Dynamic Virtual Private Network provisioning from multiple Cloud Infrastructure Service Providers

Ali Sajjad[1], Andrea Zisman[1], Muttukrishnan Rajarajan[1], Srijith K. Nair[2] and Theo Dimitrakos[2]

[1] City University London, Northamption Square,
EC1V 0HB London, UK
{Ali.Sajjad.1, A.Zisman@soi.city.ac.uk, R.Muttukrishnan)@city.ac.uk
[2] BT Innovate & Design, Adastral Park,
IP5 3RE Ipswich, UK
{srijith.nair, theo.dimitrakos)@bt.com

Abstract. The Cloud infrastructure service providers currently provision basic virtualized computing resources as on-demand and dynamic services but there is no common framework in existence that allows the seamless provisioning of even these basic services across multiple cloud service providers, although this is not due to any inherent incompatibility or proprietary nature of the foundation technologies on which these cloud platforms are built. We present a solution idea which aims to provide a dynamic and service-oriented provisioning of secure virtual private networks on top of multiple cloud infrastructure service providers. This solution leverages the benefits of peer-to-peer overlay networks, i.e., the flexibility and scalability to handle the churn of nodes joining and leaving the VPNs and can adapt the topology of the VPN as per the requirements of the applications utilizing its inter-cloud secure communication framework.

Keywords: Cloud Computing, Peer-to-Peer Overlays, Virtual Private Networks.

1 Introduction

Most of the currently available Cloud Computing solutions are mainly focused on providing functionalities and services at the infrastructure level, e.g., improved performance for virtualization of compute, storage and network resources, as well as necessary fundamental functionality such as virtual machine (VM) migrations and server consolidation etc. In the cases when higher-level and more abstract concerns need to be addressed, existing Infrastructure as a Service (IaaS) solutions tend to focus on functional aspects only. In order to progress from a basic cloud service infrastructure to a more adaptable cloud service ecosystem, there is a great need for tools and services that support and provide higher-level concerns and non-functional aspects in a comprehensive manner. The OPTIMIS project [1] is an ongoing effort in this regard which strives to provide a holistic approach to cloud service provisioning

by offering a single abstraction for multiple coexisting cloud architectures. Of the various high-level concerns being addressed by the OPTIMIS project, a major concern of high importance is the provisioning of a secure communication framework to the services utilizing the resources of different cloud Infrastructure Providers (IP). The usage pattern of these services should be quite flexible i.e. on one hand they might be directly accessed by end-users or on the other hand they might be orchestrated by other Service Providers (SP) for their customers.

There are three fundamental steps in the life cycle of a service in the cloud computing ecosystem; the construction of the service, deployment of the service to one or more IPs and the operation of the services using the IPs resources. In the resulting scenarios, the presence of the multiple infrastructure providers in the cloud computing ecosystem is the key issue that needs to be addressed by any inter-cloud security solution, as a major goal of the Infrastructure Providers (IP) is to maximize their profit from multiple tenants of their resources by making efficient use of the infrastructure and possibly by outsourcing or bursting partial workloads to partner infrastructure providers. In addition to the IPs, the other three main actors in this ecosystem are the Service Providers, end users of the services and third-party cloud brokers [2] that help simplify the use, performance and delivery of the cloud services as well as offer an intermediation layer spanning across multiple cloud providers to provide a host of optimisation and value-added services which take advantage of the myriad individual cloud services e.g., aggregation of different services or arbitration for a best-match service from multiple similar services. For the numerous interaction possibilities among these actors, whatever the usage scenarios maybe, the security of data and communication between the service consumers and its multiple providers is of paramount importance.

In the light of the above discussion, it is clear that an inter-cloud security solution is highly desirable that would provide a framework enabling seamless and secure communication between the actors of a cloud ecosystem over multiple cloud platforms. Such a solution, however, has a number of challenges associated with it because of architectural limitations, as most of the current cloud service platforms and the multi-tenants environments they offer make it difficult to give the consumers of their services flexible and scalable control over the core security aspects of their services like encryption, communication isolation and key management etc. Secure communication is also challenged by lack of dynamic network configurability in most cloud providers, caused by the inherent limitations of the fixed network architectures offered by these providers.

