ELK ASIA PACIFIC JOURNAL OF COMPUTER SCIENCE AND INFORMATION SYSTEMS

Volume 1 Issue 1, September (2014)

ELK Asia Pacific Journals

www.elkjournals.com

STUDY ABOUT NETWORK MANAGEMENT PRIMITIVE: VROOM

Prof. M. H. Pandey
Department of Computer Science & Information Technology,
Moradabad Institute of Technology,
Uttar Pradesh, India

ABSTRACT

One of the biggest challenges recognised being faced on Internet today is the Network management complexity. Point solutions for individual issues further add up to the system complexity since they don't address the underlying causes. This research article tries to argue that many network management problems spring up from the same root cause, i.e. the need to maintain consistency between the physical and logical configuration of the routers. Hence, VROOM (Virtual ROuters On the Move) was proposed in the study. This network management primitive allows virtual routers to move freely from one physical node to another with the objective to avoid unnecessary alterations to the logical topology. In addition to this, VROOM can also help in emerging challenges like reducing energy consumption.

Keywords: Design, Virtual router, Internet, Network management

1. Introduction

Management of network remains widely noted as the most significant challenge for Internet. Cost of people as well as systems getting managed for networking gets noted for being typically exceeding cost of nodes as well as links; added by outrages for various networking led by errors of operator, instead of failures of equipment. From tasks noted in routine like planned maintenance towards less-frequent service deployment related to newer protocols, struggle among network operators for offering seamless service as changes for the network. Managing change is very hard as every change towards physical infrastructure in response to corresponding modification towards logical configuration over routers, like reconfiguring tunable parameters within routing protocols. Logical approach to IP packet-forwarding acts as physical refers towards physical router tool (like line cards as well as CPU) enabling these functions. Inconsistency among logical configuration and physical configuration leading to unexpected reach or performance issues. Moreover, tight coupling in current scenario among physical topology and logical topology, at times might turn for logical changes as implied purely in the form of equipment for handling physical changes gracefully.

This can be exemplified by the increasing connection of weights within Interior Gateway Protocols towards "cost out" router within advance kind of maintenance of plan. This case is for change within logical topology without any goal, instead of indirect tool for the attainment of task handy, without any potential negative side-effects. This research argues breaking of tight coupling among physical configuration and logical configurations offering single, general mode of abstraction simplifying management of network. We in particular proposed VROOM (that is Virtual Routers on the Move), with newer management process of network primitive, whereby virtual routers can move freely from a physical router to other. In implying VROOM, physical routers hardly serve in being the carrier of substrate over actual virtual routers in the form of operation. VROOM can be migrated by virtual router on different physical router without any kind of disruption over traffic flow or changing logical topology that is obviating demands for reconfiguring virtual routers as for ignoring routing-protocol for delays in convergence. As for instance, in case of physical router there is the need for undergoing maintenance of plan as virtual routers can never move (advance) towards other physical router within same Point-of-Presence (PoP). Moreover, edge routers are liable to move from a place to other by virtual rehoming connections towards neighboring domains.

Realization about the these objectives over different kinds of challenges:

(i) migratable routers: in order to migrate virtual router, whereby the "router" actions need to remain separable from physical tools that it runs;

(ii) minimal outages: in order to avoid disruption of user traffic or otherwise triggering routing reconvergence of protocol, whereby the migration must remain away or get minimal loss of packet;

(iii) migratable links: in order to consider IP-layer topology intact, wherebyteh connections remain attached towards migrating router that relevantly "*follow*" newer location.

In reference to the 3rd challenge under recent advances within transport-layer approaches (as in Section 2), we aimed in functionality of migrate router that is noted from a tool to another, without any disruption made over IP-layer topology or otherwise data traffic as carried and keeping without router reconfiguration. Superficially, migration of virtual router can appear straight-forward mode of extension for the current migration of virtual machine approach. This can be inclusive of copying image of virtual router (added by binaries of routing-protocol, files of configuration and state of data-plane) towards newer physical router added by freezing of running processes prior to copying. State of data plane and approaches can be restored over newer physical router as well as migrated links. Still, delays in the process of accomplishing all such steps is liable to lead to unacceptable mode of disruptions in relevance with data traffic and further notes of routing protocols. In terms of migration of virtual route on practical grounds, packet forwarding never gets interrupted, even not on temporary basis. As against this, control plane remains tolerant to brief disruptions, as the protocols for routing comprises retransmission approaches. However, the control plane needs to restart itself on a faster pace and new location in order to ignore losing of adjacencies of protocol added by other routers as well as for the minimisation delay in terms of dealing with unplanned events of the network.

In case of VROOM, we are liable to minimize disruption through the process of leveraging separation made for control plane and data plane with modern routers. We hereby introduced data-plane hypervisor in the form of a migration-aware interface that is noted between control plane and data plane. This is related to the unified interface that permits us to assist migration among the physical routers along with various technologies of data plane. VROOM migrates in control plane, and forward traffic by means of older data plane. Control plane is liable to begin as well as run in newer location, along with populate newer data plane. This further updates older data plane simultaneously. In the time of transition, old router inclines to redirect routing of the protocol traffic towards newer location. As the data plane gets fully populated in newer location, migration of link starts. 2 data planes can act simultaneously to facilitate asynchronous mode of migration for such links.

In order to geenralize our derived our data plane hypervisor, we are inclined to offer 2 prototypes of the VROOM routers. These are: software data plane (Linux kernel) and hardware data plane (NetFPGA card). Virtual router runs with Quagga routing suite within OpenVZ container. In context of the software extensions there comprise 3 basic modules-

(i) separation forwarded by tables from container contexts,
(ii) push forwarding of the table entries that gets generated through Quagga within separate data plane, and
(iii)dynamical bind in reference to virtual interfaces as well as forwarding tables.

Our approach is in support of seamless mode of live migration related to virtual routers among 2 data planes. Our research leads to migration of virtual router causing no packet loss/delay as the hardware data plane gets implied over fewer seconds in terms of delaying processing of the messages from control-plane.

Remainder attained from paper gets structured as-

Section 2 offering background over flexible transport networks, added by overview of the relevant work.

Section 3 illustrating the way migration of the router getting simplified over the management of the current networking tasks, like planned maintenance as well as deployment of service, whereas addressing regular challenges are noted for management of power.

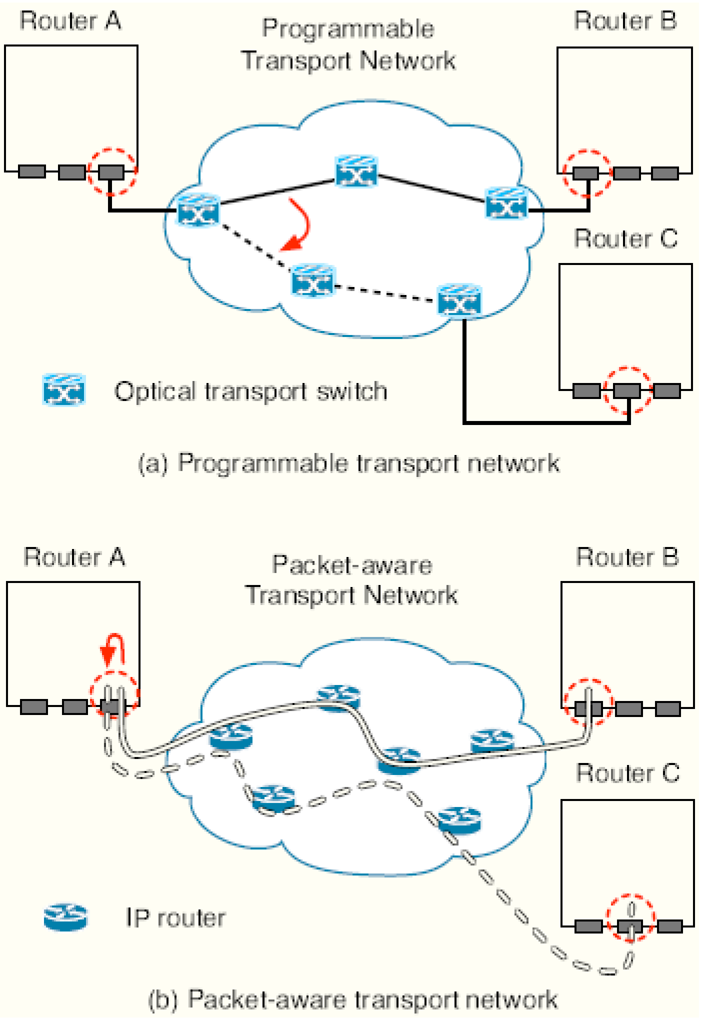


Figure 1: Link migration in the transport networks

We have presented VROOM architecture with Section 4, added by the implementation as well as assessment in Sections 5 and 6, respectively. We further discussed current work over migration scheduling in determined Section 7 as well as concluded in Section 8.

