NVP2P: Network Virtualization from Peer-to-Peer Perspective

N.M. Mosharaf Kabir Chowdhury

David R. Cheriton School of Computer Science
University of Waterloo
nmmkchow@cs.uwaterloo.ca

Abstract. Separation of policy from mechanism is a well-known principle in computing literature. Network virtualization reincarnates this very concept in the context of networking architecture. But most of the existing works either tangentially touch the issues regarding the implementation of such architecture or just ignore them completely. In this article we present NVP2P, a network virtualization framework that examines the possible use of the available P2P concepts in this respect. We also provide a brief introduction to the basics of network virtualization and present several scenarios that justify our choice of using P2P concepts in network virtualization.

1 Introduction

In recent years, the concept of network virtualization has attracted a fair share of attention in the debate on how to model the next-generation networking paradigm that can replace the existing Internet. Architectural purists view network virtualization as a tool for evaluating new architectures, while the pluralists conceive virtualization as a fundamental attribute of the next-generation architecture itself [1]. They believe that network virtualization can eradicate the so-called *ossifying forces* of the current Internet that has restricted changes to incremental updates, and consequently stimulate innovation [1, 2].

To introduce flexibility, separation of policy from mechanism is a well-tested principle. Similar approach has been pushed forward by the proponents of network virtualization [2, 3]. In this case, the role of traditional ISPs has been divided into two. *Infrastructure providers* will be in charge of the physical networks, while *service providers* will deploy customized network protocols on virtual networks by aggregating resources from multiple infrastructure providers, and offer end-to-end services to end users. Such an environment will foster implementation and deployment of multiple coexisting heterogeneous network architectures that are not bounded by some inherent limitations found in the existing Internet.

But to enforce such separation and to manage all the derived complexities (with respect to instantiation, operation, and management of virtual networks) in the real world is a daunting challenge. Existing works are mostly limited to generating buzz for network virtualization; very little work exist that actually talks about how to implement network virtualization and how to handle its challenges. From our research, we have identified several aspects of a *network virtualization environment (NVE)* that are somewhat similar to P2P phenomena, specially the micro level dynamism of virtual resources and end users due to mobility and migration. As a result, we have focused on a unified NVP2P framework that examines the possible use of the available P2P concepts in the context of an NVE.

The rest of this article is organized as follows. First, in Section 2, we provide an introduction to the basics of network virtualization concept. Following that, we identify the motivation of using P2P concepts in an NVE in Section 3. Finally, in Section 4, we present the NVP2P framework in details. We conclude in Section 5.

2 Network Virtualization

Based on the pluralist approach, we define network virtualization as an integral part of the diversified Internet architecture, which supports multiple coexisting heterogeneous network architectures from different service providers, sharing a common physical substrate managed by multiple infrastructure providers. By decoupling service providers from infrastructure providers, network virtualization introduces flexibility for innovation and change.

In this section, we cover the basics of network virtualization, that are integral to the context of this paper. A detailed survey can be found in [4].

2.1 Model

The main distinction between the players in the network virtualization model and the traditional model is the presence of two different roles: *infrastructure providers* and *service providers*, as opposed to a single role: Internet Service Provider (ISP) in the conventional model. From commercial point of view, this decoupling amortizes high fixed cost of maintaining a physical presence, by sharing capital and operational expenditure across multiple infrastructure providers.

Infrastructure Provider(InP). Infrastructure providers deploy and actually manage the underlying physical network resources. They offer their resources through programmable interfaces to different service providers. Infrastructure providers 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, infrastructure providers connecting customer premise equipments (CPEs) to the core network are the *Access Providers*.

Service Provider(SP). Service providers (SPs) lease resources from multiple facilities providers to create and deploy customized protocols by programming the allocated network resources to offer end-to-end services to end users. A service provider can also create child virtual networks by partitioning its resources, and lease those child networks to other service providers.

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

Broker. The broker plays a pivotal role in the virtual network economy. It acts as a *mediator* between infrastructure providers, service providers, and end users in the virtual network marketplace.