In this work we address the secure, flexible and scalable communication concerns that in our view must be overcome in order to provide holistic cloud provision services to consumers from multiple cloud service providers. We present the architecture and design of an inter-cloud secure communication framework that offers the features of dynamic and scalable virtual network formation, efficient and scalable key management and optimised peer discovery etc. all on top of secure and private communication between the consumers of the service across multiple cloud platforms. Our solution provides a single virtual network to the consumers of cloud services using the infrastructure and resources from multiple cloud providers and offers all the management capabilities to efficiently and transparently run services on top of this

network while catering for the dynamic growth and shrinkage of the network and its participant entities.

The rest of the paper is organised as follows: In Section 2 we outline the key motivations for our approach. In Section 3 we present the background and related works that address peer-to-peer overlays, virtual network connectivity and key management issue related to this domain. We elaborate on the detailed Inter-Cloud Virtual Private Network architecture in Section 4. We conclude in Section 5 with the directions of our future efforts, especially the implementation, experimental setup and the analysis of our idea's results.

2 Motivation

The design and architecture of our inter-cloud secure communication framework is inspired by a collection of techniques like Virtual Private Networks [3] (VPN) and Peer-to-Peer (P2P) Overlays [4]. Network virtualization techniques like VPNs and P2P Overlays have been shown to provide their users legacy communication functionalities of their native network environments, despite the topology, configuration and management architecture of the real underlying physical network. This fits perfectly with our goal of providing a secure virtual private network as a service to the consumers operating on top of multiple cloud providers. All complications and hassles related physical network management challenges can be handled by the overlay network, enabling the services deployed on multiple clouds to benefit from a customised communication network typically only available in physical local-area environments. The core technique of tunnelling of network traffic over a customised P2P overlay is inherently scalable and fault-tolerant and additionally it requires minimal administrative control. Traditionally, most of the private network solutions for the similar problem spaces require the direct and continuous control of a centralised administration entity over every aspect of the overlay network consisting of all the participants that constitute and facilitate the operation of the service being deployed and run on the multiple cloud providers. Such a central controller provides services to authenticate, secure and police the interactions amongst peers. These centralized solutions make it almost necessary to provide complex support and management functionalities to meet the user demands of smooth and continuous operation. Furthermore, to robustly handle the loads generated by a large number of users, significant infrastructure resources and services like mirroring or redundant instances and load-balancing must be set aside, which do not fit with the cloud providers' goal of optimised resource sharing. Peer-to-peer overlay networks, on the other hand, are designed to offer improved scalability, flexibility and availability in a distributed fashion without extensive reliance on centralized servers or resources. However, a basic limitation of most P2P solutions is the lack of advanced functions for authentication, access control and security, which are almost universally present in centralized solutions. Moreover, typical P2P overlays provide connectivity at the application layer, hence excluding a large number of legacy network applications from using them. For the same reason, such overlay networks have been very successfully used to provide specialized application layer services like

voice over IP (VoIP) e.g., Skype [5] and file sharing e.g., Bittorrent [6]. By the provision of a virtualized network over a private overlay network, we can enable the use of applications using even the TCP/IP socket primitives for their legacy requirement. Therefore, we promote an approach where a distributed and scalable key management framework is utilized to establish secure virtual communication channels over P2P overlay networks. The synergy of these three models produces a scalable, secure and robust inter-cloud communication framework which is able to handle a large numbers of communicating peers with considerably less management complexity.