2. Background

A basic need for VROOM is the “link migration”, which is related to virtual router and must “follow” migration from a physical node to other physical node. It has been made possible through the emergence of transport networking tools. We thus illustrate these tools or technologies in a brief way here, with an overview of related relevant work.

2.1 Migration of Flexible Link

In a very basic form a link related to IP layer remains in connection with direct physical link (as cable), connecting migration hard as this remains physically moving to the end of the connection(s). Still, on a practical note, direct connections at IP layer in general remains correspond towards series of connections by various elements of the network at transport layer. As for instance, in current ISP backbones, "*direct*" mode of physical links remain typical to the optical transport networks, that comprises IP link in relation with a circuit traversing over multiple optical switches. In recent advancements in terms of programmable transportation networks there is the room for physical links among the routers to remain dynamically placed and get torn down. As for instance, as in Figure 1(a), connection among physical routers A and Bi gets switched by programmable mode of transport networking. Through signaling transport network, similar physical port on router A gets related to router C after switch of optical path. This kind of path is for the switch-over at transport layer that can get efficiently structured as in sub-nanosecond optical span of switching time. Moreover, this can happen over wide ranged networking area of transport switches that can enable migration between inter-POP links. Added to this, basic links in ISP, there is the provision of attaining migrating accessibility of the links towards customer edge (CE) routers as well as provider edge (PE) routers, where, just PE end of respective links remains under the control of ISP. On a historic note, access links remain connected to the underlying accessibility of the network, like T1 circuit within time-division multiplexing (TDM) of the access network. In these kinds of instances, migration of access link is liable to get accomplished within same way through the process as noted in Figure 1(a). This refers to the switching towards newer circuit at switch being directly connected towards CE router. Still, traditional kind of circuit-switched for the access networks, remains dedicated towards the physical port over PE router as needed to terminate each of the TDM circuit. Thus, in case all ports over physical PE router remain inuse, it won't be able to accommodate virtual routers any more. Still, as Ethernet emerges under economical as well as flexible alternative towards legacy of the TDM services, whereby there is the evolution of access networks to packet-aware transport networks, offering significant benefits for the VROOM through the process of eliminating demands/customer physical ports over the PE routers. In case of packet-aware access network (as virtual private LAN service access network), ever consumer access port needs to remain in connection with label/"pseudo wire", that permits PE router assist multiple links for logical access over similar physical port. Pseudo-wire access migration of the link comprises newer pseudo wire as well as switching to multi-service switch adjacent towards CE. As against conventional ISP networks, there are few networks noted in terms of overlaying top of some other networks of ISP. Examples remain inclusive of commercial "*Carrier Supporting Carrier* (or the CSC)" networks, added by VINI, notably research of virtual network infrastructure laid on National Lambda Rail as well as Inter-net2 [32]. In these kinds of cases, single-hop link added by lay network remains real over multi-hop path within underlying network that can remain in terms of MPLS VPN (as CSC) or IP network (as in VINI). Process of link migration within MPLS transport network includes act of switching over towards newer label switched path (or the LSP). Migration of link in an IP network can get initiated through the process of changing IP address over tunnel end point.

2.2 Related Work

Motivation by VROOM remains same to the work of RouterFarm, especially for reducing impact of planned-maintenance through migrating router that stands for the functionality from a place within respective network to another. Still, RouterFarm performs "cold restart", as against live ("hot") migration under VROOM. On a specific note, RouterFarm router is related to migration noted through reinstantiation of router instance at newer area that demands for router reconfiguration and introduces inevitable span of downtime within both control plane and data plane. In reference to

VROOM, we perform migration of live router without reconfigurating or otherwiae discernibling disruption. From our initial prototype related to VROOM, there are router migration noted directly by the application of standard migration of virtual machine capability offered by Xen. This lacked separation of control plane and data plane as in this paper. Consequently, it comprises data-plane down time in the span of migration process. According to latest developments in the virtual machine technologies as well as live migration approaches there is the scope to leverage in server-management equipments, basically in context of data centres. As for instance, Sandpiper is subject to get virtual servers migrated automatically across pool of physical servers in order to alleviate hotspots. As per Usher permission is offered to administrators for expressing different kinds of policies in terms of managing virtual server clusters. Remus implied replication of asynchronous virtual machine in order to offer high note of availability towards respective server over hardware failures. As against this, VROOM laid emphasis over the act of leveraging the way of live migration for simplifying networking domain management. Network virtualization gets proposed under different contexts. Former work comprises concept of “switchlets”, where ATM switches get partitioned in order to enable dynamic mode of creation related to virtual networks. In the recent past, CABO architecture implies the implication of virtualization that stands for enabling multiple service providers for the act of sharing physical infrastructure. Away from research community, the router virtualization has turned available under different forms of commercial routers. In case of VROOM, we considered added step and for virtualization router functions decoupled with virtualized router in terms of physical host and thereby enabling migration.

VROOM refers to latest work over the act of minimizing disruptions of transient routing in the process of maintaining plan. Assessment of large ISP lead towards more than ½ of the changes in the routing that are planned in advance. Network operators are liable to limit disruption through the process of reconfiguring routing protocols in order to direct traffic in a different route from the equipment under maintenance. Moreover, extensions related to routing protocols permits router for the continuation of forwarding packets within data plane as reinstalling/rebooting control-plane software. Still, these approaches demands changes towards logical configuration/routing software. As against this, VROOM hides effects led by the changes of physical topology within first place, obviating demand for solutions which can increase complexity of the system as it enables new network-management approaches as illustrated in the following section.

3. Network management tasks

This section discusses about 3 case studies related to the applications of VROOM. We have showed separation among physical as well as logical aspects added by the router migration features enabled through VROOM that can immensely simplify current tasks under network management. It further offers solutions of network management to other challenges. We illustrated the reasons for current solutions (in first 2 examples) which are not of much satisfaction and overviews VROOM by means of addressing similar issues.

3.1 Planned Maintenance

The process of planned maintenance remains hidden to life in every network. Still, practices of state-of-the-art remain unsatisfactory. As for instance, software upgrades in the current scenario, still demands for booting router added by the act of resynchronizing protocol of routing from neighbors (as BGP routes), leading to outages noted within 10 to 15 minutes. Various solutions are noted for the reduction of impact led by the planned maintenance over network traffic, like “costing out” that is an advanced tool. Further there is the RouterFarm implementation for the removal of static binding among customers and the relevant access routers for the reduction of service disruption span in terms of performing maintenance over access routers. Still, we comment that neither solution remains satisfactory, as maintenance made over physical routers demands changes towards logical network topology; and demands (human interactive) appropriate reconfigurations added by routing of the protocol reconvergence. These are subject to imply more errors in configuration and further increase instability of the network. We performed as well as analysed events for planned-maintenance conducted over Tier-1 ISP supported within a week. Space limitations as mentioned by us are related to higher notes of results that remain pertinent towards VROOM, in this context. Our research states that amidst planned-maintenance of respective events, there is undesirable impact over the network in the current scenario (as routing protocol reconvergence/data-plane disruption), of which 70% can remain conducted without the impact of network, in case VROOM get used. (it leads t the state of migration among routers in reference to control planes, etc. with better migration strategies, like "control-plane hypervisor" permitting migration among routers with various implementation of control plane, the count increases by 90%.) This is caused by planned-maintenance of the respective events, in relation with hardware and without the intention to make changes any longer to configurations of logical layer. In order to perform planned maintenance in VROOM based network, administrators get the scope to migrate running virtual routers over one physical router to other, prior to maintenance as well as migrating them afterwards without demanding for reconfiguration of any routing protocols, otherwise get worried in relation with traffic disruption or otherwise protocol reconvergence.

3.2 Evolution and Service Deployment

Deployment of new services, such as IPv6/IPTV, is noted as life-blood related to any ISP. Still for ISPs need to exercise caution as the act of deploying newer services. Firstly, they need to assure newer services that will not lay adverse impact over current services. Secondly, necessary system support demands for the place prior to services that is apt for support. (Support systems comprises of management of configuration, monitoring service, provisioning as well as billing.) Thus, ISPs in general starts with smaller trial within a controlled environment over dedicated equipment, assisting some early-adopter customers. Still, the same results with "success disaster" as service warrants are wide ranged for deployment. ISP demands to provide seamless service to current customers, and restructure test network, or otherwise remove service over larger serving periphery of network for customers. Such dilemma of “trial system success” remains hard to resolve in case the logical notion over “network node” get bound to determined physical router.

VROOM offers solution through the process of enabling network operators that can migrate virtual routers freely from trial system to operational backbone. Instead shutting down of trial service, meant for ISP can still support early-adopter customers was the same grows continuously over trial system, attracting newer customers, and seamlessly migrating operational networking service. ISPs in general deploy service-oriented routers, closer to customers as much as possible, to ignore back haul traffic. Still, as services develop, geographical distribution as noted among customers can change in due course of time. With the implication of VROOM, ISPs is liable to reallocate easily over routers in terms of adapting newer demands of the customer.