2.2 Illustrative Example

In the network virtualization environment, the basic entity is a virtual network (VN); which is a collection of virtual nodes connected together by a set of virtual links forming a virtual topology. Each such virtual network is composed and managed by a single service provider. Once provisioned, a virtual network has the semblance of an actual physical network. Figure 1 depicts two possibly heterogeneous virtual networks, V1 and V2 created by service providers S1 and S2, respectively. S1 composed V1 on top of physical resources managed by two different infrastructure providers (P1, P2) and provides end-to-end services to end users U2 and U3. S2, on the other hand, deployed its virtual network V2 by combining resources from infrastructure provider P1, with a child virtual network from service provider S1. End users U1 and U3 are connected through V2.

The owner of a virtual network, i.e. a service provider, is free to implement end-to-end services by selecting custom packet formats, routing protocols, forwarding mechanisms, as well as control and management planes. End users can opt-in to any virtual network, and have more options to choose from than in the existing Internet. For example, end user U3 is subscribed to both the virtual networks V1 and V2 provided by S1 and S2, respectively.

3 Motivation

3.1 Resource Trading and Network Virtualization Economics

To sell or resell existing resources, or to create virtual networks by aggregating resources from multiple providers, or to buy services from service providers; infrastructure providers, service providers, and end users must find a common place where all can meet to trade resources. This is known as *network virtualization marketplace* (NVM). All the activities in an NVM are monitored and arbitrated by brokers.

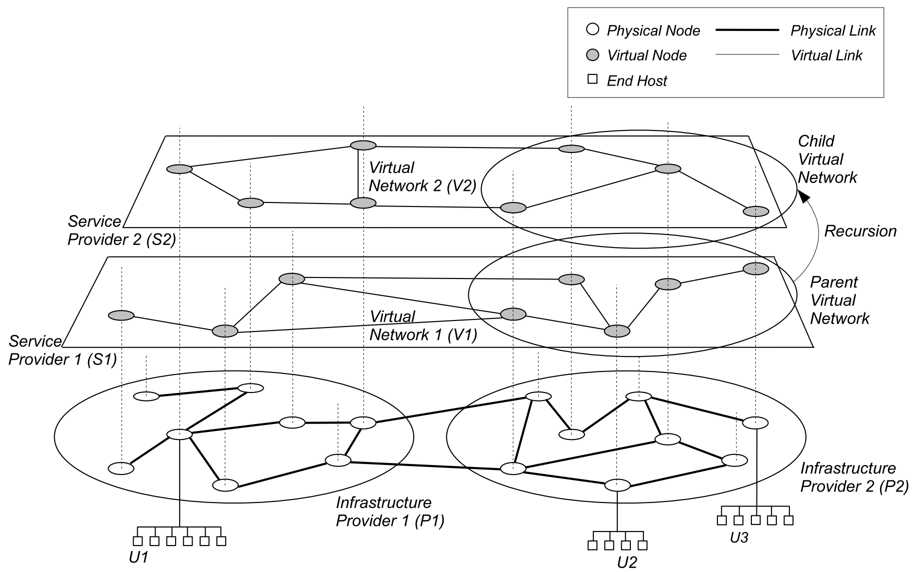

Fig. 1. Network Virtualization Architecture

An NVM infrastructure can be a distributed one, a centralized one, or a hybrid of the two. But any such infrastructure for online resource trading should meet the following two sets requirements [5]:

1. **Functional requirements:**
   - Allow multiple service providers, and infrastructure providers to trade resources
   - On-demand and in-advance trading of resources
   - Support reselling of resources

2. **Performance requirements:**
   - Economically efficient allocation of resources
   - *Robust* against individual failures, and attacks
   - *Scalable* up to a large number of participants

Existing work on resource trading marketplaces based on P2P systems (e.g. PeerMart [6], PeerMint [7], FairPeers [8] etc.) can be a good starting point for the similar problem posed in NVE.

