: For this first scenario we experimentally explore the adaptability properties of the VIS, given the diverse network environment conditions and the varying NFV Entities' requirements and constraints. We used a topology of 100 virtual routers, while the number of management information flows ranged from 5 to 30. The scenario uses up to 60% of the routers as sources and sinks for management and state information, and a number of routers for the distributed VIS nodes, thus matching a wide range of realistic NFV environment deployments, in terms of flow numbers. We executed the experiments with different communication methods, as outlined in section IV. The main goal is to quantify the impact of the information flows number on the behaviour of VIS and the performance of the respective NFV Entities. The results are shown in figures 4a-4f.

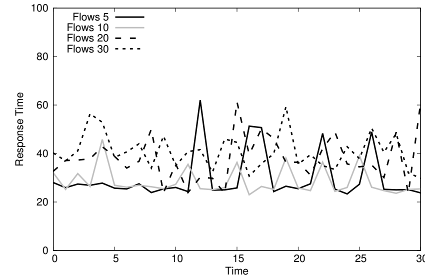
As we can see from figure 4(a), which shows CPU load, and figure 4(b), which shows memory consumption, the VIS accommodates a number of flows well, based on resource availability. We use the *Pull from Entities* method in this example, but a similar behaviour was noticed for other methods as well. There is a minor increase in the processing load and memory consumption of VIS, as the number of information flows increases. However, this increase is stable and predictable. According the figures 4(c) to 4(f), the average response time shows a minor increase as the number of information flows increases. Here, the response time may exhibit a minor jitter, (in the range of milliseconds), that can increase with information flows contention. We have determined that many of these spikes occur due to task and thread switching



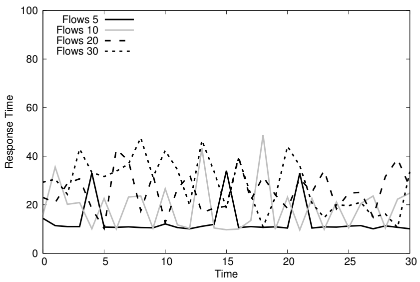
(a) VIS CPU Load (Pull from Entities Method)



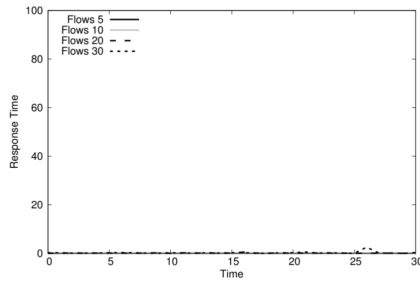
(b) VIS Memory Consumption (Pull from Entities Method)



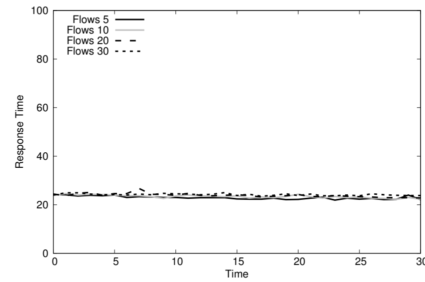
(c) Average Response Time (Pull from Entities Method)



(d) Average Response Time (Pull from Storage Method)

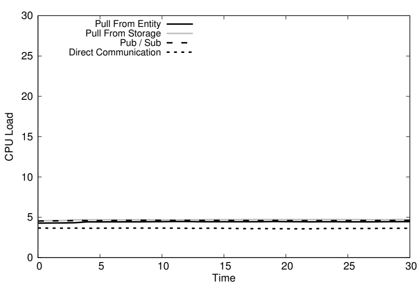


(e) Average Response Time (Publish/Subscribe Method)

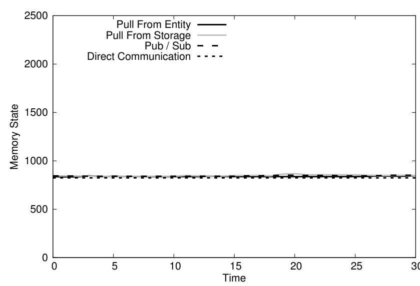


(f) Average Response Time (Direct Communication Method)

