VROOM: Virtual ROuters On the Move

Yi Wang
Princeton University
yiwang@cs.princeton.edu

Jacobus van der Merwe
AT&T Labs - Research
kobus@research.att.com

Jennifer Rexford
Princeton University
jrex@cs.princeton.edu

ABSTRACT

Network management is the proverbial "elephant in the room"—the pressing problem we all know is plaguing the Internet, yet seems intractable to solve. Each new management challenge leads to a new point solution, such as a new configuration script, measurement tool, or protocol extension. In this paper, we argue that many network-management problems stem from the same root cause—the need to maintain consistency between the physical and logical configuration of routers. Instead, we believe that future networks should break this tight coupling by allowing (virtual) routers to freely move from one physical node to another, without changing the IP-layer topology. Our VROOM (Virtual ROuters On the Move) architecture supports live virtual router migration and re-mapping of virtual links, by capitalizing on recent innovations in programmable transport networks, packet-aware access networks, virtual server migration and virtual router technologies. Preliminary experiments with a simple prototype, built using Xen and the Linux routing software, show that VROOM is feasible in practice. We believe that virtual router migration will simplify a variety of network-management tasks, including planned maintenance, service deployment, and minimizing power consumption.

1. INTRODUCTION

Network management is widely recognized as one of the most important challenges facing the Internet. The cost of the people and systems that manage a network typically exceeds the cost of the underlying nodes and links; in addition, more than half of network outages are caused by operator error, rather than equipment failures. From routine tasks such as planned maintenance to the less-frequent deployment of new protocols, network operators struggle to provide seamless service to their customers in the face of changes to the underlying network. Handling change is difficult because each change to the physical infrastructure requires a corresponding modification to the logical configuration of the routers—such as rebooting the router or reconfiguring the tunable parameters in the routing protocols. Any inconsistency between the logical and the physical configuration of the network can lead to unexpected reachability or performance problems.

In this paper, we argue that breaking the tight coupling between the physical and logical configuration of the network is the key to reducing the complexity of network management. Specifically, we propose VROOM (Virtual ROuters On the Move), a new network architecture where virtual routers can freely move from one physical router to another. In VROOM, the physical routers merely serve as the carrier substrate on which the actual virtual routers operate. VROOM can migrate a virtual router to a different physical router without disrupting the flow of traffic or changing the logical topology, obviating the need to reconfigure the virtual routers while also avoiding routing-protocol convergence delays. For example, if a physical router must undergo planned maintenance, the virtual routers could move (in advance) to another physical router in the same Point-of-Presence (PoP). In addition, edge routers can move from one location to another by virtually re-homing the edge links that connect to neighboring domains.

Virtual router migration is possible in practice, by leveraging and combining several recent technical innovations:

Virtual routers: Major router vendors have started supporting router virtualization to simplify network designs, reduce capital expenditure, and lower the barriers to co-location [13, 9]. These technologies allow the partitioning of physical router resources to enable multiple virtual routers to co-exist on the same physical routing platform. The research community is also exploring the use of virtual routers to support multiple independent virtual networks running on a shared substrate [16, 7]. VROOM extends router virtualization to support live migration from one physical router to another.

Virtual machine migration: Server virtualization technology enables a separation between physical and virtual servers[5], allowing for live migration of virtual servers across the LAN [6]. For example, virtual server migration is used in data-center environments to alleviate hot-spots [18]. In our approach, we extend these mechanisms to enable live virtual router migration across a WAN as a network-management primitive.

Programmable transport layers: Programmable transport technologies [12] enable dynamic reconfiguration of the links that interconnect routers. Reconfigurable transport systems capable of reconfiguring long-haul links are commercially available [2] and the theoretic possibility of sub-nano second switching times have been reported [14]. In our architecture we utilize the dynamic re-homing enabled by these transport systems to allow the migration of virtual routers, without causing any changes to the IP-layer topology.

Packet-aware access networks: Access networks are evolving from traditional time-division multiplexing (TDM) transport (access) networks, i.e., T1, T3, OCx etc., to access networks that are packet-aware [17, 4]. Compared with their TDM counterparts, packet-aware access networks allow for more efficient multiplexing (empty frames are not transmitted) and simplified router line cards (no need to terminate large numbers of low rate interfaces on routers). Of particular interest for the VROOM architecture is the fact that interfaces in a packet-aware access network are inherently "virtualized", e.g., associated with a packet label (rather than a physical interface) and as such much more amenable to migration.