3.2 Interactions Between Multiple SPs and InPs

One of the most important issues in an NVE is the way multiple players interact among themselves. In its simplest form, 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. To maintain such relationships, different agreements are formed between the participating entities. But still it is possible that one party may knowingly or unknowingly step on another, and cause unforeseen consequences. Moreover, when relationships are formed between multiple parties, it is very hard to determine or control the aftermath of such complex engagements.

Similar problems are faced in existing P2P overlays, where overlays and underlays interact with each other in selfish manners more often than not. In such cases, overlays try to optimize themselves, where underlays try to counter such behavior by employing complex traffic engineering solutions. As a result, a lot of works exist that address such interactions between multiple overlays [9], underlays [10], or both [11–13], and try to mitigate the condition [12–14]. Even when no one is behaving selfishly, it is still possible that one might inadvertently affect the performance goals of another [15].

At first thought, it seems that such selfish behavior can not exist in an NVE where resources are statically allocated. But to improve utilization and revenue of InPs, dynamic allocation of resources cannot be avoided. In those cases, the results obtained from the research works on interactions between P2P overlays and underlays can come in handy.

3.3 Dynamism in a Network Virtualization Environment

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 virtual networks. To cope with such dynamism, an NVE must be resilient and flexible to changing conditions. We can coarsely categorize such dynamism into two classes:

Macro Level. Virtual networks providing basic services can be dynamically aggregated and combined together to create compound virtual networks for composite services. Even though the level of dynamism is expected to be very low at this level, the complexity of adding and removing a virtual network to a collection is quite high.

To manage such dynamic creation of compound virtual networks, concepts of hierarchical P2P [16] and DHT-based systems [17–20] can be used.

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 virtual networks
- Dynamism incurred by the migration of virtual nodes and virtual routers for different purposes [21–24]

The dynamic behavior induced by the end users is exactly similar to that of in P2P systems, even though the context is different. As for dynamic behavior caused by the migration of virtual nodes, existing procedures proposed in recent works [25] are complex and require specific series of operations. If such migrations can be handled in the manner of simple peer join and leave events, the complexity can be reduced very sharply.

3.4 More P2P Concepts in Network Virtualization Context

To the best of our knowledge, the only work that has tried to directly relate P2P and virtual networks concept is P2P-XBone [26], an extension to the well known X-Bone framework [27], which itself is based on overlays. In P2P-XBone, the authors have tried to hack X-Bone framework to handle dynamic end user join and departure events and routing mechanism. Other works mostly focus on specific aspects, such as naming, addressing, routing, mobility management etc.

Virtual ring routing or VRR [28] proposes a unique intra-domain routing protocol that is based on DHT concepts, but does not necessarily work as P2P networks do. Instead of creating tunnels between end points, VRR approximates DHT like routing directly on top of link layer. An extension of VRR into inter-domain routing is ROFL [29] which arranges multiple VRRs using hierarchical DHT techniques. The combined result is a complete routing protocol suite that routes based on names or identities without the need of any name to address translation phase.

Authors in [30] proposes a P2P-based naming and mobility management mechanism for Autonomic Service Architecture. They use hierarchical DHT for naming infrastructure and supports horizontal and vertical mobility of customers and networks elements.

P6P [31] is a P2P-based internetworking protocol, mainly focused on provisioning IPv6, that decouples routing infrastructure from end sites.

4 NVP2P Framework

The NVP2P framework is designed to be a generic architecture to handle most, if not all, of the scenarios described in Section 3. Of the aforementioned design goals, we mainly focus on handling the micro level dynamism in this article with a belief that it is the most frequent event with high impact on stability and performance of the NVE.

In an NVE, end users can remain connected to multiple VNs at a time and can also use multiple InPs to connect to those VNs from different locations. As a result, support for über-homing[1] and mobility must be inherent.

NVP2P handles mobility, über-homing and resulting micro-level dynamism of end users and network elements by creating a multi-strata (e.g. local to an

[1] Similar but not exactly same as *multi-homing* as we know today. Über-homing allows end users in an NVE to simultaneously connect to multiple VNs through multiple InPs using heterogeneous technologies to access different services.

