Towards a hierarchical global naming framework in network virtualization

Yanzhe Che[1], Qiang Yang[2] and Chunming Wu[1]
[1] College of Computer Science and Technology, Zhejiang University
Hanzhou, 310027 - China
[e-mail: pomme@zju.edu.cn]
[2] College of Electrical Engineering, Zhejiang University
Hanzhou, 310027 - China
[e-mail: qyang@zju.edu.cn]
*Corresponding author: Qiang Yang

Received September 28, 2012; revised March 26, 2013; accepted March 28, 2013; published May 31, 2013

Abstract

Network virtualization enables autonomous and heterogeneous Virtual Networks (VNs) to co-exist on a shared physical substrate. In a Network Virtualization Environment (NVE), the fact that individual VNs are underpinned by diverse naming mechanisms brings about an obvious challenge for transparent communication across multiple VN domains due to the complexity of uniquely identifying users. Existing solutions were mainly proposed compatible to Internet paradigm with little consideration of their applications in a virtualized environment. This calls for a scalable and efficient naming framework to enable consistent communication across a large user population (fixed or mobile) hosted by multiple VNs. This paper highlights the underlying technical requirements and presents a scalable Global Naming Framework (GNF), which (1) enables transparent communication across multiple VNs owned by the same or different SPs; (2) supports communication in the presence of dynamics induced from both VN and end users; and (3) greatly reduces the network operational complexity (space and time). The suggested approach is assessed through extensive simulation experiments for a range of network scenarios. The numerical result clearly verifies its effectiveness and scalability which enables its application in a large-scale NVE without significant deployment and management hurdles.

Keywords: Network virtualization environment, naming framework, mapping size, lookup operation

This research was supported by the National Basic Research Program of China (973 Program) (2012CB315903), the National Natural Science Foundation of China (No. 51107113 and No. 61070157) , and the Key Science and Technology Innovation Team Project of Zhejiang Province (2011R50010).

http://dx.doi.org/10.3837/tiis.2013.05.015

1. Introduction

Though today's Internet has achieved tremendous success in the past few decadesto deliver a variety of services to residential and business users, it is considered to be suffering from "ossification" as emerging networking innovations can be hardly embedded due to intrinsic architectural and operational restrictions [1-3]. The network virtualization technique is now deemed as one of the most powerful tools of addressing this issue by enabling multiple heterogeneous and autonomous virtual networks adopting with different protocols and control mechanisms to co-exist on a shared physical substrate [4]. There already exists several projects, e.g. GENI [5] and 4WARD [6], which exploring networks of the future based on virtualization technique.

Chowdhury et al. [7-9] highlighted the key difference between the network virtualization and the existing Internet paradigm though defining a business-oriented NVE model. The role of conventional Internet Service Providers (ISPs) is portioned into two distinct segments: (1) Infrastructure Providers (InPs), who own the underlying physical network and are responsible for infrastructure deployment and maintenance as well as resource management; and (2) Service Providers (SPs), who manage rented physical resources from one or multiple InPs to deploy customized VNs for delivering tailored services to end users.

In network virtualization environment, SPs allow the on-demand deployment of the co-existing virtual networks while preserving their autonomies. Meanwhile the InPs deploy and maintain the physical infrastructure and resources which can be leased by SPs. On the other hand, SPs transparently utilize these physical resources to establish VNs to fulfill the diverse service requirements, e.g. Service Level Agreements (SLAs) and bandwidth, without revealing any detailed physical information.

It is the duty for SPs to maintain the virtual networks which involves deployment, supervision or even re-allocation of these virtual resources. SPs are also responsible for enabling transparent end-to-end communication within or across multiple VNs. It is inappropriate to simply apply the current DNS mechanism which serves as a "phone book" for translating host names into understandable IP addresses where user identity and physical location are bound together. The naming mechanism is one of the key challenges to enable the operations and managements of the network virtualization. Without an appropriate naming mechanism, the network entities cannot be uniquely identified, and hence the communication services cannot be appropriately implemented. Therefore, a novel naming framework is needed to fit for a collection of unique characteristics of NVE list as below:

Virtual network autonomy: since VNs may be deployed by different SPs, the individual co-existing VNs may have adopted heterogeneous naming schemes for protecting network autonomy or privacy, which prevents transparent end-to-end communication across multiple VNs.

Decoupling location from identity: the IP address in current Internet represents the user physical location as well as its identity which limits the user's mobile capability. In contrast, the user's identification is decoupled from its physical location in NVE.

User mobility and multi-homing: VN users can exhibit high mobility, e.g. joining or leaving a VN from time to time, due to many reasons, e.g. seeking new services, physical movement, and so forth. Moreover they can be connected with more than one VN simultaneously for accessing different services, i.e. multi-homing.

Scalability: as the scale of NVE grows with increasing number of end users as well as VNs participating in the NVE, the naming mechanism needs to be scalable to identify the massive number of users.

In this paper, we attempt to address the above technical challenges by presenting a hierarchical global naming framework (GNF) in the virtual network context. Through incorporating new entities in NVE, GNF defines a set of identifier spaces corresponding to these entities and provides mechanisms to enable communication between local and global identifiers through a collection of mapping policies managed at different entities. Through a comprehensive study with the existing solution, we confirmed that the suggested GNF solution can outperform the existing approaches in terms of efficiency and scalability with low time and space complexity.

The technical contributions made in this paper can be summarized as follows:

(1) The approach supports transparent end-to-end communications among end users across multiple VNs owned by the same or different SPs.

(2) The approach supports the communications among users with mobility and multi-homing characteristics in NVE and well copes with the high-level dynamic behaviors of VNs.

(3) In comparison with iMark [10], the proposed solution is with significantly reduced complexity, and hence enhanced scalability, e.g. greatly reduced mapping size and lookup operation time for VN hierarchy with multiple levels.