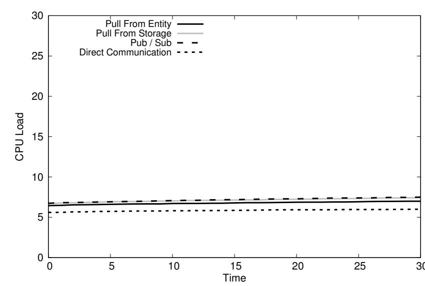
Fig. 4. Impact of Information Flows Number



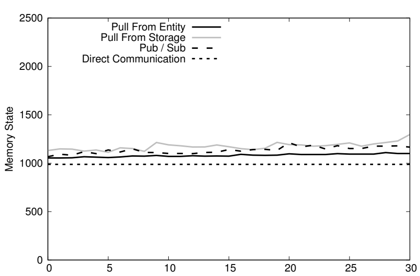
(a) VIS CPU Load (30 VRs and 3 Flows)



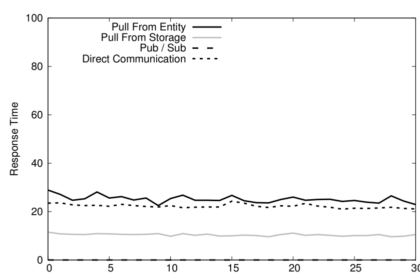
(b) VIS Memory Consumption (30 VRs and 3 Flows)



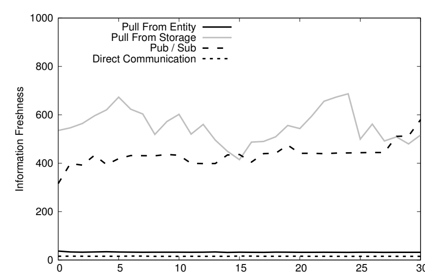
(c) VIS CPU Load (100 VRs and 10 Flows)



(d) VIS Memory Consumption (100 VRs and 10 Flows)



(e) Average Response Time (30 VRs and 3 Flows)