SP or InP, global to all the SPs and InPs etc.) architecture. In order to identify nodes in corresponding VNs, and to locate them in the underlying physical network for actually routing packets, we have defined several entities and corresponding identifier spaces (IDSes). To enable heterogeneity among different VNs, our framework focuses on mapping between different identifiers and keeping those mapping updated, leaving the decision of using those identifiers to the SPs and InPs.

4.1 Entities

In order to put together the bits and pieces of the framework we identify the major entities as follows:

1. **Service Providers:** As mentioned earlier, service providers create and manage one or more virtual networks by aggregating virtual resources from multiple InPs, and provide deployed services to end users based on specific agreements.
2. **Virtual Networks/Services:** Any virtual network, and corresponding service deployed on it, is instantiated and managed by a single SP. A virtual network has a finite timespan associated with it, and will be dissolved once its over.
3. **Virtual Resources:** Virtual resources belong to a single virtual network at a given time. We consider any end user device as a virtual resource of a VN, once it is connected to a particular VN.
4. **Infrastructure Providers/Physical Networks:** Infrastructure providers are in charge of the underlying networks and all the physical resources contained within them. Since InPs have a one-to-one relationship with the physical networks they manage, we consider both to be single entity for simplicity.
5. **Physical Resources:** Physical resources are actual routers, switches, and other network elements, as well as the end user physical devices once they connect to the access network.
6. **End Users:** End users connect to virtual networks provided by different SPs through *access networks* managed by the InPs.

Figure 2 depicts the relationships between these entities. We believe the figure itself is quite self-explanatory; hence, we omit details for brevity.

4.2 Identifier Spaces

Once the entities have been figured out, we define multiple identifier spaces (IDSes) to identify those entities. Each IDS provides different types of identifiers to uniquely define an entity in different contexts. The framework works based on the interactions between the entities and corresponding identifiers. We summarize the identifier spaces below:

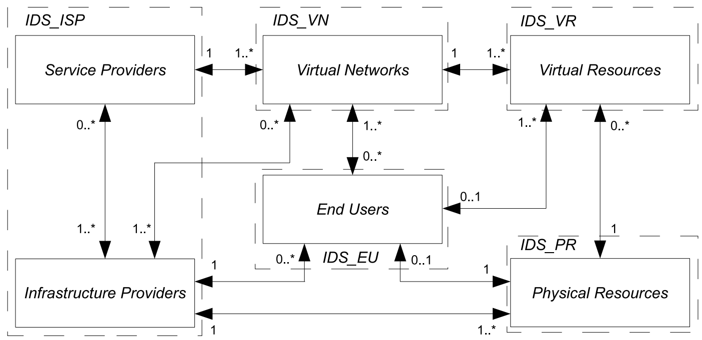

Fig. 2. NVP2P entities and relationships between them. Dashed lines mark the scope of the identifier spaces.

1. *IDS_ISP* identifies all the SPs and InPs using unique *g_isp_id* for each of them. We use a common IDS for both SPs and InPs to accommodate them in a common environment, e.g. a resource trading marketplace. An *isp_type* is used to differentiate SPs from InPs. All the SPs and InPs have representative *agents* participating in that globally shared P2P-like environment to exchange information for trading resources; to enable communication between nodes and end users; and to collaborate for creating end-to-end VNs.
2. *IDS_VNS* provides identifiers (*g_vns_id*) for all the virtual networks, and services deployed on them. The basic assumption is that only one service is deployed on a particular VN and vice versa. 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 virtual network 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 network.

   If any end user is simultaneously connected to multiple VNs at a particular time, it will have multiple *l_vr_id*. Each VN is free to use its own control and data plane protocols with its own set of *l_vr_id*s irrespective of other VNs.
4. *IDS_PR* specifies *g_pr_id* and *l_pr_id* to identify physical network elements and connected end user devices globally and locally. Each physical resource also has a *pr_type* to distinguish between end user devices and internal network elements.
5. *IDS_EU* provides globally unique location-independent identifiers *g_eu_id* for each of the end users.

