Challenges in Resource Allocation in Network Virtualization

Aun Haider
NICT Tokyo
a.haider@nict.go.jp

Richard Potter
NICT Tokyo
potter@nict.go.jp

Akihiro Nakao
The University of Tokyo/NICT
nakao@iii.u-tokyo.ac.jp

***Abstract*—Virtualization promises to overcome the weaknesses of the current Internet and to act as a torch bearer technology for enabling innovative network architectures. However, a fundamental problem in instantiation of Virtual Networks (VNs) is an optimal allocation of resources offered by a physical IP network. In order to solve this NP-hard problem, heuristics have been proposed in literature. This paper attempts to concisely present some of the major techniques for resource allocation in VNs. System level models have been employed to better understand the resource allocation process. These models can be useful for the design of efficient systems for instantiation of VNs. Relatedly, resource allocation techniques in some popular test-beds have been also briefly presented. Major contributions of this paper include: a concise survey of the latest techniques for resource allocation in VNs, proposing a new objective function for mapping of VNs, a system level description/analysis of resource allocation problems and identifying some important research challenges.**

I. Introduction

The current Internet is ubiquitous. In a period of less than two decades, it has morphed into a fundamental component of modern society, culture, knowledge, businesses and defense infrastructures. However, it has shown to be extremely resistant to wide spread adoption of new technologies, for instance: Differentiated services [1] and IP-Multicast [2]. Although, some efforts, such as IPv6, are being deployed in the Internet at extremely slow pace [3]. Various factors, such as end-to-end design principle and capital investors, [4], are responsible for this inertia to evolutionary changes. Therefore, it is widely believed that the current Internet is not performing well and hence required to be fixed.

Virtualization is being widely hailed in the networking research community as a means to overcome the weaknesses of the current Internet, [5]. It has been also seen as a harbinger for the future generation networks. Recently, it has been reported that virtualization will be at foremost position in the list of Top 10 technologies for 2009, [6]. Likewisely, virtualized environment is considered as a potent tool to implement various innovative, yet possibly disruptive technologies. Major router vendors have already started to support virtual routers and programmability to run user defined protocols, [7], [8] and [9]. Currently, up to 255 virtual routers can be configured on the physical interface of a single router by using Cisco's Virtual Router Redundancy Protocol, [10].

In a virtualization-enabled networking infrastructure, a number of diverse VNs will be sharing resources offered by an IP-network, such as the Internet, commonly referred to as Substrate Network (SN). These VNs can be constructed through the deployment of virtual routers and virtual links. Moreover, an important requirement for such infrastructures may also be a support to pluralism, [11]. It will seek to introduce virtualization as an architectural attribute of the Internet, which will enable continuous embedding of innovative technologies in the Internet. Such approaches will act as protection against ossification of the Internet, [4].

One of the most important issues in network virtualization is an efficient utilization of SN resources. It will help to improve the resource utilization as well as avoiding congestion in the SN. Intuitively the mapping of VNs onto a common substrate can be done in either static or adaptive manner. In the former case, substrate resources allocated are not changeable during lifetime of a VN; whereas, in the latter case resources allocated to a VN can be adjusted on the basis of traffic load to improve overall network performance. Generally, the resource reservation for VNs is a coarse-grained activity, the result of which is planned to last for longer periods of time; whereas, in conventional networks, resources are reserved on an end-to-end basis and are limited only to the lifetime of a flow. Once a flow expires, the resources are returned back to network. However, the interactions between VNs and substrate network are far more complicated than the case of traffic flows and conventional networks [4].

A key objective for the design of a VN instantiation is to select substrate nodes with sufficient CPU, disk and the other hardware capabilities as well as substrate links with enough spare bandwidth, while minimizing the usage of total resources of SN. Further complexity will be added if the resource allocation is not static and it dynamically takes into consideration the changes in requirements of various VNs. Any design of VN should have a sufficient factor of safety, so as to ensure that it will be able to handle the desired traffic patterns, both under normal and abnormal conditions. However, it has been known that problem of assigning nodes in Ethernet connected test-bed without violating bandwidth constraints is NP-hard [12]. Similarly, it is intuitive to conjecture that optimized resource allocation problem in VNs is NP-hard in nature and thus we have to resort to heuristic approaches.

This paper focuses on issues related to the problem of resource allocation in VNs. It provides a concise overview of various existing techniques for resource allocation in VNs.

Essential features of these resource allocation techniques have been presented through the use of approximate system-level models. These can be helpful for developing detailed designs, specifications and performance evaluation techniques for VNs. Further, a bird's eye view of resource allocation techniques in state-of-art test-bed infrastructures for experimentation on VNs, has also been presented. However, this paper does not attempt to provide an exhaustive survey on the resource allocation/management techniques in VNs.

The rest of this paper is organized as follows. Some basic principles for instantiation of VNs have been presented in Section II. A problem description for resource allocation in VNs has been provided in Section III. Various existing approaches for resource allocation have been given in Section IV. Implementation of resource management in major test-beds for VNs, has been presented in Section V. Research challenges related to resource allocation in VNs, have been indicated in Section VI. Finally, some concluding remarks have been presented in Section VII.

II. Principles for Instantiation of VNs

The process of creating a VN starts after carrying out the virtualization of physical resources. Hence, a pool of virtual resources can be created by a virtualization layer that implements the abstraction of physical resources available at SN. These virtualized network resources will be subject to three intertwined steps: resource description, resource discovery and resource provisioning. The requests for VNs are serviced and whole process can be monitored by a management plane. These steps, involved in instantiation of VNs, have been described in Fig. 1.

