Secure Cloud Computing with a Virtualized Network Infrastructure

Fang Hao, T.V. Lakshman, Sarit Mukherjee, Haoyu Song
Bell Labs, Alcatel-Lucent
{firstname.lastname}@alcatel-lucent.com

ABSTRACT

Despite the rapid development in the field of cloud computing, security is still one of the major hurdles to cloud computing adoption. Most cloud services (e.g. Amazon EC2) are offered at low cost without much protection to users. At the other end of the spectrum, highly secured cloud services (e.g. Google "government cloud") are offered at much higher cost by using isolated hardware, facility, and administrators with security clearance. In this paper, we explore the "middle ground", where users can still share physical hardware resource, but user networks are isolated and accesses are controlled in the way similar to that in enterprise networks. We believe this covers the need for most enterprise and individual users. We propose an architecture that takes advantage of network virtualization and centralized controller. This architecture overcomes scalability limitations of prior solutions based on VLANs, and enables users to customize security policy settings the same way they control their on-site network.

1. INTRODUCTION

Despite the rapid development in the field of cloud computing, security is still one of the major obstacles to cloud computing adoption [5, 4, 3]. To ease the concerns of IT managers, it is critical to ensure data privacy and integrity in the cloud at a level that is at least comparable to that in current enterprise networks.

However, the current cloud computing services fall in short on isolating computing resources and networks between customers. This is not surprising because the success of cloud computing depends on economy of large scales. It is essential for cloud service providers to take advantage of resource sharing and multiplexing among customers. Virtual machines of different customers may reside on the same physical machine, and their data packets may share the same LAN. Such lack of isolation brings security risks to users. For example, [15] has shown that it is possible for a hacker to conduct attacks towards another Amazon EC2 user who shares hardware resources with the hacker in the cloud.

On the other end of the spectrum, google has proposed "government cloud", which creates entirely separate hardware, software, and administrators (with appropriate background checks) for special customers. While such cloud service can be very secure, it is also very expensive — almost like building a separate data center for each customer.

In this paper, we explore the "middle ground", where users can still share physical hardware resource, but user networks are isolated and accesses are controlled in the way similar to that in enterprise networks. We believe this covers the need for most enterprise and individual users. More specifically, we propose a new data center architecture with following properties:

- Isolation. The architecture provides effective isolation between different customer networks. This includes supporting their private IP address spaces, which may potentially be overlapping, and isolating their traffic. Resource allocation should be managed so that customers cannot impact each other's resource usage in an uncontrolled manner.
- Transparency. The underlying data center infrastructure and hardware should be transparent to the customers. Each customer should have a logical view of its own network, independent of the actual implementation. This simplifies the administration for the customer and improves security.
- Location independence. The virtual machines (VM) and networks of customers should be "location independent", i.e., can be physically allocated anywhere in the data center. This can greatly improve resource utilization and simplify provisioning.
- Easy policy control. Each customer may have its own policy and security requirements. The architecture should allow customers to configure their individual policy settings on the fly, and enforce such settings in the network.
- Scalability. The number of customers that can be supported should be restricted only by the resources available in the data center, not by design artifacts.
- Low cost. The solution must mostly rely on off-the-shelf devices, so that new investment for cloud service providers can be reduced.

In this paper, we exploit recent advances in technologies amenable to network virtualization [1, 7, 8]. Network virtualization techniques can logically separate different networks on the same hardware and partition resources accordingly [9, 1]. This feature is useful for providing good isolation as well as network-resource sharing among different users. Furthermore, recently proposed mechanisms simplify packet-forwarding elements and make control functions more flexible and manageable [7, 8] by using centralized control. Given the high density of physical resources and demand for high manageability of devices, the centralized control architecture suits data center networks very well.

However, unlike in typical network virtualization solutions, our design does not require deploying specialized routers or switches across the entire data center network. Instead, conventional off-the-shelf Ethernet switches can be used in most parts of the network. Enhanced layer 2 switches, which we refer as Forwarding Elements (FE), are deployed only at certain aggregation points to provide the required virtualization functions. In this architecture, each customer has its own isolated virtual network in the data center, to which access is tightly controlled. But physically, such virtual network may be distributed at anywhere in the data center.