The rest of the paper is organized as follows: Section 2 reviews the related work; Section 3 introduces the architectural design of GNF in details including the incorporated NVE entities and identifier spaces, and the defined mapping mechanisms among them; Section 4 describes the operational details of GNF with VN and user dynamics; Section 5 presents the performance assessment and a set of key results through numerical simulation experiments for a range of network scenarios; finally Section 6 gives some concluding remarks with additional discussion.

2. Related Work

At present, the naming mechanism is one of the key technologies underpinning the next generation networks. This section reviews some relevant work without the intention of exhaustive overview.

Due to the fact that current Internet can no longer support communications among all network entities even with address space reuse technique, e.g. Network Address Translation (NAT), TRIAD [11] attempts to address the connectivity issue by using location-independent identifiers instead of IP addresses for entity identification and data routing. However, many restrictions limit its application in NVE, e.g. merely supporting source routing algorithm and IPv4. TurfNet [12] introduces a flexible host identity namespace allowing the adoption of different addressing and routing mechanisms in individual autonomous domains. It enables end to end connectivity across multiple federated domains via a combination of name registration, name resolution process as well as packet relaying at the network boundaries. However, the design of TurfNet cannot support mobility and multi-homing of users exhibited in NVE. The Layered Naming Architecture [13] adopts a layered approach consisting of four layers: user-level descriptors, service identifiers, endpoint identifiers and IP addresses with three levels of mapping maintained between adjacent layers. This approach enables global identification of Internet users with mobility or multi-homing. However, this solution is designed specifically for tackling Internet name resolution problem where making its direct application in NVE inappropriate. In [14], the authors provide a peer-to-peer naming

infrastructure which takes resource virtualization into account. The solution defines four identity spaces with a set of mapping between them, as well as maintains distributed naming agents (NAs) to keep the naming infrastructure up to date. In [7], the authors introduce a hierarchical geographic addressing scheme, named COST, to enable location aware forwarding. However, the scheme is based on the knowledge of all the addresses of physical substrate which is not acceptable for the network virtualization environment. In this paper, we take a further step on the naming mechanism for NVE by following the Layered Naming Architecture proposed in [13] and mapping mechanism design philosophy in [14].

Our previous work [15] also give a detailed specification about the virtual network architecture. However, it mainly focus on the access control issues within or across multiple VNs. To the author's best knowledge, iMark [10] is the only available solution addressing this open issue in literature so far. It efficiently decouples the identities of end hosts from their physical locations. By deploying a set of controllers and mappings, iMark enables universal connectivity (within and outside of one virtual network domain) without sacrificing VN autonomy and performance of the underlying physical networks. The controllers can translate back and forth for the local and global IDs of an entity. The mappings are stored by different controllers which can federate into a logical hierarchy. The controllers at top level of the hierarchy control all the entities below. As a result, the sizes of mappings at top-level controllers are always larger than those at the bottom. Therefore, two potential scalability limitations of iMark need to be considered: (1) the mapping size grows exponentially along with the increasing number of VN hierarchy levels; and (2) the lookup time becomes unacceptable for communication among end users from different VNs with different hierarchy. This implies prohibitive operational complexity (space and time) in the large scale NVE. In order to overcome these limitations, in this paper we further exploit the naming issue in NVE by proposing a novel global naming framework with enhanced scalability. The details of the approach is studied and explained in the following sections.

3. The GNF Architecture Overview

This section overviews the architecture of the proposed GNF solution, including the fundamental NVE entities, the identifier spaces, the hierarchical VN structure as well as the mapping mechanism design.

3.1 Entities and Identifier Space

In GNF, four major fundamental entities are defined as follows:

Infrastructure Provider (InP)/ Service Provider (SP): the definition of InP and SP are directly adopted from [8, 9] as described in Section 1. As SPs may potentially act as an InP to provide other SPs with the physical resources leased from InPs, they are referred as the same entity in GNF.

Virtual Network (VN): a VN is generally created and managed by a single SP. So an individual customized VN may deploy diverse protocols or network mechanisms. During a VN's life cycle, it can be created, terminated, merged with other VNs, separated into multiple VNs, aggregated into or disassociated from a VN hierarchy.

Virtual Resource: two types of virtual resources are defined: Logical Resource (LR) and End User (EU) resource. The former is the logical representation of the physical network components, e.g. routers and servers which are maintained by the InPs. The latter refers to the end users served by the VNs. Physically, end users are connected with the physical network

access points to access network services, but they may logically connect to one or multiple VNs and could join or leave VNs over time (i.e. mobility and multi-homing). As a result, one user may be a part of virtual resource for both two VNs simultaneity.

Identity Manager (IM): it is an entity playing a key role in each VN which manages all the identifiers of logical entities within the VN and maintains the mappings between different identifier spaces as well as provides with a set of specific functionalities, e.g. user search or access control. Through managing the user identifiers by the IMs, the user lookup operation can be carried out prior to setting up any communications between users in one or across multiple VNs.

Fig. 1 semantically illustrates the interactions among these entities in a network virtualization environment. Furthermore, five corresponding identifier spaces are defined as follows based on the aforementioned entities in the proposed naming framework. In this paper, we focus on the management of virtual resources of each VN, and the problem of substrate control by the InPs and the physical resources abstraction are out of the scope of this paper.