We argue that the flexibilities afforded by these technological advances enable a rethinking of the relationship between different network layers and functions. In the work presented here we embark on this rethinking process by exploring the possibilities of a clean separation between logical and physical router functionality to enable live router migration as a network management primitive.

2. NETWORK MANAGEMENT TASKS

The separation between physical and logical, and the migration capability provided by VROOM provide a simple, general solution to many network management challenges, and enables new network management applications. In this section, we briefly discuss three examples.

2.1 Planned Maintenance

Planned maintenance is a hidden fact of life in every network from the day it is commissioned. However, the state-of-the-art practices are still unsatisfactory. For example, software upgrades today still require rebooting the router and re-synchronizing routing protocol states from neighbors (e.g., BGP routes), which can lead to outages of 10-15 minutes [3]. Different solutions have been proposed to reduce the impact of planned maintenance on network traffic. One example is the "cost-out/cost-in" approach commonly used in ISPs, which manipulates IGP link weights to divert traffic away from a router before starting maintenance, and re-attract traffic back afterwards. Another example is the RouterFarm approach of removing static binding between customers and access routers to reduce service disruption time while performing maintenance on access routers [3]. However, we argue that neither solution is satisfactory, since maintenance of *physical* routers still requires changes to the *logical* network topology, and requires (typically human interactive) reconfigurations and routing protocol re-convergence, which usually implies more configuration errors [10] and increased network instability.

With VROOM, network administrators can simply migrate all the virtual routers running on a physical router to other physical routers before doing maintenance and migrate them back afterwards as needed, without ever needing to reconfigure any routing protocols or worry about traffic disruption or protocol re-convergence.

2.2 Service Deployment

Deploying new services, like IPv6 or IPTV, is the life-blood of any ISP. Yet, ISPs must exercise caution when deploying these services, to ensure they have the necessary support systems in place (e.g., configuration management, service monitoring, provisioning, and billing) and do not adversely impact existing services. As such, ISPs usually start with a small trial running in a controlled environment on dedicated equipments, supporting a few early-adopter customers. However, this leads to a "success disaster" when the service warrants wider deployment. The ISP wants to offer seamless service to the existing customers, and yet also restructure their test network, or move the service on to the larger network to efficiently serve a larger base of customers. This "trial system success" dilemma is hard to resolve if the *logical* notion of a "network node" remains bound to a specific *physical* router.

VROOM provides a simple solution by enabling network operators to freely migrate virtual routers from the trial system to the operational backbone. Rather than shutting down the trial service, the ISP can continue supporting the early-adopter customers while continuously growing the trial system, attracting new customers, and eventually moving the service completely to the operational network.

2.3 Power Savings

VROOM not only provides simple solutions to conventional network-management problems, but also enables new network-management solutions to emerging challenges, such as energy efficiency.

It is reported that the total power consumption of the estimated 3.26 million routers in the U.S. is about 1.1 TWh (Tera-Watt hours), as of the year 2000 [15]. This number was expected to grow to 1.9 to 2.4TWh in the year 2005 by three different projection models [15], which translates into an annual cost of about 178-225 million dollars [1]. These numbers do not include the power consumption of the required cooling system.

Although designing energy-efficient equipment is clearly an important part of the solution [8], we believe that network operators can also *manage* a given network in a power-efficient manner. Previous studies have reported that Internet traffic has a consistent diurnal pattern that caused by human interactive network activities [11]. We argue that these traffic variations can be exploited to reduce power consumption. Specifically, the size of the physical network can expand and contract according to traffic demand, by idling or powering down equipment that is not needed. The best way to do this today is to use the cost-out/cost-in mechanism, which inevitably introduces configuration overhead and performance disruptions due to protocol re-convergence.

VROOM provides a cleaner solution: as the network traffic volume decreases at night, operators can migrate virtual routers to a smaller set of physical routers and shutdown or hibernate unneeded physical routers to save power. When the traffic starts to increase, physical routers can be brought up again and virtual routers can be migrated back accordingly. With VROOM, the IP-level topology will stay intact during the migrations, so the power savings does not come at the price of user traffic disruption, reconfiguration overhead and protocol re-convergence. Our initial analysis show that with real traffic data from a large Tier-1 backbone network, applying the above VROOM migration scheme could save 18%-25% of power.

