iMark: An Identity Management Framework for Network Virtualization Environment

N. M. Mosharaf Kabir Chowdhury
Cheriton School of Computer Science
University of Waterloo
Waterloo, Canada
Email: nmmkchow@uwaterloo.ca

Fida-E Zaheer
Cheriton School of Computer Science
University of Waterloo
Waterloo, Canada
Email: fzaheer@uwaterloo.ca

Raouf Boutaba
Cheriton School of Computer Science
University of Waterloo
Waterloo, Canada
Email: rboutaba@uwaterloo.ca

***Abstract*—In recent years, network virtualization has been propounded as an open and flexible future internetworking paradigm that allows multiple virtual networks (VNs) to co-exist on a shared physical substrate. Each VN in a network virtualization environment (NVE) is free to implement its own naming, addressing, routing, and transport mechanisms. While such flexibility allows fast and easy deployment of diversified applications and services, ensuring end-to-end communication and universal connectivity poses a daunting challenge.**

This paper advocates that effective and efficient management of heterogeneous identifier spaces is the key to solving the problem of end-to-end connectivity in an NVE. We propose *iMark*, an identity management framework based on a global identity space, which enables end hosts to communicate with each other within and outside of their own networks through a set of *controllers*, *adapters*, and well-placed *mappings* without sacrificing the autonomy of the concerned VNs. We describe the procedures that manipulate these mappings between different identifier spaces and provide performance evaluation of the proposed framework.

I. Introduction

Recently the concept of network virtualization has attracted considerable attention in the debate on how to model the next-generation internetworking paradigm that can replace the existing Internet. Architectural purists view network virtualization as a tool for evaluating new architectures, whereas pluralists conceive virtualization as a fundamental attribute of the next-generation architecture itself [1]. They believe that network virtualization can eradicate the *ossification* of the current Internet and stimulate innovation [1], [2].

To introduce flexibility, separation of policy from mechanism is a well-known principle in computing literature. Network virtualization takes a similar approach [2], [3] by dividing the role of the traditional ISPs into two: *infrastructure providers* are in charge of the physical networks; *service providers* deploy customized VNs by aggregating resources from multiple infrastructure providers and provide end-to-end services to end users. Moreover, network virtualization allows each of these physical and virtual networks to implement heterogeneous control and management protocols. But such flexibility does not come without cost; due to the potential heterogeneity of the networks, end-to-end communication becomes almost impossible in an NVE.

We believe that the first logical step toward universal connectivity is to make heterogeneous namespaces[1] (or identifier spaces) in different physical and virtual networks interoperable. Once it becomes possible to uniquely identify and locate the end hosts irrespective of their physical and logical locations, enabling end-to-end communication boils down to creating connections with necessary address/protocol translators in place.

This paper presents iMark, an identity management framework for network virtualization environment, which focuses on interoperability of heterogeneous identifier spaces. It does not put any restriction on an individual network's choice of local naming mechanism; instead, iMark defines a globally agreed upon identifier space for the end hosts and provides mechanisms to translate back and forth between local and global identifiers through a set of mappings placed in iMark controllers. Such explicit separation of the identity of an end host from its physical and logical locations allows heterogeneous networks to interoperate without sacrificing their autonomy.

The remainder of the paper is organized as follows. In Section II we provide a brief introduction to the network virtualization environment. We present the motivation behind the design of iMark in Section III. In Section IV we describe the design choices and a high-level overview of iMark, followed by a detailed specification of the basic iMark operations in Section V. Section VI presents experimental results from initial evaluation. Section VII summarizes related works, and we conclude the paper in Section VIII.

II. Network Virtualization Environment

The main distinction between the network virtualization model and the existing model of internetworking is the presence of two distinct roles: *infrastructure provider* and *service provider* in the NVE, as opposed to a single role: *Internet Service Provider (ISP)* in the conventional model.

Infrastructure Provider(InP): Infrastructure providers deploy and actually manage the underlying *physical network resources*. They offer their resources through programmable

[1]The words 'name' and 'identifier' are used interchangeably to refer to the identity of an entity throughout this paper.

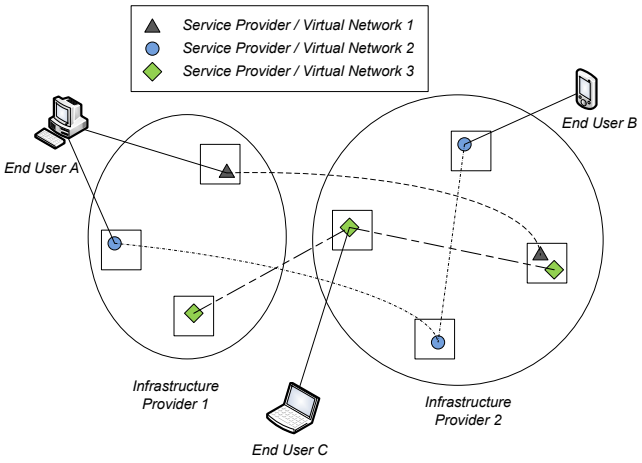

Fig. 1. Network virtualization environment.

interfaces to different service providers. InPs communicate and collaborate among themselves, based on specific agreements, to complete the underlying network. Those offering connectivity to service providers through different networking technologies are known as the *facilities providers*. On the other hand, InPs connecting end user equipments to the core network are the *access providers*.