This architecture intends to offer a more secured elastic cloud computing (SEC2) service. But the design can also naturally support virtual private cloud (VPC) service, where each user's private network in cloud is connected to its on-site network via VPN.

We describe the architecture design in the next section. We then present the design details in Section 3, followed by further discussion of several design issues in Section 4. In Section 5 we explain why existing data center designs are not sufficient for solving this problem. Concluding remarks are in Section 6.

2. ARCHITECTURE

2.1 Conventional data center architecture

A conventional data center network is typically partitioned into three layers: access, aggregation, and core. It is possible to support multiple isolated networks in this architecture by using VLANs combined with "virtual switches" in hypervisors such as VmWare and Xen. VMs on physical hosts can be configured to use different VLANs. Both virtual switches in hypervisors and physical layer 2 and 3 devices in data center can be configured to support VLANs for each customer. To extend VLANs across layer 3 networks, "VLAN trunking" can be used to tunnel packets across routers.

The main limitation of this solution is caused by the scalability issues of VLAN. The maximum number of VLANs is limited to 4K due to VLAN id size. Furthermore, overlaying thousands of VLANs on the same physical network may cause network management complications and increase control overhead. For example, one physical link failure may trigger spanning tree computation on all VLANs that run on it. In a more general sense, VLAN couples both access control and packet forwarding. We believe it is cleaner and more scalable design to separate the two.

Another limitation is that it is not easy to enable per-user security policy control. There are potentially two places where security policies can be enforced: at the middleboxes at aggregation layer, or in software setup at each hypervisor. Just relying on hypervisor setup may carry security risks given that hypervisor runs on the same physical host as the user VMs, and thus share same resources. The middlebox solution is likely to be more secure since it is more difficult to be hacked by users. But to enforce packets always traverse through given middleboxes in a given sequence requires non-trivial planing, configuration, and even tweaking physical link connections, rendering the solution not administratively feasible. The latest Cisco fiber extender solution addresses this problem by forcing all traffic to go through the aggregation switch. But this creates unnecessary traffic load in aggregation switches since in many data centers majority internal traffic is between nodes within the same rack.

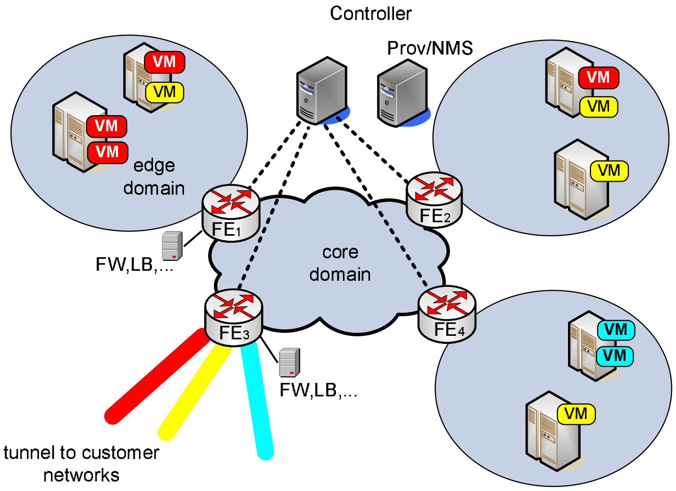

Figure 1: SEC2 architecture

2.2 Secure Elastic Cloud Computing (SEC2)

Based on the above observations, we propose a new design based on network virtualization. In this design, network virtualization is accomplished by two entities: Forwarding Elements (FEs) and Central Controller (CC). FEs are basically Ethernet switches with enhanced APIs that allow them to be controlled from a remote CC. Packet handling actions such as address mapping, policy checking and enforcement, and forwarding are done in FEs. CC stores control information such as addresses, location, and policy. Such information is distributed to different FEs as needed.

