INTERNATIONAL JOURNAL OF NETWORK MANAGEMENT
Int. J. Network Mgmt 0000; **00**:1–24
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nem

A Flexible Information Service for Management of Virtualized Software-Defined Infrastructures

Lefteris Mamatas[1], Stuart Clayman[2], and Alex Galis[2]

[1]*Department of Applied Informatics, University of Macedonia, Thessaloniki, Greece*
[2]*Department of Electronic and Electrical Engineering, University College London, London, UK*

SUMMARY

There is a major shift in the Internet towards using *programmable* and *virtualized* network devices, offering significant flexibility and adaptability. New networking paradigms such as Software-Defined Networking (SDN) and Network Function Virtualization (NFV) bring networks and IT domains closer together using appropriate architectural abstractions. In this context, new and novel Information Management features need to be introduced. The deployed Management and Control Entities (MCEs) in these environments should have a clear, and often global, view of the network environment and should exchange information in alternative ways (e.g. some may have real-time constraints, while others may be throughput sensitive). Our work addresses these two network management features.
In this paper we define the research challenges in information management for virtualized highly dynamic environments. Along these lines, we introduce and present the design details of the Virtual Infrastructure Information Service (VIS), a new management information handling framework that: (i) provides logically-centralized information flow establishment, optimization, coordination, synchronization and management with respect to the diverse MCE demands, (ii) is designed according the characteristics and requirements of SDN and NFV, and (iii) inter-operates with our own virtualized infrastructure framework. Evaluation results demonstrating the flexible and adaptable behavior of VIS and its main operations are included in the paper.

Received 31/8/2015

KEY WORDS: Information Management, Software-Defined Networking, NFV, Virtual Networks

1. INTRODUCTION

The Internet is gradually evolving by incorporating flexibility and programmability within the network infrastructure (such as Software-Defined Networks [52] and [58]); in data centers (using Cloud technologies, [64]); and in the content distribution (with Information-Centric Networking [2]). Software-Defined Networks introduce a new network paradigm that decouples the control plane from the data plane, whereby the control plane has either centralized and / or distributed programmable components that realize a global view and global management of the network. The most prominent solution for SDN so far, OpenFlow [52], is being deployed on both commercial [35] and experimental / academic networks [37], [52].

The adoption of virtualization technologies is gradually coming to network elements as Virtual Network Functions (VNFs), i.e. the Network Function Virtualization (NFV) concept [6] allows a deeper integration of networks with IT domains and their related operations, by creating network functions within a virtual machine. This approach allows significant cost savings and more flexibility

*Correspondence to: Lefteris Mamatas, Department of Applied Informatics, University of Macedonia, Thessaloniki, Greece. Email: emamatas@uom.gr

in service provisioning. SDN and NFV can be seen as mutually beneficial, as they may co-exist within an infrastructure. This co-existance is not obligatory as network functions can be virtualized and deployed without an SDN and vice-versa [49].

This unique flexibility at both the network operation level and the control level enables management solutions that bring high-level service requirements closer to the network, resulting in a proliferation of new services, while allowing traditional services to operate in more predictable environments. Given such a view for improving service-awareness in the network infrastructure, we have identified *four research challenges* for the future evolution of the SDN / NFV technologies and their unification. These challenges are:

1. to maintain a global picture of the network infrastructure at both the network level and the domain level, using logically-centralized intelligence, programmable techniques, and an abstracted design. This facility should be adaptable to the requirements and resource constraints of the involved network management and control entities.

2. to introduce new abstractions and enablers for realising sophisticated information management features on top of the flexible and programmable network control components. Such technologies bridge the gap between the local viewpoints (e.g. solutions handling network control issues) and the global viewpoints (e.g. higher-level management features).

3. to enable new SDN / NFV applications, beyond centralized traffic engineering, without being constrained by hardware characteristics. Virtualization hides the hardware-level heterogeneity and serves as a migration path towards adopting SDN / NFV-like technologies.

4. to support fast and simple deployment of network resources that collectively support a large number of services, from the level of a single node up to a large-scale network. The network environment should support an efficient, stable, and predictable operation of the services, aligned to the requirements of both the users and the infrastructure / service providers.

This paper presents the Virtual Infrastructure Information Service (VIS) – a new information management framework as a proposal targeting the above challenges. In addition, we argue that *an information management service contributes toward a high-level unification and integration of the SDN, NFV, Network Programmability and Management aspects.*

In this work, we focus on the first research challenge, mainly from the viewpoint of assisting and enabling novel management solutions, but targeting the other three challenges as well. We consider the introduction of an integrated and abstracted state information manipulation facility an important step towards realizing the above research vision. Such a facility will build the global-picture for the system, while supporting local level and domain level views, yet adaptable to the diverse requirements of the involved entities, while at the same time should be architected within the context of the relevant Software-Defined Infrastructures (SDIs).

The traditional approaches to information handling in computer networks, such as the TMF Information Framework related works, [63] [70], have been designed for fixed and static environments. Using off-the-shelf monitoring software is not effective, since these general purpose tools and plugins are not adapted to the dynamic behavior of the Software-Defined Infrastructures. Furthermore, solutions providing *Information as a Service* capabilities have been investigated in the context of clouds and are mainly focusing on data analytics or SOAs and business-aligned services (e.g. [3], [17], [20]).

There is currently no consensus from the different stakeholders on the unification of various proposed information models (e.g. CIM [18], ETSI Information Model [22], ITU-T Information Model [34], IETF YANG [32] or TMF SID [63]). Furthermore, they have not yet fully evolved for such flexible network infrastructures but they are considered as starting points [54]. In our case, VIS is designed as an information-model agnostic service, however we follow these works, in order to accommodate particular information models in the future.

VIS focuses more on the information management aspects, rather than on the information itself. It is an important distinction that VIS is proposed as an *Information Management as a Service* solution, and not an *Information as a Service* solution.

DOI: 10.1002/nem

1.1. Information Management for Virtualized Software-Defined Infrastructures

Here we consider some more detailed characteristics of Information Management for Dynamic Software-Defined Virtual Infrastructures. Both the SDN and NFV paradigms use a number of management & control components, called MCEs (Management and Control Entities), that are used for exploiting, handling, and communicating (i.e. between each other) management information. The MCEs are characterized by diverse requirements in terms of information manipulation. For example, a system network management operation that mitigates failures may require real-time constraints in information consumption, while a daily network routine process may have no such constraints and be delay-tolerant.

Currently, each software entity designer uses their own ways to communicate and process state information. These tailor-made information handling facilities may be efficient for specific software tools, but are associated with the following disadvantages:

- they are not adaptable to the existing network conditions and the QoS requirements of the participating MCEs.
- there is no way to prioritize communication of MCEs essential to overcoming a problem, when faced with systematic problems (e.g. network congestion or failure).
- the collected / processed state information may not be available to other MCEs that can benefit from the information, or the information may not be represented in a compatible format.
- the different information manipulation processes can not be optimized in a collective manner.

We propose that such information communication interchange and handling can be abstracted away within a logically-centralized information management infrastructure, that has been designed considering the unique characteristics of this new wave of challenging SDN / NFV paradigm and technologies. In other words, the information manipulation should have similar characteristics in terms of flexibility, adaptability, scalability, and stability as the network environment it supports.

One may argue that is difficult to have one solution that satisfies the many diverse requirements in information handling that MCEs may have. A characteristic here is to design the appropriate abstractions that allow a number of information handling strategies to co-exist. Such a solution should select the most appropriate strategy each time. The selection should be dynamic as well as adaptable to the network conditions and MCE requirements.

Another characteristic is the minimization of communication overheads, by evaluating the extra overhead that is potentially introduced by any change. For example, the communication cost to select and evaluate the most appropriate information handling strategy and to maintain the global-picture. Our proposal allows simpler ways to communicate information and process information, as the MCEs may choose a direct communication option, overriding the communication through the logically-centralized entities. However, the information handling service should have the option to override the MCEs choices, in case this contradicts with the global view point (e.g. a different global performance goal or systematic failure).

Other characteristics that such a new information management solution should include the support of are:

- service deployment that is adaptable to the resource availability and user / system requirements.
- seamless integration within SDN / NFV and related technologies (such as Software Defined Data Centre & Cloud Computing and Networking).
- a number of closed management control loops, including those for information manipulation.
- tuning of the involved performance trade-offs at global and local levels.
- tackling of scalability and stability issues.
- reduced footprint and lightweight messaging.

These characteristic, together with the four research challenges for the future evolution of the SDN / NFV technologies, have been factored into the design of our Flexible Information Service for Virtualized Software-Defined Infrastructures.

Prepared using nemauth.cls
DOI: 10.1002/nem

1.2. A Flexible Information Service for Virtualized Software-Defined Infrastructures

In this paper, we introduce a new framework that targets the previously defined research challenges and characteristics, manifesting an abstracted and adaptable facility and released as an open-source software (i.e. available at [48]), called the Virtual Infrastructure Information Service (VIS). The VIS provides information manipulation capabilities to the software entities acting as consumers and / or producers of information, namely, the MCEs which include SDN Applications / Controllers [58], and network and service management applications. The MCEs may be embedded at the network devices, virtual routers, or deployed at the physical hosts (as distributed or centralized components). These MCEs may contribute to building the global picture of the environment and / or retrieve either high-level information abstractions or low-level state information. To achieve such functionality, the VIS supports the following facilities:

- information collection, information aggregation, and information dissemination.
- information storage, information indexing, and information processing (such as initial attempts at knowledge production, where we define knowledge as the global-picture information for the network environment).
- information flow (i.e. the management information flow) establishment and optimisation, supporting both global and local tuning of involved performance trade-offs.
- alternative communication methods between information sources and information sinks, such as the Push/Pull, Publish/Subscribe (pub/sub), and Direct Communication methods.
- logically-centralized path optimization for the management of information flows.
- interfaces for both information exchange and management information flows regulation.
- accommodation of extensions, in an architecture aligned to both physical and virtual network environments, that can improve its behaviour further.