Service Provider(SP): Service providers lease *virtual resources* from multiple infrastructure providers to synthesize *virtual networks*. They can deploy customized protocols by programming the allocated virtual resources to offer end-to-end services to end users. An SP can also create child VNs by partitioning its resources and lease those child networks to other SPs.

End User: End users in the network virtualization model are similar to their counterparts in the existing Internet, except that the existence of multiple VNs from competing SPs enables them to choose from a wide range of services. Any end user can simultaneously connect to multiple virtual networks from different SPs for different services.

In an NVE, the basic entity is a VN, which is a collection of *virtual nodes* connected together by a set of *virtual links* forming a virtual topology. While each VN is composed and managed by a single SP, it can span over multiple physical networks. Once provisioned, a VN has the semblance of an actual physical network. Fig. 1 depicts three possibly heterogeneous VNs that span over two different InPs.

The owner of a VN, i.e. an SP, is free to implement end-to-end services by selecting custom packet formats, routing protocols, forwarding mechanisms, and other control and management protocols. End users can opt-in to any VN, or even multiple ones at the same time. For example, end user A in Fig. 1 is connected to two different VNs, VN1 and VN2.

III. Motivation

To understand the intricacies of the naming problem in an NVE, consider the following scenario: Alice is the North American continental manager of a corporate giant G with her headquarters located in New York. G maintains separate VNs for every continent and a parent VN connecting all of its continental head-quarters. As a continental manager, Alice holds frequent video conferences with the regional managers of her continent and has to visit different offices occasionally. In addition, she has to participate in monthly meetings with her counterparts in other continents regarding the global objectives and progress of the company. As a result, she must be able to connect to appropriate VNs from her home, local office, or even when she is in transit or in foreign offices. She can try to access a VN using different devices, including personal phone or laptop, or her office desktop and through different access networks. In each case, the corresponding VN must be able to identify her with a single identity irrespective of her physical location, device, or the access network.

Now consider Bob, who is the continental manager of S, the biggest supplier of G. His job requires him to reach Alice from his own VN to wherever she is at a particular moment; not to mention that he himself can connect from different places and with different devices. Since Alice's VN and Bob's VN might not use compatible naming and addressing systems, finding one another in different VNs is not as simple as it is in the existing Internet.

In addition, the InPs that are hosting Alice's and Bob's VNs can also move the virtual nodes of those VNs around, to handle failures, or to upgrade equipments, or for regular maintenance. Even though the physical locations of the virtual nodes are changing, they must maintain their identity to keep themselves reachable from other nodes in the same VN or from other VNs.

Therefore, the naming requirements for an NVE boils down to something that will be able to handle the following phenomena:

A. Dynamism in an NVE

Network virtualization introduces a dynamic environment at all strata of networking, which starts from individual end users or network elements and continues up to the level of complete VNs. We can broadly categorize such dynamism into two classes:

Macro Level: VNs providing basic services or VNs with shared interests can be dynamically aggregated together to create compound VNs. This is known as *federation* of VNs. Multiple federations and VNs can also come together to create hierarchy of VNs. Even though the level of dynamism is expected to be very low at this level, the complexity of adding a VN to a collection, or removing one, can be quite high.

Micro Level: This is the more influential of the two classes discussed here and requires more attention. Micro level dynamic behavior can basically be attributed to two broad sets of activities:

- Dynamic join, leave, and *mobility* of end users within and in between VNs.
- Dynamism incurred by the migration of virtual routers for different purposes [4].

Mobility of end users or virtual resources can again be of two types:

- *Geographical mobility* from one physical access network to another (e.g., Alice connecting to her office VN using her laptop from her home, in transit, or from her office)
- *Logical mobility* from one VN to another (e.g., Alice moving from her office VN to an online gaming VN in her spare time)

A naming framework for an NVE must, therefore, be flexible enough to handle such high level of mobility of end users while preserving their identities. Moreover, it should also provide support for federation and hierarchy of VNs to deploy complex end-to-end services.

B. Scale

Every day the number of users is increasing rapidly, and it is expected to continue along this line in the near future. Any new naming infrastructure, whether for an NVE or something else, must be scalable enough to accommodate huge influx of end users.

C. Interactions Between Multiple Heterogeneous Parties

One of the most important issues in an NVE is the way multiple players interact among themselves. Such an interaction can be between two SPs (i.e. VNs), or two InPs, or an SP and an InP, and, in the most trivial form, between an end user and an SP. Moreover, each party can have heterogeneous naming, addressing and routing mechanisms. To identify a particular node (physical or virtual) or an end user in this complex web, a naming framework must be expressive.