Figure 1 illustrates the basic SEC2 architecture. Unlike in the conventional three layer design, here the network is structured in two levels: one *core domain* and multiple *edge domains* surrounding it. The core domain consists of layer 2 Ethernet switches with high switching capacity. Its function is to transport packets between edge domains.

An edge domain contains physical hosts connected by Ethernet switches. Each edge domain is connected to the core domain through one or more FEs. The edge domain has three functions. First, it resolves packet addresses, and determines the edge domain and MAC address of an IP packet. Second, it ensures that packets of different customers are isolated, and are processed according to their configured policy settings. Packets cannot traverse across customer network boundary without being checked and enforced based on security rules. Third, depending on the destination, packets are either delivered locally, or tunneled through the core domain to reach the destination edge domain.

Note that FEs serve as gateways between core and edge domains, while CC provides control and configuration functions. Middleboxes such as firewalls and load balancers are attached to FEs, so that packets needing such treatment will be forwarded through them. In this design, each customer has its own isolated virtual network in the data center, but such virtual network may be physically distributed across any edge domains. Each customer can set up security policies for their network through a web portal, which are in turn translated into policy settings in CC.

2.3 Numbering and addressing

To distinguish between different customers, we assign each customer a unique *cnet id* (customer network id). If a customer has multiple subnets or needs to set up several different security zones, then each subnet or security zone is assigned a unique *cnet id*. A *cnet* is a logical entity where all nodes in it share the same policy rules, independent of the physical domain structure described previously.

Logically, an end point (i.e. VM) can be identified by the combination of (*cnet id*, IP). This is in turn mapped to a unique layer 2 MAC address [1]. Most existing host virtualization platforms support assigning virtual MAC addresses to VMs. In platforms that do not support such configuration, VMs usually share the same MAC address as their host machine. In this case we generate a pseudo MAC address for the VM, and use such pseudo address for identification purpose but not for packet forwarding.

Each edge domain is assigned a unique *eid*. We use VLANs to separate different customers within each edge domain. In the same edge domain, there is one to one mapping between VLAN id and *cnet id*.

VLAN configuration is done at all networking devices in the edge domain, including FEs, Ethernet switches, and virtual switches on host hypervisors. Such configuration is transparent to VMs, so that applications that run on the VMs are not aware of VLAN configurations.

The scope of a VLAN is limited within the same edge domain. Different edge domains can reuse VLAN ids. As a result, different customers in different edge domains may have the same VLAN id. Hence we eliminate the limit on the number of customers that can be accommodated due to VLAN id size. In addition, each subnet may be expanded to multiple edge domains using different VLAN ids, so the scale of each customer subnet is not limited by edge domain size. This design implicitly poses a limit on the number of VMs that can be supported in the same edge domain. In the worst case, where each VM belongs to a different customer in the same edge domain, there can be a maximum of 4K VMs in each edge domain. However in general an edge domain can accommodate many more VMs since many customers are likely to have more than one VM. Note that this limit is only for the same edge domain but the proposed architecture does not impose limits on the number of edge domains in a data center.

2.4 Integration with customer's on-site network

This architecture can be naturally extended to accommodate the service where customers need to integrate cloud resources with their existing on-site network. One example of such service is Amazon Virtual Private Cloud (VPC) service, where customers can extend their network to the data center via IPSec tunnels.

The customer's on-premise network can be treated as a special edge domain for the data center network. The customer site can be connected to the data center network using existing VPN technologies such as VPLS, layer 3 MPLS VPN, or IP tunnels such as GRE or IPSec. In particular, if a customer has already been using the VPN services from a service provider, it will be easy to add data center as an additional site of the same VPN instance.

The FEs at the data center edge can serve as Customer Edge (CE) routers since the data center is a *customer* of the service provider's VPN service. Hence such FEs are referred as CE-acting FEs for convenience.

3. DESIGN DETAILS

In this section, we present further details on each component of our design.

3.1 Central Controller (CC)

CC controls the operation of FEs. It maintains both address mapping and policy databases. The following mappings are maintained by CC:

- VM MAC ↔ (*cnet id*, IP). This resolves the IP

[1] Choice of requiring unique MAC address is just a convenience for simplifying table lookup in FEs, not a necessity, since (cnet id, IP) can uniquely identify an end point.

address of each VM to its corresponding MAC address.