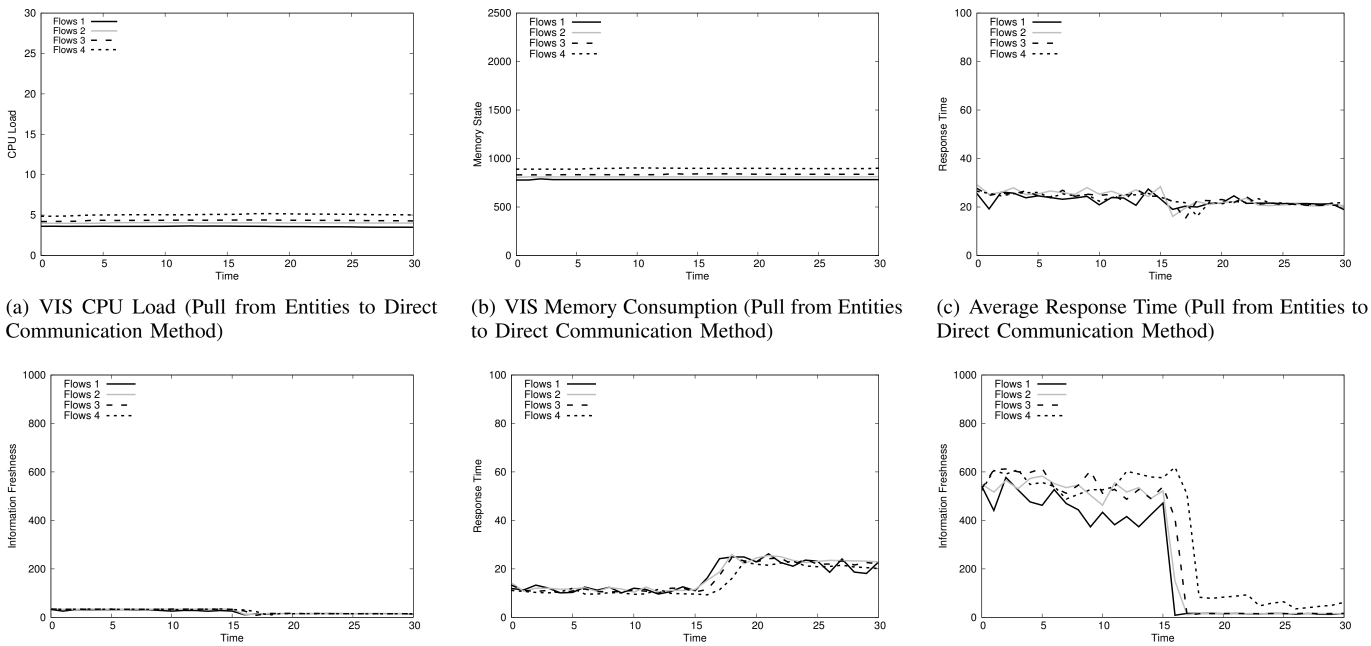


(f) Average Information Freshness (30 VRs and 3 Flows)

Fig. 5. Impact of Communication Method

and other low-level OS processes that may run in the servers, and are not VIS attributes. These spikes will not happen with dedicated hardware hosting the VNFs (i.e. using separate network processors). Furthermore, fully distributed methods (e.g. *Direct Communication*) do not have this issue (figure 4(f)). As the involvement duration of the VIS (including the centralized storage behind it) is gradually reduced, the jitter is reduced as well.

We plan to run more VIS instances on the physical hosts, and also to deploy a distributed database to see how this issue is improved. For example, the *Pull from Entities* method involves the VIS more than the *Pull from Storage* method. The *Direct Communication* method involves the VIS for information flow performance monitoring and negotiation aspects only. This gives significant advantages to the *Direct Communication* method for applications that have real-time constraints. In the



(a) VIS CPU Load (Pull from Entities to Direct Communication Method)

(b) VIS Memory Consumption (Pull from Entities to Direct Communication Method)

(c) Average Response Time (Pull from Entities to Direct Communication Method)

(d) Average Information Freshness (Pull from Entities to Direct Communication Method)

(e) Average Response Time (Pull from Storage to Direct Communication Method)

(f) Average Information Freshness (Pull from Storage to Direct Communication Method)

Fig. 6. Global Tuning of Involved Performance Trade-offs

case of the *Publish/Subscribe* method (figure 4(e)), the average response time is almost zero, because information is retrieved from local storage (however it may not be fresh, as we show in the next scenario).

At this point, we explore how the choice of communication method impacts the performance of the global system and the different NFV Entities. We executed four different sets of runs with a topology of 30 virtual routers and 3 information flows, varying the communication method used. As can be seen from figures 5(a) and 5(b), the impact on the VIS is insignificant in terms of memory consumption but varies in terms of processing load. The *Direct Communication* method produces the least load to the VIS, while the *Pull from Storage* and the *Publish/Subscribe* methods produce the most. The *Pull from Entities* method seems to be closer to the last two methods, in terms of processing load. As we increased the topology size to 100 virtual routers and the information flows size to 10, a difference in terms of memory consumption appears (see figure 5(d)). As was expected, the *Direct Communication* method requires the least consumption and the *Pull from Storage* method the most. However, the relative difference of the different methods in terms of processing load appears the same (figure 5(c)).

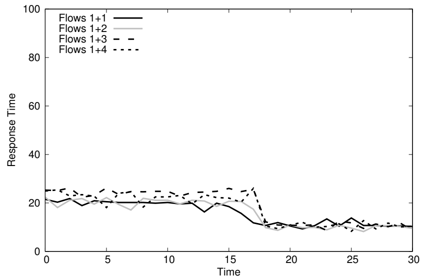
Based on the figures 5(e) and 5(f), we observe the following:

(i) the *Pull from Entities* method has the higher response time but very good information freshness.
(ii) the *Pull from Storage* method is characterized by a very low response time, but may not retrieve fresh information.
(iii) the *Publish/Subscribe* method is characterized by an almost zero response time, but may not be associated with fresh information.
(iv) the *Direct Communication* method has a lower response time compared to the *Pull from Entities* method, but higher compared to the *Pull from Storage* and *Publish/Subscribe* methods. However, it can retrieve the most fresh information.
(v) in the case of lower information flow contention, there is no response time jitter.