In this idea paper, we present the design and architecture of an inter-cloud virtual private network (ICVPN), which provides secure communication as a service to end users, service providers (SP) and cloud brokers (CB) over multiple cloud infrastructure providers. At its core, it provides the ability to automatically establish peer-to-peer overlay networks comprising of the virtual machines and other infrastructure resources constituting a cloud service. Using the same P2P techniques, we also offer a distributed key management service which facilitates the automatic discovery of the peers participating in a service and the binding of cryptographic constructs like keys, certificates and fingerprints to their identities. In all of this undertaking, the only configuration required from the users of the system is the creation and management of the service deployed on the infrastructure of multiple cloud providers, which falls under the scope of companion components of the OPTIMIS toolkit [1]. The configuration and maintenance of the VPN connections over the P2P overlay is autonomous and transparent to the consumers of this service. The ICVPN overlay is managed without bothering the users with the complicated configurations typically required to set up the key management and virtual networking infrastructures in similar problem spaces. To achieve this goal, our overlay architecture offers following unique features:-

i. Automatic allocation of IP addresses and DNS host names to the virtual machines (peers) participating in a service,
ii. Automatic discovery of peer credentials (certificates, private/public keys etc.)
iii. Scalable key management service to store and bind cryptographic credentials to individual peers
iv. End-to-end encryption of all the communication among the peers over the P2P overlay

3 Related Work

The Inter-Cloud VPN architecture incorporates peer-to-peer overlay networks and decentralized peer and key management models, but our main interest is the formation of secure virtual private networks using these models as the underlying foundations of our solution.

There is an existing vast body of research detailing how P2P systems can be used to provide better services for specific applications like voice and video communication and file sharing etc. [5][6]. For our research, we will make use of some existing structured P2P techniques that will help us overcome some inherent

difficult issues in network virtualization such as the use of Distributed Hash Table (DHT) as a data structure in which data object (or value) location information is placed deterministically, at the peers with identifiers corresponding to the data object's unique key. The P2P overlay networks support the scalable storage and retrieval of (key, value) pairs on the overlay network which is very helpful when we need to store and retrieve meta-data related to the virtual private network management. This technique has already been used for providing large-scale distributed DNS service [7] and we will be leveraging the same strengths to cater for our storage and retrieval requirements to build up a virtual private network. In theory, DHT-based systems guarantee that on average, any data object can be located in O(log N) peer hops, N being the number of peers in the network. Existing solutions like Chord [8], Pastry [9] and Tapestry [10] have been widely used to provide scalable and fast information storage and retrieval services for a vast variety of applications.

As mentioned earlier, the central thrust of our architecture is the provisioning of a secure virtual private network over multi-cloud infrastructure. VPNs have been a mainstay for providing secure remote access over wide-area networks to resources in private organizational networks for a long time. Well-known tools and softwares like OpenVPN [11] are used to create secure point-to-point or site-to-site connections for authenticated remote access. However, the main problem such client/server based approaches is that they require centralized servers to manage the life cycle of all the secure connections for the participating clients, hence suffering from a single point-of-failure. Another issue is the quite complex and error prone configuration problems especially if you want to construct and manage a large-scale network not having a relatively simple topology, as it would require customised configuration on every client and even more elaborate management and routing configuration on the server-side. Another major drawback is the complexity of key distribution among all the participating clients in a VPN, as the software itself doesn't provide any key distribution service and all keys have to be manually transferred to individual hosts and in case of PKI model, an additional requirement of a trusted Certificate Authority exists that has to issue individual certificates to all the servers and clients constituting a VPN, which incurs an additional communication overhead when forming a virtual private network.

There have been some other VPN solutions for large-scale networks aimed at grid and cluster computing environments, such as VIOLIN [12] and VNET [13], that don't follow a strict client/server model based approach. VNET is a layer 2 virtual networking tool that relies on a VNET server running on a Virtual Machine Monitor (VMM) hosting a virtual machine in a remote network which establishes an encrypted tunnel connection to a VNET server running on a machine (called Proxy) inside the user's home network. All of the remote virtual machine's communication goes through this tunnel and the goal of the Proxy is to emulate the remote virtual machine as a local host on the user's home network, in effect presenting it as a member of the same LAN. The motivation of this approach is to tackle the user's lack of administrative control at remote grid sites to manipulate network resources like routing and resource reservations etc. but it suffers from the previously discussing problem of complex and manual configuration though going for the simplicity of a private LAN. Also the scalability will be a big issue for the Proxy as the number of remote virtual machines grows as each will require a secure tunnel connection and

corresponding virtual network interface mapped to the Proxy's network interface by the VNET server software.