The VIS organizes configurable information flows between a large number of entities, acting as information sources and information sinks (i.e. the information flows communicate management information and are not the data flows of the networks data plane). Those entities initially specify their requirements and constraints (e.g. maximum data rate, requested communication method etc) at start up. The VIS matches sources with sinks, finding the most appropriate data paths based on the global-picture of the network, while at the same time specifying the details of the corresponding flows. Furthermore, the information flows are optimized at a global level, but with respect to each source / sink expressed requirements and constraints. At any point, the VIS may trigger re-negotiations and flow re-establishments of some or all of the management information flows, in the case of a different high-level performance goal decision for the system or due to an unexpected event, such as a failure.
To summarize, the Virtual Infrastructure Information Service supports the following novel features:

1. it is specially designed to meet the characteristics of dynamic environments (e.g. SDN / NFV, clouds and / or hybrid deployments such as software defined data centres / cloud computing and networking),

2. it supports information flow establishment, operation, and optimization between the MCEs,

3. it provides logically-centralized management of established information flows with respect to the diverse MCEs demands,

4. it provides explicit support for adaptability, stability, scalability, and flexibility characteristics,

5. it operates over SDN and NFV infrastructures and it augments them at both the level of their northbound interfaces and controller levels with information handling capabilities.

This paper discusses the main research challenges in information handling for integrated SDN/NFV environments. Furthermore, it introduces the first relevant abstracted solution addressing these challenges so far, namely the VIS, along with its architecture, interfaces, and important design

DOI: 10.1002/nem

details of its sub-components and their interactions. It includes detailed descriptions of the VIS software components, the sub-components, the interfaces, and the associated data exchange, interactions and operations between these components. The paper also includes representative VIS proof-of-concept results on its main operations and a functional validation analysis. We have documented the implementation details of VIS and an extensive experimental analysis of its non-functional characteristics in a complementary paper [46]. The two papers complement each other in terms of problem statement plus design details and pragmatic implementation and large-scale experimentation with functional and non-functional features, respectively. Furthermore, paper [46] presents VIS in the context of information handling for and as an extension of the MANO NFV architecture [23].

Our VIS implementation is inter-working, coupled, and tested with the Very Lightweight Software-Driven Network and Service Platform (VLSP) [15] [47], our framework which integrates NFV with SDNs. VLSP consists of two main management layers; these layers, beside the proposed VIS, are: (i) the Virtual Infrastructure Management layer which provides management and orchestration of virtual networks, and (ii) The Lightweight Network Hypervisor which provides SDN-enabled lightweight virtual routers and links that are suitable for large-scale experiments. Paper [47] introduces and motivates the complete three-layer infrastructure and includes a brief discussion of VIS. The detailed design of the VLSP features, besides the VIS, and an evaluation of its main operations is presented in [15].

The paper structure is as follows: Section 2 contrasts the proposed framework with the related works; Section 3 gives the design details of the VIS; Section 4 discusses our experimental methodology and highlights experimental results validating the VIS and its main operations in terms of adaptability and flexibility; and finally, section 5 briefly discusses our next steps and concludes this paper.

2. RELATED WORK

Software-Defined Networks are associated with a major architectural shift in the Internet, such as the decoupling of network control from the data plane. Logically-centralized control is gradually replacing the distributed self-organized way that the Internet behaves, although this is only within the boundaries of a single organization domain (enabling decision-making aligned to its organization structure). They exhibit the following properties [52]:

Logically-centralized intelligence: Network control has been decoupled from forwarding and resides at a logically-centralized component, allowing decision-making based on a global- or domain-level view of the network.

Programmability: Networks are controlled by software functionality enabling application interaction within the network.

Abstraction: Applications consuming SDN services are abstracted from the underlying network technologies.

Very elaborate surveys in the field of Software-Defined Networks have been presented in papers [31] and [43]. Most deployments so far adopt SDN applications realizing centralized traffic engineering and achieve impressive performance boosting. For example, the average utilization of network links in software-driven WANs has been increased from 30-40% to near 100% [35]. From our point of view, the research focus should gradually move from the phase of the Software-Defined Networks to the Software-Driven Networks, where novel network management approaches can bring services and networks closer [25]. This calls for new network architectures, standardized interfaces and management applications focusing on aspects beyond traffic engineering. Network Functions Virtualization (NFV) [6] concepts and systems allows a deeper integration of networks with the IT domains and their related operations. It shares with SDN the virtualisation and programmability aspects and characteristics. The relationships between ONF SDN, ITU-T SDN,

DOI: 10.1002/nem

IRTF SDN, ETSI NFV [67], OpenStack Neutron and OpenDaylight [59] architectures are presented in [47]. The approaches related to the Information-Centric Networks (ICN) [2] focus on the data flows in the network rather than on the management information flows between the MCEs.

It is important that these features should be supported by the deployed management and control components, including SDN applications, SDN controllers and information handling infrastructures. A global picture at a network- or domain-level should be maintained, using logically-centralized intelligence, programmable ways and abstracted design. This means that *network abstraction should be associated with abstracted network information manipulation*.

2.1. Abstracted Information Management for SDN/NFV

The authors of [44] assess how SDN control state inconsistency significantly degrades performance of logically-centralized control applications. They identify two associated performance tradeoffs: (i) between control application performance and state distribution overhead, and (ii) application logic complexity vs. robustness to inconsistency in the underlying distributed SDN state. In their work and many others, information management is tightly integrated within a corresponding SDN application or an SDN controller (e.g. for traffic engineering). For example, in [37], the proposed infrastructure (i.e. Procera) collects information through SNMP and particular kernel data structures exported to the user-level context of the operating system. Those event sources communicate with the controller by periodically sending files along with timestamps.

Google documented an outage incident in its SDN WAN deployment. According to [35], the outage could have been avoided if latency sensitive operations have been received higher priority, compared to the throughput-intensive ones. Furthermore, the problem was detected at a very late stage, due to the lack of enough performance profiling and reporting. Along these lines, we suggest that *SDN applications and controllers should be able to communicate information based on their own diverse requirements and constraints using an abstracted information management facility*.

The following are solutions which are moving towards more abstracted ways to communicate information between the MCEs (e.g. SDN applications and controllers). *Atlas* [29] uses mobile agents that collect netstat logs and pass that information to the control plane. The latter uses an automated data consolidation and classifier generation logic. *PANE* [24] allows users, hosts and applications to express requests for resources, hints about future traffic, and queries for current or future properties of the network to a logically-centralized network controller. It introduces a corresponding end-user API. *ALTO* [28] hosts aggregated information to which each controller has a link. Its main goal is to guide applications to select one out of several hosts capable of providing the desired resource. *HyperFlow* [71] selectively publishes events on system state changes, other controllers replay all the published events to reconstruct the state. The *XSP* [38] protocol provides general and extensive support for interactions between applications and network-based services, and between the devices that provide those services. XSP includes a "call setup" phase, indicating connectivity requirements. *Onix* [39] uses a DHT-based solution, supporting group membership and distribution mechanisms. It uses a data model that represents the network infrastructure, namely the Network Information Base (NIB).

There are recent NFV state handling proposals focusing on the VNFs state mainly and its related novel problems (e.g. VNF migration or elasticity). Some proposals, like the [26], [36], [60], manipulate the state separately and others offer coordinated state management, such as [27], [40], [61] and [62]. Existing open-source flexible infrastructures, e.g. OpenDaylight [53], use netconf [21] (or its RESTful variation restconf) for the communication of configuration and operational data, RPCs and notifications. Netconf uses the YANG information model [66].

In the following section, we discuss solutions that implement information management and monitoring over virtualized environments.

2.2. Information Management and Monitoring for Virtualized Infrastructures

Information management is an enabling technology for management functions achieving autonomicity and sophisticated resource optimization within virtualized infrastructures (such as clouds). Many different aspects relevant to information handling should be studied in parallel:

DOI: 10.1002/nem

information flow optimization, information representation, event-based and publish-subscribe mechanisms, and monitoring functionalities. Information management is considered as a crucial component of autonomic management architectural models in virtual environments, as proposed in a number of research projects and studies e.g. in AutoI [65], RESERVOIR [64] and 4WARD [56]). Clark [8] introduces a knowledge plane architecture for the Internet, while in [45], an Information Management Overlay (IMO) is proposed, which is a knowledge plane providing focused-functionality management applications with the necessary information for realizing self-management capabilities.

Monitoring is associated with information management, and is a major issue in a network virtualization [7]. Existing monitoring systems such as Ganglia [50], Zabbix [57], Nagios [33], MonaLisa [55] and GridICE [4] have addressed monitoring in large distributed systems. They are designed for the fixed, and relatively slowly changing physical infrastructure that includes servers, services on those servers, routers and switches. However, they have not addressed or assumed rapidly changing environments that VIS considers (i.e. in terms of requirements and resource constraints). In this direction, *VIS is integrated with the Lattice monitoring system*, which already been tested in diverse dynamic environments (such as clouds [10] and other virtualized infrastructures [11]).

2.3. Our Approach Contrasted to the Related Works

All of the above solutions seem to work well within their targeted contexts. The proposed VIS goes one step further: *It abstracts information manipulation further away within such environments*. This requires not only supporting alternative methods to create the network-wide state, but also a flexible way to choose the most appropriate configuration each time. It allows each MCE to specify its own requirements and constraints. The VIS infrastructure is flexible enough to accommodate diverse needs, since there is no single method that works everywhere. However, VIS maintains a global view of the management information flows and may revoke configuration settings, when the global performance is impacted. Additionally, VIS supports most of the features of the above proposals, such as alternative communication methods / data storage, an information flow setting up / negotiation phase, indexing of information, dynamic monitoring, open end-user APIs etc. A more detailed description of the proposed Virtual Infrastructure Information Service follows in the next section.

Our proposal augments the SDN / NFV architecture with a decoupled information manipulation component, the Virtual Infrastructure Information Service. It is not competing with OpenFlow oriented research, rather, it extends it at both the level of the SDN / NFV Northbound Interfaces and the SDN controller [58]. The proposed framework is not constrained by well-established SDN technology (such as OpenFlow), since it targets and operates as an augmented higher information handling framework over wider SDN / NFV infrastructures and applications base.

3. THE VIS DESIGN DETAILS

This section presents the design details of the proposed infrastructure. There is a particular focus on the features enabling logically-centralized, programmable, and abstracted information manipulation. The VIS architecture, its sub-components and the associated interfaces are described and explained in detail in the following subsections.