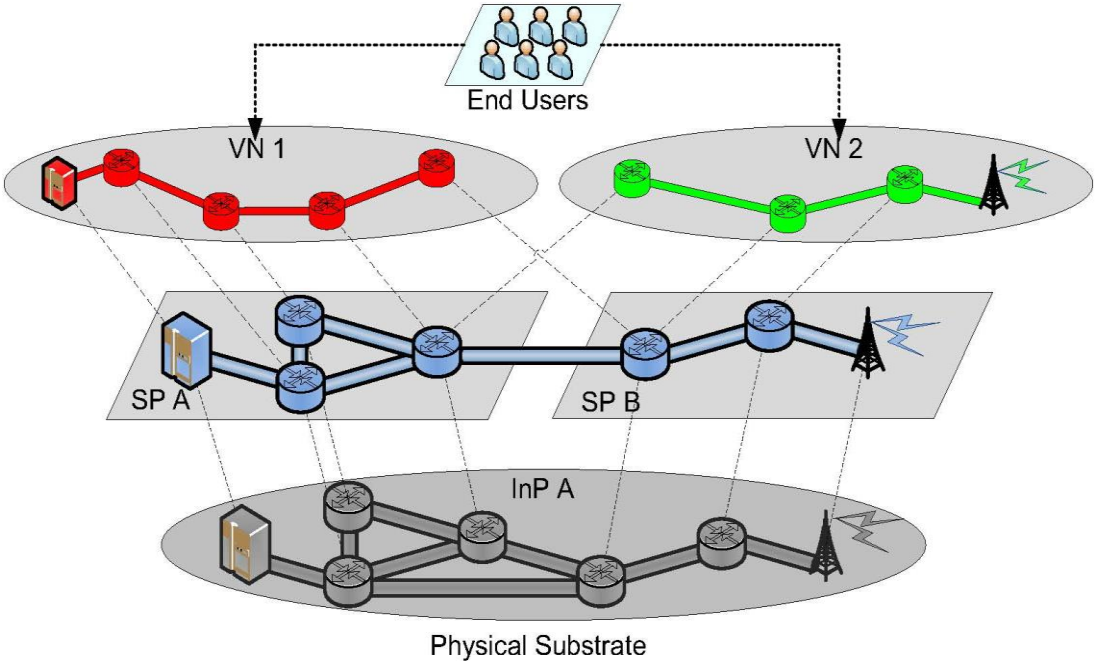

Fig. 1. Network virtualization environment

ISP-IDSpace: all the InPs and SPs can be uniquely identified by a global identifier, namely *isp-id*, which enables negotiation among InPs and SPs for creating customized VNs meeting certain criteria.

VN-IDSpace: all VNs can be uniquely identified in NVE using a global identifier, namely *vn-id*.

LR-IDSpace: the networking entities in the underlying substrate network can be identified using *lr-ids* defined by InPs. They are the original identifier for physical resource when leased to SPs. Moreover, the resources of a SP could also be leased to another SP whilst preserves its *lr-id*.

EU-IDSpace: individual end users can be identified via global location independent identifier, namely *eu-id*, irrespective of their connections to VNs.

VR-IDSpace: the identifiers of virtual resources, namely *vr-id*, can be obtained by combining *vn-id* and *lr-id* (representing the logical resource) and combining *vn-id* and *eu-id* (representing the end user resource). Since the *vn-id*, *lr-id* and *eu-id* are all uniquely defined, *vr-id* can uniquely specify a logical networking entity or an end user in the specified VNs. In NVE, a multi-homed end user is associated with a number of *vr-ids*, indicating its affiliation to multiple distinct VNs simultaneously.

As we can see from **Fig. 1**, VN1 is maintained by SP-A. However some of VN1's virtual resources are leased from SP-B. Those virtual resources can be distinguished by using the defined *vr-id*. For example, a logical resource in VN1 can be located by combining VN1's *vn-id* with the resource's *lr-id*, while the same logical resource leased to SP-B in VN2 can be found using VN2's *vn-id* and the same *lr-id*.

3.2 Hierarchical Structure of VN

In reality, along with the growth of NVE scale, the interactions among involved VNs may become extremely complex. To meet certain operational objectives, a group of VNs can be combined together to form a large VN domain, and these domains may further be associated with each other resulting in a complex hierarchical network structure. For example, a large international company may deploy individual VN for each department located at different places and associate them according to their business hierarchical relationship.

From the SP's perspective, a number of operations can be performed on VNs, e.g. *merge* operation and *aggregate* operation, as shown in **Fig. 2** With the *merge* operation, SP aggregates multiple VNs into a single VN to be administrated with a shared control platform and IM. Individual VNs may adopt naming mechanisms and protocols used in certain VNs or allow merged VNs to agree upon a new set of policies. In this case, individual IMs also need to be merged into one which maintains all the identifiers previously kept in individual VNs. On the other hand, the operation of *aggregate* allows each individual VN to preserve their autonomous nature even after they become a part of the composed VN. As the fundamental component in NVE, VN may be organized in a hierarchical structure to fulfill complex service requirements.