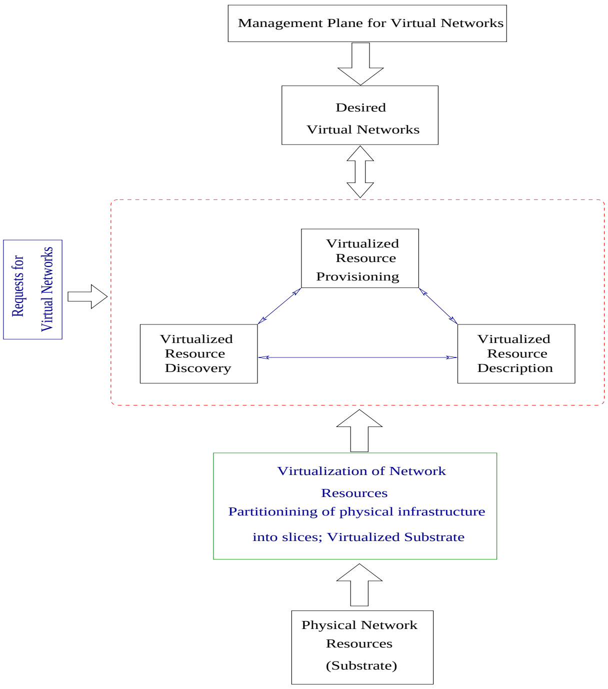
Management Plane for Virtual Networks
Desired Virtual Networks
Requests for Virtual Networks
Virtualized Resource Provisioning
Virtualized Resource Discovery
Virtualized Resource Description
Virtualization of Network Resources
Partitionining of physical infrastructure into slices; Virtualized Substrate
Physical Network Resources (Substrate)

Fig. 1. Main steps involved in instantiating VNs, adapted from [13].

Some basic goals for a policy driven resource management system for VNs include: (i) system must allow users to reserve resources diametrically across the network, for a predictable and reliable operation, (ii) system must provide enough isolation, so as to avoid users from interfering with each others' reservations, and (iii) system must have admission control mechanism so that only a limited number of requests are enqueued for receiving service and thus avoiding congestion.

III. Problem Description

Let the topology of a substrate network be represented by a graph $G_s = \{V^s, E^s, C_n^s, C_l^s\}$; where V^s is a set of nodes (vertices), E^s is a set of links (edges) along with C_n^s and C_l^s as constraints associated with nodes and links. For nodes the constraints include CPU computational capacity, physical location, and maximum possible number of instantiation of virtual machines. On the other hand, constraints of links are: delay, jitter, bandwidth and resiliency. The arrival process for VN requests can be conceptualized as a queuing system in which requests arrive in real-time and are served in a first-come-first-serve basis or by some other scheduling mechanism, cf. Fig. 6. For each VN request, the resource controller has to assign the available virtual resources extracted from the SN. The serviced request will yield a VN which can be represented by another graph $G^v(i) = \{V^v(i), E^v(i), C_n^v, C_l^v\}$. The overall process can be decomposed into two sub-problems [14] [15]: node assignment $f^n(i) : \{V^v(i), C_n^v\} \rightarrow (V', R_n)$ and link assignment $f^l(i) : \{E^v(i), C_l^v\} \rightarrow (E', R_l)$; where $V' \subset V^s$, $E' \subset E^s$ with R_n and R_l as node and link resources allocated for i-th request of VN. This process has been summarized in Fig. 2.

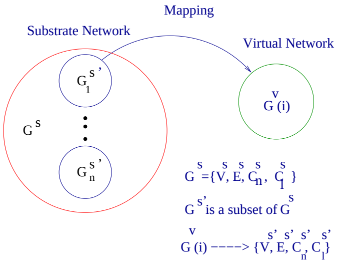
Mapping
Substrate Network
Virtual Network
$G_1^{s'}$
G^s
$G_n^{s'}$
$G^v(i)$
G^s ={V^s, E^s, C_n^s, C_l^s }
$G^{s'}$ is a subset of G^s
G^v(i) ----> {$V^{s'}$, $E^{s'}$, $C_n^{s'}$, $C_l^{s'}$}

Fig. 2. An illustration of mapping for instantiation of VNs.

This decomposition will help reduce the overall complexity of resource assignment in VNs. However, it is important to note that node-assignment and link assignment problems are not independent of each other and solving them sequentially will not produce satisfactory results [14]. Intuitively, node assignment will effect link assignment and vice versa. Therefore, heuristics should attempt to simultaneously solve the VN nodes and links assignment sub-problems.

A. Objective functions for mapping of VNs

The instantiation process of VNs is intimately related to economical usage of SN resources. Hence, the mapping of substrate resources to VN topology must be carried out in accordance to optimization of an appropriate objective function.

[14] quantifies the resource usage in a SN by introducing the notions of node and link stress. These have been defined as: number of virtual nodes or virtual links assigned to a particular node or link in a SN. Logically, minimization of weighted sum of maximum values of node and link stresses has been formulated as an objective function for assignments to construct a VN. However, guidelines for selecting the weights have not been provided in [14].

In [15], the objective has been set to maximize the revenue generated by VN instantiations. Whereas, the revenue generated by a VN will be defined according to the economical and business model adopted by the provider of VNs. Considering the link bandwidth and CPU usage as two main resources of SN, the objective function has been defined in [15] as a long term average value of the weighted sum of bandwidth and CPU requirements for all virtual nodes and links. This approach of using weighted sum based objective functions is similar to [14] and is also silent about selection of weight parameter for CPU usage.

It is important to observe that objective functions defined in both [14] and [15], do not take into consideration the resiliency factor required for reliability of SN's paths being employed by a VN instantiation. If incorporated, a resiliency factor can help to reliably maintain the important links in a VN topology, when there is a path failure in underlying IP network, i.e. substrate. Thus, a new objective function, incorporating resiliency, can be proposed by adding a cost factor in function described in [14] and [15], for reservation/usage of extra paths in SN.

Topological resiliency can also be provided by reserving a set of substrate network's paths for crucial links in a VN topology [16]. Another approach would be to provide resiliency in virtual plane as has been proposed for overlay networks, [17]. Further, if there are multiple VN instantiations, wherein each is specialized for particular type of application, a different objective function will be needed for each network. However, as described later, the problem of VN embedding is NP-hard [18], so a different set of heuristics will be needed for each type of VN instantiation.

B. VN Nodes assignment as a NP-hard problem

The problem of assigning a VN's nodes to SN, without violating bandwidth constraints, is NP-hard and similar to multiway separator problem, [12] and [15]. To solve such problems, three approaches have been identified in [12]: brute-force backtracking algorithm, simulated annealing [19] [20] and approximation algorithms such as sparse cuts or multi-commodity flow problem. The backtracking algorithm depends upon clever heuristics, and is not a scalable solution for a large number of nodes. It has been also pointed out that the major drawbacks in simulated annealing are: failure to use the domain specific information available through graph properties and slow convergence properties. On the other hand, the multi-commodity flow problem has been shown to be NP-complete for integer flows, even for a pair of commodities, [21]. Hence, several approximation methods: e.g., [22], [23] and [24], have been surveyed in [12]; the details are being omitted for brevity. In general, due to its complexity, an efficient and largely scalable solution for node assignment problem in VNs is still elusive.