VIOLIN is a small-scale virtual network with virtual routers, switches and end hosts implemented in software and hosted by User-Mode Linux (UML) enabled machines as virtual appliances. It allows for the dynamic establishment of a private layer 3 virtual network among virtual machines, however, it doesn't offer dynamic or automatic network deployment or route management to setup the virtual network. Virtual links are established between the virtual appliances using encrypted UDP tunnels that have to be manually setup and are not self-configuring, making it cumbersome to establish inter-host connections in flexible and dynamic fashion. Furthermore, the detailed performance results for VIOLIN are not currently available, to the best of our knowledge.

More recently, P2P VPN solutions like Hamachi [14] and N2N [15] have come up as peer-to-peer alternatives to centralized and client/server model based VPNs. Hamachi is a shareware application that is capable of establishing direct links between computers that are behind NAT firewalls. A backend cluster of servers are used to enable NAT traversal and establish direct peer-to-peer connections among its clients. Each client establishes and maintains a control connection to the server cluster. It is mainly used for internet gaming and remote administration but suffers from scalability issues as each peer has to maintain the connection with the server as well as any other peers it wants to communicate with, ending up with the overhead of a mesh-topology. It therefore offers limited number of peers (16 per virtual network) and limited number of concurrent clients (50 per virtual network). The keys used for connection encryption and authentication etc. are also controlled by the vendor's servers and individual users do not initially control who has access to their network.
N2N is a layer 2 VPN solution which doesn't require a centralized backend cluster of servers like Hamachi but it uses a peer-to-peer overlay network similar to Skype, where a number of dedicated super-nodes are used as relay agents for edge nodes that cannot communicate directly with each other due to firewall or NAT restrictions. The edge nodes connect to a super-node at start-up and pre-shared TwoFish [16] keys are used for link encryption. As it operates on layer 2, the users of the overlay have to configure their IP addresses etc. It also assumes node membership as relatively static with edge nodes rarely leaving or joining the network over their life cycle.

More recently, some commercial cloud computing services have been made available by different vendors that provide a virtual private network inside their public cloud offering and offering the customers some limited degree of control over this network, which is called a Virtual Private Cloud (VPC). Prime examples in this domain are Amazon Virtual Private Cloud [17], Google Secure Data Connector [18] and CohsiveFT VPN-Cubed [19]. These are aimed at enterprise customers to allow them to access their resource deployed on the vendor's cloud over an IPSec [20] based virtual private network. Although these products allow the possibility of leveraging the cloud providers' APIs to flexibly grow and shrink their networks, the management and configuration is as complex as a traditional network as component s of the VPC such as internet gateways, VPN servers, NAT instances and subnets etc. have to be managed by the customers themselves. Furthermore, the customers are required to setup an IPSec device on their premises that connects to an IPSec gateway in the VPC running as a virtual appliance which integrates the enterprise's network

with the VPC subnet in the cloud. Most importantly, with the exception of [19], these solutions are locked to single cloud vendor and [19] provides use of a selective set of cloud providers by placing its virtual appliances as IPSec gateways in these cloud infrastructures and allowing the customers to join these gateways in a mesh topology manually.