3.3 Power Savings

VROOM offers simple solutions in context of tasks related to the management of conventional network and enables newer solutions for the act of emerging challenges like power management. In 2000, total consumption of power has been estimated as 3.26 million routers within the U.S. which is about 1.1TWh (Tera-Watt hours). This will further increase to 1.9 to 2.4TWh by 2005 with 3 various projection models [28] that can translate annual cost of 178 to 225 million dollars, without including consumption of power in the cooling systems.

Though designing tools for the energy-efficiency are significant for obtaining solution, we consider network operators as efficient enough in managing network with more dominant power efficiency manner. According to former research, internet traffic turns up to remain consistent in a diurnal pattern led by human interactive activities in the network. Still, in the current scenario, routers remain surprisingly insensitive to power for traffic loads as has been handled, which remains as an idle router consuming more than 90% of necessary power at maximum capacity. We contradict it as VROOM is for variations within daily traffic volume which is liable to get exploited in terms of reducing consumptions of power. On a determined note, size of physical network can get expanded as well as shrunk as per traffic demand, through hibernation or powering down routers which are not needed. Best way is the implementation of *"cost-out/cost-in"* process that stands inevitably in terms of configuration of performance disruptions caused by protocol reconvergence.

VROOM offers cleaner solution: since the volume of the network traffic volume reduces at night, respective virtual routers are liable to migrate towards smaller physical routers as well as unnecessary physical routers can shut down/ put into hibernation for saving power. As the traffic increases, the physical routers is liable bring in and respective virtual routers can migrate back. By the assistance of VROOM, IP-layer topology remains intact in the time of migrations, in a way that the saving of power never appears at the price of disruption, relevant reconfiguration of the user traffic, overhead/protocol reconvergence. As per the analysis led by us volumes of data traffic within Tier-1 ISP backbone leads to the phase even in case of migrating virtual routers in same POP whereas considering similar link by the use of rate above VROOM approach for managing power, saving 18 to 25% of power a needed to run network routers. In Section 7 it is discussed that leading of migration over various POPs can lead to more substantial modes of saving power.

4. VROOM ARCHITECTURE

Here, we have noted VROOM architecture, by illustrating 3 building-blocks making migration of virtual router possible, whereby router virtualization, control plane and data plane get separated as well as there are binding of dynamic interface. This further leads to the process of migration of VROOM router. As against this, there are regular servers related to modern routers that are physically separating control plane and data plane. By means of such provision we came up with data-plane hypervisor among control plane and data plane in order to enable virtual routers towards migration over various platforms of data plane. Further illustrations are made about the 3 migration approaches that can minimize downtime of control plane and further eliminate disruption of data plane cloning, remotely placing control plane and doubling the data planes.

4.1 Migration of Virtual Routers

As noted in Figure 2, VROOM architecture for router assists in migration of virtual router, in 3 determined features, which make migration possible. These are - router virtualization, separation of control plane and data plane, and dynamic mode of interface binding, available within high-end commercial routers.

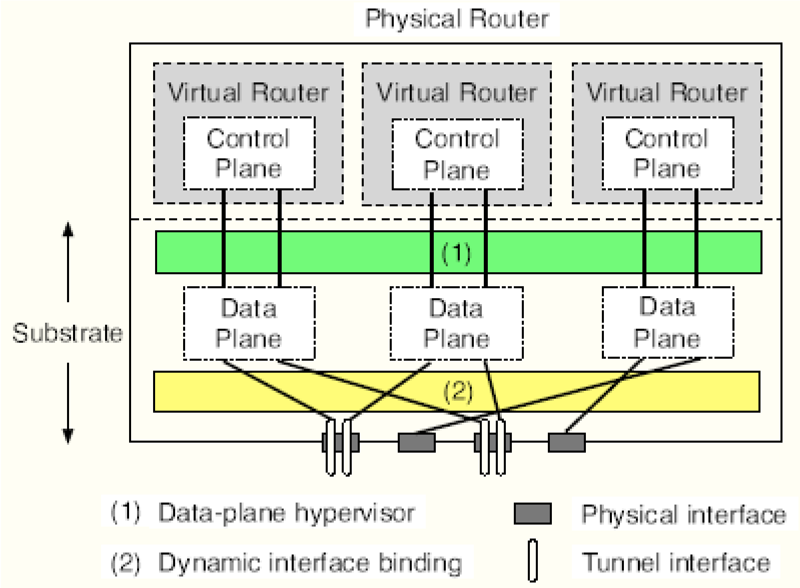
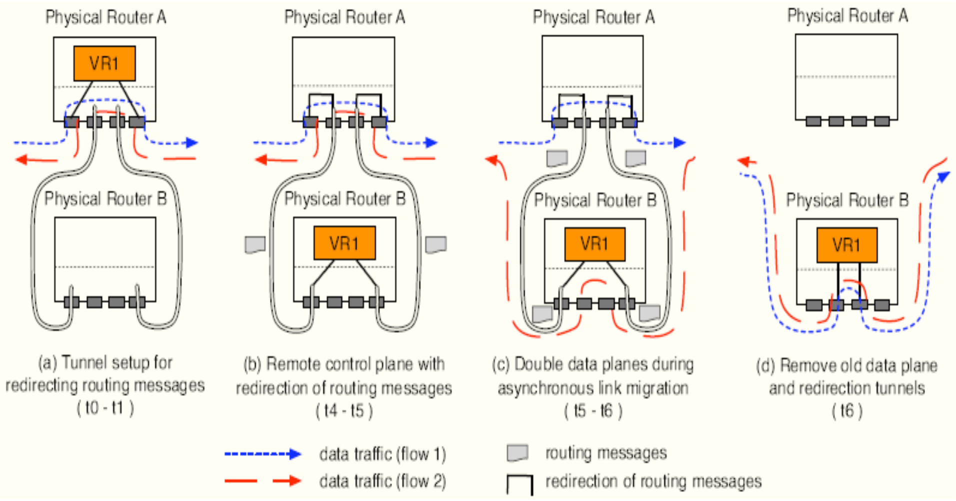


Figure 2: The architecture of a VROOM router

Router Virtualization: A determined VROOM router partitions remains as resources related to the physical router in order to assist multiple virtual routers. Every virtual router runs in a very independent way along with control plane (as instances of configurations, applications, routing protocol and routing information base (or the RIB)) added by data plane (like interfaces as well as forwarding information base (or FIB)). This kind of router virtualization assists available commercial routers in isolation among virtual routers that can turn migration of virtual router without initiating any difference over others.

Control and Data Plane Separation: In case of VROOM router, data plane and control plane run under different environments. Figure 2 illustrates control planes related to virtual routers that can host under separate "*containers*" (or "*virtual environments*"), whereas data planes reside under substrate that every data plane need to separate data structures along with lower data like FIB as well as access control lists (or ACLs). In the same way, separation made over control plane and data plane is part of current commercial routers added by control plane over CPU(s) as well as main memory, whereas data plane runs over line cards with personal computing power (packet forwarding) as well as memory (holding FIBs). Such separation permits VROOM to initiate migration of control as well as data planes over virtual router on a separate way (Section 4.2.1 and 4.2.2).

Dynamic Interface Binding: this is for enabling router migration added by link migration, whereby VROOM router must remain dynamic as well as change binding among FIB of virtual router, added by substrate interfaces (that has physical/tunnel interfaces), see Figure 2.



(a) Tunnel setup for redirecting routing messages (t0 - t1)

(b) Remote control plane with redirection of routing messages (t4 - t5)

(c) Double data planes during asynchronous link migration (t5 - t6)

(d) Remove old data plane and redirection tunnels (t6)

Figure 3: VROOM's novel router migration mechanisms (the times at the bottom of the subfigures correspond to those in Figure 4)

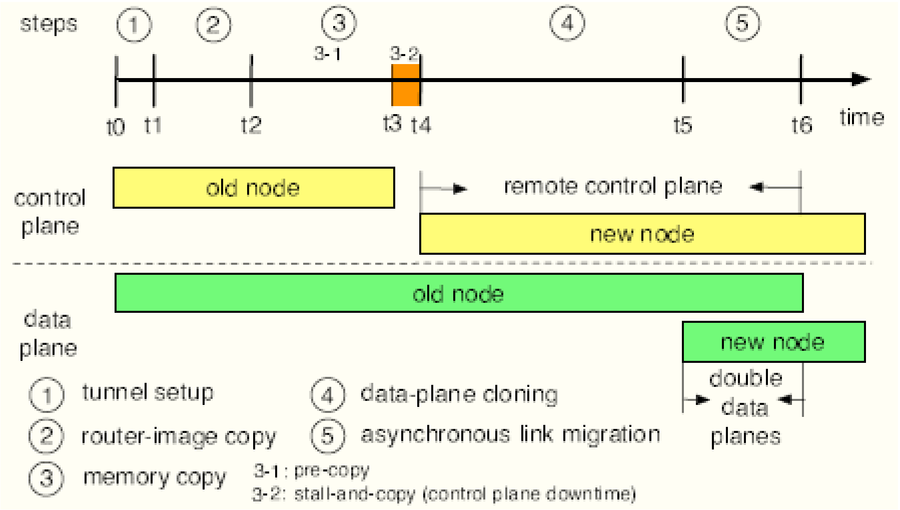


Figure 4: VROOM's router migration process

Current interface related to the process of binding mechanism under current routers maps with virtual routers, whereby VROOM demands two simple extensions: after virtual router there is the migration that demands binding for re-establishing dynamical newer physical router same to virtual router, instantiated over physical router. Secondly, link migration with packet-aware transport network comprises tunnel interfaces' changes (Figure 1), whereby router substrate demands to switch binding from old to new tunnel interface on-the-fly[1].