IV. Current Approaches to Resource Allocation in Virtual Networks

Broadly speaking, approaches to resource assignment in VNs can be categorized as: static and dynamic; whereas the former approach does not allow any change in resource assignment during the life-time of a VN, and the latter approach allows to adaptively change the resource allocations on the basis of current demand and performance of VN. The requested VNs can have diverse topologies and admission control mechanisms will be deployed by VN providers. The dynamic assignment of resources in VNs will require a constant monitoring of VN as well as dynamic updates of substrate's node and link capacities. An abstraction of this process has been depicted in Fig. 3.

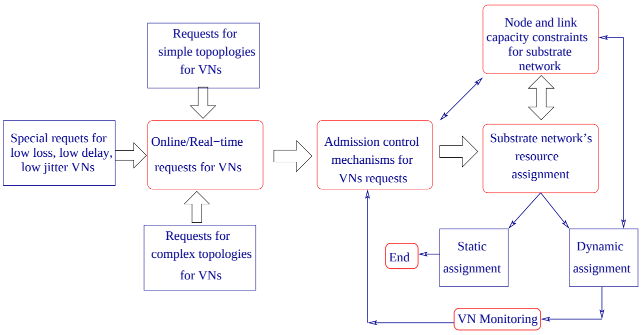
Requests for simple topoplogies for VNs
Special requets for low loss, low delay, low jitter VNs
Online/Real−time requests for VNs
Requests for complex topologies for VNs
Admission control mechanisms for VNs requests
Node and link capacity constraints for substrate network
Substrate network's resource assignment
End
Static assignment
Dynamic assignment
VN Monitoring

Fig. 3. Static and dynamic approaches for resource allocation in VNs.

A. Static Approaches

1) A basic algorithm: The problem of static assignments of resources to a VN has been investigated in [14] as an assignment without reconfiguration. It has been observed that the static mapping of VN node request can be considered as an offline load balancing problem which can be transformed into NP-hard unsplittable flow problem, [25]. Next, it has been conjectured that complexity of resource assignment problem in VNs will further increase due to stringent requirements of minimizing node and link stresses. Therefore, a heuristic approach has been adopted, i.e. by selecting a cluster of nodes that have low stress (lightly loaded) and will likely to cause a lower link stress when connected in a VN topology.

The shortest distance path algorithm [26], has been employed in [14] for evaluation of various available paths on SN. It depends upon a distance function, which has been originally defined as sum of reciprocals of available bandwidth on various available paths. It dynamically balances the impact of hop count and the path load on the computation of path distance for various candidate paths. This original distance function has been adapted for substrate link stress in [14]. Then, after computing minimum distance for all paths, node potential is computed, which has been defined as a ratio of sum of minimum distances of all substrate links in a cluster and maximum node stress. The next step consists of determining the potential of all the nodes in a cluster and mapping between substrate nodes and VN nodes is performed in such a way that virtual nodes with higher potential are connected to substrate nodes with higher neighbourhood resource availability. The last step involves the connection of selected SN nodes according to VN topology; for which again the shortest-distance path algorithm has been employed. It has also been pointed out that the basic assignment algorithm becomes inefficient for sparse topologies of VNs.

A simple modification to basic algorithm consists of subdividing the complete topology of a VN into smaller star topologies. These sub-topologies will be easier to be fitted into regions of low stress in the SN that will reduce the computation time. Further, optimization in resource assignment has been also attempted by identifying critical nodes and links in SN and then switching between the link and node optimization through the use of a threshold value [14].

2) Traffic constraints based algorithm: A cost effective method for designing VNs, though which may not yield optimal results due to NP-hard nature of problem, has been presented in [27]. The size of search space is reduced by restricting the VN topologies to back-bone star topologies. In such topology, some of the nodes are designated as back-bone while others are referred to as access nodes. The backbone nodes form the center of the star, to which access nodes are connected. The back-bone nodes can be connected in an arbitrary fashion, but they have been constrained to form a complete graph consisting of ring or star. The whole algorithm can expressed as an iterative loop shown in Fig. 4.

Next, three types of traffic constraints have been defined in [27]: (i) termination constraints that describe the total traffic terminating at the VN's access nodes and described by incoming and outgoing traffic from an access node [28], (ii) pairwise traffic constraints which provides upper bound on traffic flow from one access node to other, and (iii) distance constraints that specify the upper bounds on traffic flow outside the neighbourhood of a node. Link dimensioning has been done by following these constraints. The back-bone nodes mapping has been formulated and solved as a mixed integer quadratic program. Finally, VN designs are compared by cost metric defined by product of shortest path distance and fair traffic share function.

3) Splitting and Migration of Paths: A greedy node mapping algorithm with an objective to maximize revenue has been presented in [15]. It defines amount of resources available at node as a product of CPU capacity and link bandwidth. Link mapping is performed by k-shortest path algorithm, [29]. In order to improve efficiency of link assignment, path splitting has been proposed. Further, in order to achieve an efficient resource utilization in a scenario of time dependent requests for VN, path migration has also been proposed. For path migration the node mapping is needed to be kept fixed and either path splitting ratio can be varied or a completely new path in SN is selected. Also, in order to avoid out-of-order packet delivery, hash-based splitting schemes have been proposed, [30].

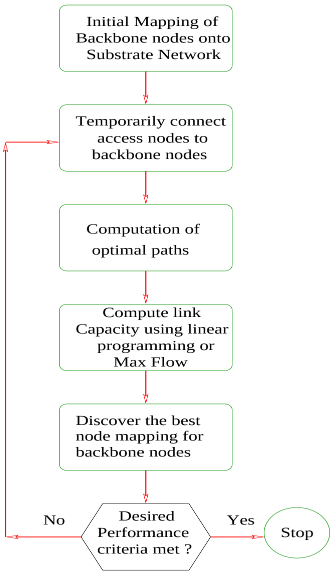
Initial Mapping of Backbone nodes onto Substrate Network
Temporarily connect access nodes to backbone nodes
Computation of optimal paths
Compute link Capacity using linear programming or Max Flow
Discover the best node mapping for backbone nodes
Desired Performance criteria met ?
No
Yes
Stop

Fig. 4. Iterative method for resource assignment in VNs, adapted from [27].