3. VIRTUAL ROUTERS ON THE MOVE

In Section 2 we showed live router migration to be an attractive network management primitive that can enable a number of network management tasks. We argue that, far from a merely hypothetical possibility, a network architecture that supports live router migration can be realized through the combination and extension of a number of enabling technologies.

3.1 Objectives and Challenges

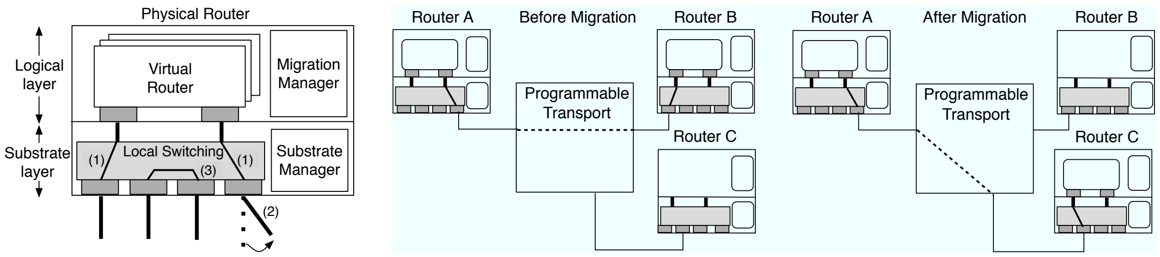
The primary objective of our approach is to be able to move router functionality from one physical piece of equipment to another without any discernible impact, i.e., without requiring the router to be reconfigured, without disturbing the IP-level topology, without causing downtime and without triggering reconvergence of protocols in the logical topology. Further, we would like to not only migrate between routers internal to an ISP network (i.e., between core routers), or between routers co-located in the same PoP, rather, we want to be able to migrate edge routers that connect to customer networks and be able to migrate routers between PoPs if needed.

Realizing these objectives presents a number of challenges: (i) The first obvious challenge is that the functionality to be migrated has to be separable from the physical equipment that it is migrated to and from (migratable routers). (ii) Because we want the topology to stay intact, the next challenge is for link level connectivity to "follow" the migrated router to its new location so that packets to and from other routers can continue their traversal through the network (migratable links). (iii) So as not to trigger any protocol outages or re-convergence, the migration should happen rapidly and with minimal packet loss (minimal outages). (iv) Finally, to enable the migration of edge routers, the migration should not require the cooperation of connected routers, i.e., customer or peer routers, to enable the migration (edge migration).

3.2 VROOM Architecture

We now describe the VROOM architecture and show, with the aid of the nodal building blocks illustrated in Figure 1a, how each of the above challenges are addressed.

Migratable Routers: Being able to move router functionality from one piece of physical equipment to another calls for clean separation between logical router functionality and the physical equipment on which that functionality executes. Further, any binding between the logical functionality and the physical nodes and links should be dynamically changeable. This separation is shown in nodal architecture in Figure 1a: Each router in the VROOM architecture principally consists of a logical layer and a substrate layer. The logical layer contains *virtual routers*, which are configured as core or access routers. They form the equivalent of today's physical IP level routers and topology by running an IGP to maintain the intra-AS topology, maintaining BGP peering sessions with routers in neighboring ASes, etc. The primary purpose of the substrate layer is to provide a dynamic binding between the interfaces of the virtual routers and interfaces and links in the substrate layer. In-



(a) Node architecture

(b) Programmable transport re-homing with interface re-binding

Figure 1: Node architecture and consolidation examples

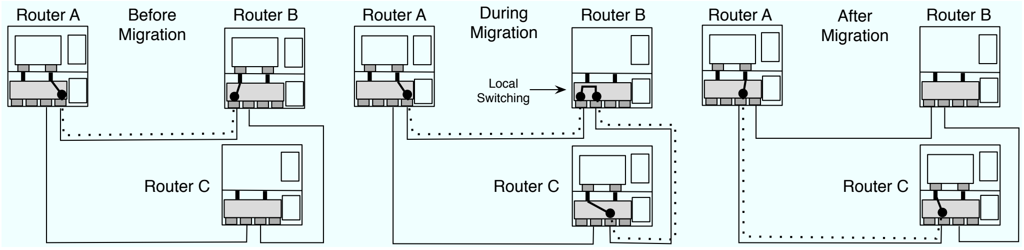


Figure 2: Tunnel re-homing with interface re-binding

terfaces and links in the substrate layer can be either physical or virtual (e.g., tunnel based interfaces and links).