4.2 Process of Virtual Router Migration

As noted in Figures 3 and 4, the process of virtual router migration by VROOM considers the establishment of tunnels among source physical router A added by destination of physical router B over migration, at the first place (Figure 3(a)). These tunnels permit control plane for sending as well as receiving routing messages followed by migration (in steps 2 and 3) yet prior to link migration (in step 5) completes. They permit migration of control plane in terms of keeping data plane over A (Figure 3(b)). Though the control plane experience short period among the downtime by the end of step 3 (that is memory copy), continues for data

---

[1] In the case of a programmable transport network, link migration happens inside the transport network and is transparent to the routers.

plane working in the time of entering the process of migration. As a matter of fact, step 4 (that is data-plane cloning) gets followed by data planes over A and B forwarding traffic (Figure 3(c)). This migrates links from A to B within asynchronous way (Figure 3(c) and (d)), that is followed by data plane over A being disabled (Figure 4). We illustrate migration process.

4.2.1 Control-Plane Migration

The process of migrating control plane demands: image of router, like routing-protocol binaries and the files for network configurations, and memory that is inclusive of all sorts of running approaches. In the process of copying image and memory of router, it can opt for minimizing total time of migration added by lessened control plane downtime (time between check point of control plane over source node and as the same gets restored over destination node). Though routing protocols in general can tolerate brief network glitch through retransmission (BGP implies TCP retransmission, as OSPF implies reliable retransmission), long control-plane outage adjacencies of break protocol added by reason for protocols to reconverge. We illustrate the way VROOM leverages migration of virtual machine (VM) techniques over control plane in step 2 (copy router-image) and step 3 (memory copy) for migration process (see Figure 4).

As against this, VMs potentially runs under various programs, and have vendor running the same set of programs (as routing protocol suites). VROOM makes assumptions about binaries are noted in every physical router. Prior to virtual router for migration, there are binaries noted locally and copied in file system over destination node. Thus, router configuration demands to get copied over network that can reduce total migration time (local-copy remains faster than network-copy).

Migration of memory over virtual router is about the check-point of the router, copy memory to destination, and restore router, a.k.a. stall-and-copy, leading to downtime proportional to memory size of router. Better way is to add iterative pre-copy prior to final stall-and-copy, (Figure 4). All pages remain transferred in first round of precopy phase, added by following rounds, whereby only pages get modified in former round being transferred. It reduces page counts for being transferred in stall-and-copy phase, minimising control plane downtime over virtual router (control plane being “frozen” among t3 and t4; Figure 4).

4.2.2 Data-Plane Cloning

Migration of control-plane, as discussed above can be extended toward migration of data plane, which copy all kinds of data-plane to newer physical node. Still, this proceeding got two drawbacks. The foremost is copying status of data-plane (as FIB and ACLs) being unnecessary and absolutely wasteful, as used data is meant to generate such states (like RIB and files of configuration) available already within control plane. Secondly, copying data-plane directly can turn up hard to initiate, in case the destination and the source routers imply different data-plane technologies. As for instance, some routers can be used TCAM (as the Ternary Content-Addressable Memory) in respective data planes, whereas others can opt for regular SRAM. Consequently, data structures holding the state can turn different.

VROOM has formalized interface among data planes and control planes through data-plane hypervisor that permits migrated control plane towards the process of re-instantiation of data plane over newer platform, a proceeding we can term as data-plane cloning. This is just control plane for router that gets migrated. As control plane gets migrated towards newer physical router, it is subject to copy the original data plane through repopulating FIB by means of using RIB and the act of reinstalling ACLs, along with other data-plane states[2] by data-plane hypervisor (Figure 2). Data-plane hypervisor offers unified interface towards control plane that eventually hides the matter of heterogeneity towards the underlying data-plane which gets implied for enabling virtual routers get migrated among varied data planes.

4.2.3 Remote Control Plane

---

[2] Data dynamically collected in the old data plane (such as NetFlow) can be copied and merged with the new one. Other path-specific statistics (such as queue length) will be reset as the previous results are no longer meaningful once the physical path changes.

Figure 3(b) shows that after control plane of VR, there is a migrated status from A to B, followed by the step of repopulate (clone) approach of data plane over B and migrate connections from A to B. However, this is not happening as the new data plane is unable to act instantaneously, because of time needed for the installation of FIB entries that takes between some 100 and 1000s seconds. Thus, installation over full Internet BGP routing table (250k routes) can consume 20 seconds. In this time, though data traffic can be forwarded through old data plane over A, every act of routing within control plane of VR1 which cannot be send/receive routing messages. In case the duration stays for longer span of time, control plane remains unreachable, and can eventually lose protocol adjacencies added by neighbours.

In order to get over such dilemma, substrate of A begins redirecting every routing messages towards VR1 to B by the last part of migration of control plane within time t4 (Figure 4).

It gets accomplished by means of establishing tunnel among A and B as substrate interfaces of VR1. In order to ignore the act of introducing added downtime within the control plane, there is the need for tunnels for the act of establishing prior to the migration of control-plane (see Figure 3(a)). Followed by such redirection control plane of VR1 exchange routing messages among the neighbours and act in the form of remote control plane meant for the old data plane over A and keep on updating old FIB, as routing changes.

4.2.4 Double Data Planes

On a theoretical basis, in the last step of data-plane cloning, VR1is liable to make a switch over from old data plane over A in order to attain newer over B through the process of migrating all the connections from A to B on a simultaneous basis. Still, to act by accurate synchronisation with the migration over all the links turns up challenging. Moreover, it can increase system complexity at a very large scale (as there is the emergence to implement determined synchronization mechanism).

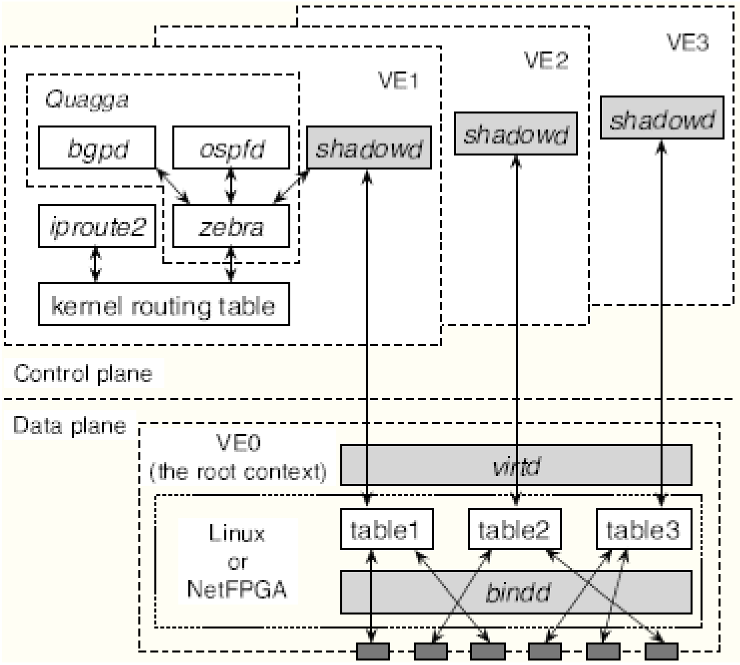


Figure 5: The design of the VROOM prototype routers (with two types of data planes)

On a better side, due to VR1 have two data-planes for forwarding traffic by the end of cloning step of data-plane (Figure 4), migration of determined connections never happen instantaneously. Rather every connection can be migrated on an independent basis over others, under synchronous fashion (Figure 3(c) and (d)).

Firstly, router B is responsible for the creation of newer outgoing connection for every neighbour of VR1, whereas every data-traffic keeps on flowing by router A. Then, incoming links can migrate synchronously and safely. This will be assisted by some traffic flow by router B, whereas the rest of the traffic will still continue flowing by router A.

Finally, as all the connections of VR1 get migrated towards router B, older data plane as well as outgoing links over A, and temporary tunnels can get safely removed.