3.1. VIS Architecture, Interfaces, and Functions

The VIS architecture is described here, firstly discussing the type of entities that use the VIS, and then highlighting the interfaces and the main core functions.

As we show in figure 1, three types of MCEs communicate with the VIS: (1) *The High-Level Management Applications*, which are responsible for the efficient operation of the whole system and take optimization decisions based on the global picture, such as the SDN Applications, (2) *The*

DOI: 10.1002/nem

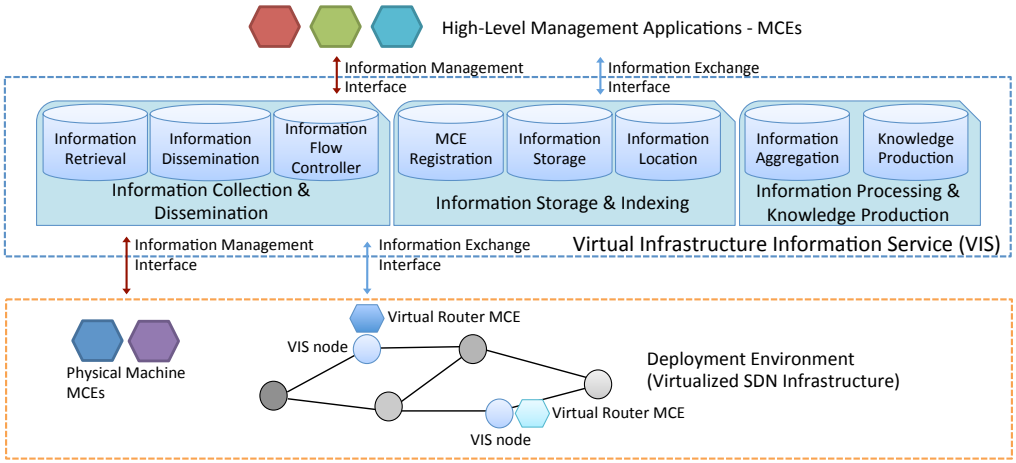


Figure 1. The Virtual Infrastructure Information Service Architecture coupled with a virtualized infrastructure

MCEs deployed at the physical hosts, such as the SDN Controllers, which usually have a domain-level or server-level view, and (3) *The MCEs deployed at virtual routers*, which are responsible for resource-facing operations at the virtualization hypervisor level. The third MCE category covers the need to deploy a number of entities close to the resources. For example, in [69], the authors place an embedded module on OpenFlow switches close to the failure's source in order to reduce computational and network resource consumption.

The VIS uses the following two separate interfaces for communication with the MCEs: (1) the *Information Management Interface* is used for information manipulation configuration, including the MCEs registration to the VIS (i.e. expressing requirements, capabilities and constraints in terms of information handling). This interface is associated with both internal VIS management activities and external to the VIS information flow configurations and acts as the interface to the proposed *logically-centralized information management facility*; and (2) the *Information Exchange Interface* offers the actual information exchange capability to the deployed MCEs. These two interfaces allow a *unified yet abstracted information handling* in all types of participating entities.

The VIS uses four main functions (see figure 1):

- **Information Collection and Dissemination:** Responsible for organizing communication of information.
- **Information Storage and Indexing:** Provides storage and indexing capabilities to the VIS.
- **Information Processing and Knowledge Production:** Augments VIS with information processing, aggregation and knowledge production capabilities.
- **Information Flow Establishment and Optimization:** Handles information flow manipulation (including information flow characteristics negotiation between potential sources and sinks) and VIS optimization aspects.

Detailed descriptions on each of the main VIS functions are included in the following subsections.

3.1.1. ***The Information Collection and Dissemination function (ICD)*** The *Information Collection and Dissemination function* is responsible for information collection, sharing, retrieval, and dissemination. The *ICD* is the communication front-end of the VIS, residing behind the *Information Exchange Interface* and supporting alternative ways to communicate information, which can be configured in order to meet certain performance profiles. The basic information communication mode between MCEs and VIS imply a proxy-like interaction, where all communication is being handled via the VIS. However, VIS also supports a communication method where it can

DOI: 10.1002/nem

redirect information querying MCEs towards the appropriate resources (i.e. other MCEs) for direct interaction. The interested MCEs may either request such a communication method from the VIS or the *IFEO function* may enforce it transparently for optimization purposes.

New *ICD* algorithms (e.g. for collection or dissemination) can be added in a plug-and-play fashion. This process gives flexibility to the VIS infrastructure to meet new information manipulation demands, as soon as they arrive. The main *ICD* operations are elaborated below.

Information Retrieval The VIS retrieves information from a number of MCEs according to their information collection constraints, while at the same time it regulates communication to meet certain information collection and information sharing requirements.

Information *collection* is triggered from the VIS, whereby the VIS gathers the relevant information from an information source. Such a collection could take place in response to an information retrieval request from an MCE (often in the case where the requested information is not available in the VIS storage).

Information *sharing* is triggered from the MCE information sources. In this situation, the VIS gets given the relevant information, as such an information source shares the information with the VIS. An information source may share new information with the VIS, or can update existing information.

Information *retrieval* can be one of four types:

(i) 1-time queries, which collect information that can be considered static, e.g. the number of CPUs in a server,

(ii) N-time queries, which collect information periodically for a certain number of times,

(iii) continuous queries that collect information in an on-going manner, and

(iv) unsolicited acquisition of subscribed information units.

and for each of these four types there can be information collection or information sharing.

The MCEs can provide information for collection or sharing with the VIS, set their information collection constraints during their registration phase, or update the constraints during an MCE configuration update phase (e.g. to respond to a network event like congestion).

Information *collection* is where the VIS gathers the relevant information from an information source, and is triggered from the VIS. Such a collection could take place in response to an information retrieval request from an MCE (often used when the requested information is not available in the VIS storage).

Information *sharing* is where the information source shares the information with the VIS and the VIS gets given the relevant information. It is triggered by the MCE information sources, and they may share new information with the VIS, or can update existing information.

Information Dissemination The ICD function performs information *dissemination* to a number of MCEs that act upon this information, e.g. performing configuration changes. The MCEs can request information from the VIS, which may be available in the VIS storage or can be indexed. The information sinks register their information retrieval requirements during their VIS registration or configuration update. This process enables the corresponding MCEs to act as information sinks. The information can be disseminated using one of the following methods:

i) *Pull:* The MCEs may request information using the Pull method by explicitly requesting a particular type of information, and getting the data as the response. They can either make these requests on a periodic basis (polling) or when a certain demand arises. Two main types of pull method are currently supported: (1) the *Pull from Entity* method, where the VIS pulls that information from the source on behalf of the sink, and (2) the *Pull from Storage* method, where the VIS pulls that information from the VIS storage. The VIS can be configured to dynamically switch between the two methods (i.e. to retrieve the information from the source when not in storage or vice-versa);

DOI: 10.1002/nem

ii) *Publish/Subscribe (pub/sub):* The MCEs can subscribe to receive a certain type of information, i.e. they are automatically informed when this information appears or changes. Filtering and information accuracy objectives may be applied (e.g. to follow changes that are higher than a particular threshold). The information sinks define their information requirements during their MCE registration or a configuration update. The MCEs maintain the information in their local storage, from which they service either further information processing or act upon the new information;

iii) *Direct Communication:* The MCEs can communicate directly, when necessary, rather than sending data via the VIS. The MCEs are signaled by the VIS to communicate directly, when the information flow establishment phase decides to use that particular method. Although, MCEs can have direct information exchange, this process may be revoked by the VIS as it is still in overall control of all the management information flows. This may occur in the case of changes in the network or to the requirements.

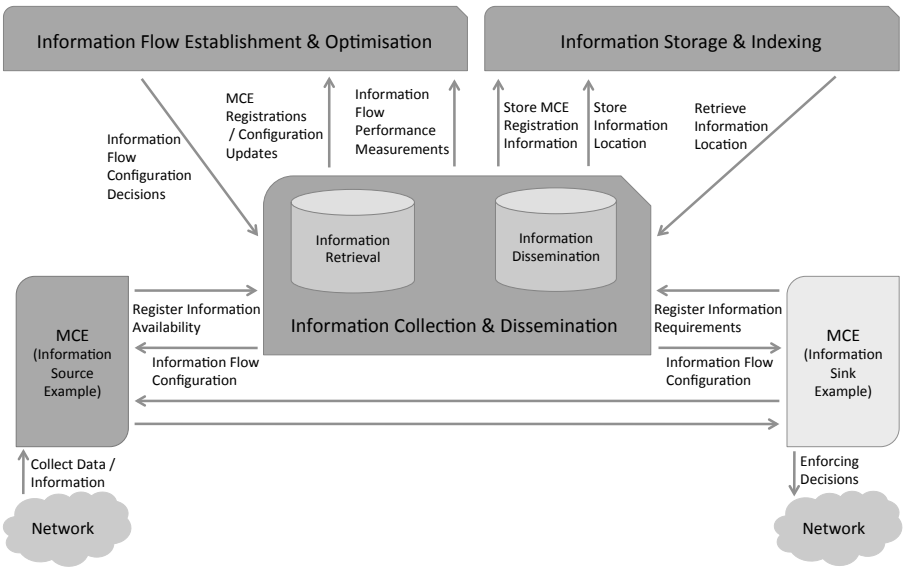


Figure 2. The Information Collection and Dissemination Interactions

An example of ICD usage is shown in figure 2. Following the registration of an information source MCE (left) and an information sink MCE (right), the VIS communicates management information flow configurations to both entities signaling direct communication, containing the location of the information source as well. Such location information, is part of the information availability registration of the source entity, and is retrieved through the ISI function. The *IFEO function* oversees this information establishment process, considering active information flow communication measurements, the global performance goal in the system, and the most recent registration information of the available entities.

3.1.2. ***The Information Storage and Indexing function (ISI)*** The *Information Storage and Indexing (ISI) function* is a logical element representing a repository for registering MCEs, storing and indexing information. The *ISI function* stores information at various levels of abstraction, ranging from MCE registration information and raw network state data to processed and global picture information. The *ISI* functionality includes methods and functions for keeping track of MCEs, including information registration and naming, MCEs configuration, information directory and indexing.

An important storage aspect, which can assist the production of higher information abstractions handled by the *Information Processing and Knowledge Production* function, is the inherent support of historical capabilities. For example, an MCE could request information that was stored in the past using an appropriate time-stamp.