We note that the essence of this separation between logical and physical is available in commercial routers [9]. For our architecture we require two extensions: (i) to dynamically change the binding between a virtual router and its physical host (i.e., to re-map logical interfaces of the virtual router to physical interfaces and links), and (ii) to migrate the virtual router from one physical router to another. These actions are respectively depicted by (1) and (2) in Figure 1a. Given the sophisticated forwarding that modern routers are capable of, the dynamic binding appears imminently doable. Virtual router migration involves migration of both the control plane and the lookup tables that constitute the forwarding information base (FIB) in the data plane. For migrating the control plane we assume that the control plane functions of each virtual router runs in a virtual server which allows us to make use of virtual server migration technologies for its migration [6]. We extend this functionality to also migrate the FIB to ensure a consistent control plane and data plane is activated after migration.

Migratable Links: Migrating links to "follow" the migrated virtual router involves the physical or logical re-homing of links between routers. Programmable transport networks [2, 3] allow physical port links to be switched. For example, as shown in Figure 1b, a port on router A that connects to a port on router B, can be switched at the transport layer, so that the port is now connected to a port on router C. Further, this switching can be performed across a system of transport switches, so that this link re-homing can be performed across wide area links to enable inter-PoP link re-homing. Virtual links (or tunnels) can also be utilized to effect link re-homing. For example in the case of GRE tunnels, re-homing involves the changing of tunnel destination addresses. In such cases, the substrate layer can itself be an IP network (separate from the logical layer), whose sole purpose is to provide connectivity between physical routers and serve as tunnel endpoints for tunnels used by the logical layer.

Minimal Outages: For the virtual router migration part, the key to minimizing outages is to minimize the time that the virtual router is not operational. Once the virtual router is operational on the new physical platform, the challenge is to migrate appropriate links as quickly as possible and re-map the virtual router interfaces as appropriate. If re-homing and re-mapping could be performed in an instantaneous, coordinated and distributed manner across all routers involved in a migration, that would take care of the migration functionality at the substrate level. However, given the technical challenges of meeting such requirements we introduce *local switching* between substrate interfaces ((3) in Figure 1a). The use of this primitive is shown in Figure 2. In this case we utilize the local switching function to redirect traffic on the existing tunnel between router A and B, to and from a new tunnel between routers B and C, as soon as the virtual router migration is complete. This local switching can be done instantaneously when the final migration happens, while re-homing of links and re-mapping of virtual routers to physical interfaces can then be performed at more relaxed time scales.

Edge Migration: As is the case with "migratable links", programmable transport can be utilized to migrate access routers. Again referring to Figure 1b, a programmable transport access network can be used to re-home a router interface to a different physical router interface to facilitate virtual router migration. The appeal of this approach is that no cooperation is required from customer or peer routers [3]. Further, evidence exists that such switching can be performed at sub-nano second time scales [14]. Current programmable transport networks are "circuit" oriented in nature. I.e., either a TDM circuit or an optical wavelength is being manipulated in the programmable network. The downside of circuit based transport networks is that customer access ports are "directly" connected to the access routers, which means that each customer access port need to be terminated on the PE router to which it connects. This is illustrated in the top part of Figure 3 which shows a (significantly simplified) view of a TDM based transport network connecting two customer edge (CE) routers to a provider edge (PE) router. As is shown in the figure, the low speed access links can be multiplexed together through the transport network, however, these low speed circuits need to be de-multiplexed again to terminate the links on the PE[1]. In contrast, the bottom part of Figure 3 shows a packet-aware transport network, e.g., a VPLS access network, where de-multiplexing at the physical interface level is not necessary. Rather, each access port is associated with a label, or "pseudo wire", which allows the PE to disambiguate each access port as a separate virtual interface, without requiring a separate physical port on the PE. Commercial access networks are evolving to packet-aware transport networks [17], which is of importance for the VROOM architecture because it reduces the need for a per customer physical interface on PE routers, thus greatly simplifying the migration of virtual routers.