   For each virtual network an end user is connected to, there is an *l_vr_id* with an appropriate *vr_type* from the IDS_VR. In addition, each end user has a *l_pr_id* from IDS_PR within the access network for the device it used to connect to a virtual network.

Table 1. Mappings between different identifiers: Why and Where (Not exhaustive)

Mapping	Why	Where
$\langle g\_eu\_id \leftrightarrow l\_vr\_id \rangle$	Identify an end user within a virtual network and vice versa for routing back and forth	Inside corresponding virtual network
$\langle g\_eu\_id \leftrightarrow l\_pr\_id \rangle$	Identify an end user within an access network and vice versa	Within the access network the end user is connected to
$\langle g\_eu\_id \leftrightarrow \{g\_vns\_id\} \rangle$	Obtain the set of VNs an end user is über-homed to	Globallly shared P2P environment
$\langle l\_vr\_id \rightarrow l\_pr\_id \rangle$	Get the local identifier of the physical host of a virtual resource within an InP	Inside corresponding InP
$\langle g\_vns\_id \rightarrow g\_isp\_id \rangle$	Find the owner SP of a virtual network	Globally shared P2P environment
$\langle g\_vns\_id \rightarrow \{g\_isp\_id\} \rangle$	Obtain the set of InPs that form the underlying network to create a virtual network	Globally shared P2P environment
$\langle g\_pr\_id \rightarrow g\_isp\_id \rangle$	Find the managing physical network of a network element belongs to	Within corresponding physical network

4.3 Mappings

In order to keep track of a particular node or end host in the collection of mobility-enabled virtual networks and to route to the current location accordingly, a set of mappings between different identifiers must always be kept updated. In addition, a crucial decision is where and how these mapping must be kept. Table 1 presents a list of such mappings extending the insights from Figure 2 .

4.4 Basic Concepts

In an NVE, each of the VNs must be able to implement its customized topologies, protocols and algorithms irrespective of other cohabiting VNs. NVP2P enables that by providing a bare bone framework for the SPs and InPs to do whatever they want within a certain boundary. For that purpose, we have already identified the entities participating in an NVE, defined the necessary identifier spaces, and laid out the required mappings between identifiers from different contexts. Here we wrap it up with a discussion on how the framework works and reacts to different events.

Steady State. In the steady state, representative *agents* from the SPs and InPs participate in a globally shared P2P environment to create the *substrate* for the

NVP2P framework using their *g_isp_id* identifiers. This substrate is intended to be used for control and management purposes. For example, a distributed resource trading marketplace can be easily set up here [5]. Similarly, creation of compound VNs can be easily managed at this level.

As for routing inside a VN, each VN uses its own set of protocols together with identifiers from its IDS_VR. *l_vr_id*s can be used in conjunction with architectures similar to VRR[28] to enable *identity* based routing inside a VN. NVP2P framework assists by ensuring that the mappings between the virtual resources and actual underlying elements are valid.

Join. An end user connects to a VN by carrying out the following steps:

1. First of all, the end user connects to an access network, which assigns the end user with an *l_pr_id*, specific to that access network. The *gateway access router* the end user connects to, updates and keeps track of the *l_pr_id* assigned to that particular *g_eu_id* in an internal P2P or VRR like environment of that access network.
2. Next, the end user provides the unique identifier *g_vns_id* of the VN it intends to be connected to. The access network, which itself is an InP, looks up the *g_isp_id* of the SP which runs that VN in the global P2P substrate, and forwards the end user to the VN.
3. At this point two situations can arise: first, the *gateway access router* actually hosts a *virtual gateway router* belonging to the VN the end user wants to connect to and can directly bootstrap the end user to that VN. But more frequently that will not be the case; the *gateway access router* must find a way to connect the end user device to another physical router that hosts a *virtual gateway router* and thus can bootstrap it. The details depend on the policy followed by the InP.
4. Once the end user is connected to the VN it is assigned an *l_vr_id* unique within the VN. As mentioned earlier, this *l_vr_id* is then used by the SP to identify the end user inside the VN.