DOI: 10.1002/nem

It should be noted that the knowledge production functionality is not part of the *ISI function*, rather the *ISI* offers storage facilities for knowledge derived due to some earlier calculations. The *ISI* optionally stores knowledge produced from the *Information Processing and Knowledge Production function*, in the form of global picture information regarding the network (e.g. as an input to sophisticated inference mechanisms).

The different MCEs, which request or store information to the VIS, do not directly communicate with the *ISI*. The *ICD function* handles information collection or dissemination between the storage points and the MCEs.

The VIS parameterizes the *ISI function*, based on the current conditions or global performance goals. The *ISI* may be characterized by alternative structures or configurations (e.g. number of storage nodes, in case of a distributed storage) that are suitable to a particular environment or network condition (e.g. the presence of congestion in the network). The main *ISI* operations are elaborated below:

MCE Registration All MCEs must be registered to the VIS. This process allows the MCEs to express their information manipulation requirements and their capabilities. The *ISI function* maintains an MCE registry, including specifications for the available information to be collected, retrieved, or disseminated. If the MCE is already registered, the particular MCE configuration is updated. This process may result into re-negotiations of existing management information flows, in order to meet the updated requirements and constraints. In this paper, a similar case is studied experimentally in the section 4.

Information Storage/Location The information communicated through the *ICD function* is optionally stored in the Information Storage. Once stored, it can be passed back to the *ICD function* for dissemination. Alternatively, information need not be stored immediately but, could be stored after the completion of an information aggregation or knowledge (i.e. we consider as knowledge the global-picture information) production process. The information communicated through the *ICD function* is optionally processed to create *Information Locaters*. These Locaters are pointers to the original data rather than containing the actual data. Information locaters can be collected as part of an information aggregation or knowledge production operation. Furthermore, they may be used in the establishment of a direct communication flow between MCEs.

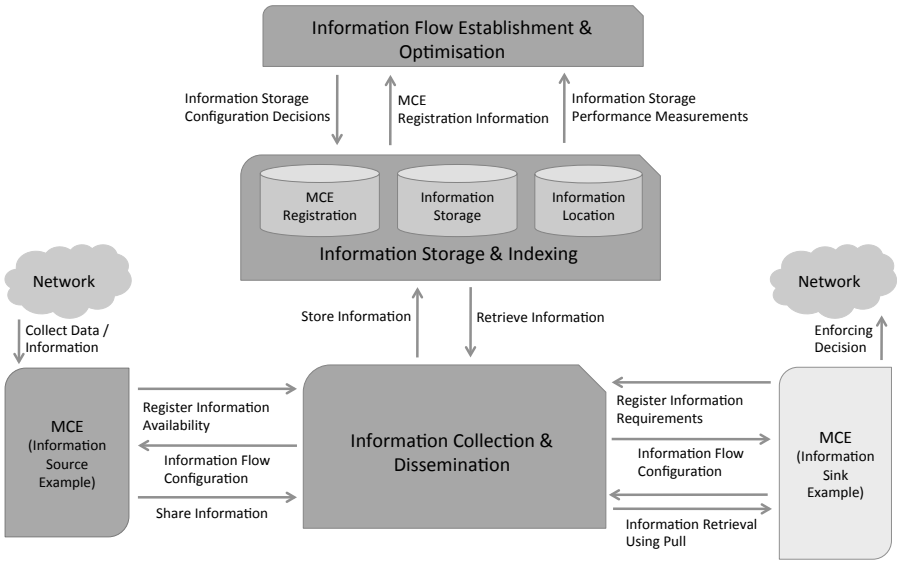


Figure 3. The Information Storage and Indexing Interactions

DOI: 10.1002/nem

In figure 3, we show example interactions of the *ISI function*. We show an MCE collecting information from the network and passing it through the *ICD function* to the VIS storage, handled by the *ISI function*. At some later stage, another MCE retrieves that information through the ICD function, which in turn retrieves it from the *ISI function* storage. This example is using the *Pull from Storage* method. From the figure we see the *IFEO function* retrieves periodic storage-related measurements from the *ISI* (i.e. system-oriented input) and the updated MCE registration information (i.e. local-oriented input) and can trigger configuration changes in the *ISI*, such as additional or relocated distributed storage nodes, or the choice to use an alternative storage technology, etc).

3.1.3. ***The Information Processing and Knowledge Production function (IPKP)*** The *Information Processing and Knowledge Production function (IPKP)* is responsible for operations related to information processing (e.g. aggregation) and knowledge production. In network environments, there is a frequent need to aggregate information in order to take decisions concerning a wider network scope (e.g. those based on the average link load in a network domain). In VIS, this is overseen from the *Information Aggregation* (IA) component of the *IPKP function*. The IA receives the collected data from the *ICD function* and processes them out before they are stored through the *ISI function* or disseminated. The data may be filtered at the IA level, i.e. this reduces the volume of measurements by only sending values that are significantly different from previous measurements. Furthermore, the IA component itself can be flexible enough to be given different aggregation specifications by a higher level management application (e.g. a governance application) in order to process the data in a varying way. For example, it can be configured to wake up once an hour and select data for the last day, and then apply an aggregation function. This is achieved using a mechanism that relies on plugins.

As well as requesting information, an MCE may subscribe to the event-based notification service (i.e. the publish/subscribe mechanism) by setting an appropriate threshold to a specific type of information. Whenever this threshold is exceeded, the subscribed MCE is notified. Aggregation is done during the information collection phase, in order to minimize overhead. More details on the information aggregation facility of the VIS and relevant optimization aspects can be found in papers [9], [16].

Accordingly, the *Knowledge Production* (KP) component handles and produces global picture information. This type of information is produced from aggregated information and / or by combining processed information coming from several parts of the network. In both cases, reasoning and inference mechanisms and associated software components are required. These components may use a number of algorithms depending on the exact problem that is addressed, the type of inputs that are used, and the type of output that needs to be acquired. Such techniques come from scientific areas like statistics, clustering, reasoning, Fuzzy or machine learning (including supervised, unsupervised and reinforcement learning techniques). The produced knowledge from the *IPKP function* can be optionally stored through the *ISI function* so as to be available for other MCEs when it is needed.

In figure 4, we show example interactions involving the *IPKP function*. An Information Aggregation or Knowledge Production process is triggered from the *ICD function* if this is required by a newly arrived MCE registration. The Information Aggregation and Knowledge Production operations are optimized: (i) from a high-level viewpoint, through updates to the information aggregation and knowledge production algorithms, guided by a corresponding high-level application and based on global performance/accuracy measurements as an input; and (ii) from a lower-level viewpoint, guided by the *IFEO function* and having as an input the MCE registration information, the global performance goals in the system and the current status of the topology.

In practice, the high-level view is responsible for the general behaviour of the information aggregation and knowledge production processes and the *IFEO function* for the efficient adaptation to local demands. Such optimizations include efficient placement of aggregation points, information filtering based on accuracy objectives, and other information flow configuration aspects. Details of similar experimental evaluation can be found in papers [9] and [16].

DOI: 10.1002/nem

The main *IPKP* operations are being summarized below.

Information Aggregation This applies aggregation functions to the collected data / information. The aggregation process increases the level of information abstraction, thereby transforming the data into a structured form, but at the same time reducing the load on the network. Aggregation works in situations where MCEs do not need a continuous stream of data from the VIS, but can get by with an approximation of the data values. For example, getting an occasional measurement with the average volume of traffic on a network link, may be enough for some MCEs. Some common aggregation functions include SUM, AVERAGE, STDDEV, MIN, and MAX. Although these are most common, arbitrary functions can be passed in, which gives considerable flexibility when determining aggregations. The *Information Aggregation* operation uses information filtering based on accuracy objectives [9] and [16]

Knowledge Production This produces higher-level global picture information through processing and/or aggregating information. In both cases, reasoning and inference mechanisms and associated software components are required. Such techniques include statistical methods, clustering, reasoning, fuzzy or machine learning (e.g. supervised, unsupervised and reinforcement learning techniques). The necessary input information can be available in *storage* or can be produced in real-time, using an information collection operation.

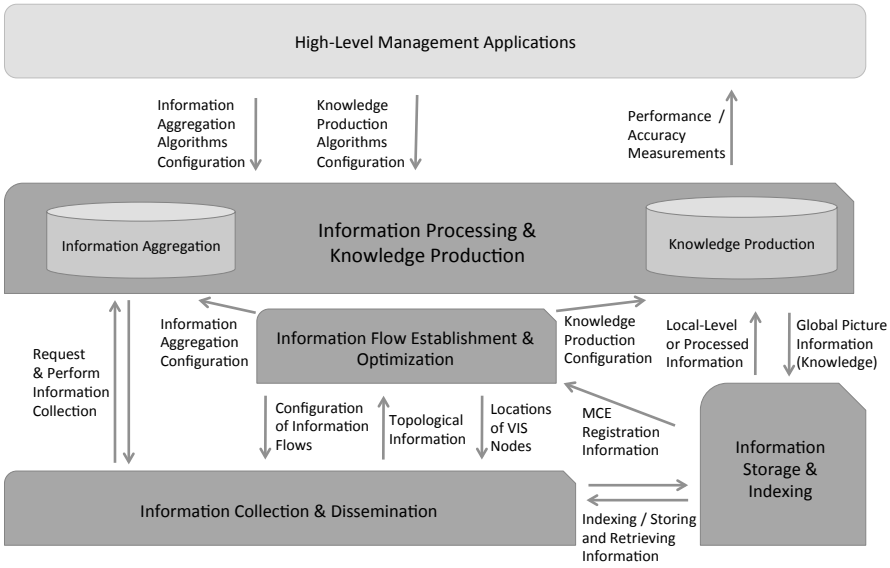


Figure 4. The Information Processing and Knowledge Production Interactions

We have investigated the issue of knowledge production in the context of the UniverSELF project [72], using a number of alternative knowledge production problems and solutions as an input and targeting a generalized knowledge production facility for VIS. We demonstrated two solutions involving such generalized knowledge production capabilities: [41] and [42]. This is a very complicated aspect that deserves an independent study, which we consider as a future work. As an example, all of the required information that enables the VIS knowledge production capabilities should be described in a relevant ontology, ready to be looked up from the IPKP function when such a demand appears (including the problems addressed, the types of inputs / outputs, the inference / reasoning mechanisms used, etc). A first definition of a relevant ontology have been released in [72].