4 Inter-Cloud VPN Architecture

The Inter-Cloud VPN architecture consists of the following main components:-

4.1 Peer-to-Peer Overlay

The core technique employed by the ICVPN concept is the use of a universal P2P overlay to provide a scalable and secure service infrastructure to initiate and bind multiple VPN overlays to different cloud services. The universal overlay itself is initiated by the Service Provider or Cloud Broker and helps providing bootstrapping of VPN peers and providing the possibilities of service advertisements, service discovery mechanisms and service code provisioning for them, with minimal manual configuration and administration. This approach acts as an aggregation service for the eventual peered overlay resources (which in this case are virtual machines) spanning across multiple cloud domains to help form a virtual private network. The peers of the universal overlay act as super peers for the nodes of the underlying overlays and let new nodes enroll, authenticate, bootstrap and join a particular ICVPN overlay based on the cloud service requiring a VPN service. As depicted in Figure 1, the SP or the Cloud Broker could itself be a peer in the universal overlay and a subset of the universal overlay peers can act as super-peers for the peer nodes of the VPN overlay for a particular cloud service. The universal overlay peers can join and leave the system dynamically and additional VMs from the cloud providers can be provisioned by the SP or Cloud Broker to act as the universal overlay peers as well. As both the universal and the VPN overlay nodes are basically VMs provisioned from different cloud providers, they can be demoted or promoted from these overlays respectively based on parameters like performance and availability.

To join the universal overlay, each peer needs to acquire a unique identification number (peerID). In most structured P2P systems, this is done by the peer itself by choosing a random number from a large identity space, however, this approach is vulnerable to Sybil attacks [21]. Due to the security constraints of our solution, we require some trusted authorities to allocate peerIDs to the participating peers. This identity management problem can be solved by various methods but we focus on two i.e. either by using Trusted Third Parties in a PKI framework or the Hierarchical Identity-Based Cryptography (HIBC) [22]. In the traditional PKI approach, either the service provider or the cloud broker acts as a Certificate Authority (CA) and issues signed certificates to super-peers and peers, or the super-peers can themselves act as the CA for the underlying overlays' peers. The CA assigns a random peerID to the peers and signs a certificate that binds the ID of the cloud service utilizing the VPN (serviceID) and peerID with the public key of the peer for a limited time duration.

The peer then can use the corresponding private key to authenticate itself with other peers in the overlay. It is important to note that the peerID should not be based on the IP address of the peers as these are liable to change in the cloud environment due to the possibility of virtual machine migration. In the OPTIMIS toolkit, we utilize the MAC addresses for this purpose, which are assigned to the VMs by the SP or Cloud Broker when initially provisioning them from the cloud infrastructure providers.
To join the ICVPN overlay within the universal overlay for a particular cloud service, a peer first needs to obtain the address of at least one peer already in the same overlay. To achieve this, for each service, a small list of member peers is inserted in the universal overlay under the service key. The peer looks up the service key in the universal overlay and chooses a member peer at random to be its contact peer in the overlay. The oldest peer in the list is replaced by the new joining peer to keep the list fresh. For sake of redundancy, the membership list is replicated in the universal overlay. The overlay can be further leveraged to customize the virtual topology of the network, by either emulating a completely distributed mesh topology at one end, or a centralized star topology by automatically electing a super-peer to act as the gateway of the virtual network at the other end, or an in-between hybrid topology using a number of selected super-peer nodes. The adaptation of a suitable network topology is indeed a major communication benefit for services using P2P overlays for their operational requirements [23]. Figure 1 shows the presence of all the above mentioned topologies in the three cloud infrastructure providers.