4. PROTOTYPE EVALUATION

We implemented a prototype that demonstrates the feasibility of the live virtual router migration mechanisms. Our evaluation results show that, even with a Linux based software router executing in a standard virtual machine monitor (Xen), which is not optimized for routing, the proposed VROOM mechanisms can provide live virtual router migration with very small downtime.

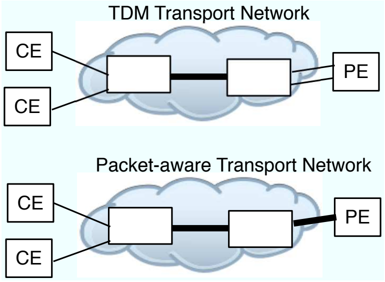


Figure 3: TDM versus Packet-aware access networks

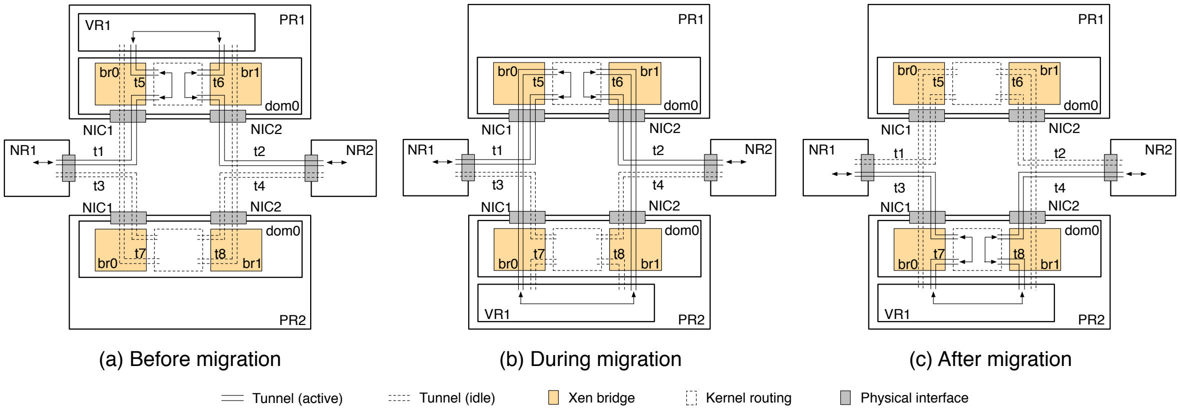
4.1 Prototype Setup

Our prototype platform has a four-node physical topology that consists of four servers, as shown in Figure 4. PR1 and PR2, which have identical installations of Xen 3.1 on Linux 2.6.18, serve as the source and destination physical routers for the virtual router migration. NR1 and NR2 are traffic generators. PR1 and PR2 each have two single-core Intel Xeon 3.6GHz processors, 2.6GB memory, and a dual port Intel Pro/1000 MT network adapter. The Xen virtual machine (which serves as a virtual router in our experiments), is configured with access to one Xeon 3.6GHz processor and 512MB memory.

The prototype is configured in such a way that NR1, NIC1 of PR1 and NIC1 of PR2 are on a subnet A, and NR2, NIC2 of PR1 and NIC2 of PR2 are on another subnet B. As part of Xen's network virtualization solution (the bridging mode), when PR1 and PR2 boot up, Xen associates each physical interface with a bridge in its privileged domain 0. When the virtual router VR1 is initially created on PR1, it associates its eth0 with bridge br0, and eth1 with bridge br1, and appears on both subnet A and B as an individual host. Xen's built in live server migration mechanism relies on the use of an unsolicited ARP reply from the migrated host to advertise that its IP has moved to a new location (which limits the Xen virtual server migration to a single LAN). To demonstrate that VROOM's virtual router migration mechanism is generally feasible regardless of the underlying physical/link layer technologies (ethernet, SONET, etc.), we create GRE tunnels between NR1/NR2 and PR1/PR2, so that the evaluation results we get do not rely on Xen's ARP reply mechanism.

In our prototype, we use the Linux kernel routing functionality (*iptables* and *iproute2*) of domain 0 to emulate the substrate layer of the VROOM architecture. As shown in Figure 4a, the traffic between NR1 and NR2 flows through a series of four tunnels: t1–

[1] In reality each low speed circuit/link does not require a separate physical interface as several circuits can terminate on a so called channelized router interface. However, the complexities involved with the de-multiplexing process means that channelized line cards suffer from low port densities so that the argument holds.