In figure 5, we show an overview of the designed knowledge production operation. The basic interactions involving the main IPKP components are described here. Consider that an MCE that

DOI: 10.1002/nem

requires the VIS IPKP functionalities requests to utilize either an Information Aggregation (IA) or a Knowledge Production (KP) operation. The ICD function handles the communication of the MCE with the internal IPKP functionalities and the IPKP controller is responsible to control the internal IPKP components. The two IPKP operations (namely information aggregation & knowledge production) require a number of fundamental steps, which are outlined:

- *Step 1:* Determining the information aggregation or knowledge production parameters (e.g. information filtering configuration, the inference / reasoning algorithm to use, translation requirements, whether aggregation is required and / or information / knowledge post-processing requirements). This process is handled from the IPKP controller, which matches the MCE's requirements and the type of problem to solve with the relevant information. The parameters are being communicated to all relevant internal IPKP components.
- *Step 2:* Collection of input information either from an MCE that produces it or from the ISI function (i.e. the VIS storage). A collection request is being passed back from the IPKP controller to the ICD function.
- *Step 3:* Pre-processing of the input information (e.g. applying information filtering) that may be required. The pre-processing requirements are being set from the IPKP controller. More details can be found in [9] & [16].
- *Step 4:* The input information is being passed to the IA operation in case of information aggregation, where an aggregation process takes place according the requirements (e.g. aggregation function used) being set from the IPKP controller. In case of knowledge production, this step may be bypassed or not (i.e. the knowledge production processes may require aggregation before the inference / reasoning process).
- *Step 5:* In case of knowledge production, the input information may need to be translated into a convenient representation, such as OWL [51]. The translation configuration is being set from the IPKP controller to match the requirements of the inference / reasoning mechanism identified from a relevant ontology, such the one described in [72].
- *Step 6:* The actual inference / reasoning process takes place in this step. The input information (i.e. in an appropriate form) and the relevant knowledge production rules are being passed to the identified inference / reasoning mechanism. A rule description language that can be used is the Semantic Web Rule Language (SWRL) [30]. The output of this process is the produced Knowledge. This step may be by-passed, in case of a request for information aggregation without knowledge production.
- *Step 7:* The produced knowledge or aggregated information may need a post-processing (e.g. filtering). This step is optional.
- *Step 8:* At this stage, the result is being communicated to the ICD function, to find its way to the requesting MCE. The produced knowledge or aggregated information can be optionally stored in the ISI function so as to be available for MCEs when requested / needed.

A relevant knowledge production workflow is described in [72].

3.1.4. ***The Information Flow Establishment and Optimisation function (IFEO)*** The *Information Flow Establishment and Optimization (IFEO) function* regulates the management information flows based on the current state and the locations of the participating components (e.g. the MCEs producing or requiring information). In particular, it controls how the information collection and dissemination are handled from the *ICD function* (e.g. communication method and data path used), the information aggregation in the IA operation (including optimal aggregation point placement), and storage optimization configuration parameters in the *ISI function*. Furthermore, it guides a filtering system for information collection and aggregation points that can significantly reduce the communication overhead. Both Information dissemination and collection processes should meet certain information collection / dissemination constraints, being communicated to the VIS during the MCE registration processes. For example, a number of MCEs may trade information accuracy for communication cost. Such accuracy objectives should also meet global performance

DOI: 10.1002/nem

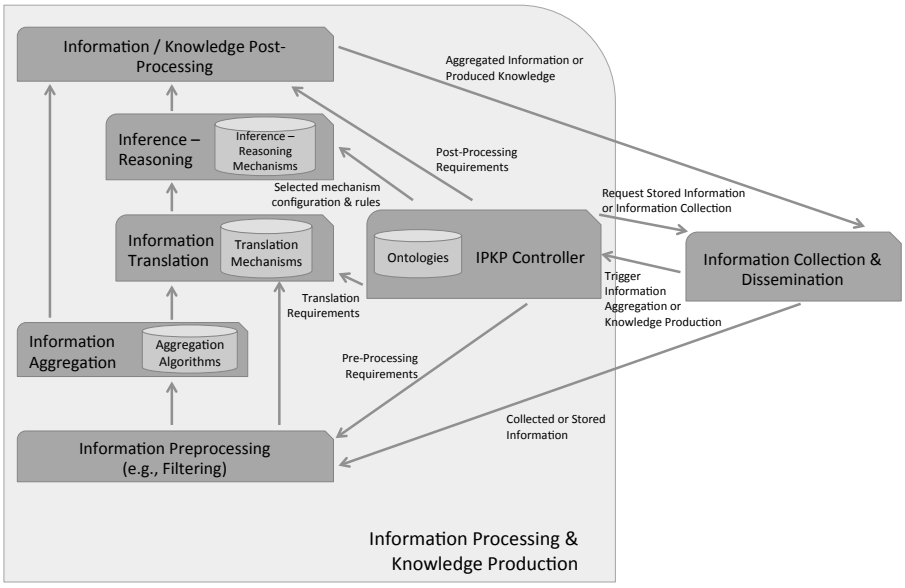


Figure 5. Overview of the Knowledge Production Operation

requirements (i.e. harmonizing management information flows to the global performance goals). The *Information Flow Establishment and Optimization function* is responsible for such quality enforcing functionalities.

In the *IFEO function*, a number of negotiations take place that match interests with constraints. The outcome of each negotiation is the management information flow parameters, balancing the capabilities of the information sources, with the requirements of the information sinks and the potential global performance goals in the system. The *IFEO function* enforces optimization decisions by communicating with the corresponding MCEs and the VIS functions, in order to satisfy the performance optimization requirements. The above processes are part of the quality enforcement functionality of the VIS and all corresponding decisions are being taken from the Information Quality Controller (IQC) operation of the *IFEO function*.

The basic *IFEO* operations are detailed below.

Information Flow Establishment & Optimization This is responsible for the establishment and optimization of each management information flow. Flow establishment takes place either pro-actively (during an MCE registration) or re-actively (during a MCE configuration update phase). The *IFEO* oversees the negotiation activity between potential information sources and sinks. The outcome is one or more flows parameterized appropriately to meet the particular optimization requirements. The management information flow constraints / requirements come from the MCEs during their MCE registration / configuration update phases. The constraints / requirements are aligned with the high-level performance objectives of the system. This operation applies the optimization decisions coming from the Information Quality Controller operation, described below.

Information Quality Control This operation is responsible for taking the management information flow optimization decisions. The IQC requires sophisticated heuristics that match requirements with constraints (i.e. local optimization) with respect to the global requirements in the system (i.e. global optimization). Different prioritization levels are set in these two optimization aspects, in case of conflicting goals. An MCE may participate in a number of co-dependent management information flows. So, a change in the network environment or the requirements may trigger a re-establishment of a group or all existing management information flows. The *IFEO*

DOI: 10.1002/nem

function enforces the optimization decisions by communicating with the corresponding MCEs and the VIS functions, in order to satisfy the performance optimization requirements.

In figure 6, we show how the *IFEO function* acts as the heart of VIS efficient operation, including optimizing: (i) the deployed / overseen management information flows through the ICD function, (ii) the storage and indexing capabilities through the ISI function, and (iii) the information aggregation and knowledge production features through the *IPKP function*. All optimization aspects consider the global performance goals in the system, which come from corresponding high-level management applications. These management applications may trigger changes in all optimization algorithms, based on their specialized but global performance measurements (e.g. storage efficiency, communication cost, accuracy of processed information, etc).

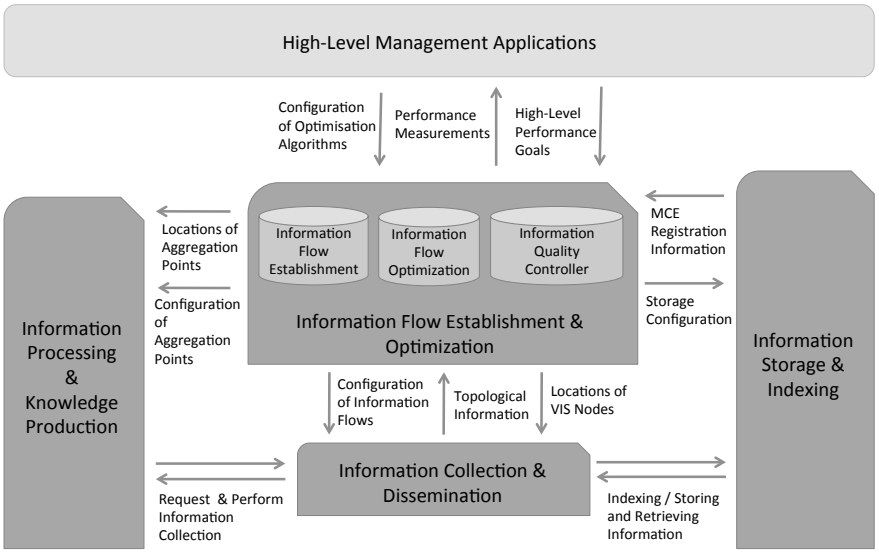


Figure 6. The Management Information Flow Orchestration Interactions

4. EXPERIMENTAL EVALUATION OF MAIN VIS OPERATIONS

VIS as an open source solution that is available at [48]. In this section, we present an evaluation exercise of the main operations of the Virtual Infrastructure Information Service (VIS). First, we discuss our experimental methodology, then we give our experimental results in terms of flexibility and adaptability of the information exchange process.

In order to experiment with a *complete virtualized environment*, we built a test-bed that integrated our VIS implementation with the *Very Lightweight Service Platform (VLSP)* platform [47]. In paper [46], we provide the VIS implementation details and a functional extensive experimental analysis in an NFV MANO [23] context. The VLSP is our unified approach to NFV and SDN, which is also available as an open-source solution at [13].

VLSP supports: (i) software-defined virtual *network topologies*, and their associated management, and (ii) efficient service deployment & operation with the relevant management aspects, rather than just flow management. The VLSP collects measurement data using the Lattice monitoring system, presented in [11], and optimizes distributed node placement using the algorithms proposed in [16]. The resource allocation mechanism used in VLSP was presented and evaluated in [12]. The same paper describes an early implementation of some components used in the VIM and LNH layers. The VIS, VLSP, and Lattice components can be all downloaded from [14] where more details of the components can be found.