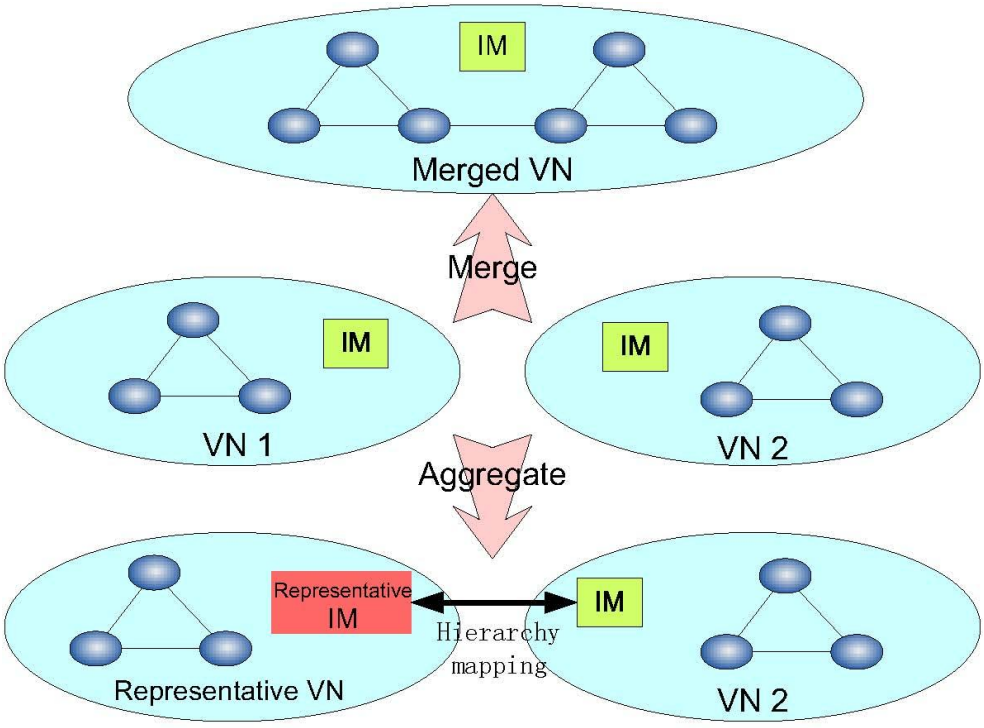

Fig. 2. Merge and aggregate operation in NVE

In this case, one VN will be selected to be the representative VN with its IM chosen as the representative IM which collects all the required mapping information from other IMs. Once a number of VNs are aggregated, the IMs of all the involved VNs (including the representative IM) only need to add a few mapping entries to record the relationship among VNs. Meanwhile, a number of VN aggregations may continually aggregate with other VNs, leading to a higher level of hierarchy. For the suggested solution in this paper, the process of *aggregate* operation can be effectively seen as aggregating their IMs of individual VNs. To achieve robust performance in federated VNs, advanced technique, e.g. loosely-coupled network structure of IMs, can be adopted to prevent IM failures in the hierarchical structure. **Fig. 3** depicts an example to show the IMs in a three-level hierarchical VN domain. With IM1, IM2 and IM3 are aggregated to form an aggregation domain at level 1, and IM3 elected to be the representative IM. At the second level, IM3, IM4 and IM5 are aggregated together with IM4 as the representative IM of this domain.

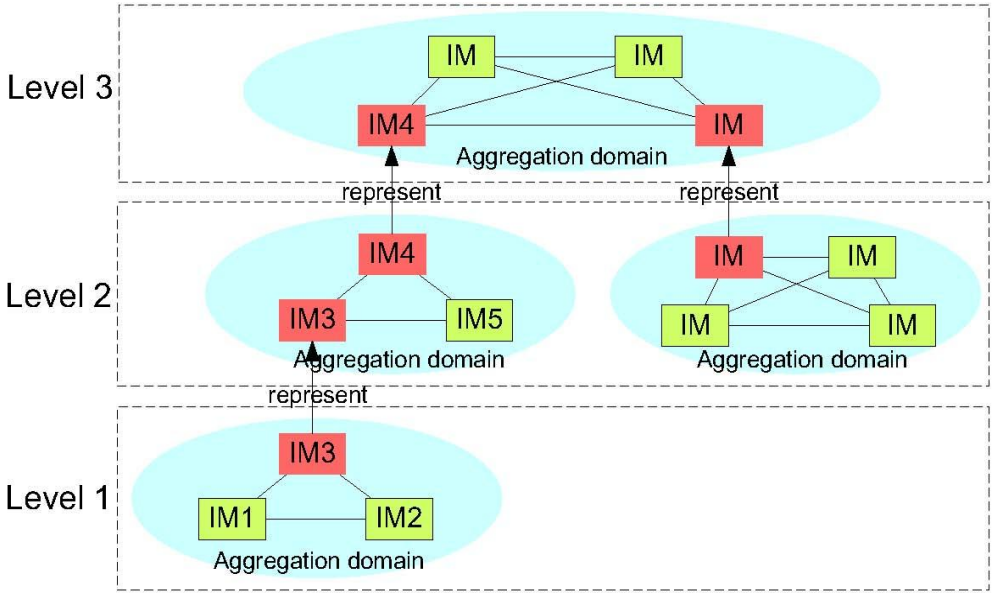

Fig. 3. IMs in aggregated domains at different hierarchies

3.3 Mapping Design

VN mapping information needs to be maintained up to date to indicate the relations among the defined identifier spaces and entities. With GNF, different entities in NVE, e.g. IMs and end users, need to maintain their local mappings, e.g. end user records the name of its associated VNs. The list of mappings defined in GNF is shown in **Table 1**.

Table 1. Mappings between defined Identifier Spaces

No.	Mapping	Specification
1	User → VN mapping	*eu-id* → { *vn-id* }
2	VN → User mapping	*vn-id* → { *eu-id* }
3	LR → VN mapping	*lr-id* → { *vn-id* }
4	VN → LR mapping	*vn-id* → { *lr-id* }
5	Hierarchical mapping	*vn-id* → { (*vn-id*, *vn-type*) }
6	Trace mapping	*eu-id* → (*from-vn-id*, *to-vn-id*)

The first two mappings are between end user global identifiers *eu-id* and virtual network global identifiers *vn-id*, where mapping 1 which supports user multi-homing can find the identifiers of all associated VNs and mapping 2 keeps the identifiers of all affiliated end users of any VN. Consider the scale of current Internet, mapping 2 could be extremely large in size due to the massive number of users. Therefore, GNF allows keeping the mapping 1 information locally to users, whereas mapping 2 is maintained at IM. Mapping 3 and 4 are between two identifiers of logical resources and virtual networks, i.e. *lr-id* and *vn-id*, respectively. Due to the fact that the logical resources of one SP can be shared by multiple VNs simultaneously, mapping 3 keeps the information of all VNs which share certain logical resource while mapping 4 records all the logical resources used by a specific VN.

In addition, hierarchical mapping is used to record the information regarding the hierarchical structure of individual VNs. With GNF, the representative IM of an aggregation domain is not required to have a copy of all the mapping information of other IMs within the domain, otherwise the mapping size of the representative IM could be very large [10].