D. Über-homing

In an NVE, any end user can simultaneously connect to multiple VNs through multiple InPs using heterogeneous technologies to access different services. We refer to this phenomenon by *über-homing*. Über-homing has significant impact in cross VN routing. In that case, multiple routes might exist to reach a particular node through different VNs and InPs. The decision to prefer one over another can be taken based on the agreements between concerned SPs and InPs. Any naming framework for an NVE must provide additional level of indirection to support über-homing.

IV. iMark Overview

In this section, we discuss the decisions we have made in designing the iMark framework, followed by an architectural overview of its components. A detailed description of how iMark works can be found in the next section.

A. Design Choices

The design choices made for iMark are inspired by the three key tenets of a next-generation architecture described in [5] and aim toward separation of identity and location, isolation of conflicting interests, and minimizing global functionalities.

Separation of Identity and Location: In the existing Internet, IP addresses denote both the identity of a node and its topological location. But mixing identity with location limits host mobility and restricts multihoming among many other problems [5]. Several proposals exist in the literature that separates a host's identity from its location. iMark takes a similar stance with a focus on supporting logical and physical mobility, federation and hierarchy of VNs, and überhoming.

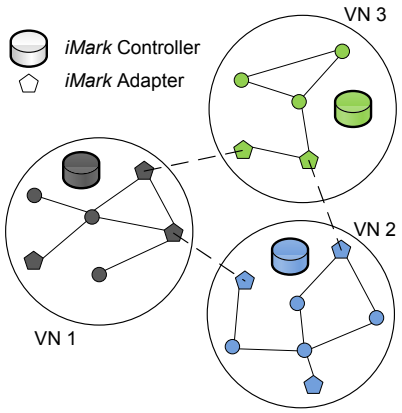

Fig. 2. Overview of the iMark framework.

Local Autonomy: iMark does not impose any requirements or restrictions on individual physical or virtual networks; rather it provides a set of defined interfaces and mechanisms to enable end-to-end connectivity across heterogeneous physical and virtual networks.

Global Identifier Space: Since each VN can implement its own naming mechanism, local identifiers have little end-to-end significance. Hence, in order to provide end-to-end communication between nodes in different VNs, there must be a globally agreed upon identification mechanism. Moreover, in order to ensure trust and security, end hosts must have unique identities that always remain the same, irrespective of whichever VN they are in or however they are connected. This requirement calls for the only globally agreed state in iMark. iMark does not impose any structure on these identifiers though; it only requires them to be unique.

B. iMark Components and Concepts

In order to identify nodes in corresponding VNs and to locate them in the underlying physical networks, iMark defines several entities and corresponding identifier spaces. To enable connectivity between heterogeneous identifier spaces, iMark stores mappings between different identifiers and keeps those mappings updated for address/protocol translation. This allows all the networks to be completely autonomous in their internal choices of naming, addressing, and routing.

Fig. 2 depicts the essential components of iMark, which are discussed in the following:

Controllers: Controllers are logical entities in each VN that provide traditional control functionalities, e.g., address allocation, name resolution etc., along with other network specific additional services. A controller can be centralized (e.g., DNS) or distributed (e.g., DHT) based on the design of its VN.

Adapters: Adapters are special entities that act as gateways between two adjoining VNs. When adjoining VNs use

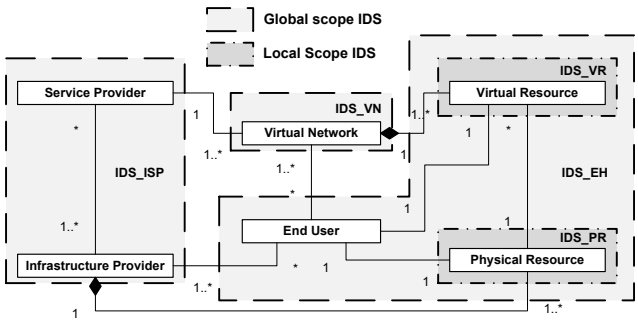

Fig. 3. iMark entities and relationships between them. Shaded rectangles denote the identifier spaces.

different addressing schemes and/or protocol suites, adapters perform required address and protocol translations to relay traffic between them. If both networks use the same mechanism, adapters just forward data without modifying them.

Entities and Identifier Spaces: Given the NVE concepts presented previously (Section II), we identify the major entities that constitute iMark as follows:

1) **Service Provider:** Service providers create and manage one or more VNs by aggregating virtual resources from multiple InPs and provide deployed services to end users based on specific agreements.
2) **Virtual Network:** Any VN is instantiated and managed by a single SP. A VN has a finite timespan associated with it and is dissolved after that period.
3) **Virtual Resource:** Virtual resources belong to a single VN at a given time. Any end user device connected to a particular VN is logically considered to be a virtual resource of that VN.
4) **Infrastructure Provider/Physical Network:** Infrastructure providers are in charge of the underlying networks and all the physical resources contained within them. InPs have one-to-one relationships with the physical networks they manage; hence, they can be considered a single entity.
5) **Physical Resource:** Physical resources are actual network elements, e.g., routers and switches, that host the virtual resources.
6) **End User:** End users connect to VNs provided by different SPs through *access networks* managed by the InPs.