DOI: 10.1002/nem

4.1. Experimental Methodology

Every experiment starts with the creation of a virtual network topology over a physical network of 11 servers with 4 CPU cores and 8-32GB of physical memory each. The network consists of 100 Virtual Routers (VR) and a number of randomly created virtual links. The connected routers are chosen randomly, using the Barabasi-Albert (BA) preferential attachment model [5] which explains some features of the real Internet topology. Each new virtual router is assigned dynamically from the VLSP to the physical machine with the least processing load. We have investigated more sophisticated resource allocation algorithms, results of which can be found in [12].

We implemented MCEs as simple management applications, with diverse information management requirements for testing purposes. All these applications support three communication methods (namely push/pull, pub/sub, and direct communication) and each can specify and update their own requirements at any point of the communication. This is done by changing: the requested communication method, the local performance goal, or the minimum / maximum data rates. By using the dynamic node selection algorithms presented in [16] and having as input the global network view, the output of the negotiation determines the most appropriate data paths for the management information flows.

The MCEs periodically transmit performance measurements to the VIS over the established management information flows using 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.
- *Information Freshness:* The time taken from the production of the new information to the point it reaches the requesting MCE. This is one way to quantify the "quality of information".
- *Average CPU Load:* Reflects 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 (the redis NoSQL database [74]).

The average values of all the above metrics are calculated every 10 seconds in a separate metric collection aggregator. Due to the stochastic nature of our experiments, we evaluated the statistical accuracy of our results using ten runs and calculated the average values for each measurement. This number of runs was deemed appropriate in order to produce very low standard deviations.

We deployed distributed VIS nodes using the PressureTime placement algorithm [16], whereby the number of VIS nodes is based on the topology size. After that, the VLSP assigns all of the MCEs to the most appropriate VIS node, i.e. the closest to them. Finally, after a small warm-up period, the communication starts.

4.2. Evaluation Results

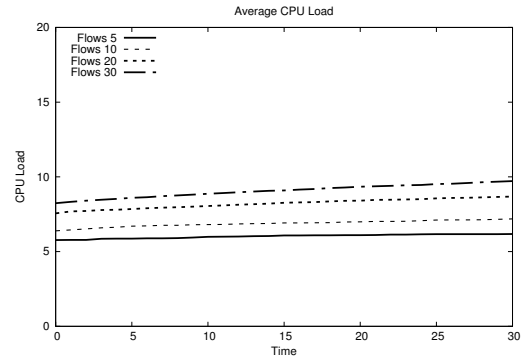
In order to validate the main VIS capabilities, we have defined the following two scenarios:

- *Scenario 1 – VIS Adaptability:* demonstrates how the VIS adapts to different conditions in terms of MCE requirements and management information flows number.
- *Scenario 2 – VIS Flexibility:* highlights how the VIS supports concurrent diverse needs, while serving a global performance goal.

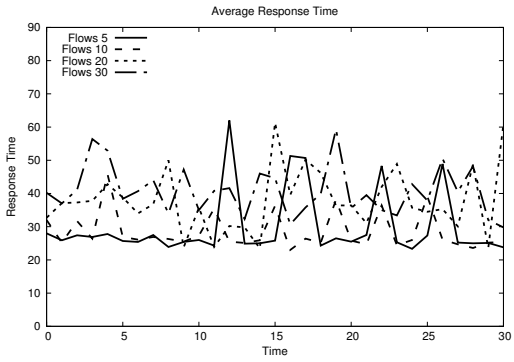
The results of these two scenarios are presented.

4.2.1. Scenario 1 – VIS Adaptability For this first scenario we experimentally explore the adaptability properties of the VIS, given the diverse network environment conditions and the varying MCEs' requirements and constraints. We used a topology of 100 virtual routers, while the number of information flows ranged from 5 to 30. The scenario uses up to 60% of the routers as sources and sinks for network management and control information, and an additional number of routers for the distributed VIS nodes, thus matching a wide range of realistic distributed management and

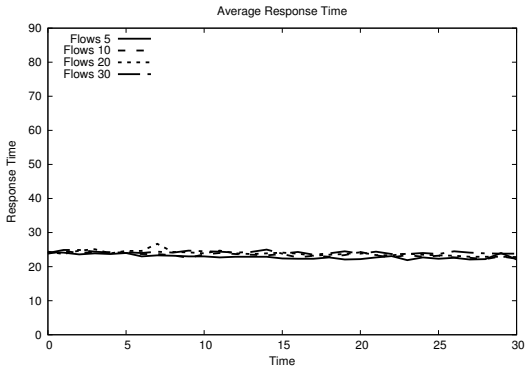
Prepared using nemauth.cls
DOI: 10.1002/nem



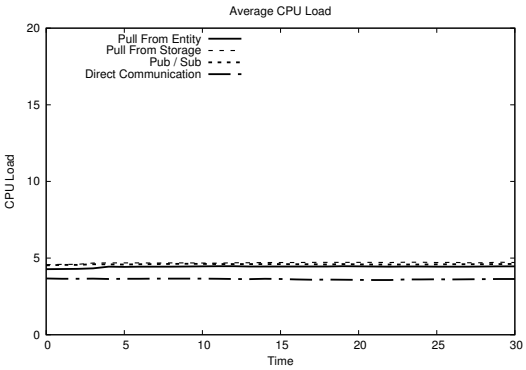
(a) CPU Load - Pull from Entities Method



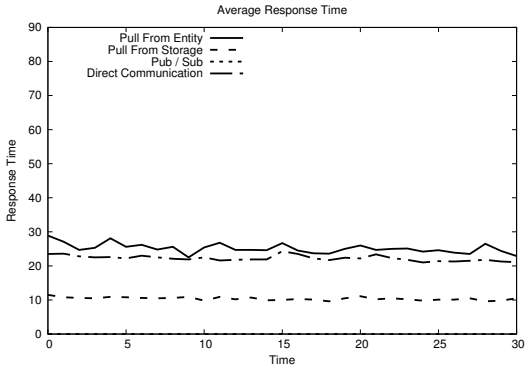
(b) Response Time - Pull from Entities Method



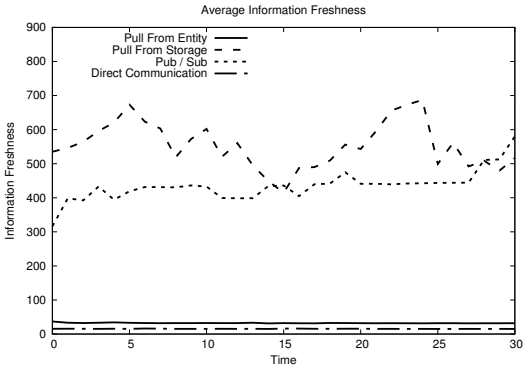
(c) Response Time - Direct Communication Method



(d) CPU Load - Varied Communication Methods



(e) Response Time - Varied Communication Methods



(f) Information Freshness - Varied Communication Methods

Figure 7. Impact of Information Flows Number and Communication Method

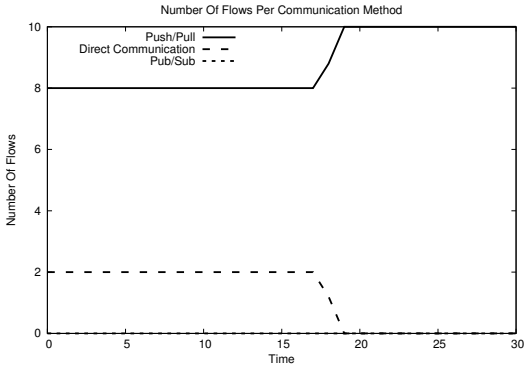
control software deployments, in terms of flow numbers. We executed the experiments with the different communication methods outlined in section 3. The communication method is one of the many information flow configuration options that could be demonstrated here (i.e. see subsection 3.1.4 for other information flow optimization aspects).

According figure 7(a), which shows CPU load, 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 of VIS, as the number of management information flows increases. However, this increase is stable and predictable. According the figure 7(b), the average response time shows a minor increase as

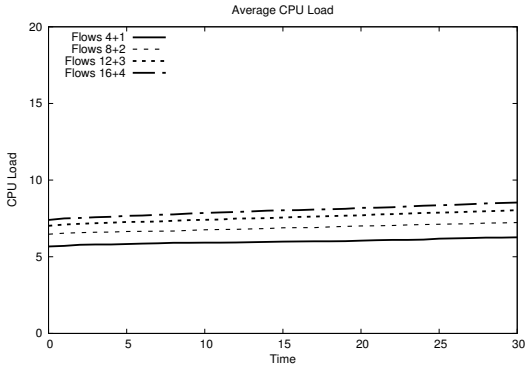
Prepared using nemauth.cls
DOI: 10.1002/nem

the number of management information flows increases. Here, the response time may exhibit a minor jitter, in the range of milliseconds, that can increase with management information flows contention. We have determined that many of these spikes occur due to task and thread switching and other low-level OS processes that may run in the servers, and are not VIS attributes. These spikes will not happen with dedicated SDN communication devices that use separate network processors. Furthermore, fully distributed methods do not have this issue (figure 7(c)). As the involvement duration of the VIS (including the centralized storage behind it) is gradually reduced, the jitter is reduced as well.