Path Splicing [31], a recently proposed routing primitive to allows network paths to be constructed by combining multiple routing trees to each destination over a single network topology, can also be experimented with efficient utilization of link resources in VNs.

B. Dynamic Approaches

A static resource assignment to multiple VNs, where each network is customized for a particular traffic class, can lead to lower performance and under utilization of substrate resources. It can lead to inefficient scenarios causing wastage of resources; e.g. over a same SN, one VN is experiencing a high packet loss, whereas the other VN is operating under a light traffic load. It will also effect delay and jitter sensitive VNs such as overlays for video transmission. Thus, it is important that an adaptive mechanism should be adopted to re-allocate the substrate network resources to various VN instantiations.

Taking inspiration from rerouting in circuit switched networks, [32], the problem of dynamic assignment of resources to VNs have been studied in [14]. However, it has been observed that the reconfiguration process in VN assignment is much more complex than flow routing. The event of reconfiguration, i.e. reallocation of resources to VNs, may involve a significant change in node and path switching in the SN. Thus, in order to quantify reconfigurations in VNs, a cost metric has been defined in [14]. It is a weighted sum of reconfiguration rate, node and path switching. It is important to realize that the number of reconfigurations of set of VNs over a substrate can be limited due to stability reasons and computational overheads. Hence, a selective reconfiguration process has been adopted, which gives priority to those parts of VNs that are highly loaded. The selective reconfiguration algorithm depends on: (i) periodic marking of critically stressed nodes and links of substrate and (ii) per VN reconfiguration and performance monitoring.

1) DaVinci: Recently, a framework for Dynamically Adaptive Virtual Networks for a Customized Internet (DaVinci) has been proposed in [33]. This architecture advocates a periodic reassignment of bandwidth among multiple VNs, which are sharing virtual resources derived from a common SN. In parallel, each VN runs its own distributed protocol to maximize its objective function. It allows the use of multiple (virtual) paths for reaching another node, which can cause packet reordering problem. Another weakness in this framework is that the links in SN need to know the performance objective function of all VNs, which may not be possible in the real world settings. Also, node assignment problem for VNs has not been considered by DaVinci.

C. Miscellaneous Approaches

1) Autonomic Systems based: A combined approach comprising of VNs and Autonomic computing [1], [34], has been proposed in [35]. It provides automated services and network resource management in DiffServ [36] enabled IP/MPLS [37] based transport networks.

In this architecture, a customer will request for the creation of a VN that is capable of delivering a desired level of a service. After the VN has been instantiated, its performance will be measured at regular intervals. The performance evaluation metrics comprise of packet loss rate, delay, jitter and end-to-end bandwidth. Also the provider will strive to achieve the optimized usage of his resources. Keeping in view of these stringent requirements, VN based Autonomic network Resource control and Management System (VNARMS) has been proposed in [35]. Two types of autonomic components defined are: Virtual Network Resource Manager (VNRM) and Resource Agents (RAs), where former is responsible for control/management and later performs the element-level resource control and management. In a nut shell, the four major components in an autonomic control loop formed by VNRM, are: resource manager for monitoring/executions, operations manager for monitoring and analysis, VN manager and topology manager for planning. However, the proposed autonomic system has not been yet implemented and thus, its performance in real systems is unknown.

2) Control Theoretic based systems: In [38], it has been suggested that computer systems should be designed in a way that they are amenable to feed-back control laws; for which several off-shelf adaptive controllers already exist. In the same breath, one of the promising techniques of resource allocation in virtualized network environments is adaptive control theory, [39] and [40]. However, no complete systematic approach exists for designing an optimal resource allocation paradigm for VNs, [15]. This open problem is further complicated by existence of NP-hardness in node and link assignment, cf. Section III. Despite of its complexity, some essential components of adaptive control system based resource assignment technique can be traced and has been depicted in Fig. 5.

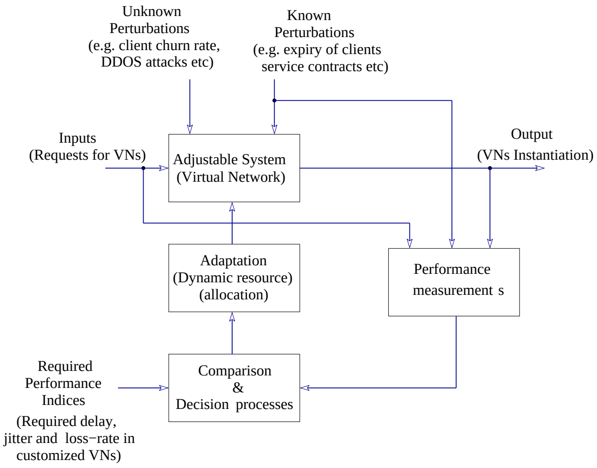

Fig. 5. Essential components of adaptive control system for VNs; adapted from [41].

The resource allocation process in VNs can also be modeled as closed loop system shown in Fig. 6. In this setup, the requests for instantiation of VNs arrive in real time; currently, it is difficult to find the arrival distribution as no commercial VNs provider exists [33]. The requests for VNs will be enqueued at provider's master queue and will be scheduled at an appropriate time according to service level agreement. In such setups, queueing will be an essential component of the loop as high computational costs may be associated with each VN request. Several possibilities exist for the design of scheduler for servicing VN requests, such as: weighted round robin, weighted fair queueing or priority queueing. A maximum time-to-live field in provider's queue may also be introduced as in VN request as in IP. After instantiation of the desired VN, either open loop or closed policy may be adopted. The former is similar to static resource assignment and the latter resembles dynamic resource assignment techniques.

Control theoretic approach has been also employed in [42] for self adaptation of virtual machines by employing the well known Additive Increase Multiplicative Decrease (AIMD) principle of Transmission Control Protocol (TCP). A virtual clock time, which is globally available to all virtual machines, has been introduced to detect overloading. The congestion signals are generated by taking a ratio of current virtual clock time and minimum of value of its exponentially weighted moving average. A proportional integral controller has been used for admission control of new threads in virtual machines.

Problem of automated control of virtualized resources has been investigated in [43]. A resource control system, Auto-Control, has been designed for adaptation to dynamic changes in shared virtualized infrastructure to achieve the desired levels of service level objectives for enterprise applications. It is a combination of an online model estimator, auto-regressive moving average, and multiple input multiple output controller.