Fig. 3 depicts the relationships between these entities using standard notations.

Based on the proposed entities, we define multiple identifier spaces (IDSes) to provide identifiers for those entities. Each IDS provides different types of identifiers to uniquely identify an entity in particular contexts. We summarize the IDSes below:

1) *IDS_ISP* identifies all the SPs and InPs using unique *g_isp_id* for each one of them. A common IDS for both SPs and InPs enables them to participate in a common environment, e.g., a resource trading marketplace. An *isp_type* is used to differentiate SPs from InPs.
2) *IDS_VN* provides identifiers (*g_vn_id*) for all the virtual networks. Each VN also has a set of characterizing attributes that can be used to search for VNs with particular properties.
3) *IDS_VR* identifies all the virtual resources connected to and contained within a VN using l_vr_id. These identifiers are unique *within* a virtual network. Each virtual resource has an associated *vr_type* that defines whether it is an end user or an actual virtual resource inside the VN.
   If any end user is simultaneously connected to multiple VNs at a particular time, it will have multiple local l_vr_ids. Each VN is free to use its own control and data plane protocols with its own set of l_vr_ids irrespective of other VNs.
4) *IDS_PR* specifies l_pr_id to locally identify physical network elements and connected end user devices. Each physical resource also has a *pr_type* to distinguish between end user devices and internal network elements. If any end user is simultaneously connected to multiple physical networks, i.e., multi-homed, it will have multiple l_pr_ids.
5) *IDS_EH* provides globally unique location-independent identifiers, g_eh_id, for every end user and nodes that a particular network wants to expose to the outside world.

TABLE I
MAPPINGS BETWEEN DIFFERENT IDENTIFIERS

$(\langle from\_id \leftrightarrow to\_id \rangle)$	Purpose
$\langle g\_eh\_id \leftrightarrow l\_vr\_id \rangle$	Identifies any resource within a virtual network and vice versa.
$\langle g\_eh\_id \leftrightarrow l\_pr\_id \rangle$	Identifies any resource within a physical network and vice versa.
$\langle g\_eh\_id \rightarrow g\_vn\_id \rangle$	Stores the virtual network an end host is connected to.
$\langle l\_vr\_id \rightarrow l\_pr\_id \rangle$	Finds the local identifier of the physical host of a virtual resource within a physical network.
$\langle g\_vn\_id \rightarrow \{l\_pr\_id\} \rangle$	Gets the local identifiers of the access nodes of a virtual network inside a physical network.
$\langle g\_vn\_id \rightarrow g\_isp\_id \rangle$	Finds the owner SP of a virtual network.
$\langle g\_vn\_id \rightarrow \{g\_isp\_id\} \rangle$	Obtains the set of InPs that host the virtual network in the underlying network.

Mappings: In order to locate all the entities in an NVE and to route to their current locations based on their global identifiers, a set of mappings between different IDSes are required. Mappings are stored at controllers and updated based on micro-level events (e.g., node join, leave, and mobility) as well as macro-level ones (e.g., VN creation, expiration etc.). Table I presents a list of mappings required by iMark.

Federation and Hierarchy of iMark Controllers: Federation allows multiple autonomous VNs in an NVE to connect

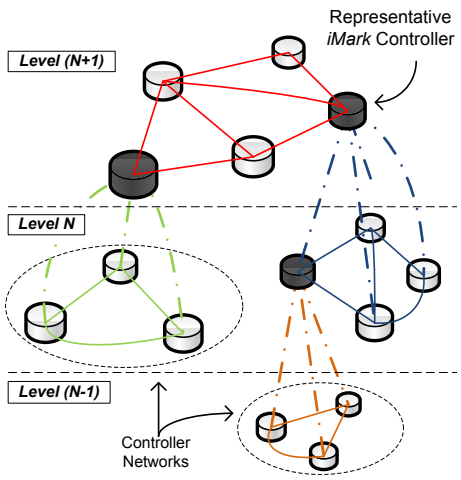

Fig. 4. Federation and hierarchy of iMark controllers.

and logically merge together to provide end-to-end services. An example of federation is the peering relationship between Alice's VN and Bob's VN described in Section III. iMark creates a common control space, known as *controller network*, connecting controllers of each of the participant VNs to support federations. The controller network can itself be a VN.

Multiple federations and VNs can also create a logical hierarchy of VNs for different reasons. For example, Alice's corporation G in Section III has created a hierarchy of VNs for administrative purposes. iMark proposes the concept of *representative controllers* of federations to support such *controller hierarchy*. A representative controller can either be an elected member of the federation, or it can be a separate entity altogether. Each representative controller has knowledge of all the end hosts that belong to any of the VNs in its subtree.

V. iMark Operations

Due to the autonomy and potential heterogeneity of the VNs in an NVE, ensuring end-to-end connectivity across VN boundaries is a nontrivial task. Since local identifiers (l_vr_id) are not meaningful outside a VN's domain, connectivity is provided based on the global identifiers (g_eh_id) of the end hosts using different iMark mappings mentioned earlier. In order to create these mappings, a joining procedure is required that binds end hosts to physical access networks as well as to VNs of their choice. To communicate across VN boundaries, an explicit connection setup procedure is followed that looks up the destination host and sets up relaying states. This section describes these basic procedures along with the compound ones like *überhoming* and *mobility* handling.