- VM MAC ↔ edge domain id *eid*. This identifies the edge domain where the VM is located at the present time.

- *eid* ↔ FE MAC list. FE MAC refers to the MAC addresses of the FEs to which the edge domain connects. Note that it is possible for an edge domain to have multiple FEs for load sharing and reliability reasons.

- (*cnet id*, *eid*) ↔ VLAN id. This identifies the VLAN used by each customer in each edge domain.

CC also maintains policy rules. An example of policy rule can be: packets from customer A are first forwarded to firewall F, and then to its destination. In such case, the first hop FE that enforces this policy needs to tunnel such packets to firewall F.

Although CC is conceptually one entity in this architecture, it can be implemented in a distributed way. For example, different customers can be assigned to different CCs by using Distributed Hash Table (DHT). Since the management and policy control of different customer networks are relatively independent, such partition does not affect the functionality of CC.

3.2 Forwarding Elements (FE)

FEs are gateways between the core domain and the edge domains. Each edge domain may have more than one FEs for redundancy purpose. Functions of FE include the following:

- Address lookup and mapping. When a packet is originated from its edge domain to other domains, it looks up the FE MAC of the destination domain and VLAN ID in destination domain. This is done by first checking its local cache, and if there is no hit, it inquires the CC.

- Policy enforcement. FEs enforce policy by applying filtering, QoS treatment, or redirecting to middleboxes that are attached to them. By default, packets designated to a different customer are dropped.

- Tunneling. FE tunnels the packet across the core domain to the destination FE via MAC-in-MAC. The source FE adds another MAC header to the original Ethernet frame. The destination FE strips off the outer layer header upon receiving the packet. Most modern Ethernet switches allow larger frame sizes (jumbo frames) to be used so the extra few bytes of MAC header is not a problem. This is especially true for the core domain since we expect high-end Ethernet switches to be deployed to meet capacity needs in the core.

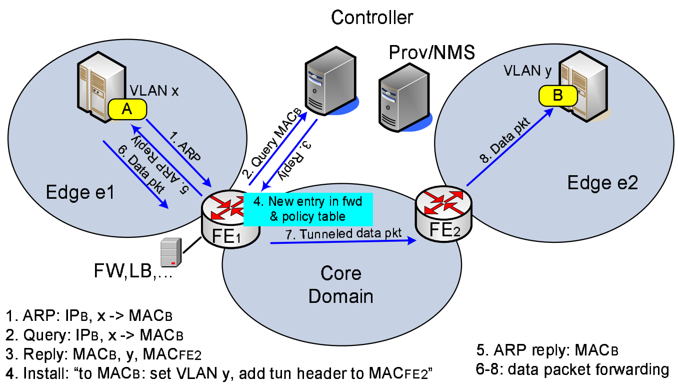

Figure 2: Data forwarding

3.3 Data forwarding path

Packets sent in the same VLAN are directly delivered to the destination VM without policy checking [2]. Otherwise they first reach FEs, which in turn enforce security policies and make forwarding decisions. This holds even for packets between different VLANs in the same edge domain.

Figure 2 illustrates how VMs that belong to the same customer network but reside in different edge domains communicate. Suppose VMs A and B belong to the same subnet of the same customer. A resides in VLAN x of edge domain $e1$, and B resides in VLAN y of edge domain $e2$. Before A can send packets to B, it first discovers B's MAC address by an ARP request. The ARP request reaches FE1 at $e1$'s edge. FE1 looks up its ARP table. If no entries are cached, it inquires CC. By using the mapping tables defined in Section 3.1, CC looks up B's MAC MAC_B, its edge domain $e2$, and corresponding VLAN id y and FE MAC_{FE2} in $e2$. Based on CC's response, FE1 then installs entry "to MAC_B: tunnel to MAC_{FE2}, dest VLAN y" along with other policies in its forwarding table, and returns A with the ARP reply that contains MAC_B.