Leave. Whenever an end user *gracefully* leaves the system, all the corresponding records are removed that were stored during the join procedure.

In case of end user connections failures, some sort of heart beat protocol can be implemented by the framework that can check for availability with a certain period. Once a failure is detected, it is handled as a normal leave procedure.

Mobility. Mobility in an NVE can be of different types: geographical mobility of the end user physical devices from one access network to another, logical mobility of end user devices from one VN to another, mobility of virtual network elements using migration techniques etc. Geographical mobility removes the old *l_pr_id* and creates a new one in the current InP, whereas logical mobility does the same for *l_vr_id*. Such mobility is boosted by the über-homing capability of the NVP2P framework. During the transition period, the migrating node can remain über-homed to both the source and destination networks to reduce transition loss.

Über-homing. When an end user is über-homed, it can simultaneously have multiple *l_vr_id*s in each of the connected VNs along with multiple *l_pr_id*s, if needed, for the access networks it used to connect to those VNs. 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. But unlike the existing network where different IP addresses might be assigned to the same node by different ISPs, in an NVE each end user has a unique *g_eu_id*. Once a route is selected to create a connection through particular VNs and InPs, *l_vr_id*s and *l_pr_id*s corresponding to that *g_eu_id* in those VNs and InPs are used to locate and perform routing.

5 Conclusion

NVP2P is a P2P inspired network virtualization framework mainly focusing on handling the micro-level dynamism caused by the mobility and migration of the network elements as well as the end users. In addition, it supports über-homing and provides a shared P2P substrate for handling resource trading, VN composing, and cross VN routing. Even though it is in its incipient stage, we believe NVP2P to be promising enough to explore it further in the future. Possible future works include: defining macro level dynamism attributes; finalizing the global P2P substrate for resource trading marketplace and VN composition; and most importantly, the actual implementation of NVP2P in real life.

Of course, there might be other possible ways to create a framework for realizing an NVE. But at the absence of any other significant work, we believe that NVP2P can at least act as a good starting point that will spark wider interest among researchers in the networking area.

References

1. Anderson, T., Peterson, L., Shenker, S., Turner, J.: Overcoming the Internet impasse through virtualization. Computer **38**(4) (2005) 34–41
2. Turner, J., Taylor, D.: Diversifying the internet. In: Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM'05). Volume 2. (2005)
3. Feamster, N., Gao, L., Rexford, J.: How to lease the Internet in your spare time. SIGCOMM Computer Communication Review **37**(1) (2007) 61–64
4. Chowdhury, M., Boutaba, R.: A survey of network virtualization. Computer Networks (In Submission) (March 2008)
5. Hausheer, D., Stiller, B.: Auctions for virtual network environments. In: Workshop on Management of Network Virtualisation. (2007)
6. Hausheer, D., Stiller, B.: PeerMart: The technology for a distributed auction-based market for peer-to-peer services. In: Proceedings of IEEE ICC'2005. (2005) 1583–1587