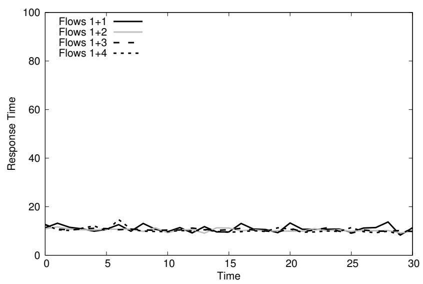
According the above, we see that VIS supports a number of communication options for information handling with alternative behaviour in terms of resource utilization, response time, and quality of information. The VIS can adapt to diverse NFV Entity requirements and global system characteristics.

2) Flexibility of VIS: In this section we demonstrate the flexibility aspects of the VIS. For the same VIS global behaviour, in terms of memory consumption and CPU load (i.e. figures 6(a), 6(b) for *Pull from Entities* method), we can tune the relevant performance trade-offs to meet the needs of the NFV Entities. If we trigger a change in the global performance goal, at one point in time, in order to switch the communication method of flows, this allows a global tuning of the performance of some selected or all established information flows. As an example, switching from the *Pull from Entities* method to the *Direct Communication* method improves average response time and information freshness, while at the same time it minimizes response time jitter (figures 6(c), 6(d)). It also involves tuning performance trade-offs as well. If we switch, at some point in time, from the *Pull from Storage* method to the *Direct Communication* method, we trade average response time for information freshness (figures 6(e), 6(f)). Therefore, we improve the quality of information if we tolerate more delays in order to retrieve the information. Such performance updates can be maintained by an autonomic control loop at the VIS level.

In the previous example, a change is triggered in the global performance goal that impacts all existing flows. The VIS implementation supports changes to the local performance goals at the NFV Entity level as well as global goal changes that impact a subset of the established information flows.



(a) Average Response Time (The Renegotiating Flow)



(b) Average Response Time (The Rest of Flows)

Fig. 7. Local Tuning of Involved Performance Trade-offs

Figure 7(a) shows that a particular NFV Entity may request a renegotiation of its own information flow(s). This may involve a different tuning of the local performance trade-offs (i.e. improving response time in this example) but with a minor or zero impact in the performance of other co-existing information flows (i.e. see figure 7(b)). In this example, we range the total number of flows from two to five. Such behaviour can be associated with autonomic control loops at the NFV Entity level.

3) Scalability and Stability of VIS: In this section, we stress test our infrastructure with large topologies (up to 500 virtual routers). The main goal here is to investigate its behaviour in terms of scalability and stability. As is shown in figures 8(a), 8(b), 8(c), large scales can be reached. Figure 8(a) highlights how VIS CPU load increases with the topology size. Since the number of information flows remains the same (10 in this example), there is no impact on the memory consumption (figure 8(b)). The next figures (i.e. figures 8(d), 8(e), 8(f), 8(f)), show how VIS can trade an increased jitter in response-time for a slight increase in the average response time (figure 8(f)), in the case of a large scale topology and gradual resource exhaustion. In this example, we enforced a global performance goal change that switches the communication method from *Pull from Storage* to *Direct Communication*. This strategy can be associated with a control loop that detects and tackles systematic stability problems. We plan to introduce such a management capability in the near future.

VI. CONCLUSIONS

In this paper, we have argued that abstracted logically centralized information manipulation should be a fundamental feature of NFV MANO, and that it should follow the underlying dynamics of the NFV environments. We have architected and implemented a solution along these lines, the Virtual Infrastructure Information Service (VIS), which exhibits management and state information flow establishment, operation, and optimization between the NFV entities. We have experimentally validated the behaviour of VIS in terms of (i) its *Adaptability*, (ii) its *Flexibility*, and (iii) its *Stability and Scalability*.