After A receives MAC_B, it can send data packet to B with MAC_B as its layer 2 destination. The packet is forwarded to FE1. FE1 enforces appropriate policies and performs QoS treatments, and then change the VLAN id to y and add an outer MAC header MAC_{FE2}. The packet is then forwarded across the core domain, reaching FE2. FE2 strips off outer MAC header, and delivers the packet to B.

The customer can offer public access to node B by requesting a public IP address, which is in turn NATed to B's private address. In this case, external packet will be forced by FE to traverse through firewall and NAT middleboxes before reaching the private network.

3.4 Connectivity across sites

[2] As explained previously, nodes with different policy requirements or nodes that cannot directly communicate are put into different VLANs

The customer network can be connected to the data center via layer 2 or 3 VPN tunnels, depending on whether the customer requires the cloud resource to be part of the same LAN of its on-site network or not. Layer 2 VPLS tunnels can be used to attach the data center VMs into customer's existing LAN. In the VPLS setup, at the customer end, customer edge (CE) router or switch is connected to a VPLS provider edge (PE) router of the service provider. Similarly at the data center end, CE-acting FE is connected to another PE router of the Internet service provider. For CEs at both ends, the PE router appears the same as a local Ethernet switch. However, this setup may have a scalability issue since different ports need to be allocated at both CE and PE router interfaces for different customers, so that the total number of customers that can be supported by PEs and CE-acting FEs is limited by the number of ports. To scale it beyond the port limit, QinQ encapsulation can be used between CE-acting FE and PE routers. By doing so, each port can support up to 4K customers, each customer with 4K VLANs.

For customers that allocate cloud resources as different subnets, layer 3 MPLS connection can be used. In this case, the CE-acting FE can serve as a virtual router that connects customer VPN with its data center subnets.

For small businesses or individual users, L2TP/IPSec can be used between the sites to provide basic connectivity and security without assistance from network service providers.

4. FURTHER DISCUSSION

4.1 Security via isolation and access control

In SEC2, each user has its own private network and IP address space. The only way different users can communicate with each other is through FEs, which in turn use firewall, NAT, and other middleboxes to ensure proper access. Such isolation significantly reduces the security risk caused by resource sharing in cloud computing.

For example, the attack outlined in [15] relies on determining co-resident VMs (i.e. VMs on same physical host). This is done by jointly using three methods: (1) determine matching Dom0 IP address via traceroute; (2) test for small round-trip time; and (3) check for numerically close IP addresses.

None of the three methods would work in SEC2. For (1), traceroute is disabled between different customer networks. For (2), all packets across networks are forced to go through FEs even if source and destination colocate at the same physical host, so the round-trip time cannot reveal location. (3) also does not work because each customer has private IP addresses.

Like in many other systems, the overall security of the architecture depends on the security of its components. In SEC2 architecture, isolation among customers can still be compromised if hypervisors, switches, or middleboxes are compromised. Fortunately, security issues in switches and middleboxes are well studied, and hypervisor security has also received attentions from both industry and research community [21, 22].

4.2 Cached vs. installed forwarding table

The MAC forwarding and policy table each FE is a fraction of the global table maintained at CC. There are two options to set up the FE table: either to maintain a cached copy of entries that are actively in use, or to install a complete table that includes MAC addresses of all those within the same subnet of the local VMs in this edge domain, and hence can potentially communicate with the local VMs.

We have assumed using cached table in our discussion in Section 3.3. Cached table is more scalable since typically a node only talks to a small fraction of other nodes, especially at a given point of time. Cached entry is created at FE triggered by an unknown MAC address contained in ARP or data packets. Cached entry is removed after a certain idle time, and can also be replaced by newer entries. The drawback of caching is that it may not be easy to keep track of which entry is cached where. Which option is better largely depends on how subnets are split across edge domains.