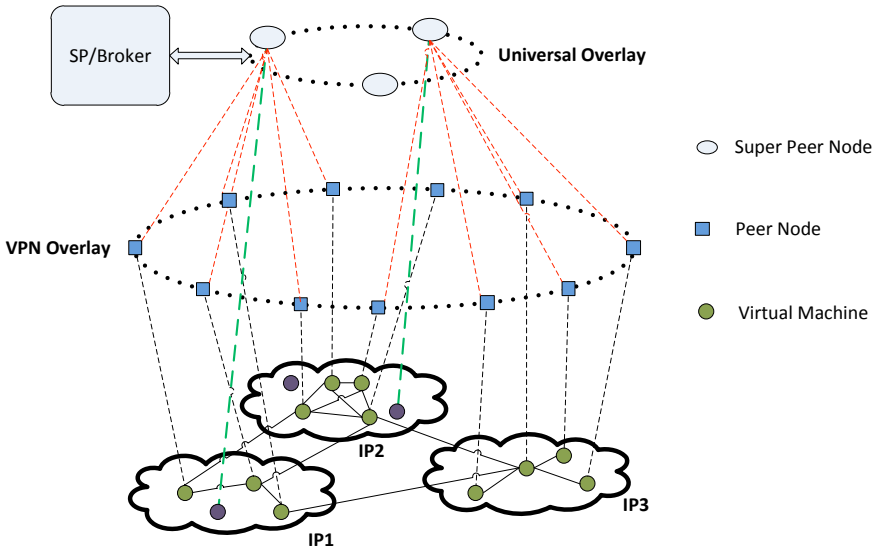

Fig. 1. Establishment of a peer-to-peer overlay based virtual network on top of multiple cloud infrastructure providers.

In typical usage scenarios, the end-users or the SPs are responsible for provisioning virtual machines from cloud providers to deploy and run their services. These virtual machines are considered as the peers of the overlays and the complete

lifecycle of the peers is handled by a P2P client embedded in the appliance image used to instantiate a virtual machine on a cloud platform. However, a further advantage of the universal overlay approach is that the peers of a VPN overlay can get, update and modify the P2P client program dynamically from the peers in the universal overlay. The program to be run is signed by the super-peers or the CA for validity and it can check for updated versions of itself by querying for the associated serviceID in the persistent store of the universal overlay's DHT.

Upon joining the overlay network to become part of a service, a peer starts the process of creating secure tunneled connections to the other peers of the service according to the network topology selected for the operation of that particular service. The typical constructs used for this purpose are TAP and TUN devices, which are virtual network kernel devices set in the operating system of the peer node [24]. A TAP device simulates an Ethernet interface card and operates with link layer datagrams such as Ethernet frames. A TUN device simulates a network layer device and operates with network layer packets such as IP packets. TAP is mainly used to create a virtual network bridge, while TUN is used with routing, so choice of their selection mostly depends upon the network topology chosen for the overlay network. Packets sent by an operating system via a TUN/TAP [25] device are delivered to a user-space program that attaches itself to the device. A user-space program may also pass packets into a TUN/TAP device. In this case TUN/TAP device injects these packets to the operating system network stack thus emulating their reception from an external source.

4.2 Secure Virtual Private Connections

The key feature of our ICVPN is establishing a secure communication network between the peers of the overlay formed over a collection of cloud providers' infrastructure. To achieve this, we can make use of public key cryptography [26] which is supported by both the PKI model and the HIBC model. This makes it possible to create authenticated and confidential end-to-end tunnels which provide protection against eavesdropping, message tempering and message forgeries. Another practical advantage of using these is the reuse of existing frameworks and tools which have been thoroughly tried and tested in a myriad of different domains, are widely used and have been adopted and integrated in countless security techniques. The main components of the P2P client used to construct a virtual private network topology in our model are shown in Figure 2.

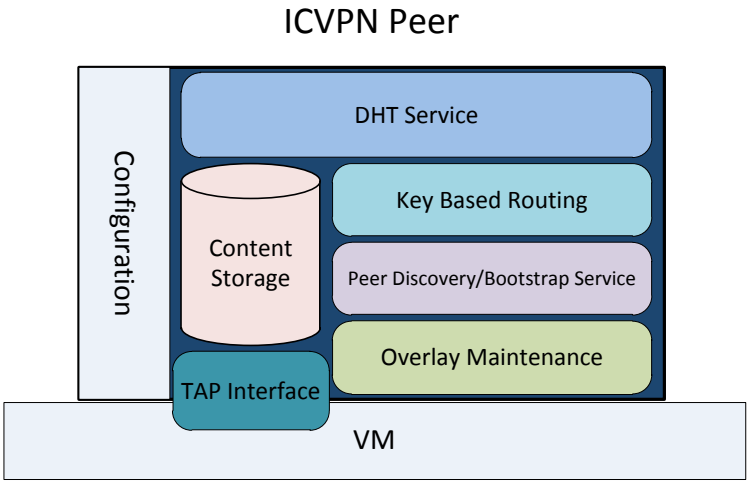

Fig. 2. Logical architecture of the internal components of a peer node in a P2P overlay based virtual network.