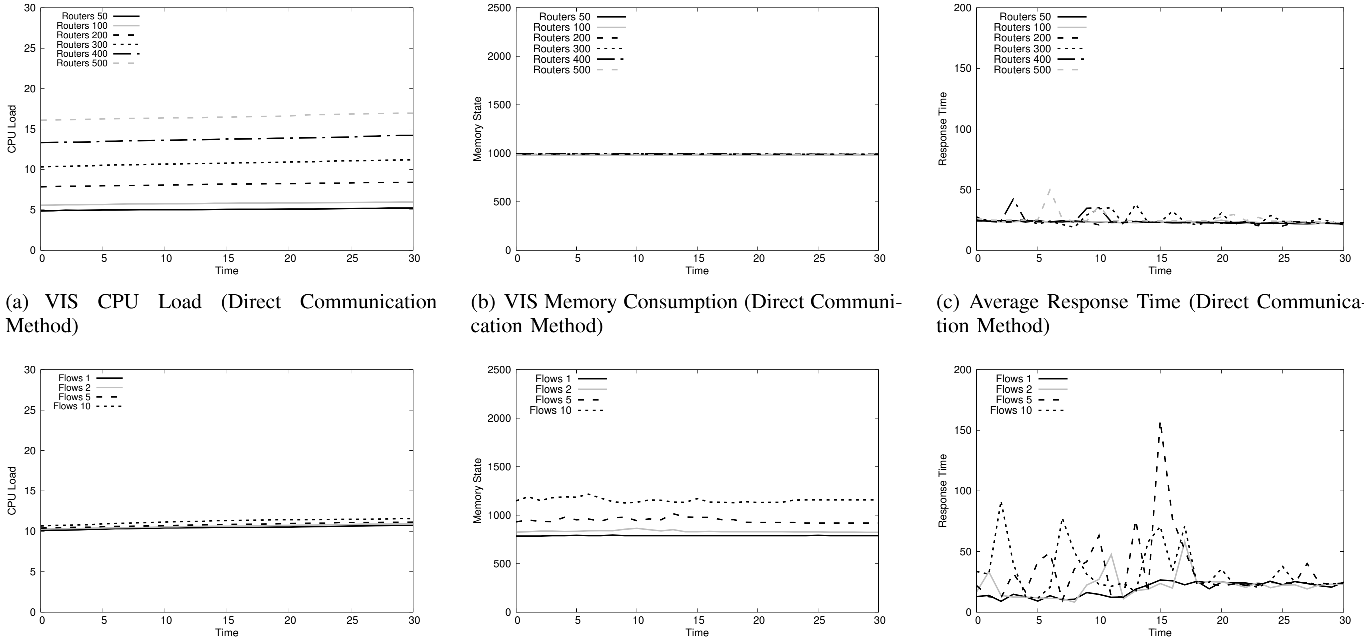
The design of VIS has been presented and the experiments undertaken here have shown that:

(i) A global picture of the information flow manipulation aspects in the system can be maintained. This allows an appropriate tuning of the relevant performance trade-offs, at a local or a global level.
(ii) The local requirements of the NFV Entities can be met, while the global behaviour of the system can be monitored and predicted.
(iii) The global behaviour of the system can adapt, often with a minor impact on the local requirements, to the different NFV Entities.

Consequently its appropriateness to NFV MANO has been validated and confirmed.

To extend our work and to continue our investigations, we plan to implement and research the following VIS aspects:

- The full integration of VIS with the OpenMANO [48], codebase [49].
- Investigate issues related to the co-existence of alternative information models, including the negotiation of the information model to use and the required model translators.
- Research a number of NFV state synchronisation scenarios for stateful network functions, i.e. VNF migration or SDC adaptability.
- Investigate a number of optimization strategies and associate them with different high-level performance goals, including improving energy efficiency in the system. Evaluate complete autonomic control loops, at both local and global levels, for tackling performance and stability problems.
- Determine how VIS behaves in dynamic environments and the involved trade-offs being associated with the information flow negotiation complexity and the delay-sensitive applications.
- Observe the impact of resource allocation algorithms for different types of virtual resources, allowing us to reach even larger scales. Experiment with more VIS instances and alternative allocations of VIS functions between the VIS nodes at the virtual and physical space, i.e. reach even larger scales with less performance spikes (as highlighted in section V-B1 of Experimental Results).
- Consider ideas inspired from the Information-Centric Networks (ICNs) paradigm [50], e.g. data applications can be communicating over negotiated flows, while the



(a) VIS CPU Load (Direct Communication Method)

(b) VIS Memory Consumption (Direct Communication Method)

(c) Average Response Time (Direct Communication Method)

(d) VIS CPU Load (Handling Jitter Issue)

(e) VIS Memory Consumption (Handling Jitter Issue)

(f) Average Response Time (Handling Jitter Issue)

Fig. 8. Impact of Topology Scale

global behaviour of the system will be monitored and controlled in a logically centralized manner.