[1] Autonomic computing helps to address the complexity of large systems by using technology to manage technology, with a minimal amount of human intervention [34].

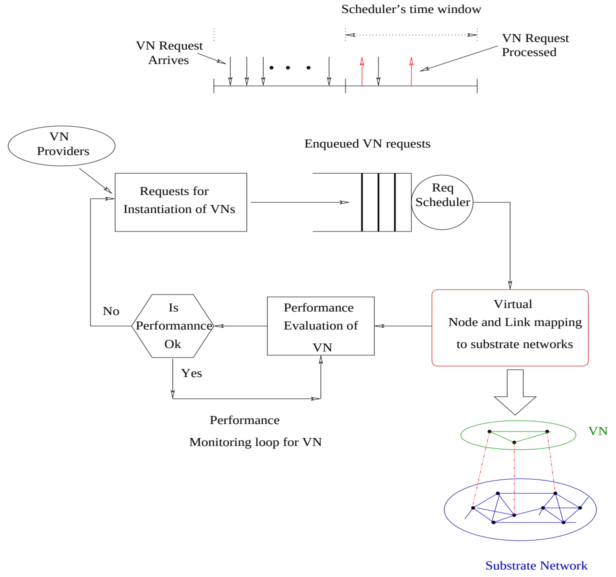

Fig. 6. A system-theoretical model for resource allocation processes in VNs.

Its performance has been evaluated through a test-bed consisting of three virtualized nodes, each running multiple VMs hosting multiple applications. The test-bed is running XEN virtual machines [44], benchmarks and various single/multi-tiered applications. It has been reported that AutoControl can detect CPU and disk bottlenecks across multiple nodes, adjust the resource allocation, can provide service differentiation and can enforce various performance targets.

A mechanism, named as QoSMap, attempting to incorporate both QoS and resiliency in constructing VNs over SN has been presented in [45]. It provides path resiliency by constructing alternate one hop overlay routes via intermediary nodes. The reported results are limited in scope and QoSMap is required to be tested thoroughly. In [46] a multi-commodity flow based approach has been applied for resource allocation in VNs. It has been aimed to connect this approach to VNRM system, cf. subsection IV-C1, though details of proposed algorithm have not been fully divulged.

V. Implementation of Resource Management

Resource management is a crucial issue in recently developed federated computing infrastructures, such as: PlanetLab [47], Emulab [48], Virtual Internet Infrastructure (VINI) [49] and Global Environment for Network Innovation (GENI) [50]. Such platforms, aiming to provide the Internet scale distribution and accessibility, require an effective resource management system for a fair, isolated, predictable and adaptive sharing of community's research resources.

For a solution of problem of resource discovery and allocation, several systems have been developed recently, [51]. These include: (i) centralized architectures such as: Condor Scheduling system [52] and Virtual Grids [53], (ii) hierarchical architectures such as Ganglia [54], and XenoSearch [55], and (iii) Decentralized architectures such as SWORD [56]. Secure Highly Available Resource Peering (SHARP) [57], Sirius [58], Bellagio [59] and Tycoon [60] have been modelled after a virtual market place where users can spend currency to obtain a share in system resources.

Basically, a VN is constructed by virtual hosts and virtual links. A virtual host is a network node that can add or remove indirection infrastructure. It experience illusion of dedicated physical physical host. Several virtual hosts can run over the same hardware of a physical host. Similarly, a virtual link mimics an isolated physical link. Although, several virtual links may have been instantiated from the same physical link. There can be a support of full virtualization where each node runs its own instance of Operating System (OS) or an OS level virtualization where some of resources of OS are isolated per virtual host [61]. An improvement to full virtualization (in terms of reducing virtualization overhead) is paravirtualization, which requires modifications to guest OS, [62]. VMWare server [63] and Kernel-based Virtual Machine (KVM) [64] provide full virtualization, whereas XEN [44] and Denali [62] provide support for paravirtualization. Some examples of OS level virtualization, also known as containers, are Linux VServers [65], FreeBSD Jails [66], Solaris Zones [67] and OpenVZ [68]. The virtual links can be implemented by virtual tunnels [69] or by sending Ethernet Frames over Generic Routing Encapsulation (GRE) tunnels [70], as has been employed in [61].

A. *PlanetLab and VINI*

PlanetLab, [47], users usually choose nodes for their experiments, from a set of all available nodes, on an ad-hoc basis. Although monitoring services are available, Ganglia [54], most users select the nodes in an arbitrary manner, [51]. This static allocation of nodes does not cope very well with rapidly changing network conditions, [71].

A resource allocator has been proposed in [51], which let users to specify the characteristics of their slices for which it automatically discovers the optimal resources. Further, it has been envisioned that a resource allocator should be asynchronous in nature, can adopt various allocation policies such as: providing hard guarantees, employing complex models such as economic market places and employing scheduling architectures.

A resource discovery mechanism, known as SWORD [56], has been employed in PlanetLab [47]. It is an advisory service, but not resource allocation, which consists of a distributed query service and an optimizer. The nodes satisfying the user submitted specifications are searched using a peer-to-peer network and then optimizer attempts to find a mapping with the lowest penalty function. Also, it carries out a continuous search and attempt to find new sets of re-matching nodes, which are returned to users for usage.

Fair sharing of overall bandwidth, memory and CPU capacity are also big challenges in PlanetLab. It employs Sirius [58] for brokerage services, CoStat [47] to gather data about state of local node which is then used by CoMon [72], Planetary scale Event Propagation and Routing (PsEPR) [73] and SWORD [56] to process information. Additionally, Stork [74] deploys, updates and configure services/experiments and PlanetFlow [75] is used for auditing services that log information about packet flows in PlanetLab, [76]. PlanetLab has employed Scout in Linux Kernel (SILK) scheduler in v2 [77], Class-based Kernel Resource Management (CKRM) in v3.0 and a modification of VServers CPU rate limiter to implement fair scheduling, [76].

VINI builds on PlanetLab (PL-VINI), [49]. However, it has been reported that PL-VINI is showing poor performance due to a need to implement the forwarding infrastructure in user mode in PlanetLab Kernel, [78]. A new implementation of VINI, called Trellis [61], has shown better performance because of moving network virtualization into Linux Kernel and enabling fast packet forwarding, [78]. In GENI [50], substrate resources are allocated by slice embedding service, [79].