Furthermore, keeping other IM's mapping records in the representative IM may violate the privacy or rights of individual end users in other VNs, as the end user's location information could be disclosed by internal or external domain operation. Moreover, to maintain the redundant mapping information across different IMs results in significant additional storage space. Under the circumstance that once changes are made in an IM's mapping, all the involving representative IMs have to modify their mappings in response to the changes to avoid mapping inconsistency.

In GNF, each IM only needs to keep a hierarchical mapping of other IMs which are in the same aggregation domain or its parent or child nodes in the hierarchical structure. By using this mapping, one VN can find its *parent*, *child* or *sibling* VNs' id. So we distinguish these different VN relationships by a parameter named *vn-type*. In the example shown in **Fig. 3**, VN3 is the parent node of VN1 whilst VN1 is the sibling node of VN2. This mechanism does not keep any redundant mapping information therefore it greatly reduces the average mapping size of each IM as well as potentially enhances the performance of lookup operation.

This is confirmed by the numerical results from a set of simulation experiments given in Section 5.

Finally trace mapping is designed to cope with the high-level mobility of end users. When an end user leaves a VN, e.g. "log out", other users who still use the old address may suffer from communication failure. In this case, a *lookup* operation is needed to be carried out to search for that address at other IMs in a sequential order. The existing solution, such as iMark, will first search within the VN (horizontally), if not find, then search in the upper level of VN hierarchy (vertically). In case the total number of end users is large, such a *lookup* operation procedure becomes onerous.

Aiming to solve this problem, we introduce the trace mapping strategy in which each IM records the move-in and move-out behavior of every end users. When a user connect to a VN, the IM will create or update (if exist) a record in the trace mapping by applying user's last connected VN's id, if exist, to the *from-vn-id* and leave the *to-vn-id* empty. Then the current IM will inform the user's last connected IM (known through *from-vn-id*) which can complete the corresponding record in the trace mapping by applying the current connected VN's id to the *to-vn-id*. As a result, one can find a target user's current VN id through querying one or a chain of trace mappings without performing an additional time-consuming lookup operation. However, adopting trace mapping may result in increased space complexity and extra entities management. These issues will be further evaluated in Section 5.

4. The operations of GNF

This section describes the GNF operations in details which can be classified into two types: VN operations, e.g., *create*, *terminate*, *merge*, *separate*, *aggregate* and *dissociate*, and end user operations, e.g., *join*, *leave*, *switch* and *lookup*. In this paper we mainly focus on the aspect of naming mechanism of these operations.

4.1 VN Operations

The design of GNF takes the VN dynamics into account which is one of the main contributions made in this paper. To create a VN, InPs and SPs need to negotiate by using their unique *isp-ids*, then trade their physical resources by using the *lr-id*, afterwards a global *vn-id* will be generated for the new VN which remains unchanged during its life cycle. During this operation, VN→LR mapping is created to identify the allocation of physical resources for creating the VN, at the same time LR→VN mapping is updated to record the VNs which share

the physical resource. Similarly, for the *terminate* operation, VN→User and VN→LR mappings maintained by the VN will be deleted, and the mappings of User→VN and LR→VN will be updated too.

In many cases, a number of VNs will need to be merged into a new VN for service delivery or other operational purposes. If fact, the *merge* operation can be seen as a combination of two VN operations, namely *create* and *terminate*, as a new VN is created and the old ones are to be terminated. The new VN can reuse one of its merged VN's *vn-id* or generate a new one. *Merge* operation also updates all the mappings associated with the involved VNs. Moreover, a single VN can be decomposed into multiple VNs via *separate* operation with new *vn-ids* generated for each new VN. At the same time, other relative VNs whose mapping records still refer to the old invalid *vr-id* should update their mappings (e.g. hierarchical mapping or trace mapping) accordingly.

The *aggregate* and *dissociate* operations on VNs are both related to the VN hierarchical structure. Existing solutions do not consider the scalability limitation and require each representative IM has the knowledge of all the end users in VNs at its sub-level in the hierarchical structure. This implies that the representative IM must have a copy of the mappings of other VNs in the aggregation domain. Along with the growth of hierarchical level, the mapping size at representative IM will increase exponentially which becomes complex to manage. In contrast, with GNF, IM only maintain its own mapping and some additional hierarchical information. When a group of VNs are aggregated together, one IM from certain VN is selected as the representative IM while making other IMs as the child nodes in the hierarchy so to form an aggregation domain.

Following the event that VN dissociates from an aggregation domain, only the hierarchical mappings of the VN domain need to be updated in order to keep the hierarchical relationship up to date. It should be noted that the issues of unified service access control and information consistency need to be addressed following to the operation of *merge* or *aggregate*, due to the various polices adopted in different VNs.

4.1 VN Operations

Another key contribution of this paper is that the proposed approach is designed with great flexibility to cope with the dynamic behaviors of end users, i.e. mobility and multi-homing. An end user may *join*, *leave* or *switch* VNs from time to time or associate with multiple VNs simultaneously. For *join* operation, the end user needs to have the knowledge of VN's *vn-id*, and adds an entry in the User→VN mapping. At the same time, the VN's IM adds an entry in its VN→user mapping. When a user joins in one of the VNs which are organized in hierarchical structure, only the corresponding IM needs to add an entry in its VN→user mapping. However, all of the representative (parent) IMs in iMark should record this mapping information which results in prohibitive complexity in networks with the Internet-scale. The *leave* operation is similar except for deleting instead of adding a mapping entry. The *switch* operation somehow can be treated as a combination of *leave* and *join* operations. During *switch* operation, the trace mapping should also be create and updated in both involved VNs as described before.