We will continue working towards further releases of VIS implementations as open-source software, including the documentation of its detailed design and implementation artefacts.

References

[1] DOLFIN, "DOLFIN FP7 project," 2013, http://www.dolfin-fp7.eu.
[2] UniverSELF, "UniverSELF FP7 project," 2010, http://www.univerself-project.eu/.
[3] 5GEX, "EU H2020 - 5G Multi-Domain Exchange (5GEx) project," 2015, https://5g-ppp.eu/5GEx.
[4] SONATA, "EU H2020 - 5G Service Programing and Orchestration for Virtualized Software Networks," 2015, https://5g-ppp.eu/sonata/.
[5] M. Chiosi, D. Clarke, P. Willis, A. Reid *et al.*, "Network Functions Virtualisation," White paper at the SDN and OpenFlow World Congress, ETSI, Tech. Rep., 2012.
[6] ETSI, "GS NFV-MAN 001," 2014, http://www.etsi.org/deliver/etsi_gs/NFV-MAN/001_099/001/01.01.01_60/gs_nfv-man001v010101p.pdf.
[7] R. Mijumbi, J. Serrat, J. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, "Network function virtualization: State-of-the-art and research challenges," *Communications Surveys Tutorials, IEEE*, vol. 18, no. 1, pp. 236–262, 2016.
[8] ETSI, "NFV Information Model WI IFA015 (working document)," 2015, http://docbox.etsi.org/ISG/NFV/Open/Drafts/IFA015_NFV_Information_Model/NFV-IFA015v020.zip.
[9] ——, "GS NFV-IFA 005," 2016, http://www.etsi.org/deliver/etsi_gs/NFV-IFA/001_099/005/02.01.01_60/gs_NFV-IFA005v020101p.pdf.
[10] ——, "GS NFV-IFA 006," 2015, http://www.etsi.org/deliver/etsi_gs/NFV-IFA/001_099/006/02.01.01_60/gs_NFV-IFA006v020101p.pdf.
[11] ——, "GS NFV-REL 002," 2015, http://www.etsi.org/deliver/etsi_gs/NFV-REL/001_099/002/01.01.01_60/gs_nfv-rel002v010101p.pdf.
[12] TMF, "TMF Information Framework (SID)," http://www.tmforum.org/information-framework-sid/.
[13] J. P. Reilly, "Implementing the TM Forum Information Framework (SID): A Practitioner's Guide," 2011.
[14] L. Mamatas, S. Clayman, and A. Galis, "A Flexible Information Service for Management of Virtualized Software-Defined Infrastructures," 2016, Wiley International Journal of Network Management, In Press.
[15] ——, "The Virtual Infrastructure Information Service Open Source Software," University College London, Tech. Rep., 2015. [Online]. Available: http://clayfour.ee.ucl.ac.uk/ikms/index.html
[16] A. Gember, A. Krishnamurthy, S. S. John, R. Grandl, X. Gao, A. Anand, T. Benson, V. Sekar, and A. Akella, "Stratos: A network-aware orchestration layer for virtual middleboxes in clouds," *arXiv preprint arXiv:1305.0209*, 2013.
[17] D. A. Joseph, A. Tavakoli, and I. Stoica, "A policy-aware switching layer for data centers," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 4, pp. 51–62, 2008.
[18] Z. A. Qazi, C.-C. Tu, L. Chiang, R. Miao, V. Sekar, and M. Yu, "Simple-fying middlebox policy enforcement using sdn," in *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4. ACM, 2013, pp. 27–38.
[19] A. Gember-Jacobson, R. Viswanathan, C. Prakash, R. Grandl, J. Khalid, S. Das, and A. Akella, "OpenNF: Enabling innovation in network function control," in *SIGCOMM 2014*. ACM, 2014, pp. 163–174.
[20] B. Kothandaraman, M. Du, and P. Sköldström, "Centrally Controlled Distributed VNF State Management," in *SIGCOMM Workshop on Hot Topics in Middleboxes and Network Function Virtualization*. ACM, 2015, pp. 37–42.
[21] S. Rajagopalan, D. Williams, and H. Jamjoom, "Pico replication: A high availability framework for middleboxes," in *Proceedings of the 4th annual Symposium on Cloud Computing*. ACM, 2013, p. 1.
[22] S. Rajagopalan, D. Williams, H. Jamjoom, and A. Warfield, "Split/merge: System support for elastic execution in virtual middleboxes," in *Presented as part of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13)*. Lombard, IL: USENIX, 2013, pp. 227–240. [Online]. Available: https://www.usenix.org/conference/nsdi13/technical-sessions/presentation/rajagopalan
[23] J. Batalle, J. Ferrer Riera, E. Escalona, and J. A. Garcia-Espin, "On the implementation of nfv over an openflow infrastructure: routing function virtualization," in *Future Networks and Services (SDN4FNS), 2013 IEEE SDN for*. IEEE, 2013, pp. 1–6.
[24] A. Dan, R. Johnson, and A. Arsanjani, "Information as a service: Modeling and realization," in *Systems development in SOA environments, 2007. SDSOA'07: ICSE workshops 2007. International workshop on*. IEEE, 2007, pp. 2–2.
[25] V. Dwivedi and N. Kulkarni, "Information as a service in a data analytics scenario-a case study," in *Web Services, 2008. ICWS'08. IEEE International Conference on*. IEEE, 2008, pp. 615–620.