B. Emulab

In Emulab [48], the set of nodes to be used in an experiment is determined by the `assign` facility, which is built around a simulated annealing core [19], [80]. It has been reported in [78] that `assign` facility has been enhanced by relaxing conservative resource assignment policies. Flexible resource specification has been carried out by adding more packing schemes for nodes and links. The scalability has been improved by finding sets of homogeneous physical nodes and combining them into equivalence classes. Also, instead of randomly selecting a physical node, `assign` selects it with some probability to which a neighbour has already been mapped. Coarsening virtual graph is also performed by pre-passing and dividing it into smaller sub-topologies to improve the performance. For coarsening, two algorithms have been implemented: (i) finding and combining all leaf nodes in a LAN to a conglomerate and (ii) using graph partitioner METIS [78], [81].

However, coarsening algorithms cannot completely capture the intricacies of mapping problem in VNs which will result into returning of set of nodes that cannot be mapped into physical network. This problem has been dealt by using multi-dimensional bin packing approximation algorithms [78]. A potential problem with this approach is fragmentation, i.e., coarsening algorithms generate set of nodes that cannot be packed into a physical node. It has been avoided by carefully selecting the size of returned conglomerate; wherein the worst case fragmentation caused 13 % increase in resource usage [78]. Emulab resource management techniques do not assure timeliness of events. However, user's application specific metrics can serve as safety thresholds and also a Kernel based mechanism for detecting resource usage over small time scales has been proposed. Currently, node support in Emulab is limited to FreeBSD with a partial support for XEN, [48].

VI. Research Challenges

After surveying existing work on resource allocation in VNs, we have identified the following five major research challenges:

- The basis of multiway separator approximation to multi-commodity flow problem, arising in resource allocation in VNs, [14] and [15], is sparse cuts for which several approximations are known to exist in literature, [12], [25] and [82]. A comparative study of various theoretical approximations for cuts in graphs will be useful for designing optimal resource allocation algorithms. Towards this end, an important aspect will be an in-depth investigation of tradeoffs existing between algorithmic complexity, speed, accuracy and maximum time-to-live for enqueued requests for instantiating VNs. Also, Divide-and-Conquer approximation algorithms via spreading metrics, [83], need to be considered for resource allocation in VNs, [12]. It will help to create better heuristics for static allocation of resources in VNs. However, transforming of these static allocations techniques to dynamic scenarios is an open problem.
- In sequel, an important research challenge consists of designing traffic adaptive techniques for resource allocation in VNs. Initial resource allocation to VNs can be made on the basis of long-term characteristics of network traffic. Later on, the initial allocation will be desired to be adaptive to shorter time scales of traffic variations. However, an important issue is the frequency of adaptation in allocated resources to various VNs. In such a dynamically adaptive environments, it would be worth to study tradeoffs among important aspects, such as: stability margins, optimality, signalling overheads and hardware limitations and resiliency of services, of overall system performance.
- After allocation of resources by intelligent heuristics, it would be interesting to apply the principles of adaptive feed-back control theory and/or game theory to further manage bandwidth and routing resources to varying traffic conditions in competing VNs, instantiated over a same SN. The recent emergence of Virtual Routers On the Move (VROOM), [84], demands the design of highly dynamic control techniques for managing the SN's routing resources. Moreover, not much work has been done for the application of autonomic computing techniques to resource allocation problem in VNs.
- For VNs test-beds, five major requirements for resource discovery and allocation have been identified in [51] as: (i) the resource allocator should be asynchronous, i.e., accepting jobs as they are submitted, (ii) interface for resource allocator should be fully QoS enabled, i.e., supporting immediate, queued and reserved styles of servicing resource requests, (iii) system should be able to provide guarantees for availability and quality of resources, (iv) responsible users should be given incentives, and (v) separation between resource discovery and allocation should not be mandatory. Incorporating these requirements, while maintaining Internet level scalability, into VN test-beds is a challenging task for future research.
- A fundamental issue in the design of future VN test-beds would be managing of trust among global users

[76], for which it would be desired to find a correct balance between centralized and decentralized architectural extremes e.g., PlanetLab Central and MyPLC [47]. Currently, a federation of all the major test-beds, through a unified interface for global authentication, is also under development [47]. Such efforts can help to materialize a fully flexible and globally automated resource allocation paradigm, that would be able to work smoothly within the complete spectrum of management policies adopted for VN test-beds.

It is important to clarify that the above presented list of research challenges is not exhaustive, but only an indicative overview of some major open problems. It does not attempt to encompass all of the research issues related to allocation and management of resources in VNs.

VII. Concluding Remarks

Virtualization has been proposed to overcome the weaknesses of the current Internet. Taking cues from current trends in industry, it can be anticipated that virtualization will be an essential part of future networks operation and designs. In this regard, an important challenge is allocation of substrate network resources to instantiate multiple VNs. An optimal allocation of SN's resources to instantiate VNs has been shown to be a NP-hard problem. Hence heuristic based approaches have been widely used in literature.

Resource allocation in VNs can be carried out either in a static or dynamic fashion; wherein the former case, allocated resources are dynamically controlled and adjusted according to the current requirements of VNs. Thus, the dynamical assignment approach for VNs is much more difficult to design. However, one of the important requirements for optimal usage of resources is adaptation and responsiveness to changing traffic patterns in different VNs sharing a same SN; which will require a dynamic approach towards resource allocation. A semi-dynamic approach for resource allocation is possible by adopting path migration, splitting and splicing in the topology of the SN.

The complexity of the resource allocation problem can be reduced by decomposing it into two sub-problems: node and link assignment. However, an attempt to solve these two sub-problems in a sequential manner will not yield satisfactory results, [14]. Thus, intelligent heuristics, which can solve both sub-problems synchronously, are desired. A promising technique for resource assignments in VNs is offered by autonomic systems. However, such approaches are still in their infancy and not readily available. Recently, adaptive feedback control systems have been employed to control the virtualized resources. Finally, resource allocation concepts and techniques as being applied in VNs test-bed infrastructures, such as in Emulab [48] and GENI [50], can also be very useful for designing future resource allocation methodologies for VNs.

References

[1] Sylvia Ratnasamy, Andrey Ermolinskiy and Scott Shenker, *Revisiting IP Multicast*, ACM SIGCOMM Computer Communication Review, Vol. 36 , Issue 4, pp. 15-26, October 2006.