(a) Before migration

(b) During migration

(c) After migration

Figure 4: Experiment of live virtual router migration on a four node prototype network

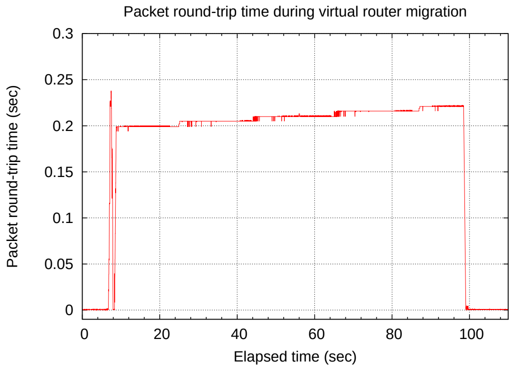


Figure 5: Effect on packet round-trip time of migrating a virtual router

t5–t6–t2 before the migration. Note that there are four other tunnels set up but idle (with no traffic) in the same figure: t3 and t4 emulate the IP connectivity between NR1/NR2 and PR2. t7 and t8 are tunnels pre-setup for the migration. Since the data plane of our Linux-based router is also embedded in the virtual server, the virtual router migration is complete after the virtual router VR1 is migrated from PR1 to PR2 using Xen's live server migration mechanism (Figure 4b). At this point all traffic to and from VR1 is still taking a detour via PR1. Then, three re-mapping commands are executed to make NR1 switch to t3 from t1, NR2 switch to t4 from t2, and VR1 to route between t7 and t8 instead of t5 and t6, respectively (Figure 4c). The kernel routing table of domain 0 of PR2 can be preset and thus is out of the critical path.

4.2 Experiment Results

We migrate VR1 from PR1 to PR2 while sending continuous *ping* packets from NR1 to NR2 with 1 millisecond interval. Figure 5 illustrates the change of packet round-trip time during the migration. At the end of the migration process (around t = 98.47 sec), there is a downtime of 563 msec. This is likely due to the Xen migration implementation because similar behavior is also reported in the original Xen live migration experiments [6]. The increase of delay during the migration is likely due to the increased queueing delay at PR1's network buffer, as a result of the migration traffic.

In practice, we expect that VROOM can hide the delay for migrating a virtual router from one physical router to another. Conventional routers have a clear separation between the control plane (running in user space) and the data plane (running in the kernel, or on dedicated line cards). As such, it is possible to migrate a virtual router to a new location, while continuing to forward data packets through the old location during the transition. Once the migration process completes, the packets can start traveling through the new location and the original forwarding table can be safely flushed. We are building a prototype of this solution as part of our ongoing work.

5. DECIDING WHERE TO MIGRATE

Ideally, a virtual router should be able to move to any physical router. However, in practice, certain physical constraints limit where a virtual router can go. These constraints include:

Latency: The new home for the virtual router should not substantially increase the propagation delays of the virtual links, to avoid degrading application performance. For example, moving a virtual router from New York City to Washington, D.C. adds just 2 msec of round-trip propagation latency, but moving a virtual router over thousands of miles would lead to noticeable increases in latency. We envision that, in practice, a virtual router would move to another physical router in the same geographic region.

Link capacity: Migrating a virtual router moves its traffic load to a new set of underlying links. These links should have sufficient unused capacity to accommodate the extra traffic. We envision that automated tools would combine the topology information with traffic measurement data to determine where a virtual router can move without overloading any of the underlying links.

Platform incompatibilities: Routers from different vendors may not support the same operating system, routing-protocol features, or migration techniques, making it difficult to move a virtual router from one vendor platform to another. As such, a virtual router may be limited to migrating to another physical router built by the same vendor. Fortunately, most networks consists of routers by one or at most two different vendors, leaving many possible physical routers that can host each virtual router. (Though today's routers are far from being binary compatible, we could envision a future where all routers run a common, vendor-independent "router hypervisor" to support seamless migration across routers from different vendors.)

Router capabilities: Even when all routers come from the same vendor, different physical routers might have different capabilities. For example, routers may differ in the number of access-control lists (ACLs) they can support, or in the measurement functionality implemented on the line cards. As such, a virtual router could only move to physical routers that support the required features.