[26] O. D. C. Alliance, “Master Usage Model: Information as a Service Rev. 1.0,” 2013, http://www.opendatacenteralliance.org/docs/Information_as_a_Service_Master_Usage_Model_Rev1.0.pdf.

[27] J. Medved, R. Varga, A. Tkacik, and K. Gray, “Opendaylight: Towards a model-driven SDN controller architecture,” in *2014 IEEE 15th International Symposium on*. IEEE, 2014, pp. 1–6.

[28] R. Enns, M. Bjorklund, J. Schoenwaelder, and A. Bierman, “Network configuration protocol (netconf),” *Internet Engineering Task Force, RFC*, vol. 6241, 2011.

[29] J. Schönwälder, M. Björklund, and P. Shafer, “Network configuration management using netconf and yang,” *IEEE communications magazine*, vol. 48, no. 9, pp. 166–173, 2010.

[30] ATIS, “ATIS NFV Forum,” 2016, http://www.atis.org/nfv/index.asp.

[31] DMTF, “DMTF Common Information Model (CIM),” 2016, www.dmtf.org.

[32] ITU-T, “ITU-T Information Model,” 2016, http://www.itu.int/rec/T-REC-M.3100/en.

[33] MEF, “MEF Common Information Model,” 2013, https://www.mef.net/Assets/Technical_Specifications/PDF/MEF_7.2.pdf.

[34] IETF, “YANG Data Model for Interface Management,” 2014, https://tools.ietf.org/html/rfc7223.

[35] OASIS, “OASIS TOSCA,” 2015, http://docs.oasis-open.org/tosca/tosca-nfv/v1.0/csd01/tosca-nfv-v1.0-csd01.pdf.

[36] ETSI, “GS NFV-IFA 010,” 2016, http://www.etsi.org/deliver/etsi_gs/NFV-IFA/001_099/010/02.01.01_60/gs_NFV-IFA010v020101p.pdf.

[37] ——, “GS NFV-REL 003,” 2016, http://www.etsi.org/deliver/etsi_gs/NFV-REL/001_099/003/01.01.01_60/gs_NFV-REL003v010101p.pdf.

[38] L. Mamatas, S. Clayman, M. Charalambides, A. Galis, and G. Pavlou, “Towards an Information Management Overlay for Emerging Networks,” in *Network Operations and Management Symposium (NOMS), 2010*. IEEE, 2010, pp. 527–534.

[39] S. Clayman, A. Galis, and L. Mamatas, “Monitoring virtual networks with lattice,” in *Network Operations and Management Symposium Workshops (NOMS Wksps), 2010 IEEE/IFIP*. IEEE, 2010, pp. 239–246.