A. Macro Level Operations

In order to let end hosts join different VNs and communicate between themselves, VNs must be instantiated first. Here we briefly describe how SP, InP, and VN specific mappings accommodate VN instantiation as well as formation of federation and hierarchy in an NVE. How VNs are provisioned before being instantiated is out of the scope of this paper.

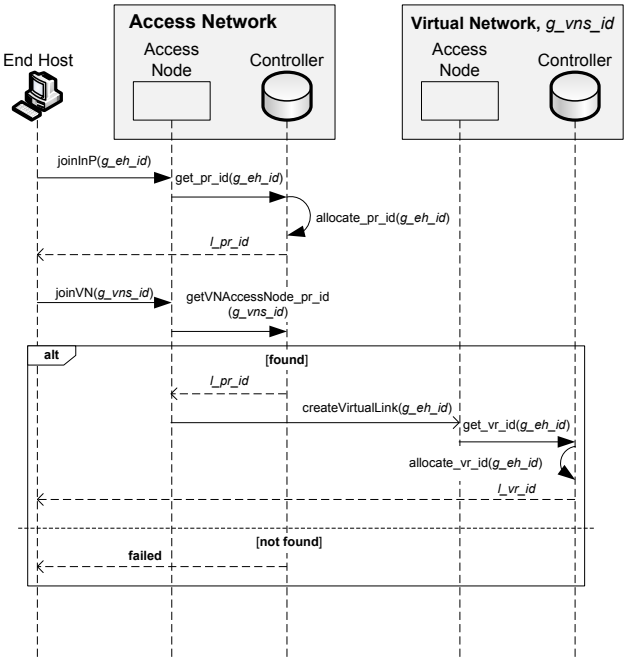

Fig. 5. Sequence diagram: join.

It is well understood that in order to create VNs, SPs and InPs must have a common marketplace to trade resources [6] which itself can be a VN. iMark provides globally unique identifiers (g_isp_id) for the SPs and the InPs to participate in such an environment. We refer to this VN as the *administrative VN* and its controller as the *administrative controller*.

When an SP wants to create a VN, it contacts one or more InPs and provides its requirements. Once a VN is provisioned and instantiated, it is assigned a unique identifier (g_vn_id) which is used for later identification, e.g., during the join operation. Two mappings, $\langle g\_vn\_id \rightarrow g\_isp\_id \rangle$ and $\langle g\_vn\_id \rightarrow \{g\_isp\_id\} \rangle$ are stored in the administrative controller at this point; the first one identifies the owner of the VN and is required to form federations; the later identifies the InPs that host the VN's resources and is used by the InPs to setup cross-InP virtual links for the VN. Physical networks store $\langle l\_vr\_id \rightarrow l\_pr\_id \rangle$ mappings to route in the underlay to create virtual links for the VN request.

Each VN has several access nodes that are located in different physical access networks and are used as gateways to that VN. Access networks store information about these access nodes using $\langle g\_vn\_id \rightarrow \{l\_pr\_id\} \rangle$ mapping and use this mapping during the join procedure.

B. Join

In order to connect to a VN, an end host must join a physical access network first. The controller of the access network assigns and stores an l_pr_id corresponding to the g_eh_id of the end host based on its naming and addressing mechanism. This l_pr_id will be used in the underlay to create virtual links between the end host and VN access nodes.

Next, the end host provides a globally unique identifier of

the VN (g_vn_id) it wants to connect to. The access network finds out the l_pr_id of the access node of that particular VN and forwards the request. The controller of the VN then assigns the end host an l_vr_id, stores a $\langle g\_eh\_id \leftrightarrow l\_vr\_id \rangle$ mapping, and assimilates the end host.

Fig. 5 provides further details of the joining procedure.

C. Lookup and Connection Setup

When an end host e_s wants to initiate communication with another end host e_d, it gives the global identifier of e_d (g_eh_id) to its local iMark controller with a request to setup a connection. The controller first looks up its tables to see whether e_d belongs to its own network. In that case, it sets up a connection and returns the l_vr_id of e_d to e_s

If e_d does not belong to the same VN and the VN belongs to a VN federation or hierarchy, the controller communicates with other controllers in the controller network (first horizontally, then vertically toward the topmost level of VN hierarchy). If any controller can resolve g_eh_id, it returns a positive response to the originating controller. Consequently, a cross VN connection is setup by creating necessary states in the inter-VN adapters.

Since lookup is expensive, after every lookup operation, the originating VN's controller caches the $\langle g\_eh\_id \rightarrow g\_vn\_id \rangle$ mapping for a certain time period as a performance optimization measure.

In case e_d is simultaneously connected to multiple VNs, one is chosen as the destination VN based on inter-VN agreements and VN-specific policies.

Fig. 6 depicts the lookup and connection setup procedure using a sequence diagram. Note that, searching in the VN hierarchy as well as the destination host is omitted from the diagram for brevity.