The *lookup* operation is a key task in finding the user's addresses before establishing the communication connection between any two VN users. In iMark, the target node provides its global unique ID *eu-id* to other nodes who intend to connect. With GNF approach, each node communicates with *vr-id* instead which is the combination of both *vn-id* and *eu-id*. For example, if Alice conducts a *lookup* operation for Bob, Alice will firstly query her VN's IM

for Bob's *vr-id*. If they are in the same VN, it's easy to find him and hence setup a connection. Otherwise, i.e., they are in different domain in VN hierarchy, a two-step *lookup* operation will be performed.

First step: given on Bob's *vn-id*, Alice will first search it within the hierarchical mapping. If the *vn-id* is found in the mapping, it indicates both of them are in the aggregation domain, and then Bob's VN can be obtained. Otherwise, the search request is delivered to the representative IM in the domain for further *lookup* operation outside its domain. This request would be continually forwarded to the VN in higher level of the hierarchy until the *vn-id* is found.

Second step: After the target VN is located, the target VN's IM will examine its VN→user mapping for *eu-id*. If the user is found, the *lookup* operation terminates and the end-to-end communication can be started. If not found, as Bob can be known to be moved from this target VN to another VN, the trace mapping will be used to find out Bob's current VN, *vn-id*, and return to execute the first step. The primary benefit of this two-step lookup scheme is that rather than searching an end user in every IM, it firstly determines the target VN and then to lookup the target end user. This process would take less time compared with iMark which searches for *eu-id* in every VN across the VN hierarchy.

5. Performance Evaluation and Numerical Result

This section experimentally evaluates the proposed naming framework, GNF, in comparison with a recent notable solution, iMark, through a set of simulation experiments for a range of NVE scenarios. In this work, we select iMark as the representative method is due to its superiority over other similar state-of-the-art approaches in the literature. Also, our comparison study with other existing solutions has obtained the similar findings. This paper considers the iMark based naming framework has two major potential scalability problems: (1) the large mapping size of IM for VNs which are located in complex hierarchical architecture; and (2) the performance degradation of *lookup* operation.

In this work, the simulation experiments are carried out with a PC (Intel Core 2, Quad Q6600, 2.4GHz, 3G RAM, Windows XP) by using a simulator implemented in C++. In these experiment scenarios, VNs are organized with a hierarchical structure, refer to **Table 2** and **Table 3**. For mapping size evaluation, we suppose that all the end users in one VN have been connected to at least other ten different VNs on average due to the mobile nature of users and each IM already have trace mapping for users who have ever connected with the VN. The reason for these experiment setups is to take fully into account all the negative aspects of GNF. Because GNF need additional computation resource and storage space to maintain these large-scale mappings, while it has no influence for iMark.

5.1 Mapping Size Evaluation

In mapping size evaluation, we compare the average mapping size of iMark against GNF for scenarios with different levels of VN hierarchy. Two network scenarios are considered based on the different types of VN hierarchical structure: balanced and unbalanced hierarchy. The former is defined as the VNs can form aggregated domain among themselves with the lowest level of hierarchy and only the representative VNs appear at higher hierarchical level; and the latter is defined as VNs can randomly form aggregated domain with VNs (representative VN or ordinary VN) at any level of the hierarchy.

Table 2. Experimental Setup For VN Mapping Size of Balanced and Unbalanced Hierarchy

Scenario A		Scenario B	
Parameter	**Value**	**Parameter**	**Value**
Number of VNs	10000	Number of VNs	10000
Number of Users per VN	5w ~ 10w	Number of Users per VN	5w ~ 10w
Aggregation domain size at level 1	35~45	Aggregation domain size at level 1	200~5000
Aggregation domain size above	2~4	Hierarchy level	8
Hierarchy level	7	Hierarchy type	unbalanced
Hierarchy type	balanced		

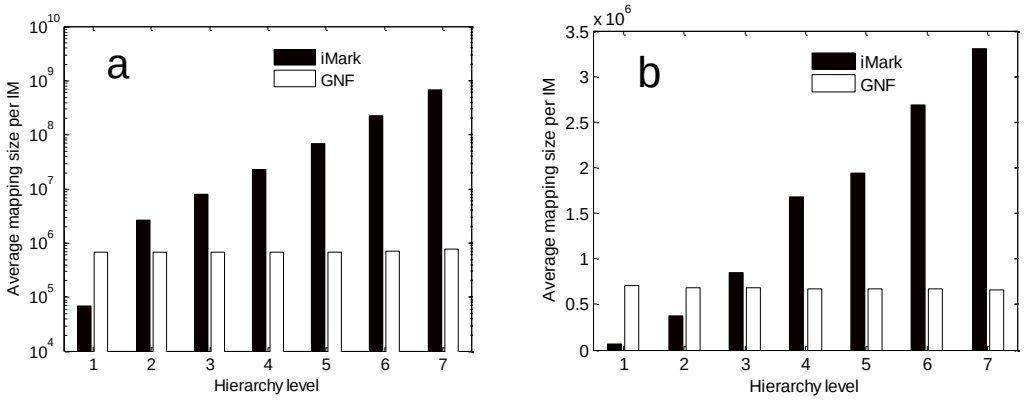

Fig. 4. (a) Scenario A: Average mapping size at different level of a balanced hierarchy; (b) Scenario B: Average mapping size at different level of an unbalanced hierarchy.

For scenario A (see **Table 2**), 10000 VNs are randomly generated to form a 7-level hierarchy with balanced structure where each VN associated with 50000-100000 users. At hierarchical level 1, 35~45 VNs form an aggregated domain and at above hierarchy levels, 2~4 representative VNs aggregate to form a domain. **Fig. 4** (a) shows that at the level 1 (i.e. the lowest level of hierarchy), each VN of iMark only keep its individual user mapping while the mapping size of GNF is significantly more than that of iMark, as GNF not only keep a user name mapping but also hierarchical mapping and Trace mapping. The Trace mapping size grows with the increasing mobility of the end users. At higher levels, mapping size of iMark increases exponentially with the maximum value at the top level 7. On other hand, mapping size of GNF remains almost the same along the level grows. This is because as the level grows, the representative VNs of iMark will record all the mapping information of the sub-level VNs below whereas GNF's mappings remain the same on average.