5. Implementation of prototype

This section is about the implementation of two VROOM prototype routers. The foremost id built over commodity PC hardware added by Linux-based virtualization solution Open VZ. Next is on same software, yet by the use of Net FPGA platform since hardware data plane is present. We considered the offered design as a readily applicable content for commercial routers, that can typically attain similar cleaned mode of separation among control as well as data planes. Implementation of our prototype comprises 3 new programs (see Figure 5), which include virtd for enabling packet forwarding to the location outside virtual environment (separation of control plane and data plane); shadowd for enabling every VE towards the installation of routes in FIB; and lastly, bindd (cloning data plane) in order to offer bindings among physical inter-faces along with virtual interfaces added by FIB of every VE (data-plane hypervisor). We analysed the mechanisms for enabling migration of virtual router within our prototypes and further offered added mechanisms for the implementation of realization towards the act of migration.

5.1 Enabling Virtual Router Migration

We decided to select Open VZ, which is identified as a Linux-based Operating System level for virtualized solution, since the environment of virtualization is meant for our prototypes. With the running of multiple operating systems, under various virtual routers, light-weight OS-level virtualization gets better to our approach as against any other virtualization approach, especially like full virtualization added by paravirtualization. In case of Open VZ, there are multiple virtualised environment of (VEs) running over same host is subject to share same kernel, yet attain separated virtualized resources like those of namespaces, devices, process trees and network stacks. Instance of Open VZ further offers live migration capability in the process of running VEs[3].

Further descriptions in a top-down order have been initiated here with three components related to our two prototypes which are liable to enable migration of virtual router. We initiated with the mechanism that is there for separating

---

[3] The current OpenVZ migration function uses the simple “stall-and-copy” mechanism for memory migration. Including a “pre-copy” stage in the process will reduce the migration downtime.

control planes and data planes, added by illustration of data-plane hypervisor which permits updates of control planes with FIBs under shared data plane. Lastly, we illustrate mechanisms with dynamically bind interfaces along with FIBs along with set up of data path.

5.1.1 Separation of Control Plane and Data Plane

In order to mimic the separation of control plane and data plane under commercial routers, we initiated with FIBs that remain out of VEs added by the act of placing them with shared, yet virtualized data plane (see Figure 5). This specifies that packet forwarding is not happening anymore within respective context meant for every VE, thus remain unaffected as VE gets migrated.

As mentioned above, we implied 2 prototypes added by various data planes, software-based data plane (known as SD) and hardware-based data plane (known as HD). SD prototype router considers data plane for residing root context (or “VE0”) of respective system and implies Linux kernel in terms of packet forwarding. As Linux kernel (2.6.18) supports 256 separate kinds of routing tables, SD router is liable to virtualize data plane through association with VE added by various kernel routing table as from FIB. In terms of implementing HD router, we implemented NetFPGA platform that is liable to configure in association with reference router a sled by Stanford. Net FPGA card attains 4-port gigabit ethernet PCI card added by Virtex 2-Pro FPGA over it. Along with Net FPGA in the form of data plane, forwarding packet in HD router never implies host CPU, therefore stand same as commercial router. NetFPGA reference router never supports virtualization, in the current scenario. Thus, our HD router implies current limited approach to virtual router/physical node.

5.1.2 Data-Plane Hypervisor

Section 4 had already illustrated that VROOM is meant for extending standard over control plane or data plane interface for migration that has data-plane hypervisor. Our determined prototype offers rudimentary data-plane hypervisor implementation supporting FIB updates. (Full-fledged data-plane hypervisor can permit configuration over other data plane). We hereby implemented virtd program since data-plane hypervisor. The noted virtd runs within VE0, offering interface meant for virtual routers for either installing or removing routes within shared data plane (Figure 5). We further implied shadowd program running within every VE and then pushing route updates from respective control plane towards FIB by virtd. We initiated Quagga routing software suite as inside control plane of every VE. The Quagga assists various routing protocols, added by BGP as well as OSPF. Moreover, it remains inclusive of protocols, Quagga offers interface within zebra, added by routing manager for permitting added newer protocol daemons. We implied such interface in order to initiate the act of shadowd, as in zebra' client that offers both ability for notification about route changes by zebra, and being notified. As illustrated in shadowd, the process is not for routing protocol but just for shadowing daemon by the application of capability for route redistribution. By such interface, shadowd gets notified about relevant changes within RIB and further mirrors towards virtd by the application of Remote Procedure Calls (RPCs). Instances of each shadowd gets configured along with unique ID (as the ID of virtual router), that is part of every message as sent by it towards virtd. On the basis of this ID, virtd is liable to get installed or removed correctly within FIB over attained updates from shadowd instance. In case of SD prototype, such inclusion by Linux iproute2 utility for setting routing table entry. In case of HD prototype, such inclusion implies device driver towards the process of writing to registers within NetFPGA.

5.1.3 Dynamic Interface Binding

By means of the separation in control plane and data plane, and further sharing of same data plane over multiple virtual routers, respective data path for every virtual router need to set up in an apt way assure-

(i) data packets that is liable to get forwarded as per right FIB, and
(ii) routing messages being delivered to apt control plane.

We have implied bindd program through the scope of 2 basic approaches. The foremost is the mapping among substrate interfaces of virtual router and its FIB followed by virtual router within instantiated/migrated status, for the confirmation of correct forwarding of the packet. (Note: substrate interface of virtual router can be either dedicated by physical interface or otherwise through tunnel interface for the purpose of sharing physical interface added by other tunnels.) In case of SD prototype, establishment of bindd over respective binding gets implied for the application of managing routing policy function ("*ip rule*") offered through the utility of Linux iproute2. From the aforementioned HD prototype being under current limited single table, it is specific to note NetFPGA in terms of supporting virtualization, in terms of a mechanism that stands same as the function of "*ip rule*" to be implied to bind interfaces in FIBs. The next approach led by bindd is to initiate binding of substrate interfaces along with virtual interfaces form control plane. In reference to both the prototypes, such binding is attained through the process of connecting every pair of substrate as well as virtual interfaces towards different bridge by implying through the utility of Linux brctl. In case of HD prototype, every 4 physical ports over NetFPGA gets offered through Linux in terms of separating physical interface, so that respective destined packets for control plane of local VE get passed from NetF-PGA towards Linux by corresponding interface.

5.2 Realization of Virtual Router Migration

The aforementioned approach for the VROOM virtual router migration within OpenVZ environment, is liable to get illustrated in terms of the implementations of determined remote control plane, data-plane cloning and double data planes. Though migration stands transparent towards routing means of running within VE, shadowd demands for notification at end of migration process of the control plane for "data plane cloning". We implied a shadowd function, termed as triggers shadowd in order to request zebra towards resending of routes and further push them down to the virtd state to repopulate FIB. (Note: virtd runs over fixed (private) IPaddress along with fixed port over every physical node.) Thus, after migration of virtual router to newer physical node, respective route updates shadowd state for being seamlessly routed towards local virtd over new node. In order to enable migrated control plane towards continuation of updating old FIB ("*remote control plane*"), we implied in virtd with the capability to forward route updates towards another virtd instance by the use of RPC mechanism implied by shadowd. As the virtual router VR1 gets migrated from respective node A to determined node B, script of migration notifies virtd instance over B of A led IP address as well as ID of VR1. virtd of B, apart from updating newer FIB, begins forwarding updates of the route from control plane of VR1to A, that has virtd over updates old FIB of VR1. As all the links of VR1 get migrated, old data plane never gets used any more, thus, virtd of B gets notified for stopping of forwarding updates. Updates related to B's virtd over new and old FIBs meant for VR1 ("double data planes"), comprises of 2 data planes that are forwarded by packets in the time of asynchronous connection within the migration process.

Note: data-plane hypervisor is liable to implement control planes being under the state of unawareness meant for detailed particularised underlying process of data plane. Thus, migration is liable to appear among combined SD and HD prototypes (SD to SD, SD to HD, HD to HD, and HD to SD.

6. EVALUATION

Here, the core approach is to assess the performance related to the application of VROOM through routers of SD and HD prototypes. Initially we assesses performance led by basic aspects over the migration approach on an individual ground, and further place VROOM router within network and then evaluate migration effect over data plane and control plane. On a specific note, we answer answered –

1. What is the impact led by virtual router migration over data forwarding?

   --- Our process of evaluation confirmed the importance of having bandwidth among migration and data traffics. Considering different and separate bandwidth, migration led by HD router got no impact of the performance over data forwarding. The process of migration over SD router leads to the minimisation of delay increase added by loss of no packet towards data traffic.