D. Leave

Whenever an end host wants to leave a VN, it notifies the concerned controller, and all the corresponding mappings stored during the join procedure are removed. In addition, controllers can implement heart beat protocols to periodically check the availability of the connected virtual resources. It also allows controllers to handle failures as a normal leave events.

E. Über-homing

When an end host is über-homed, it can have multiple l_vr_ids in each of the connected VNs along with multiple l_pr_ids, if necessary, in each of the access networks it used to connect to those VNs. Unlike the multihoming scenario in the existing Internet where different IP addresses might be assigned to the same node by different ISPs, in an NVE each end user has a unique identifier g_eh_id which is free from its logical and physical location.

Once an end-to-end connection to an end host is setup through a particular pair of physical and virtual networks, l_vr_id and l_pr_id corresponding to that g_eh_id in those physical and virtual networks are used to locate the end host and to perform routing.

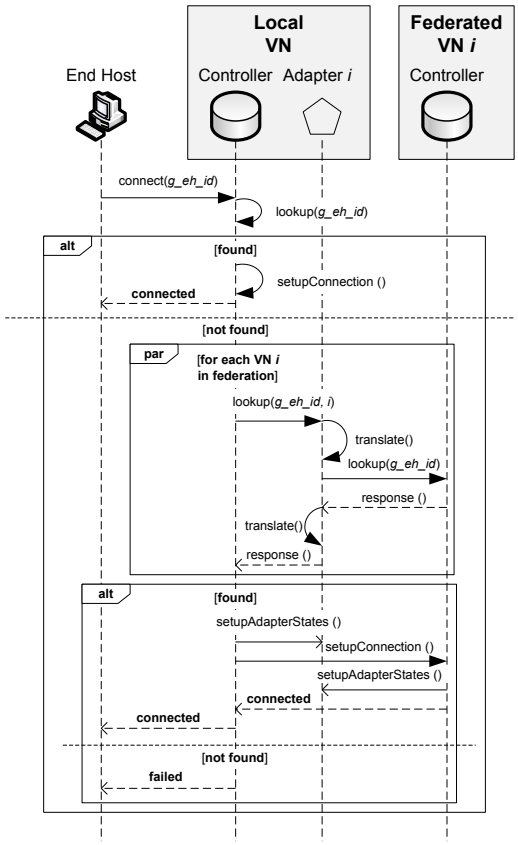

Fig. 6. Sequence diagram: connecting to an end host.

F. Mobility

As mentioned earlier, mobility in an NVE can be of two types: geographical mobility of the end host physical devices from one access network to another, and logical mobility of the end hosts from one VN to another. iMark supports both types of mobility through necessary manipulations of the related mappings, with some assistance from the überhoming capability of the NVE.

In case of geographical mobility, an end host moves from one access network to another by *soft handoff*. First, it *joins* the new access network without leaving the old one and gets a new l_pr_id. Then it requests the new access network to create a connection to the same VN that it is already connected to through the old access network. When the controller of the VN gets the new request, it updates its $\langle g\_eh\_id \leftrightarrow l\_vr\_id \rangle$ mapping with a new l_vr_id based on the l_pr_id of the new access network. The end host finally leaves the old access network to complete a seamless transition.

Logical mobility can be handled by a simpler two step process: *leave* from the old VN, and then *join* a new one.

VI. Performance Evaluation

iMark can face performance challenges from two main sources: size of the mappings stored at different controllers, and lookup frequency at different levels of the controller

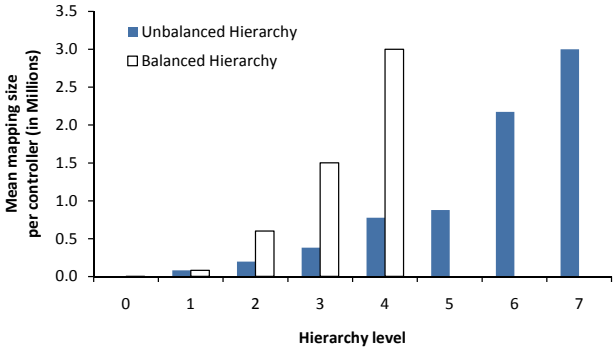

Fig. 7. Mean mapping size per controller at different levels of balanced and unbalanced controller hierarchies.

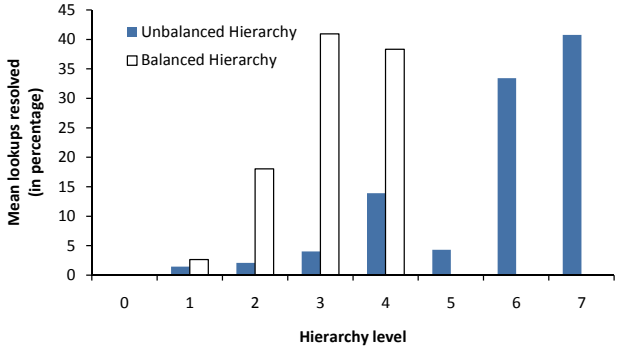

Fig. 8. Mean lookups resolved at different levels of balanced and unbalanced controller hierarchies.

hierarchy. In this section, we evaluate iMark's performance in both cases through simulation.