[2] Jon Crowcroft, Steven Hand, Richard Mortier, Timothy Roscoe and Andrew Warfield, *QoS's Downfall: At the bottom, or not at all !*, Proc. of the ACM SIGCOMM workshop on Revisiting IP QoS: What have we learned, why do we care?, pp. 109-114, 2003.

[3] Craig Labovitz, *The End is Near, but is IPv6?*, Arbor Networks Security Blog, August 2008, http://asert.arbornetworks.com/2008/08/the-end-is-near-but-is-ipv6/, http://penrose.uk6x.com/

[4] Jonathan S. Turner and David E. Taylor, *Diversifying the Internet*, Proc. of IEEE Globecom'05, Vol. 2, pp. 755-760, 2005.

[5] L. Peterson, S. Shenkar and J. Turner, *Overcoming the Internet Impasse Through Virtualization*, IEEE Computer, Vol. 38, No. 4, pp. 34-41, 2005.

[6] Gartner, *Gartners Top 10 Strategic Technologies for 2009*, http://blogs.gartner.com/david_cearley/category/top-10/, October 2008.

[7] Juniper Networks, Inc., *Intelligent Logical Router Service*, 2004, http://www.juniper.net/solutions/literature/white_papers/200097.pdf

[8] Juniper Networks, Inc., *Partner Solution Development Platform*. http://www.juniper.net/partners/osdp.html

[9] Network World, *Cisco opening up IOS*, December 2007. http://www.networkworld.com/news/2007/121207-cisco-ios.html

[10] Cisco, *Virtual Router Redundancy Protocol*, http://www.cisco.com/en/US/docs/ios/12_0st/12_0st18/feature/guide/st_vrrpx.html#wp1035071

[11] Jon Crowcroft, Steven Hand, Richard Mortier, Timothy Roscoe and Andrew Warfield, *Plutarch: An Argument for Network Pluralism*, Proc. of the ACM SIGCOMM workshop on Future directions in network architecture, pp. 258-266, 2003.

[12] D. G. Andersen, *Theoretical Approaches to node assignment*, Unpublished Manuscript, http://www.cs.cmu.edu/~dga/papers/index.html, 2002.

[13] N. Niebert, I. E. Khayat, S. Baucke, R. Keller, R. Rembarz and J. Sachs, *Network Virtualization: A Viable Path Towards the Future Internet* Springer Wireless Pers. Commun., pp. 511-520, March 2008.

[14] Yong Zhu and Mostafa Ammar, *Algorithms for Assigning Substrate Network Resources to Virtual Network Components* Proc. of IEEE INFOCOM'06, pp. 1-12, 2006.

[15] Minlan Yu, Yung Yi, Jennifer Rexford and Mung Chiang, *Rethinking Virtual Network Embedding: Substrate Support for Path Splitting and Migration*, ACM SIGCOMM CCR, Vol. 38 , Issue 2, pp. 17-29, 2008.

[16] Peter Key, Laurent Massoulie and Don Towsley, *Combining Multipath Routing and Congestion Control for Robustness*, 40th Annual Conference on Information Sciences and Systems, pp. 345-350, 2006.

[17] Cui Weidong, Ion Stoica, Randy H. Katz. *Backup Path Allocation Based On A Correlated Link Failure Probability Model In Overlay Networks* Proc. of 10th IEEE Intl. Conf. on Network Protocols, pp. 236-245, 2002.

[18] Michael R. Garey and David S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*, Publ. W. H. Freeman, 1979.

[19] S. Kirkpatrick, C. D. Gelatt, Jr., M. P. Vecchi, *Optimization by Simulated Annealing*, Science, Vol. 220, No. 4598, pp. 671-680, May 1983.

[20] Lester Ingber, *Very Fast Simulated Re-Annealing*, J Mathl. Comput. Modelling, Vol. 12, pp. 967-973, 1989. ftp://ftp.ingber.com/asa89_vfsr.pdf

[21] S. Even and A. Itai and A. Shamir, *On the Complexity of Timetable and Multicommodity Flow Problems*, SIAM Jour. on Comp., Vol. 5, No. 4, pp. 691-703, 1976. http://www.cs.technion.ac.il/~itai/publications/2CF.pdf

[22] Tom Leighton, Fillia Makedon and Spyros Tragoudas, *Approximation Algorithms for VLSI Partition Problems* IEEE Intl. Symposium on Circuits and Systems, Vol. 4, pp. 2865-2868, 1990.

[23] Satish Rao, *Finding Near Optimal Separators in Planar Graphs*, 28th-Annual Symp. on Foundations of Computer Science, pp. 225-237, 1987.

[24] Tom Leighton and Satish Rao, *Multicommodity Max-Flow Min-Cut Theorems and Their Use in Designing Approximation Algorithms*, Journal of the ACM, Vol. 46, No. 4, pp. 787-832, November 1999.

[25] Stavros G. Kolliopoulos and Clifford Stein, *Improved Approximation Algorithms for Unsplittable Flow Problems*, Proc. of 38-th IEEE Symposium on Foundations of Computer Science, pp. 426-436, 1997.

[26] Qingming Ma and Peter Steenkiste, *On Path Selection for Traffic with Bandwidth Guarantees*, Proc. International Conference on Network Protocols, pp. 191-202, 1997.

[27] Jing Lu and Jonathan Turner, *Efficient Mapping of Virtual Networks onto a Shared Substrate*, WUCSE-2006-35, Washington University in St. Louis, 2006. http://cse.seas.wustl.edu/Research/FileDownload.asp?503

[28] N. G. Duffield, Pawan Goyal, A. Greenberg, P. Mishra, K. K. Ramakrishnan, J. E. van der Merwe, *A Flexible Model for Resource Management in Virtual Private Networks*, Proc. ACM SIGCOMM'99, pp. 95-108, 1999.

[29] David Eppstein, *Finding the k Shortest Paths*, SIAM Journal on Computing, pp. 652-673, 1999.

[30] David Karger, Eric Lehman, Tom Leighton, Rina Panigrahy, Matthew Levine and Daniel Lewin, *Consistent Hashing and Random Trees: Distributed Caching Protocols for Relieving Hot Spots on the World Wide Web*, Proc. of the 29-th Annual ACM symposium on Theory of computing, pp. 654-663, 1997.

[31] Murtaza Motiwala, Megan Elmore, Nick Feamster and Santosh Vempala, *Path Splicing*, ACM SIGCOMM CCR, Vol. 38, Issue 4, pp. 27-38, 2008.