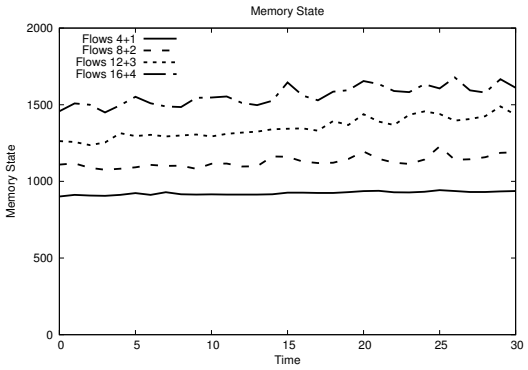
4.2.2. Scenario 2 – VIS Flexibility In this scenario, we created a topology of 100 virtual routers with random virtual links. Using this topology, we experimented with 5, 10, 15, and 20 management information flows. Each time, 20% of the management information flows (i.e. the selected flows) were configured to use the Direct Communication method, where the rest were using the Pull from Storage. At some point in time, the application associated with the Direct Communication MCEs requests a change in its own requirements – this request triggers a re-negotiation of the corresponding flows. The new requirement is for an *improvement in the average response time*. The re-negotiation process ends up converting the direct communication flows into Pull from Storage flows – communicating through the VIS. Using this scenario, we demonstrate how an application may change its requirements in terms of information manipulation and how VIS responds to that, in a way that is aligned to its own global performance goal requirements. In other words, how the local optimization with the global optimization aspects are being balanced. This strategy can be associated with a control loop that detects and tackles performance problems. We plan to introduce such a management capability in the near future.



(a) Type of Flows



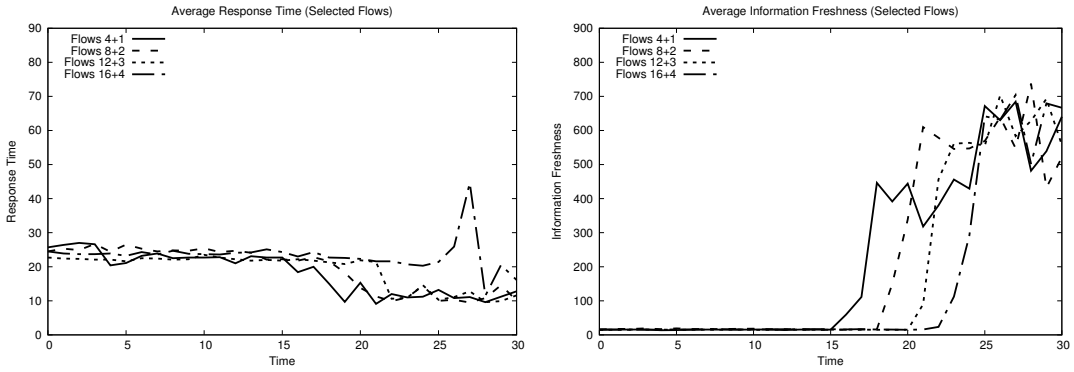
(b) VIS CPU Load



(c) VIS Memory Consumption

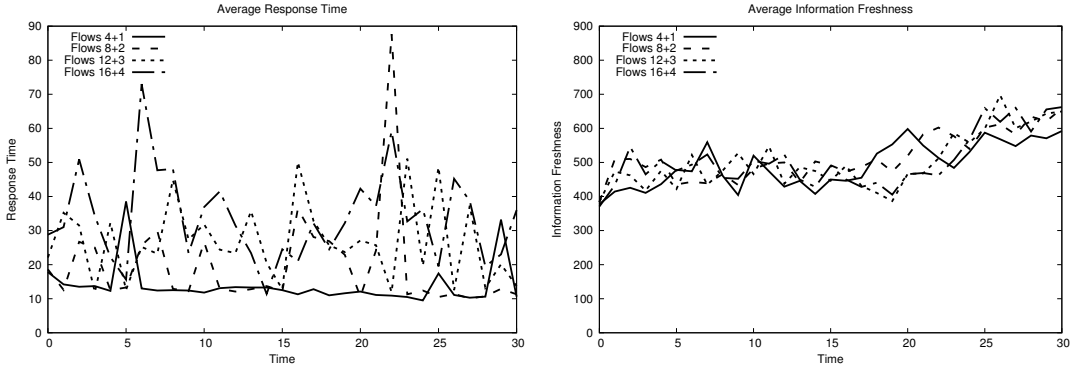
Figure 8. System Status

Prepared using nemauth.cls
DOI: 10.1002/nem



(a) Average Response Time of Selected Flows (b) Average Information Freshness of Selected Flows

Figure 9. Impact on Selected Flows



(a) Average Response Time (b) Average Information Freshness

Figure 10. Global System Behavior

In the beginning of the experiment, we have 8 flows communicating via the VIS (using Pull from Storage method) and 2 flows communicating directly. After the 17th second, the latter two flows request a change in their communicating method, which is granted from VIS. After approximately 1 more second, the two flows have been re-established and re-configured to use the Pull from Storage method. In figure 8(a), we see a snapshot of how the type of flows changes with time, showing a total of 10 flows. As is shown in figures 8(b), 8(c), the VIS accommodates the different number of management information flows, with a slight increase in processing load and memory consumption per flow. This is an insignificant impact of this process on the cpu load and memory state of the VIS.

The observed impact on the performance of the two flows and the global performance is in figures 9(a) and 9(b). We see that a trade-off between response time and information freshness is tuned in a way that satisfies the application using the selected flows. There is a reduction in the average response time after the flow re-establishment but with an increase in information freshness. Regarding the global view (see figures 10(a), 10(b)), there is no statistical important impact in the average response time. However, the average information freshness increased, due to the higher information freshness of the selected flows.

As a bottom line, VIS is able to orchestrate the management information flows and balance local with global requirements and grant requests to tune performance trade-offs in a way that satisfies local applications without damaging the global performance.

Prepared using nemauth.cls
DOI: 10.1002/nem

5. CONCLUSIONS

In this paper, we have identified four research challenges for the future evolution of SDN / NFV technologies towards improving service-awareness and information management in virtualized highly dynamic environments. As an important step towards tackling the identified open research issues, we propose the introduction of an abstracted information manipulation facility that is both adaptable to the requirements of the diverse management applications, as well as the constraints of the underlying resources.

Additionally, we have introduced and highlighted the design details and the characteristics of VIS – the Virtual Infrastructure Information Service. VIS is a new information management and orchestration infrastructure and service for highly dynamic virtualized environments, such as SDN and NFV enabled Virtual Infrastructure and network clouds. The VIS is the first proposal, to our knowledge, that resides between the applications and services and the network infrastructure, and is able to orchestrate the management information flows, while at the same time balancing the the global requirements with the local requirements. We have provided a proof-of-concept platform demonstrating these challenging features. Evaluation results demonstrating the behaviour of the main VIS operations are also included in the paper.

For future work we consider the following:

- to thoroughly experiment and improve the negotiation heuristics of management information flows in order to support a wider range of parameters, considering a selection of important management problems.
- to investigate other local and global performance trade-offs that can be tuned, such as between energy efficiency and QoS.
- to improve our facility towards the better support of service-chaining capabilities (such as [49] and [73]).
- to investigate a number of optimization strategies and associate them with different high-level performance goals.
- to determine how VIS behaves in dynamic environments and the when considering trade-offs being associated with the information flow negotiation complexity and delay-sensitive management applications.
- to observe the impact of resource allocation algorithms for different types of virtual resources, allowing us to reach even larger scales.
- to evaluate complete autonomic control loops, at both local and global levels, for addressing performance and stability problems.
- to consider aspects from the Information-Centric Networks (ICN) paradigm [2], by having data applications that can communicate over negotiated flows, while the global behaviour of the system will be monitored and controlled in a logically centralized manner.

There is more research needed in the area of information management and the provisioning of an *Information Management as a Service* solutions for highly dynamic environments.

ACKNOWLEDGEMENT

This work was partially supported by the European Union DOLFIN [19] project of the 7th Framework Program, and the EU H2020 projects: 5GEX – “5G Multi-Domain Exchange” [1] and SONATA – “Service Programing and Orchestration for Virtualized Software Networks” [68].

REFERENCES

1. 5GEX. EU H2020 - 5G Multi-Domain Exchange (5GEx) project, 2015. https://5g-ppp.eu/5GEx.
2. B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman. A survey of information-centric networking. *Communications Magazine, IEEE*, 50(7):26–36, 2012.
3. 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.