Each of these practical constraints limit the set of physical routers that can serve as a new home for the virtual router. Fortunately, these constraints limit the size of the search space, making it easier to determine an appropriate place for the virtual router to go. In our ongoing work, we plan to formulate the router migration problem as a constrained optimization problem, and to propose automated solutions for a variety of practical contexts, including the three examples in Section 2.

6. CONCLUSION

In this paper we argue that the need to maintain consistency between the logical and physical topologies in IP networks is the root cause of many network management challenges. VROOM breaks the coupling by allowing a virtual router to move from one physical router to another. We argue that virtual router migration can be realized by combining and extending a number of emerging technologies, and illustrate the feasibility of our approach with a prototype system.

We are investigating several aspects of VROOM in depth in our ongoing work, including (1) formulating the migration scheduling as a constrained optimization problem and solving it in an automatic fashion, (2) minimizing the forwarding disruption introduced by migration by migrating the control plane and data plane of a router in separate steps, and (3) exploring other applications that could benefit from VROOM, such as tolerating unplanned network failures. We expect our approach to foster a new breed of network management applications which have been inhibited by current network architectures.

7. REFERENCES

[1] Average retail price of electricity from Department of Energy. `http://www.eia.doe.gov/cneaf/electricity/epm/table5_6_a.html`.

[2] Ciena CoreDirector Multiservice Switch. `www.ciena.com`.

[3] M. Agrawal, S. Bailey, A. Greenberg, J. Pastor, P. Sebos, S. Seshan, J. van der Merwe, and J. Yates. RouterFarm: Towards a dynamic, manageable network edge. In *Proc. ACM SIGCOMM Workshop on Internet Network Management (INM)*, September 2006.

[4] W. Augustyn and Y. Serbest. Service Requirements for Layer 2 Provider-Provisioned Virtual Private Networks. IETF RFC 4665, Sept 2006.

[5] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebar, I. Pratt, and A. Warfield. Xen and the Art of Virtualization. In *Proc. Symposium on Operating Systems Principles (SOSP)*, October 2003.

[6] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield. Live Migration of Virtual Machines. In *Proc. Networked Systems Design and Implementation*, May 2005.

[7] N. Feamster, L. Gao, and J. Rexford. How to lease the Internet in your spare time. *ACM SIGCOMM Computer Communications Review*, Jan 2007.

[8] M. Gupta and S. Singh. Greening of the Internet. In *Proc. ACM SIGCOMM*, August 2003.

[9] Juniper Networks. Juniper Logical Routers. `www.juniper.net`, 2007.

[10] Z. Kerravala. Configuration Management Delivers Business Resiliency. The Yankee Group, November 2002.

[11] A. Lakhina, M. Crovella, and C. Diot. Diagnosing Network-Wide Traffic Anomalies. In *Proc. ACM SIGCOMM*, August 2004.

[12] G. Li, D. Wang, J. Yates, R. Doverspike, and C. Kalmanek. Detailed study of IP/reconfigurable optical networks. First International Conference on Broadband Networks (BROADNETS'04), 2004.

[13] D. McPherson et al. Core Network Design and Vendor Prophecies. In *NANOG 25*, Toronto, Ontario, Canada, June 2003.

[14] A. Rostami and E. Sargent. An optical integrated system for implementation of NxM optical cross-connect, beam splitter, mux/demux and combiner. IJCSNS International Journal of Computer Science and Network Security, July 2006.

[15] K. Roth, F. Goldstein, and J. Kleinman. Energy Consumption by Offic and Telecommunications Equipment in commercial buildings Volume I: Energy Consumption Baseline. National Technical Information Service (NTIS), U.S. Department of Commerce, Springfield, VA 22161, NTIS Number: PB2002-101438, 2002.

[16] J. S. Turner. A proposed architecture for the GENI backbone platform. In *Proc. Architecture for Networking and Communications Systems*, December 2006.

[17] J. Wei, K. Ramakrishnan, R. Doverspike, and J. Pastor. Convergence through packet-aware transport. *Journal of Optical Networking*, 5(4), April 2006.

[18] T. Wood, P. Shenoy, A. Venkataramani, and M. Yousif. Black-box and Gray-box Strategies for Virtual Machine Migration. In *Proc. Networked Systems Design and Implementation*, April 2007.