For scenario B (see **Table 2**), we generate the VN hierarchy by adopting the same set of parameters used in Scenario A, but in an unbalanced structure with 8 hierarchical levels and 200~5000 VNs per aggregated domain at each level of the hierarchy. **Fig. 4** (b) shows the result of average mapping size. Comparing the result for balanced and unbalanced scenarios, it can be seen that with iMark, the mapping size increasing speed of scenario one is higher than that of scenario two. This is mainly because in the unbalanced hierarchical structure, VNs can aggregate at higher level hierarchy which can bring down the average mapping size at higher levels. With GNF, the mapping size at level 1 is larger than that of iMark due to maintaining additional mapping (hierarchical mapping and trace mapping), however as the hierarchical level gets higher, the mapping size of iMark exceeds that of GNF rapidly. These two scenarios demonstrate the GNF effectiveness for both balanced and unbalanced structures.

5.2 Lookup Operation Evaluations

This section assesses the *lookup* operation performance for GNF by examining the average *lookup* operation time against iMark. *Lookup* operations are taken when setting up a communication connection between the end users from different VNs at different hierarchical locations. We define the hop count as the number of IMs between users on the end-to-end communication path. With iMark, the *lookup* operation searches its own IM first, if not found, forwards the lookup request to the VNs of the same aggregated domain before resorts to the VNs in the upper hierarchical level. For GNF, it provides a quicker two-step *lookup* operation which locates the target VN before searching the end user as described in Section 4.

The *lookup* operation is assessed through three different network scenarios (C, D, and E) as shown in **Table 3**. For scenario C, 32 VNs are generated with 5000~15000 users in each VN which are randomly organized in a 5-level hierarchical structure. The size of the aggregated domain is 1~4 at each level. The time consumed for accomplishing a *lookup* operation is measured between two randomly selected users. In the experiments, we repeat this process 5000 times and classify the measured time records in terms of hop counts. **Fig. 5** (c) shows the simulation result indicating that the average time for *lookup* operation remains almost the same in GNF in spite of the different hop counts. While the average time of iMark increases significantly along with increase of hop counts. It is noted that at the hop count of 1, although they both search within their native VN, iMark takes longer average lookup time in case it is a representative VN which contains all VN user mappings at its sub-levels.

In scenario D, 10000 VNs are randomly generated to form a 5-level hierarchy in a balanced structure. For each aggregated domain, its average size is about 10 VNs per domain, and for each VN, it contains about 500~1500 users. We simulate 5000 *lookup* operations between a certain user in a specified VN and users randomly selected in other VNs in the hierarchical structure with different hop counts. The result of average lookup time is shown in **Fig. 5** (d). It shows that the lookup time increases exponentially for iMark, while the time for GNF is minimal due to the fact that iMark spends more time querying huge number of VNs on the searching path as the hop count increases.

Table 3. Experimental Setup for Lookup operation for scenario c, d and e

Parameter	Scenario A Value	Scenario B Value	Scenario C Value
Number of VNs	32	10000	5000
Number of Users per VN	5000 ~ 15000	500 ~ 1500	1500~2500
Aggregation domain size	1~4	10	500
Hierarchy level	5	5	11
Hierarchy type	unbalanced	balanced	unbalanced
Total Lookup repeat times	5000	5000	11000

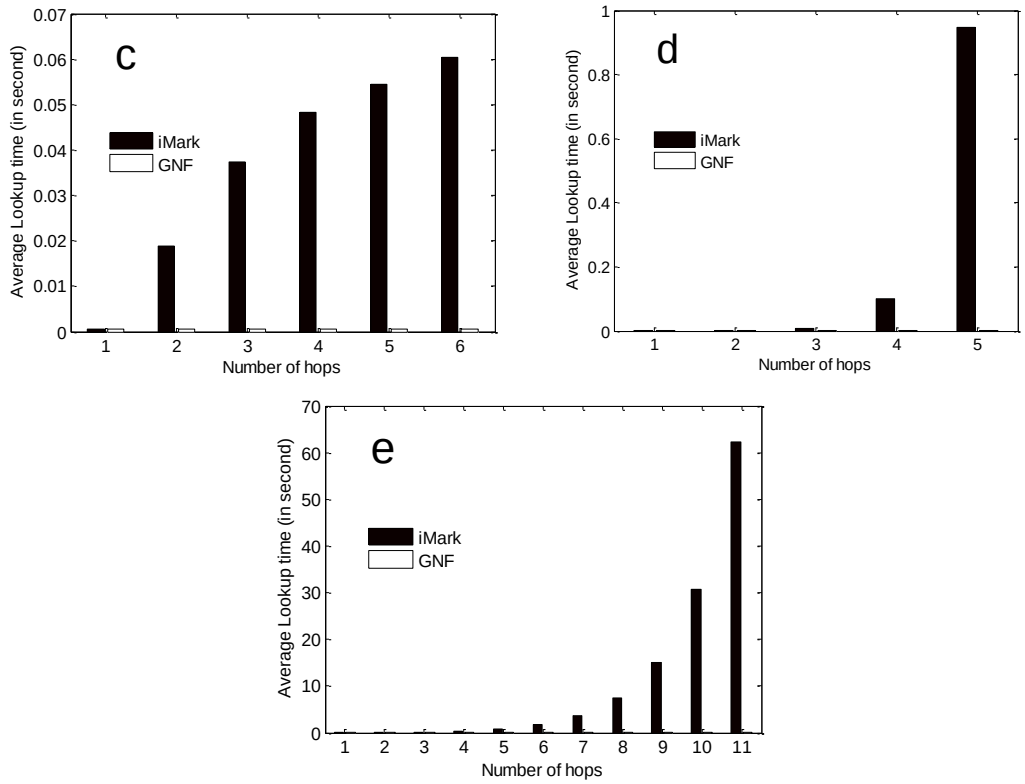

Fig. 5. The average Lookup times of Scenario C, D and E.