The P2P client software sets up and configures the TAP/TUN device automatically and assigns unique IP addresses to their virtual interfaces and DNS hostnames to the peers themselves and this record is put on the distributed hash table (DHT) for the operation of the peer discovery service of the overlay network, which works by obtaining the list of unique peer identifiers and mapping them to the locally assigned IP addresses and host names. The same DHT is used for binding security credentials like certificates and keys to the unique peers and identify which service the peers are a part of. Once a peer has obtained the certificate or HIBC identity of a peer it wants to communicate with from the DHT, they can directly negotiate and generate symmetric session keys for encrypted tunnelled communication.

5 Future Work

The logical step towards further development of this idea is its implementation. To implement a prototype of our solution, we will pursue the following simplified usage model.

5.1 Prototype Use Case Model

The service provider or cloud broker wants to provision an Inter-Cloud VPN for a distributed web service which is deployed on multiple virtual machines on different cloud providers. We assume that the P2P client software was already packaged in the ISO image of the VMs when they were initialized and the service provider has provided the UUID identifying the web service the peers are participating in, the P2P

topology it wants to create for the peers, and a list of peers for the P2P bootstrapping process. For a centralized key management approach, the service provider is also responsible for setting up a Certificate Authority that can issue the peers with keys and certificates which are used for encryption and authentication; otherwise, the peers would create their own self-signed certificates which can be located by the interested peers using the DHT lookups on the overlay network. The DHT will be used to store all the data necessary for creating and maintaining the VPN among the peers according to the network topology selected by the service provider for the particular web service.

Once equipped with the required security credentials and peer locations, the peers can proceed to form secure tunnels with other peers to complete the construction of the VPN. For example, if the requirement from the service provider is to provide a star-topology VPN to emulate a LAN, one of the peers is elected as the designated gateway and rest of the peers open secure tunnels with the gateway peer. The DHT is used as the control channel that provides the meta-data to carry out the necessary configurations and connection setups

5.2 Prototype Evaluation

We will evaluate a prototype implementation based on experiments conducted to measure the performance of the network communications over the P2P overlay. The experiments will be carried out in realistic conditions, over multiple cloud infrastructure environments. We will measure the bandwidth and latency overhead of creating and maintaining the ICVPN connections against the increasing number of peers participating in the service. We will also detail metrics like how long it takes for a peer to join the overlay, the bandwidth overhead per peer node to maintain the VPN connections, the throughput of the encrypted communication, the effect of peers joining and leaving the overlay as the demand and number of users varies, the effect of some of the peers failing, etc.

Acknowledgments. We acknowledge financial support for this work provided by the European Commission's Seventh Framework Programme ([FP7/2001-2013]) under grant agreement number 257115, OPTIMIS.

References

1. Ferrer, A.J., Hernández, F., Tordsson, J., Elmroth, E., Zsigri, C., Sirvent, R., Guitart, J., Badia, R.M., Djemame, K., Ziegler, W.: OPTIMIS: a Holistic Approach to Cloud Service Provisioning. Presented at the First International Conference on Utility and Cloud Computing , Chennai, India December 14 (2010).
2. Gartner: Cloud Consumers Need Brokerages to Unlock the Potential of Cloud Services, http://www.gartner.com/it/page.jsp?id=1064712.
3. Andrew S. Tanenbaum, David J. Wetherall: Virtual Private Networks. Computer Networks. p. 821. Prentice Hall (2010).
4. Andersen, D., Balakrishnan, H., Kaashoek, F., Morris, R.: Resilient overlay networks. SIGCOMM Comput. Commun. Rev. 32, 66–66 (2002).