A. Experimental Setup

Since the existing testbeds and popular network simulation tools do not support network virtualization concepts, we developed a simulator and used a quad CPU Sun V440 Server with 8GB of memory to perform the numerical simulations. In order to explore the problem space, we ran a large set of experiments by varying the total number of VNs from 1000 to 15000, the size of federations from 30 to 300, and the total number of end hosts from 10 thousand to 10 million. To explore the impact of controller hierarchies, we experimented with two different types of hierarchies: in a balanced hierarchy, we allowed VNs to form federations only among themselves; whereas, in an unbalanced hierarchy, we let VNs join randomly at any level of the hierarchy. Of the two, balanced ones resulted in shorter hierarchies.

Our main goal was to show iMark's correctness and to provide an indication of its performance trends in heterogeneous NVE. Hence, we did not put any restrictions on the use of any particular protocol or algorithm in individual VNs. Instead, we focused on two basic operations: *join* and *lookup* from Section V. The *leave* operation is simply removing entries from the mapping tables of the controllers in the hierarchy and therefore has little impact on the performance and scalability of the system. And mobility of the end hosts is a combination of join and leave operations. For simplicity, we did not employ any optimization, e.g., caching, and did not consider überhoming of end hosts. However, the experimental setup can be extended in the future to handle this operation.

After running a large set of experiments by varying different parameters, we observed definite trends in the size of the mappings stored and the lookup frequency. We picked one representative result of each case to discuss our findings.

B. Mapping Size

When considering the total amount of mapping information stored in a representative controller, the contribution of $\langle g\_eh\_id \rightarrow g\_vn\_id \rangle$ easily dominates the rest, since each representative controller aggregates this mapping from all of its child controllers. We, therefore, focused on finding out how the size of this mapping increases as we move upward in the controller hierarchy.

Fig. 7 depicts the simulation results showing the mean mapping size per controller at different levels of the controller hierarchy, level 0 being the lowest level consisting only of individual VNs. For this particular experiment, we considered 3000 VNs with an average of 80 VNs per federation, and 3 million end hosts with an average of 1000 end hosts per VN.

As expected, the size of the mapping increases gradually from lower to higher levels of the hierarchy with the topmost level controllers having the maximum. Since the unbalanced hierarchy has more levels than its balanced counterpart, each level has fewer participating VNs and federations; this results in a greater number of controllers, each with smaller loads.

C. Lookup

Whenever there is a lookup request, a controller first tries to resolve it using its own mapping information. In case of a failure, it forwards the request to its peers in the federation before resorting to the upper level controllers. The more requests a controller forwards to the upper level, the higher the number of messages generated. This also results in higher lookup resolution time. So we examined the percentage of lookup requests that are resolved at different levels of the controller hierarchy to gain an insight into the performance of iMark.

Fig. 8 presents the simulation results showing the mean percentage of lookups resolved at different levels of the controller hierarchy after 1 million lookup operations. For this experiment, we considered 3000 VNs with an average of 80 VNs per federation, and 50000 end hosts.

Since the upper level controllers store more information, a large percentage of the lookup requests find their way to the top two layers of the hierarchy. The smaller height of the balanced variant gives it a competitive advantage over its unbalanced counterpart, because in this case requests can reach the topmost levels faster.

A sudden decrease in the lookup resolution percentage is observed in level 5 of the unbalanced hierarchy. Due to the randomness in the formation phase, we believe that in

this particular instance, more individual VNs than federations ended up forming the federations of level 5, which resulted in less information being stored at this level on the average.

VII. Related Work

To the best of our knowledge, there is no existing work in the literature that addresses the exact problems posed by the unique naming and addressing requirements of an NVE. But there are several proposals that share some common aspects with our one and have influenced as well as motivated us.

Similar to our proposal, TRIAD [7] uses location independent identifiers instead of addresses for node identification. But TRIAD relies on the presence of IPv4 in all network domains, which is completely in contrast with the basic requirement of heterogeneity. In addition, TRIAD's dependence on semantics and hierarchy of domain names is completely opposite to our choice of flat global identifier space.

Plutarch [8], on the other hand, provides explicit support for heterogeneity through the concept of *contexts* and uses *interstitial functions* to translate communication between them, which is similar to our proposal. But Plutarch does not consider mobility of end hosts between multiple contexts and überhoming.

IPNL [9] and 4+4 [10] try to isolate independent IP-based networks through loose integration. IPNL provides three stage communication path consisting of originating and terminating *private realms* and a global *middle realm*. 4+4 generalizes it by supporting multiple middle realms. But both schemes primarily focus on the address depletion problem faced by the existing Internet and are not concerned about the requirements of the NVE.