[40] R. G. Clegg, S. Clayman, G. Pavlou, L. Mamatas, and A. Galis, “On the selection of management/monitoring nodes in highly dynamic networks,” *Computers, IEEE Transactions on*, vol. 62, no. 6, pp. 1207–1220, 2013.

[41] J. Zawodny, “Redis: Lightweight key/value store that goes the extra mile,” *Linux Magazine, August*, vol. 31, 2009.

[42] S. Clayman, “Time indexing - an introduction,” May 2003, http://www.timeindexing.com/documentation/papers.html.

[43] L. Mamatas, S. Clayman, and A. Galis, “A Service-aware Virtualized Software-Defined Infrastructure,” *IEEE Communications Magazine*, vol. 53, no. 4, pp. 166–174, 2015.

[44] N. Koutsouris, K. Tsagkaris, P. Demestichas, Z. Altman, R. Combes, P. Peloso, L. Ciavaglia, L. Mamatas, S. Clayman, and A. Galis, “Conflict free coordination of SON functions in a Unified Management Framework: Demonstration of a proof of concept prototyping platform,” in *2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*, May 2013, pp. 1092–1093.

[45] N. Koutsouris, K. Tsagkaris, P. Demestichas, L. Mamatas, S. Clayman, and A. Galis, “Managing software-driven networks with a unified management framework,” in *2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*, May 2013, pp. 1084–1085.

[46] A. Fischer, J. Botero, M. Beck, H. De Meer, and X. Hesselbach, “Virtual network embedding: A survey,” *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 1888–1906, 2013.

[47] A.-L. Barabási and R. Albert, “Emergence of scaling in random networks,” *science*, vol. 286, no. 5439, pp. 509–512, 1999.

[48] D. Lopez, “Openmano: The dataplane ready open source nfv mano stack,” in *IETF Meeting Proceedings, Dallas, Texas, USA*, 2015.

[49] T. N. Labs, “OpenVIM & OpenMANO,” 2016, https://github.com/nfvlabs/openmano.

[50] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, “A survey of information-centric networking,” *Communications Magazine, IEEE*, vol. 50, no. 7, pp. 26–36, 2012.

Lefteris Mamatas is an Assistant Professor at the Department of Applied Informatics, University of Macedonia, Greece. Before that, he was a researcher at the University College London, Space Internetworking Center / Democritus University of Thrace and DoCoMo Eurolabs in Munich. He received his Ph.D. from the Department of Electrical and Computer Engineering, Democritus University of Thrace in Greece. His research interests lie in the areas of software-defined networks, network management, opportunistic networks and energy efficient communication. He participated in several international research projects, such as Dolfin (FP7), Autonomic Internet (FP7), UniverSELF (FP7) and others. He published around 50 papers in international journals and conferences. He served as a General co-chair for the WWIC 2016 conference, General co-chair for the INFOCOM SWFAN 2016 workshop, TPC co-chair for the WWIC 2012 and E-DTN 2009 conferences and as a guest editor for the Elsevier Ad Hoc Networks Journal.

Stuart Clayman received his PhD in Computer Science from University College London in 1994. He has worked as a Research Lecturer at Kingston University and at UCL. He is currently a Senior Research Fellow at UCL EEE department. He co-authored over 40 conference and journal papers. His research interests and expertise lie in the areas of software engineering and programming paradigms; distributed systems; virtualised compute and network systems, network and systems management; networked media; and knowledge-based systems. He has been involved in several European research projects since 1994. He also has extensive experience in the commercial arena undertaking architecture and development for software engineering, distributed systems and networking systems. He has run his own technology start-up in the area of NoSQL databases, sensors and digital media.

Alex Galis is a Professorial Research Associate In Networked and Service Systems at University College London. He has co-authored 10 research books and more that 200 publications in the Future Internet areas: system management, networks and services, networking clouds, virtualisation and programmability. He acted as a Vice Chair of the ITU-T SG13 Group on Future Networking (www.itu.int/ITU-T/focusgroups/fn/index.html) and TPC co-chair of IEEE Network Softwarization 2015 (NetSoft 2015 http://sites.ieee.org/netsoft/). He is currently the co-chair of the IEEE SDN publication committee (http://sdn.ieee.org).