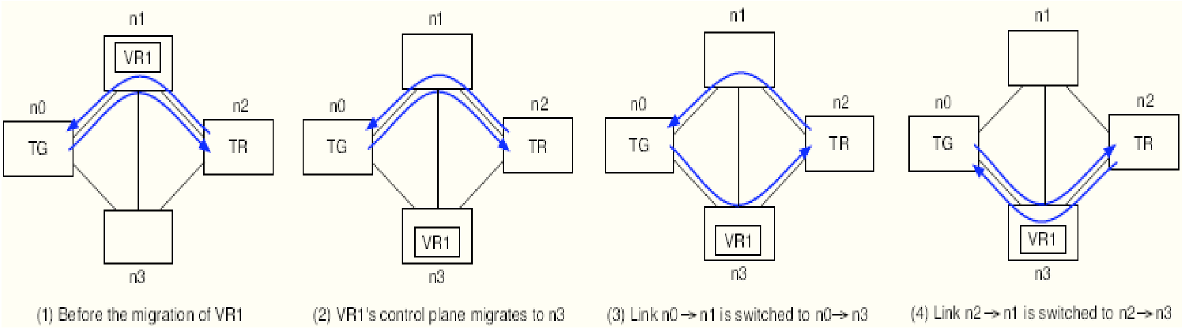
2. What is the impact led by the virtual router migration over routing protocols?

   ---- From our evaluation, it has been showed that virtual router is at the state of running over just OSPF within Abilene-topology network that is liable to assist for a second OSPF hello-interval that too without losing protocol adjacencies in the time of migration. Similar router gets loaded with full Internet BGP routing table and the same can remain supportive with minimal OSPF hello-interval for a span of 2 seconds without losing any of BGP or OSPF adjacencies.

6.1 Methodology

Assessment and evaluations led by us are inclusive of experiments in Emulab tesbed. Basically, we implied PC3000 machines in terms of physical nodes within our experiments. PC3000 remains in the form of IntelXeon 3.0 GHz 64-bit platform along with 2GB RAM added by 5 Gigabit Ethernet NICs. In case of HD prototype, every physical node remains added to the equipped mode with NetFPGA card. All the selected nodes meant for our experiments are based on the OpenVZ patched Linux kernel 2.6.18-ovz028stab049.1. In some of the experiments, we implied lower performance led by PC850 for the physical nodes, with the attempt to create an Intel Pentium III 850MHz platform along with 512MBRAM as well as 5 100Mbps Ethernet NICs. We applied 3 diversified testbed topologies within our experiments:

Diamond testbed: We implied 4-node diamond-topology testbed (see Figure 6) in order to assess performance related to individual migration approaches as well as impact led by migration on data plane. Testbed with 2 different configurations follow similar machines as well as physical node with n0 and n2, yet differ in terms of hardware over node n1 and n3.



(1) Before the migration of VR1 (2) VR1's control plane migrates to n3 (3) Link n0→n1 is switched to n0→n3 (4) Link n2→n1 is switched to n2→n3

Figure 6: The diamond testbed and the experiment process

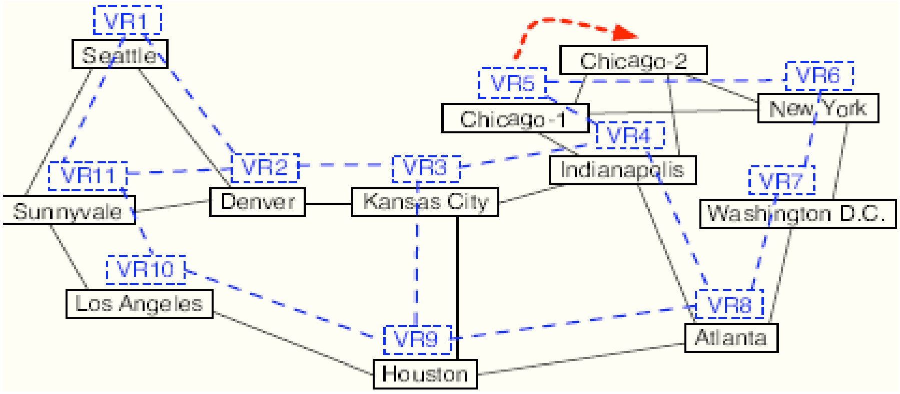


Figure 7: The Abilene testbed

Table 1: The memory dump file size of virtual router with different numbers of OSPF routes

Routes	0	10k	100k	200k	300k	400k	500k
Size (MB)	3.2	24.2	46.4	58.4	71.1	97.3	124.1

In case of SD configuration, n1= n3 under regular PCs where we have installed SD prototype routers. In case of HD configuration, n1 as well as n3 are PCs added by NetFPGA card, over the act of installation related to our HD prototype routers. By the experiments, virtual router that is VR1 gets migrated from the instance of n1 to n3 by linkn1!n3.

Dumbbell testbed: We implied 6-node dumbbell-shaped testbed towards the research of bandwidth contention among migration and data traffics. In testbed, the act of round-trip UDP data traffic gets noted among nodes' pair whereas virtual router gets migrated among other nodes' pair. Migration and data traffics get forced in order to share similar physical connection.

Abilene testbed: We implied 12-node testbed (Figure 7) for the assessment of migration impact over control plane. This comprises of a topology that stands same as 11 node Abilene network backbone [1]. The difference is by the addition of added physical node (Chicago-2), that virtual router is related to Chicago-1 (V5) being migrated.

Figure 7 illustrates initial topology meant for virtual network, comprising 11 virtual routers (V1 to V11) managed over 11 physical nodes (excluding Chicago-2).

6.2 Migration Steps: Performance

Here we assess performance led by 2 basic migration functions of prototypes, which are memory copy and FIB repopulation.

Memory copy: it is for the assessment of time for memory copy in connection with memory usage of virtual router that has been loaded for ospfd in VR1 with different routes. Table 1 enlists sizes of respective memory dump file related to VR1.

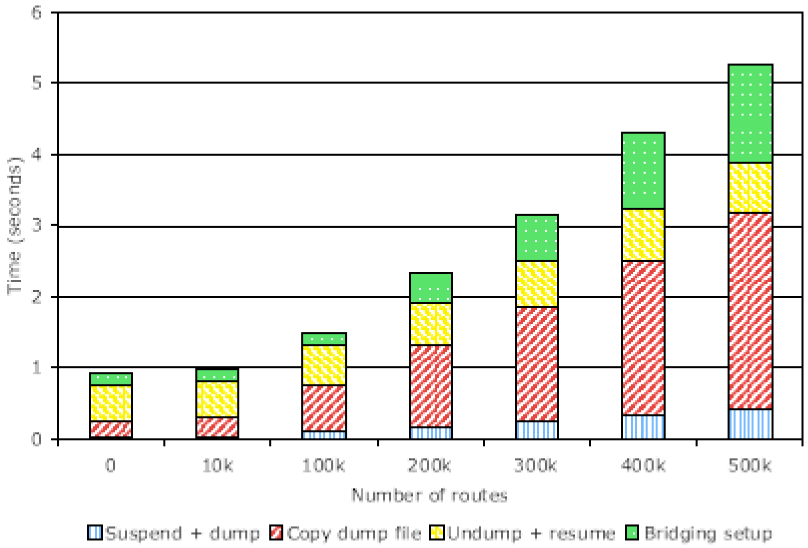


Figure 8: Virtual router memory-copy time with different numbers of routes

Figure 8 notes total time for completing memory-copy step, along with-

1)suspend or dump VR1 over n1,
2)copy dump file from n1 to n3,
3)resume the state of VR1 over n3, and
4)set up bridging (as interface binding) for VR1 over n3.

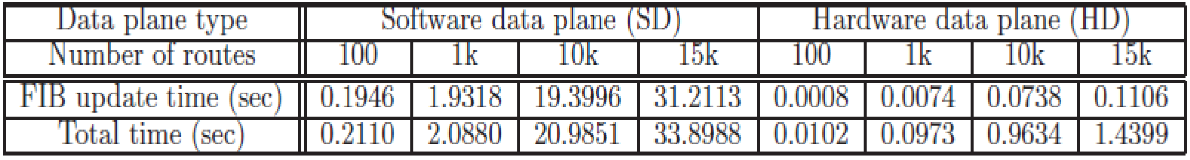
We have noted various routes getting wider, since it takes the act of copying dump file turning to the act of dominating total time for memory copy. We further marked that as memory usage gets large, respective bridging time too grows in a remarkable way and can be due to CPU contention added by virtual router for the process of restoration. That is liable to happen simultaneously.

FIB repopulation: We have by now assessed time consumed by VR1 in order to repopulate newer FIB over n3 followed by migration. This experiment configure virtual router added by various static routes as well as assess time that is needed for the installation over routes with FIB as software/hardware data plane. It is in Table 2 that there is a comparison of time for FIB update as well as time needed for FIB repopulation. Time for FIB update against virtd consider entries of install route within FIB, whereas entire time further remains inclusive of total shadowd in terms of sending routes towards virtd. Results attained by us show installation of FIB entry in NetFPGA hardware (7.4 microseconds) that turns to be more than 250 times faster against installing FIB entry within Linux kernel routing table (with 1.94 milliseconds). As expected, update time grows linearly with routes.