In scenario E, 5000 VNs are generated with 500 VNs on average per aggregated domain which forms an 11-level hierarchical structure with each VN of 1500~2500 users. We construct an unbalanced hierarchical structure of 11 levels. In the experiments, 110,000 *lookup* operations are simulated from a user to other users in the hierarchy with different hop counts, as shown in **Fig. 5** (e). The result confirms the results and meets our expectation that a significant performance improvement in terms of lookup time can be achieved by GNF in comparison with iMark.

Through the performance evaluation carried out by the numerical experiments with different scenarios, it can be seen that the average mapping size is smaller than that of iMark and the average lookup time is much shorter than iMark. The result clearly confirmed that the suggested naming solution can perform well with better scalability and low complexity than the exiting solutions.

5. Conclusion and Discussion

In this paper a novel global naming framework, GNF, is presented which is dedicated for user identification in the network virtualization environment. The architectural and operational details of the proposed solution are provided and its performance is assessed through numerical simulations. Its scalable design aims to enable flexible and transparent end-to-end communication across multiple VNs, in the presence of dynamic behaviors from both VN and end users in NVE. The results demonstrate its enhanced scalability in respect to a recent solution as the operational complexity (space and time) can be significantly reduced, indicating its great potential to be adopted in large-scale NVE. The key findings from this

work can be summarized as follows: it is noted that the saving of lookup time through trace mapping in GNF can be achieved at the price of extra space at each IM to maintain the relevant mapping information and time for trace mapping update. The mobility of users in a NVE will increase the trace mapping size in IMs. Therefore, a tradeoff needs to be properly made between the communication setup time and the extra cost for maintaining trace mapping.

In respect to the future work, several research directions are of particular interests. One aspect is to design a naming framework prototype based on the proposed GNF solution and assess its performance through more extensive experiments in network test-bed with a particular focus on scenarios with mobile and multi-homed users. Moreover, the cooperation and negotiation among IMs in a loosely-coupled network structure need to be investigated to prevent "single-point" failure and cope with a variety of operational uncertainties in NVE.

References

[1] D. Clark, J. Wroclawski, K. R. Sollins and R. Braden, "Tussle in cyberspace: defining tomorrow's internet," in *Proc. of ACM Special Interest Group on Data Communication (SIGCOMM)*, pp. 347-356, August 19-23, 2002.

[2] L. Peterson, S. Shenker, and J. Turner, "Overcoming the internet impasse through virtualization," in *Proc. of 3rd ACM Workshop on Hot Topics in Networks*, pp. 1-6, November 15-16, 2004.

[3] G. Schaffrath, C. Werle, P. Papadimitriou, A. Feldmann, R. Bless, A. Greenhalgh, and A. Wundsam, "Network virtualization architecture: proposal and initial prototype," in *Proc. of 1st ACM SIGCOMM Workshop on Virtualized Infrastructure Systems and Architectures*, pp. 63-72, August 17-21, 2009.

[4] J. Turner and D. Taylor, "Diversifying the internet," in *Proc. of IEEE Global Telecommunications Conference (GLOBECOM)*, pp. 755-760, November 28-December 2, 2005.

[5] GENI: Global Environment for Network Innovations, *http://www.geni.net*.

[6] 4WARD Project, *http://www.4ward-project.eu*.

[7] N. M. K. Chowdhury, F. Samuel, and R. Boutaba "PolyViNE: policy-based virtual network embedding across multiple domains," in *Proc. of 2nd ACM SIGCOMM Workshop on Virtualized Infrastructure Systems and Architectures*, pp. 49-56, August 30-September 3, 2010.

[8] N. M. K. Chowdhury and R. Boutaba, "Network virtualization: state of the art and research challenges," *IEEE Communications Magazine*, vol. 47, no. 7, pp. 20-26, July, 2009.

[9] N. M. K. Chowdhury and R. Boutaba, "A survey of network virtualization," *Computer Networks*, vol. 54, no. 5, pp. 862-876, April, 2010.

[10] N. M. K. Chowdhury, F. E. Zaheer and R. Boutaba, "iMark: An identity management framework for network virtualization environment," in *Proc. of the 11th IFIP/IEEE international conference on Symposium on Integrated Network Management*, pp. 5-12, June 1-5, 2009.

[11] D. R.Cheriton and M. Gritter, "TRIAD: a scalable deployable NAT-based Internet architecture," *Stanford Computer Science Technical Report*, January, 2000.

[12] S. Schmid, L. Eggert, M. Brunner and J. Quittek, "TurfNet: An architecture for dynamically composable networks," *Lecture Notes in Computer Science*, vol. 3457, pp. 211-211, 2005.

[13] H. Balakrishnan, K. Lakshminarayanan, S. Ratnasamy, S. Shenker, I. Stoica and M. Walfish, "A layered naming architecture for the Internet," in *Proc. of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, pp. 343-352, August 30-September 3, 2004.

[14] R. Farha and A. Leon-Garcia, "A novel peer-to-peer naming infrastructure for next generation networks," in *Proc. of the 7th IEEE international conference on IP operations and management*, pp. 1-12, October 31- November 2, 2007.

[15] Y. Che, Q. Yang, C. Wu and L. Ma, "BABAC: An access control framework for network virtualization using user behaviors and attributes," in *Proc. of the 2010 IEEE/ACM Int'l Conference on Green Computing and Communications*, pp. 747-754, December 18-20, 2010.