TurfNet [11], [12] is the most closely related proposal to our work in the existing literature. Conceptually, it supports heterogeneous autonomous network domains; separation of identifiers from locators; encapsulation of internal naming, addressing and routing mechanism and policies of an autonomous domain; and dynamic network composition (vertical and horizontal). But since TurfNet does not consider network virtualization, it is free from the issues arising from InP-SP interactions. Also it does not consider mobility of end hosts and virtual resources.

In addition, our work is motivated by the recent works on flat identifiers and location independent (or identity-based) naming and routing mechanisms [13]–[16]. Last but not the least, our mapping mechanism and identifier space selection was highly influenced by the P2P-based naming architecture proposed in [17] for autonomic networks.

VIII. Conclusion

In this paper, we have presented iMark, a novel identity management framework for network virtualization environment. iMark manages identifiers for entities at different levels: at macro level, it assists creation of independent VNs and formation of VN federations and hierarchy of VNs by ensuring cooperation between SPs and InPs; whereas, at micro level, iMark enables end-to-end communication between end hosts in different VNs. iMark separates identity of the end hosts from their physical and logical locations, and with the help of a global identifier space, it provides universal connectivity without revoking the autonomy of the concerned physical and virtual networks.

To demonstrate iMark's correctness and to provide an indication of its performance, we have done a simulation-based study of the framework. Current experience with iMark suggests that it can indeed enable end-to-end connectivity in a highly heterogeneous NVE.

In an ongoing effort, we are investigating additional performance and robustness aspects of iMark along with possible optimizations, e.g., caching, to improve its scalability. Our future work includes developing a working prototype of iMark to evaluate the runtime performance of lookup and join operations, and to study the effects of end host mobility and überhoming on iMark in a heterogeneous NVE.

Acknowledgment

This research was supported by WCU (World Class University) program through the Korea Science and Engineering Foundation funded by the Ministry of Education, Science and Technology (Project No. R31-2008-000-10100-0).

References

[1] T. Anderson, L. Peterson, S. Shenker, and J. Turner, "Overcoming the Internet impasse through virtualization," *Computer*, vol. 38, no. 4, pp. 34–41, 2005.

[2] J. Turner and D. Taylor, "Diversifying the Internet," in *GLOBECOM'05*, vol. 2, 2005.

[3] N. Feamster, L. Gao, and J. Rexford, "How to lease the Internet in your spare time," *SIGCOMM CCR*, vol. 37, no. 1, pp. 61–64, 2007.

[4] Y. Wang, E. Keller, B. Biskeborn, J. van der Merwe, and J. Rexford, "Virtual routers on the move: Live router migration as a network-management primitive," in *ACM SIGCOMM*, 2008, pp. 231–242.

[5] D. D. Clark, K. Sollins, J. Wroclawski, and T. Faber, "Addressing reality: An architectural response to real-world demands on the evolving Internet," *SIGCOMM CCR*, vol. 33, no. 4, pp. 247–257, 2003.

[6] D. Hausheer and B. Stiller, "Auctions for virtual network environments," in *Workshop on Management of Network Virtualisation*, 2007.

[7] M. Gritter and D. Cheriton, "An architecture for content routing support in the Internet," in *USITS*, 2001, pp. 37–48.

[8] J. Crowcroft, S. Hand, R. Mortier, T. Roscoe, and A. Warfield, "Plutarch: An argument for network pluralism," in *FDNA*, 2003, pp. 258–266.

[9] P. Francis and R. Gummadi, "IPNL: A NAT-extended Internet architecture," in *ACM SIGCOMM*, 2001, pp. 69–80.

[10] Z. Turányi, A. Valkó, and A. T. Campbell, "4+4: An architecture for evolving the Internet address space back toward transparency," *SIGCOMM CCR*, vol. 33, no. 5, pp. 43–54, 2003.

[11] S. Schmid, L. Eggert, M. Brunner, and J. Quittek, "TurfNet: An architecture for dynamically composable networks," in *Proceedings of the First IFIP TC6 WG6.6 International Workshop on Autonomic Communication (WAC'04)*, 2004.

[12] J. Pujol, S. Schmid, L. Eggert, M. Brunner, and J. Quittek, "Scalability analysis of the TurfNet naming and routing architecture," in *DIN*, 2005.

[13] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana, "Internet indirection infrastructure," in *ACM SIGCOMM*, 2002, pp. 73–88.

[14] H. Balakrishnan, K. Lakshminarayanan, S. Ratnasamy, S. Shenker, I. Stoica, and M. Walfish, "A layered naming architecture for the Internet," in *ACM SIGCOMM*, 2004, pp. 343–352.

[15] M. Caesar, M. Castro, E. B. Nightingale, G. O'Shea, and A. Rowstron, "Virtual ring routing: Network routing inspired by dhts," in *ACM SIGCOMM*, 2006, pp. 351–362.

[16] M. Caesar, T. Condie, J. Kannan, K. Lakshminarayanan, and I. Stoica, "ROFL: Routing on flat labels," in *ACM SIGCOMM*, 2006, pp. 363–374.

[17] R. Farha and A. Leon-Garcia, "A novel peer-to-peer naming infrastructure for next generation networks," in *IPOM*, 2007, pp. 1–12.