DOI: 10.1002/nem

4. S. Andreozzi, N. De Bortoli, S. Fantinel, A. Ghiselli, G. L. Rubini, G. Tortone, and M. C. Vistoli. Gridice: a monitoring service for grid systems. *Future Generation Computer Systems*, 21(4):559–571, 2005.
5. A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.
6. M. Chiosi, D. Clarke, P. Willis, A. Reid, et al. Network Functions Virtualisation. Technical report, White paper at the SDN and OpenFlow World Congress, ETSI, 2012.
7. N. M. M. K. Chowdhury and R. Boutaba. A survey of network virtualization. *Computer Networks*, 54(5):862–876, 2010.
8. D. D. Clark, C. Partridge, J. C. Ramming, and J. T. Wroclawski. A knowledge plane for the internet. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 3–10. ACM, 2003.
9. S. Clayman, R. Clegg, L. Mamatas, G. Pavlou, and A. Galis. Monitoring, aggregation and filtering for efficient management of virtual networks. In *Proceedings of the 7th International Conference on Network and Services Management*, pages 234–240, 2011.
10. S. Clayman, A. Galis, C. Chapman, G. Toffetti, L. Rodero-Merino, L. M. Vaquero, K. Nagin, and B. Rochwerger. Monitoring service clouds in the future internet. In *Future Internet Assembly*, pages 115–126, 2010.
11. S. Clayman, A. Galis, and L. Mamatas. Monitoring virtual networks with lattice. In *Network Operations and Management Symposium Workshops (NOMS Wksps), 2010 IEEE/IFIP*, pages 239–246. IEEE, 2010.
12. S. Clayman, E. Maini, A. Galis, A. Manzalini, and N. Mazzocca. The dynamic placement of virtual network functions. In *1st IEEE / IFIP International Workshop on SDN Management and Orchestration*, 2014.
13. S. Clayman, L. Mamatas, and A. Galis. The Very Lightweight Software-Driven Network and Services Platform (VLSP) Open Source Software, 2015. http://clayfour.ee.ucl.ac.uk/usr/index.html.
14. S. Clayman, L. Mamatas, and A. Galis. The VIS, VLSP and Lattice Open Source Software, 2015. http://clayfour.ee.ucl.ac.uk.
15. S. Clayman, L. Mamatas, and A. Galis. Efficient Management Solutions for Software-Defined Infrastructures. In *First IFIP/IEEE International Workshop on Management of 5G Networks (5GMan 2016) in conjunction with the IFIP/IEEE Network Operations and Management Symposium (NOMS 2016)*. IEEE/IFIP, 2016.
16. 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*, 62(6):1207–1220, 2013.
17. 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*, pages 2–2. IEEE, 2007.
18. DMTF. DMTF Common Information Model (CIM), 2016. www.dmtf.org.
19. DOLFIN. Data centres optimization for energy-efficient and environmentally friendly internet (dolfin fp7 project), 2013. http://www.dolfin-fp7.eu.
20. 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*, pages 615–620. IEEE, 2008.
21. R. Enns, M. Bjorklund, J. Schoenwaelder, and A. Bierman. Network configuration protocol (netconf). *Internet Engineering Task Force, RFC*, 6241, 2011.
22. ETSI. NFV Information Model WI IFA015, 2015. http://docbox.etsi.org/ISG/NFV/Open/Drafts/IFA015_NFV_Information_Model/NFV-IFA015v020.zip.
23. ETSI. NFV MANO WG, 2015. http://portal.etsi.org/portal/server.pt/community/NFV/367?tbId=79.
24. A. D. Ferguson, A. Guha, C. Liang, R. Fonseca, and S. Krishnamurthi. Participatory networking: An api for application control of sdns. In *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM*, pages 327–338. ACM, 2013.
25. A. Galis, J. Rubio-Loyola, S. Clayman, L. Mamatas, S. Kukliński, J. Serrat, and T. Zahariadis. Software enabled future internet – challenges in orchestrating the future internet. In D. Pesch, A. Timm-Giel, R. A. Calvo, B.-L. Wenning, and K. Pentikousis, editors, *Mobile Networks and Management*, volume 125 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 228–244. Springer International Publishing, 2013.
26. 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:1305.0209*, 2013.
27. 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*, pages 163–174. ACM, 2014.
28. V. K. Gurbani, M. Scharf, T. Lakshman, V. Hilt, and E. Marocco. Abstracting network state in software defined networks (sdn) for rendezvous services. In *Communications (ICC), 2012 IEEE International Conference on*, pages 6627–6632. IEEE, 2012.
29. C.-Y. Hong, S. Kandula, R. Mahajan, M. Zhang, V. Gill, M. Nanduri, and R. Wattenhofer. Achieving high utilization with software-driven wan. In *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM*, pages 15–26. ACM, 2013.
30. I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, M. Dean, et al. Swrl: A semantic web rule language combining owl and ruleml. *W3C Member submission*, 21:79, 2004.
31. F. Hu, Q. Hao, and K. Bao. A survey on software defined networking (sdn) and openflow: From concept to implementation. *Communications Surveys Tutorials, IEEE*, PP(99):1–1, 2014.
32. IETF. YANG Data Model for Interface Management, 2014. https://tools.ietf.org/html/rfc7223.
33. E. Imamagic and D. Dobrenic. Grid infrastructure monitoring system based on nagios. In *Proceedings of the 2007 workshop on Grid monitoring*, pages 23–28. ACM, 2007.
34. ITU-T. ITU-T Information Model, 2016. http://www.itu.int/rec/T-REC-M.3100/en.
35. S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, et al. B4: Experience with a globally-deployed software defined wan. In *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM*, pages 3–14. ACM, 2013.
36. D. A. Joseph, A. Tavakoli, and I. Stoica. A policy-aware switching layer for data centers. *ACM SIGCOMM Computer Communication Review*, 38(4):51–62, 2008.

37. H. Kim and N. Feamster. Improving network management with software defined networking. *Communications Magazine, IEEE*, 51(2):114–119, 2013.
38. E. Kissel, G. Fernandes, M. Jaffee, M. Swany, and M. Zhang. Driving software defined networks with xsp. In *Communications (ICC), 2012 IEEE International Conference on*, pages 6616–6621. IEEE, 2012.
39. T. Koponen, M. Casado, N. Gude, J. Stribling, L. Poutievski, M. Zhu, R. Ramanathan, Y. Iwata, H. Inoue, T. Hama, et al. Onix: A distributed control platform for large-scale production networks. In *OSDI*, volume 10, pages 1–6, 2010.
40. 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*, pages 37–42. ACM, 2015.
41. 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. In *Integrated Network Management (IM 2013)*, pages 1084–1085, May, 2013.
42. N. Koutsouris, K. Tsagkaris, P. Demestichas, L. Mamatas, S. Clayman, and A. Galis. Managing software-driven networks with a unified management framework. In *Integrated Network Management (IM 2013)*, pages 1084–1085, May, 2013.
43. D. Kreutz, F. M. V. Ramos, P. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig. Software-defined networking: A comprehensive survey. *CoRR*, abs/1406.0440, 2014.
44. D. Levin, A. Wundsam, B. Heller, N. Handigol, and A. Feldmann. Logically centralized?: state distribution trade-offs in software defined networks. In *Proceedings of the first workshop on Hot topics in software defined networks*, pages 1–6. ACM, 2012.
45. 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*, pages 527–534. IEEE, 2010.
46. L. Mamatas, S. Clayman, and A. Galis. Information Management as a Service for Network Function Virtualization Environments, 2015. Submitted to IEEE TNSM on 22/2/2016, available to the reviewers at http://tinyurl.com/zqvlvrr.
47. L. Mamatas, S. Clayman, and A. Galis. A service-aware virtualized software-defined infrastructure. *Communications Magazine, IEEE*, 53(4):166–174, 2015.
48. L. Mamatas, S. Clayman, and A. Galis. The Virtual Infrastructure Information Service Open Source Software, 2015. http://clayfour.ee.ucl.ac.uk/ikms/index.html.
49. A. Manzalini et al. Software-Defined Networks for Future Networks and Services. Technical report, White paper based on the IEEE Workshop SDN4NS 2013, 2014.
50. M. L. Massie, B. N. Chun, and D. E. Culler. The ganglia distributed monitoring system: design, implementation, and experience. *Parallel Computing*, 30(7):817–840, 2004.
51. D. L. McGuinness, F. Van Harmelen, et al. Owl web ontology language overview. *W3C recommendation*, 10(10):2004, 2004.
52. N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. Openflow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2):69–74, 2008.
53. J. Medved, R. Varga, A. Tkacik, and K. Gray. Opendaylight: Towards a model-driven SDN controller architecture. In *2014 IEEE 15th International Symposium on*, pages 1–6. IEEE, 2014.
54. 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*, 18(1):236–262, 2016.
55. H. B. Newman, I. C. Legrand, P. Galvez, R. Voicu, and C. Cirstoiu. Monalisa: A distributed monitoring service architecture. *arXiv preprint cs/0306096*, 2003.
56. N. Niebert, S. Baucke, I. El-Khayat, M. Johnsson, B. Ohlman, H. Abramowicz, K. Wuenstel, H. Woesner, J. Quittek, and L. M. Correia. The way 4ward to the creation of a future internet. In *Personal, Indoor and Mobile Radio Communications, PIMRC 2008. IEEE 19th International Symposium on*, pages 1–5. IEEE, 2008.
57. R. Olups. Zabbix 1.8 network monitoring, 2010.
58. ONF. Software-Defined Networking: The New Norm for Networks. Technical report, Open Network Foundation, 2012.
59. Open Source initiatives. OPNFV, "Open Platform for NFV, https://www.opnfv.org/.; OpenDaylight, "SDN and NFV platform that enables network control and programmability," http://www.opendaylight.org/.; OpenStack, "OpenStack Networking (Neutron)," https://wiki.openstack.org/wiki/Neutron. Technical report.
60. 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*, volume 43, pages 27–38. ACM, 2013.
61. 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*, page 1. ACM, 2013.
62. 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)*, pages 227–240, Lombard, IL, 2013. USENIX.
63. J. P. Reilly. Implementing the tm forum information framework (sid): A practitioner's guide. 2011.
64. B. Rochwerger, D. Breitgand, E. Levy, A. Galis, K. Nagin, I. M. Llorente, R. Montero, Y. Wolfsthal, E. Elmroth, J. Cáceres, et al. The reservoir model and architecture for open federated cloud computing. *IBM Journal of Research and Development*, 53(4):4–1, 2009.
65. J. Rubio-Loyola, A. Astorga, J. Serrat, W. Chai, L. Mamatas, A. Galis, S. Clayman, A. Cheniour, L. Lefèvre, O. Mornard, et al. Platforms and software systems for an autonomic internet. In *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pages 1–6. IEEE, 2010.
66. J. Schönwälder, M. Björklund, and P. Shafer. Network configuration management using netconf and yang. *IEEE communications magazine*, 48(9):166–173, 2010.

DOI: 10.1002/nem

67. SDN De-facto / Dejure Standards. IRTF, "Software-Defined Networking (SDN): Layers and architecture terminology," Tech. Rep., January 2015.; ONF, "Software-Defined Networking: The New Norm for Networks," Open Network Foundation, Tech. Rep., 2012.; ITU-T, "Recommendation y.3300 (2014) - Framework of Software-Defined Networking; Recommendation y.3001 (2012) - Future Networks: Objectives and Design Goals, Recommendation y.3011 (2012) - Framework of Network Virtualization for Future Networks"; ETSI NFV : 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. Technical report.
68. SONATA. EU H2020 - 5G Service Programing and Orchestration for Virtualized Software Networks, 2015. https://5g-ppp.eu/sonata/.
69. S. Song, S. Hong, X. Guan, B.-Y. Choi, and C. Choi. Neod: network embedded on-line disaster management framework for software defined networking. In *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on*, pages 492–498. IEEE, 2013.
70. TMF. TMF Information Framework (SID). http://www.tmforum.org/information-framework-sid/.
71. A. Tootoonchian and Y. Ganjali. Hyperflow: a distributed control plane for openflow. In *Proceedings of the 2010 internet network management conference on Research on enterprise networking*, pages 3–3. USENIX Association, 2010.
72. UniverSELF. Deliverable D2.4 - Unifed Management Framework (UMF) Specifications, 2010. Available at http://tinyurl.com/mr6rhgn.
73. N. Yadav, J. Guichard, B. McConnell, C. Jacquenet, M. Smith, A. Chauhan, M. Boucadair, P. Quinn, R. Manur, P. Agarwal, et al. Network service chaining problem statement. *Network*, 2013.
74. J. Zawodny. Redis: Lightweight key/value store that goes the extra mile. *Linux Magazine, August*, 31, 2009.

DOI: 10.1002/nem