7. Hausheer, D., Stiller, B.: PeerMint: Decentralized and secure accounting for peer-to-peer applications. In: Proceedings of Networking'2005. (2005) 40–52
8. Ruffo, G., Schifanella, R.: FairPeers: Efficient profit sharing in fair peer-to-peer market places. Journal of Network and Systems Management **15**(3) (2007) 355–382
9. Jiang, W., Chiu, D.M., Lui, J.C.S.: On the interaction of multiple overlay routing. Performance Evaluation **62**(1-4) (2005) 229–246
10. Wang, J., Chiu, D.M., Lui, J.: Modeling the peering and routing tussle between isps and p2p applications. In: Proceedings of IWQoS'2006. (2008) 51–59
11. Li, Z., Mohapatra, P.: QRON: Qos-aware routing in overlay networks. IEEE JSAC **22**(1) (2004) 29–40
12. Li, Z., Mohapatra, P., Chuah, C.N.: Virtual multi-homing: On the feasibility of combining overlay routing with BGP routing. In: Proceedings of Networking'2005. (2004)
13. Liu, Y., Zhang, H., Gong, W., Towsley, D.: On the interaction between overlay routing and traffic engineering. In: Proceedings of IEEE INFOCOM'2005. (2005)
14. Seetharaman, S., Hilt, V., Hofmann, M., Ammar, M.: Preemptive strategies to improve routing performance of native and overlay layers. In: Proceedings of IEEE INFOCOM'2007. (2007) 463–471
15. Keralapura, R., Chuah, C.N., Taft, N.T., QIannacco, G.: Can coexisting overlays inadvertently step on each other? In: Proceedings of ICNP'05, Washington, DC, USA, IEEE Computer Society (2005) 201–214
16. Garces-Erice, L., Biersack, E., Felber, P., Ross, K., Urvoy-Keller, G.: Hierarchical peer-to-peer systems. In: Proceedings of EuroPar'03. (2003) 1230–1239
17. Ganesan, P., Gummadi, K., Garcia-Molina, H.: Canon in G major: Designing DHTs with hierarchical structure. In: Proceedings of ICDCS'04. (2004) 263–272
18. Montresor, A.: A robust protocol for building superpeer overlay topologies. In: Proceedings of P2P'04. (2004) 202–209
19. Artigas, M.S., Lopez, P.G., Ahullo, J.P., Skarmeta, A.F.G.: Cyclone: A novel design schema for hierarchical DHTs. In: Proceedings of P2P'05. (2005) 49–56
20. Artigas, M.S., Lopez, P.G., Skarmeta, A.F.: A comparative study of hierarchical DHT systems. In: Proceedings of LCN'07. (2007) 325–333
21. Travostino, F., Daspit, P., Gommans, L., Jog, C., de Laat, C., Mambretti, J., Monga, I., van Oudenaarde, B., Raghunath, S., Wang, P.Y.: Seamless live migration of virtual machines over the MAN/WAN. Future Generation Computer Systems **22**(8) (2006) 901–907
22. Bradford, R., Kotsovinos, E., Feldmann, A., Schiöberg, H.: Live wide-area migration of virtual machines including local persistent state. In: Proceedings of VEE'07. (2007) 169–179
23. Huang, W., Gao, Q., Liu, J., Panda, D.: High performance virtual machine migration with RDMA over modern interconnects. In: Proceedings of Cluster'07. (2007)
24. Wood, T., Shenoy, P., Venkataramani, A., Yousif, M.: Black-box and gray-box strategies for virtual machine migration. In: Proceedings of NSDI'07. (2007) 229–242
25. Wang, Y., van der Merwe, J., Rexford, J.: VROOM: Virtual routers on the move. In: SIGCOMM HotNets-VI. (2007)
26. Fujita, N., Touch, J.D., Pingali, V., Wang, Y.S.: A dynamic topology and routing management strategy for virtual ip networks. IEICE Transactions on Communications **E89-B**(9) (2006) 2375–2384
27. Touch, J.D., Wang, Y.S., Eggert, L., Finn, G.: A virtual internet architecture. Technical Report TR-570, USC/Information Sciences Institute (2003)

28. Caesar, M., Castro, M., Nightingale, E.B., O'Shea, G., Rowstron, A.: Virtual ring routing: Network routing inspired by dhts. In: Proceedings of SIGCOMM'06. (2006) 351–362
29. Caesar, M., Condie, T., Kannan, J., Lakshminarayanan, K., Stoica, I.: ROFL: Routing on flat labels. In: Proceedings of SIGCOMM'06. (2006) 363–374
30. Farha, R., Leon-Garcia, A.: A novel peer-to-peer naming infrastructure for next generation networks. In: IPOM'2007. (2007) 1–12
31. Zhou, L., van Renesse, R.: P6P: A peer-to-peer approach to internet infrastructure. In: IPTPS'2004. (2004) 75–86