6.3 Impact of Data Plane

Here, we attempt to evaluate impact of router migration over data traffic. We have initiated tests for HD case and SD case and further compared the derivations. We studied relevance of bandwidth isolation among data traffic and migration.

Table 2: The FIB repopulating time of the SD and HD prototypes



Data plane type	Software data plane (SD)				Hardware data plane (HD)			
Number of routes	100	1k	10k	15k	100	1k	10k	15k
FIB update time (sec)	0.1946	1.9318	19.3996	31.2113	0.0008	0.0074	0.0738	0.1106
Total time (sec)	0.2110	2.0880	20.9851	33.8988	0.0102	0.0973	0.9634	1.4399

6.3.1 Zero impact: HD router having separate migration bandwidth

Initially, we evaluated performance of data plane over migration of virtual router from HD prototype router as considered by us. We then configured HD testbed in a way that migration traffic gets noted from n1 to n3 by direct link among n1 to n3, by means of eliminating potential kind of bandwidth link among migration and data traffics.

We run D-ITG traffic generator over n0 and n2 in order to generate round-trip related to UDP traffic. We further evaluated rate of maximum packet for D-ITG traffic generator over n0 for handling (sending as well as receiving 64-byte UDP packets at a range of 91k packets/s), for the migration of virtual router VR1from the point of n1 to n3 (added by migrations of link and control plane) without any impact led by performance of data traffic as the same is forwarding, without no increase in delay or packet loss[4]. Such

---

[4] We hard-wire the MAC addresses of adjacent interfaces on each physical nodes to eliminate the need for ARP request/response during link migration.

derivations are simple, unlike packet forwarding being handled by NetFPGA, instead migration gets handled by CPU. Thus, hardware routers along with separate migration bandwidth are liable to migrate virtual routers added by zero impact over data traffic.

6.3.2 Minimal impact: SD router added by separate migration bandwidth

In case of SD router, CPU remains as the resource for potential aspect for scarcity in the time of migration, as control plane as well as data plane is related to the sharing of virtual router over same CPU. We have studied the case, where packet and migration forwarding get saturate through CPU of physical node.

Through HD experiments as noted above, we implied link n1 to n3 for migration traffic, with the purpose to eliminate bandwidth contention. For the creation of CPU bottleneck over n1, we implied PC3000 machines over n0 and again n2, added by lower performance by the PC850 machines over n1 and n3. We have migrated VR1 from the range of n1 to n3, as sending round-trip for the UDP based data traffic among the nodes n0 and n2 remains relevant. We diversified rate of packet for data traffic from 1k to 30k packets per second and noted impact of performance on the experiences of data traffic caused by migration (30k packets per second gets noted as maximum bi-directional rate of packet over PC850 machine, which can handle without the process of dropping packets.)

On a surprising note increase in delay is due to migration being the only noticeable aspect as rate of packet gets relatively low. As the rate of the UDP packet gets noted at 5k packets per second, migration of control plane leads to sporadic round-trip delay which is liable to increase till 3.7%. Still, as the rate of packet gets higher (as in 25k packet per second), delay changes in the time of migration being negligible (< 0.4%). The reason is packet forwarding being handled through kernel threads, against the OpenVZ migration being handled by user-level proceedings (like ssh, rsync, etc.). Though kernel threads attain higher priority against processes at user level scheduling, Linux follows the mechanism, preventing processes at user-level from the instance of starving as the rate of packet remains high. This illustrates increase in delay as migration is under progress.

Table 3: Packet loss rate of the data traffic, with and without migration traffic

Data traffic rate (Mbps)	500	600	700	800	900
Baseline (%)	0	0	0	0	0.09
w/ migration traffic (%)	0	0	0.04	0.14	0.29

Still, with higher rate of packet, process of user level migration gets more frequently interrupted, and the more frequently the handler of packet gets called. Thus, higher rate of packet gets least additional delay to the process of migration along with forwarding of packet. This illustrates the reason for the rate of packet being 25k packets per second, increase in delay led by migration turning negligible. Further it illustrates the reason for migration for not causing any drop of the packet under experiments. Lastly, our experiments showed migration having no connection for affecting forwarded delay.

6.3.3 Significance of Reserved migration bandwidth

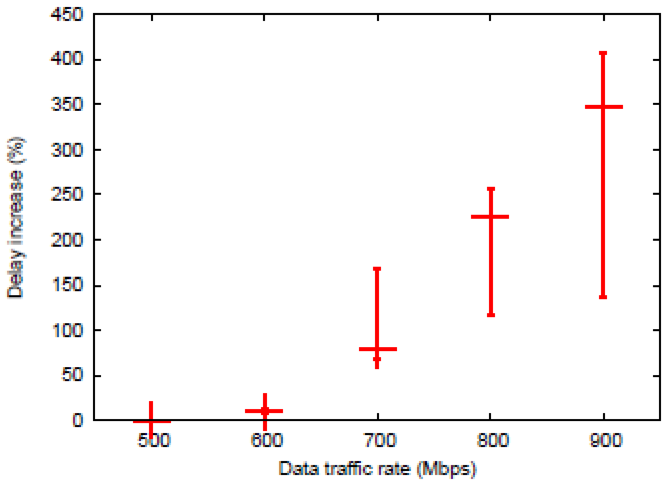


Figure 9: Delay increase of the data traffic, due to bandwidth contention with migration traffic

As noted in 6.3.1 and 6.3.2, traffic of migration is offered to respective connection (with separated bandwidth). It is here that we study relevance of respective demands and implications of performance for data traffic, in case the same is not met. We implied dumbbell testbed for this experiment that holds migration traffic as well as sharing of data traffic same for the bottleneck connection.

We load ospfd over virtual router added by 250k routes.

We started rate of data traffic from 500 Mbps, and further increased the same to 900 Mbps. as OpenVZ implied TCP (scp) for respective memory copy, migration traffic just got leftover bandwidth from UDP data traffic. Since, there is a decrease in the available bandwidth below 300 Mbps, time for migration increases, and translates in the form of a longer control-plane downtime in terms of virtual router.

Figure 9 makes a comparison between increase in delay of data traffic in diversified rates. These average delay and jitter increase in a very dramatic way as bandwidth contention get severe. Table 3 makes a comparison between the rate of packet loss for data traffic at various rates, with as well as without migration traffic. Obviously, bandwidth contention (data traffic rate being 700 Mbps) leading to loss of data packet. The derivation show that for minimization of control-plane downtime for the virtual router, as well as eliminating impact of performance over data traffic, operators needs to offer separate bandwidth in terms of migration traffic.

6.4 Impact of Control Plane

Here, we inspect dynamics of control plane focussed by another router migration; particularly the way migration is liable to affect protocol adjacencies. We have assumed backbone networking of MPLS, where its edge routers are subject to run OSPF as well as BGP, whereas the core routers are running just the OSPF. Derivations attained by us specifies default timers, adjacencies of protocol OSPF and protocol BGP as intact, and one specialised OSPF LSA retransmission as needed in the worst case scenario..

6.4.1 Core Router Migration

We have configured virtual routers as VR1, VR6, VR8 and VR10 over Abilene testbed (as in Figure 7) in terms of edge routers, added by rest of the virtual routers in the form of core routers. Through the provision of migrating VR5 from respective physical node as in Chicago-1 to Chicago-2, we have noted impact led by migrating of the core router over OSPF dynamics.

There seemed to be no occasion in the phase of migration. We considered the case, where there is no network event in the time of migration. Experiment showed downtime of control-plane in terms of VR5 remaining between 0.924 and then 1.008 seconds, added by average 0.972 seconds on 10 runs.

We started default OSPF timers for the Cisco routers, comprising hello-interval meant for 10 seconds and the dead-interval for a span of 40 seconds. Further, we reduced hello-interval to respective 5, 2, and 1 second under subsequent runs, yet kept dead-interval same as 4 times hello-interval. We derive OSPF adjacencies among migrating VR5 added by neighbours (VR4 and then VR6) being up in every case. even in terms of most restrictive 1 second for the hello-interval case, there can be 1 OSPF at max for hello message being lost and VR5 being the back up for Chicago-2, which stands prior to dead timers expiration for the neighbours.

Events occur in the time of migration. We further investigate the case where events in the time of migration as well as migrating router VR5 eventually misses LSAs triggered through events. We relevantly trigger newer LSAs by means of flapping connection among VR2 and VR3. We have noted VR5 missing LSA as LSA gets generated in the time of VR5’s 1 second downtime. During such a situation, VR5 attains retransmission over missing LSA in a span of 5 seconds, as default LSA retransmit-interval.