4.3 VM migration

There are two ways VM migration may come into play. First, within the data center, the cloud provider may need to move re-optimize the placement of VMs to balance load, save power, or avoid resource fragmentation. In this case, VMs in the data center can be moved across physical machines either within or across edge domains. This is done by transferring dynamic VM state from source to destination hosts. Once state transfer is complete, a gratuitous ARP message is sent from the destination host to announce the VM's new location. Note that this announcement only reaches hosts in the same VLAN in the same edge domain. If both source and destination hosts are in the same domain, FE and CC are not involved in the process. If the VM is migrated to a different domain, then CC updates the VM's location in its table, including both *eid* and VLAN id.

In the second case, the customer may need to migrate VMs between its on-site network and data center. This can be easily achieved if the customer's network and its data center network are configured to be in the same LAN, connected through layer 2 VPN such as VPLS. In this case, FE and CC will register the VM's new location. But from the customer's point of view, the migration procedure is the same as migrating VMs within its on-site LAN. Migration between customer site and data center across different subnets is more challenging. Here a solution proposed in [16] can be used. We can deploy an FE at the edge of customer site network, which can register the location of the VM, and tunnel packets to and from FEs in the data center.

4.4 Implementation considerations

Conventional Ethernet switches can be used in both the core domain and the edge domains. The switches in the core domain are purely for packet forwarding, and hence the topology design is not limited by any policy. In order to ensure better resource usage, shortest-path frame routing [20] or other schemes that allow multiple paths being used should be deployed.

The switches in edge domains need to handle different VLANs. They can be configured in a conventional tree-based topology, rooted at FEs. Conventional L2 forwarding is used in such switches. NIC teaming or Spanning Tree Protocol (STP) can be used to provider redundancy within the edge domain.

The new elements are the CC and FEs. The CC is essentially a directory server for storing configuration information. Standard techniques can be employed for its scalability and reliability. FEs are essentially high capacity Ethernet switches with flow forwarding and tunneling capabilities. They needs to expose an API to allow CC to install flow forwarding entries and policy rules. An OpenFlow or SoftRouter like device is suitable for these functions.

5. RELATED WORK

Much recent work has been focusing on improving data center networks [19, 18, 17, 13, 12]. Some of our design goals are similar to what has been addressed in recent work. Examples are location independence of VMs [19, 18], scalability [19, 18, 17, 13, 12], and utilization of off-the-shelf devices [17]. However, they do not address the issue of creating isolated private networks in data centers, which is the focus of our design. Since our design does not pose any restrictions on the data center fabric, it is possible to apply existing fabric designs such as [12] to the edge or core domain of our design.

Joseph at al. have proposed a policy-aware switching mechanism for the data center network[6]. This fits well into our design: FEs can function as policy-aware switches, and middleboxes such as load balancers and firewalls can be attached to FEs. Depending on the requirement of each customer, traffic can be treated differently and forwarded through different middleboxes.

CloudNet was recently proposed to support virtual private cloud service for enterprises [14], where different customers are assigned different VLANs in the data center. The main limitation is caused by the scalability issues of VLAN, as we have discussed before. This makes CloudNet more suitable for a relatively small number of enterprises.

It is possible to solve VLAN scalability problem by using VPLS. Instead of using FEs to do VLAN remapping, VPLS capable routers can be used to extend VLANs across edge domains. The main differences that we see are: (1) VPLS is a distributed solution, where it may be more challenging to maintain and update policy settings of each customer on-the-fly; and (2) the switches in the core domain need to support MPLS when using VPLS, which is not required in SEC2.

6. CONCLUSION

Security is one major hurdle that cloud computing services need to overcome before their mass adoption. While certain aspects of security rely on solutions outside the technology domain, such as laws, regulations, human resource management and so on, it is important to explore technical solutions to this problem.

In this paper, we have proposed SEC2, a scalable data center network architecture that intends to support secure cloud computing for both enterprise and individual users. It offers effective isolation between different customers while allowing physical resource sharing. Users can specify and manage their individual security and QoS policy settings the same way as they manage the conventional on-site networks. This architecture can also enable users to combine cloud-based resources seamlessly with their existing network infrastructure through VPN.