5. Baset, S.A., Schulzrinne, H.G.: An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol. Presented at the April (2006).
6. Cohen, B.: The BitTorrent Protocol Specification, http://www.bittorrent.org/beps/bep_0003.html.
7. Cox, R., Muthitacharoen, A., Morris, R.: Serving DNS Using a Peer-to-Peer Lookup Service. In: Druschel, P., Kaashoek, F., and Rowstron, A. (eds.) Peer-to-Peer Systems. pp. 155-165. Springer Berlin / Heidelberg (2002).
8. Stoica, I., Morris, R., Karger, D., Kaashoek, M.F., Balakrishnan, H.: Chord: A scalable peer-to-peer lookup service for internet applications. Presented at the , San Diego, California, United States (2001).
9. Rowstron, A., Druschel, P.: Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems. In: Guerraoui, R. (ed.) Middleware 2001. pp. 329-350. Springer Berlin / Heidelberg (2001).
10. Zhao, B.Y., Huang, L., Stribling, J., Rhea, S.C., Joseph, A.D., Kubiatowicz, J.D.: Tapestry: a resilient global-scale overlay for service deployment. Selected Areas in Communications, IEEE Journal on. 22, 41 - 53 (2004).
11. Yonan, J.: OpenVPN - an open source SSL VPN solution, http://openvpn.net/.
12. Jiang, X., Xu, D.: VIOLIN: Virtual Internetworking on Overlay INfrastructure. Presented at the (2003).
13. Sundararaj, A.I., Dinda, P.A.: Towards virtual networks for virtual machine grid computing. Presented at the , Berkeley, CA, USA (2004).
14. Hamachi - A zero-configuration virtual private network, https://secure.logmein.com/products/hamachi2.
15. Deri, L., Andrews, R.: N2N: A Layer Two Peer-to-Peer VPN. In: Hausheer, D. and Schönwälder, J. (eds.) Resilient Networks and Services. pp. 53-64. Springer Berlin / Heidelberg (2008).
16. Schneier, B., Kelsey, J., Whiting, D., Wagner, D., Hall, C., Ferguson, N.: The Twofish encryption algorithm: a 128-bit block cipher. John Wiley & Sons, Inc., New York, NY, USA (1999).
17. Amazon: Virtual Private Cloud, http://aws.amazon.com/vpc.
18. Google: Secure Data Connector, http://code.google.com/securedataconnecto.
19. CohesiveFT: VPN-Cubed, http://www.cohesiveft.com/vpncubed.
20. Doraswamy, N.: IPSec : the new security standard for the Internet, intranets, and virtual private networks. Prentice Hall PTR, Upper Saddle River NJ (2003).
21. Douceur, J.: The Sybil Attack. In: Druschel, P., Kaashoek, F., and Rowstron, A. (eds.) Peer-to-Peer Systems. pp. 251-260. Springer Berlin / Heidelberg (2002).
22. Gentry, C., Silverberg, A.: Hierarchical ID-Based Cryptography. In: Zheng, Y. (ed.) Advances in Cryptology — ASIACRYPT 2002. pp. 149-155. Springer Berlin / Heidelberg (2002).
23. Dinger, J., Hartenstein, H.: On the Challenge of Assessing Overlay Topology Adaptation Mechanisms. Presented at the Fifth IEEE International Conference on Peer-to-Peer Computing (P2P'05) , Konstanz, Germany July 6 (2011).
24. VTun: Virtual Tunnel, http://vtun.sourceforge.net/.
25. TUN/TAP, http://en.wikipedia.org/wiki/TUN/TAP.
26. Diffie, W., Hellman, M.: New directions in cryptography. IEEE Trans. Inform. Theory. 22, 644-654 (1976).