Further, we reduced LSA retransmit-interval from a span of 5 sec to 1 sec, for the reduction of time that VR5 might have attained a stale view over respective network. This kind of change is liable to bring maximum interval among link flap occurrence and reception of VR5 over resulting LSA to 2 secs (which stands for 1 sec control plane downtime, added by 1 sec LSA retransmit-interval).

6.4.2 Migration of Edge Router

In this section, we configured VR5 at 5th edge router within respective network running BGP with OSPF. VR5 attains full routing table of Internet BGP added by 255k routes (from Route Views on 12th Dec 2007) from peer of eBGP that is not part of Fig. 7, forming iBGP full mesh added by other 4 edge routers. Along with full BGP table, growth of the file size of memory dump is noted from a range of 3.2

MB to 76.0 MB. Consequently, it considers longer span to get suspended or dumped over virtual router, being copied over respective dump file, added by the mode to resume it. Average downtime for control plane in the time of migration increases from 3.484 and 3.594 seconds, added by average of 3.560 seconds in 10 runs.

We noted BGP sessions of VR5 staying intact in the time of migration. Minimal integer hello-interval VR5 is liable to assist without breaking adjacencies of OSPF in the time of migration being 2 seconds (added by set of dead-interval to 8 seconds).

On a practical note, ISPs hardly set timers at much lower than the default values, to get shield from faulty equipment or connection.

7. Scheduling of migration

The basic discussion here is about the migration mechanisms (the way it function) for VROOM. Added to this, there is the significant query about the scheduling of ("to where"). We have noted here the relevant constraints seemed important as the scheduling of migration as well as various optimization issues remain integral to the ongoing work over VROOM way of migration scheduling. As the decision for migration is taken over virtual router, there are various physical constraints that must be noted the foremost concern is about the "*eligible*" physical router's destination for the migration by compatibility of software platform, added by original physical router, along with same (or rather greater) capabilities (like supported accessibility control lists). Moreover, destination of the physical router should have enough resources, added by the processing power (if physical router is hosting maximum virtual routers, already) added by the connection capacity (if the connection is towards the physical router with enough unused bandwidth to get handled towards migrating traffic load of the virtual router). Moreover, de3maaands of redundancy for virtual router also gets noted as a router, and gets connected to 2 diversified routers (primary and the backup) in terms of redundancy. In case the primary and backup get migrated towards similar node, physical redundancy gets lost. However, fortunately, ISPs in general leave enough "*head room*" in terms of link capacities for the purpose of absorbing increased volume of traffic. Furthermore utilisation of ISPs for routers from 1 or 2 vendors are noted with least count of models that can leave larger eligible physical routers for the selection of migration.

With the presence of a physical router with the demand for maintenance, the query is about where to migrate virtual routers. Here, current hosts can get formulated in the form of optimized issue, above constraints. Considering the operator's preference, there are many objectives that are liable to get picked for the best destination router, like minimizing load of CPU for physical router, maximum load related to the physical connections with the network, minimizing stretch (latency increase) related to virtual connections initiated by migration, or otherwise maximization of network reliability (as ability for survival to failure over physical link or node).

Still, deriving optimal solutions for such issues problems might turn computationally intractable. Yet, simple local mode of search algorithms must get performed reasonably, as count of physical routers is limited (such as few 100s to 1000s, even in case of large ISPs) added by the derivation of "*good*" solution (instead of optimal point) being accepted. Against this, scheduling of migration in terms of planned maintenance leads us to scheduling issue with savings of power as well as traffic engineering. For power savings, we considered prices of power under diversified geographic areas and further lessened consumption of power with some migration granularity (as oncew/hour, as per hourly traffic matrices). For traffic engineering, we decided to migrate virtual routers towards load shift away from congested mode of physical links.

8. Conclusions

VROOM has been identified as a new mode of network-management that has primitive for supporting live migration over the virtual routers from a physical router to the other. In order to minimize disruptions, the act of VROOM is to permit plane of migrated control for cloning data-plane at newer area, at the same time continuing updates about old location's status. VROOM on a temporary way leads to packets through the application of both data planes in order to support asynchronous mode of migration related to links. Such designs remain readily applicable towards commercial mode of router platforms. Various experiments added by our prototype system aims in illustrating VROOM for not disrupting data plane added by briefly freezing control plane. Against this act of control-plane occurring while freezing process, whereby the effects remain hidden to a great extend through the mode of noting current mechanisms related to the retransmission of routing-protocol messages.

Our research related to VROOM leads to various wide ranged questions related to design of various future routers along with relationship noted for underlying transportation network. In recent innovations for supporting transport networks, there is a rapid set-up as well as tearing-down of connections, that can enable network topology for changing underneath IP routers. Modes of dynamic topologies added by migration of VROOM of control plane as well as cloning of data plane turns router on an increasingly ephemeral concept, not being tied to specialised area or hardware. In reference to future activity, the router hypervisors can consider the idea a step further.

Like the current commercial routers there is a note of clear separation among data planes and controls, whereby the future routers can get decouple software control-plane from state of control-plane (as the bases for routing information). Likewise, “control-plane hypervisor” can turn it easier in terms of upgrading router software as well as virtual routers for the migration among physical routers that is liable to run in a diversified code bases.

References

[1] 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.

[2] A. Shaikh, R. Dube, and A. Varma. Avoiding instability during graceful shutdown of multiple OSPF routers. IEEE/ACM Trans. Networking, 14(3):532–542, June 2006.

[3] Average retail price of electricity. http://www.eia.doe.gov/cneaf/electricity/epm/table5_6_a.html.

[4] B. Cully, G. Lefebvre, D. Meyer, M. Feeley, N. Hutchinson, and A. Warfield. Remus: High availability via asynchronous virtual machine replication. In Proc. NSDI, April 2008.

[5] 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. NSDI, May 2005.

[6] Ciena CoreDirector Switch. http://www.ciena.com.

[7] Cisco Logical Routers. http://www.cisco.com/en/US/docs/ios_xr_sw/iosxr_r3.2/interfaces/command/reference/hr32lr.html.

[8] D-ITG. http://www.grid.unina.it/software/ITG/.

[9] E. Chen, R. Fernando, J. Scudder, and Y. Rekhter. Graceful Restart Mechanism for BGP. RFC 4724, January 2007.

[10] Emulab. http://www.emulab.net.

[11] G. Iannaccone, C.-N. Chuah, S. Bhattacharyya, and C. Diot. Feasibility of IP restoration in a tier-1 backbone. IEEE Network Magazine, Mar 2004.

[12] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright. Power awareness in network design and routing. In Proc. IEEE INFOCOM, 2008.

[13] J. van der Merwe and I. Leslie. Switchlets and dynamic virtual ATM networks. In Proc. IFIP/IEEE International Symposium on Integrated Network Management, May 1997.

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

[15] Juniper Logical Routers. http://www.juniper.net/techpubs/software/junos/junos85/feature-guide-85/id-11139212.html.

[16] K. Roth, F. Goldstein, and J. Kleinman. Energy Consumption by Office 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.

[17] 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.

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

[19] M. McNett, D. Gupta, A. Vahdat, and G. M. Voelker. Usher: An extensible framework for managing clusters of virtual machines. In Proc. USENIX LISA Conference, November 2007.

[20] MPLS VPN Carrier Supporting Carrier.

http://www.cisco.com/en/US/docs/ios/12_0st/12_0st14/feature/guide/csc.html.

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

[22] NetFPGA. http://yuba.stanford.edu/NetFPGA/.

[23] O. Bonaventure, C. Filsfils, and P. Francois. Achieving sub-50 milliseconds recovery upon BGP peering link failures. IEEE/ACM Trans. Networking, October 2007.

[24] OpenVZ. http://openvz.org.

[25] 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. SOSP, October 2003.

[26] P. Francois, M. Shand, and O. Bonaventure. Disruption-free topology reconfiguration in OSPF networks. In Proc. IEEE INFOCOM, May 2007.

[27] Quagga Routing Suite. http://www.quagga.net.

[28] R. Teixeira, A. Shaikh, T. Griffin, and J. Rexford. Dynamics of hot-potato routing in IP networks. In Proc. ACM SIGMETRICS, June 2004.

[29] S. Bryant and P. Pate. Pseudo wire emulation edge-to-edge (PWE3) architecture. RFC 3985, March 2005.

[30] T. Afferton, R. Doverspike, C. Kalmanek, and K. K. Ramakrishnan. Packet-aware transport for metro networks. IEEE Communication Magazine, March 2004.

[31] T. Wood, P. Shenoy, A. Venkataramani, and M. Yousif. Black-box and Gray-box Strategies for Virtual Machine Migration. In Proc. NSDI, April 2007.

[32] The Internet2 Network. http://www.internet2.edu/.

[33] VINI. http://www.vini-veritas.net/.

[34] Y. Wang, J. van der Merwe, and J. Rexford. VROOM: Virtual ROuters On the Move. In Proc. ACM SIGCOMM Workshop on Hot Topics in Networking, Nov 2007.

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