Unlike prior solutions, SEC2 eliminates the scalability limitation caused by VLANs. The architecture takes advantage of network virtualization and centralized control, using enhanced FE switches at certain aggregation points (i.e., between edge and core domain). For most part of the network, it relies off-the-shelf layer 2 devices for both within each domain and in the core domain.

7. REFERENCES

[1] Global Environment for Network Innovations. *http://www.geni.net*, 2006.

[2] M. Caesar, D. Caldwell, N. Feamster, J. Rexford, A. Shaikh, and J. van der Merwe, Design and Implementation of a Routing Control Platform. In *Networked Systems Design and Implementation*, 2005.

[3] M. Armbrust et al, Above the Clouds: A Berkeley View of Cloud Computing. *http://www.eecs.berkeley.edu*, 2009.

[4] Dark Reading, Security is chief obstacle to cloud computing adoption, study says, *http://www.darkreading.com.*

[5] Network World, Are security issues delaying adoption of cloud computing?, *http://www.networkworld.com.*

[6] D. A. Joseph, A. Tavakoli, and I. Stoica, A Policy-aware Switching Layer for Data Centers. In *ACM SIGCOMM*, 2008.

[7] T. Lakshman, T. Nandagopal, R. Ramjee,

K. Sabnani, and T. Woo, The SoftRouter Architecture. In *ACM HOTNETS*, 2004.

[8] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, Openflow: Enabling innovation in campus networks. *http://www.openflowswitch.org*, 2008.

[9] Juniper Networks, Logical Router Overview. *http://www.juniper.net*.

[10] J. Rexford, A. Greenberg, G. Hjalmtysson, D. A. Maltz, A. Myers, G. Xie, J. Zhan, and H. Zhang, Network-Wide Decision Making: Toward a Wafer-Thin Control Plane. In *ACM SIGCOMM HotNets Workshop*, 2004.

[11] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana, Internet Indirection Infrastructure. In *ACM SIGCOMM*, 2002.

[12] N. Farrington, E. Rubow, and A. Vahdat, Scaling Data Center Switches Using Commodity Silicon and Optics. In *ACM SIGCOMM*, 2008.

[13] C. Guo, H. Wu, K. Tan, L. Shi, Y. Zhang, and S. Lu, DCell: A Scalable and Fault-Tolerant Network Structure for Data Centers. In *ACM SIGCOMM*, 2008.

[14] T. Wood, P. Shenoy, K.K. Ramakrishnan, and J. Merwe, The Case for Enterprise-Ready Virtual Private Clouds. In *HotCloud*, 2009.

[15] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage, Hey, You, Get Off of My Cloud: Exploring Information Leakage in Third-Party Compute Clouds. In *CCS*, 2009.

[16] F. Hao, T.V. Lakshman, S. Mukherjee, and H. Song, Enhancing Dynamic Cloud-based Services using Network Virtualization, In *VISA*, 2009.

[17] C. Guo, G. Lu, D. Li, H. Wu, X. Zhang, Y. Shi, C. Tian, Y. Zhang, and S. Lu, BCube: A High Performance, Server-centric Network Architecture for Modular Data Centers. In *ACM SIGCOMM*, 2009.

[18] A. Greenberg, J. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. Maltz, P. Patel, S. Sengupta, VL2: A Scalable and Flexible Data Center Network. In *ACM SIGCOMM*, 2009.

[19] R. Mysore, A. Pamboris, N. Farrington, N. Huang, P. Miri, S. Radhakrishnan, V. Subramanya, and A. Vahdat, PortLand: A Scalable Fault-Tolerant Layer 2 Data Center Network Fabric. In *ACM SIGCOMM*, 2009.

[20] J. Touch, and R. Perlman, RFC 5556: Transparent Interconnection of Lots of Links (TRILL): Problem and Applicability Statement. *http://www.ietf.org*, 2009.

[21] The Invisible Things Lab's blog, http://theinvisiblethings.blogspot.com.

[22] Z. Wang, and X. Jiang, HyperSafe, a Lightweight Approach to Provide Lifetime Hypervisor Control-Flow Integrity. In *Proceedings of the IEEE Symposium on Security and Privacy*, 2010.