[32] Larry Peterson, Andy Bavier, Marc E. Fiuczynski and Steve Muir, *A Taxonomy of Rerouting in Circuit-Switched Networks*, IEEE Communication Magazine, Vol. 37, Issue 11, pp. 116-122, 1999.

[33] Jiayue He, Rui Zhang-Shen, Ying Li, Cheng-yen Lee, Jennifer Rexford and Mung Chiang, *DaVinci: Dynamically Adaptive Virtual Networks for a Customized Internet*, Accepted for ACM CoNEXT 2008.

[34] IBM Corporation *An architectural blueprint for autonomic computing*, 2006, http://www-01.ibm.com/software/tivoli/autonomic/pdfs/AC_Blueprint_White_Paper_4th.pdf, http://www.research.ibm.com/autonomic/overview/elements.html

[35] Myung Sup Kim, Ali Tizghadam, Alberto Leon-Garcia and James Won-Ki Hong, *Virtual Network based Autonomic Network Resource Control and Management System*, Proc. of IEEE Globecom, pp. 1075-1079, 2005.

[36] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang and W. Weiss, *An Architecture for Differentiated Services*, RFC 2475, Dec. 1998. http://www.ietf.org/rfc/rfc2475.txt

[37] IP/MPLS forum. http://www.ipmplsforum.org/

[38] Christos Karamanolis, Magnus Karlsson and Xiaoyun Zhu, *Designing Controllable Computer Systems*, Proc. of 10th Conference on Hot Topics in Operating Systems, 2005.

[39] Magnus Karlsson, *Design Rules for Producing Controllable Computer Services*, Proc. of 10th IEEE/IFIP Network Operations and Management Symposium, pp. 1-14, 2006.

[40] Karl J. Astrom and B. Wittenmark, *Adaptive Control*, Addison-Wesley, 1995.

[41] V. V. Chalam, *Adaptive Control Systems: Techniques and Applications*, CRC Press, 1987.

[42] Yuting Zhang, Azer Bestavros, Mina Guirguis, Ibrahim Matta and Richard West, *Friendly Virtual Machines: Leveraging a Feedback-Control Model for Application Adaptation* Proc. of the 1st ACM/USENIX Intl. Conf. on Virtual execution environments, pp. 2-12, 2005.

[43] Pradeep Padala, Kai-Yuan Hou, Kang G. Shin, Xiaoyun Zhu, Mustafa Uysal, Zhikui Wang, Sharad Singhal and Arif Merchant, *Automated Control of Multiple Virtualized Resources* HP-Lab Technical Report HPL-2008-123, October 2008. http://www.hpl.hp.com/techreports/2008/HPL-2008-123.pdf

[44] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt and A. Warfield, *Xen and the Art of Virtualization* ACM SOSP'03, pp. 164-177, 2003

[45] J. Shamsi and M. Brockmeyer, *QoSMap: QoS aware Mapping of Virtual Networks for Resiliency and Efficiency*, Proc. of IEEE Globecom Workshops, pp. 1-6, 2007.

[46] W. Szeto, Y. Iraqi and R. Boutaba, *A Multi-Commodity Flow Based Approach to Virtual Network Resource Allocation*, Proc. of IEEE Globecom, Vol. 6, pp. 3004-3008, 2003.

[47] PlanetLab, http://www.planet-lab.org/

[48] Emulab - Network Emulation Testbed, http://www.emulab.net/

[49] Virtual Network Infrastructure (VINI), http://www.vini-veritas.net/

[50] GENI, http://www.geni.net/.

[51] R. Ricci, D. Oppenheimer, Jay Lepreau and Amin Vahdat, *Lessons from Resource Allocators for Large-Scale Multiuser Testbeds*, ACM SIGOPS Operating Systems Review, Vol. 40 , Issue 1, pp. 25-32, January 2006.

[52] Michael J. Litzkow, Miron Livny and Matt W. Mutka, *Condor- A Hunter of Idle Workstations*, Proc. of IEEE 8-th Intl. Conf. on Dist. Computing systems, pp. 104-111, 1988.

[53] Yang-suk Kee and Carl Kesselman *Grid Resource Abstraction, Virtualization, and Provisioning for Time-targeted Applications*, Proc. of 8-th Intl. Symposium on Cluster Computing and Grid, pp. 324-331, May 2008. http://vgrads.rice.edu/publications/conferencereference.2008-05-12.6886931242

[54] Matthew L. Massie, Brent N. Chun and David E. Culler, *The ganglia distributed monitoring system: design, implementation, and experience*, Parallel Computing, Vol. 30, pp. 817840, 2004.

[55] David Spence and Tim Harris, *XenoSearch: Distributed Resource Discovery in the XenoServer Open Platform*, Proc. of 12-th IEEE Symposium on High Performance Distributed Computing, pp. 216-225, 2003.

[56] Jeannie Albrecht, David Oppenheimer, Amin Vahdat and David A. Patterson, *Design and Implementation Trade-offs for Wide-area Resource Discovery*, ACM Transactions on Internet Technology, Vol. 8 , Issue 4, Article 18, September 2008.

[57] Yun Fu and Jeffrey Chase, Brent Chun, Stephen Schwab and Amin Vahdat, *SHARP: An Architecture for Secure Resource Peering*, Proc. of the 9-th ACM symp. on operating systems principles, pp. 133 - 148, 2003.

[58] Sirius Calendar Service, https://snowball.cs.uga.edu/~dkl/pslogin.php

[59] Alvin AuYoung, Brent N. Chun, Alex C. Snoeren and Amin Vahdat, *Resource Allocation in Federated Distributed Computing Infrastructures*, Workshop on Operating Systems and Arch. Support for the on demand IT InfraStructure, 2004. http://www.theether.org/papers/oasis04.pdf

[60] Kevin Lai, Lars Rasmusson, Eytan Adar, Stephen Sorkin, Li Zhang and B. A. Huberman, *Tycoon: an Implementation of a Distributed, Market-based Resource Allocation System*, HP-Labs Technical Report, Dec. 2004. http://www.hpl.hp.com/research/tycoon/doc/csDC0412038.pdf

[61] S. Bhatia, Murtaza Motiwala, Wolfgang Muhlbauer, V. Valancius, Andy Bavier, Nick Feamster, Larry Peterson, and Jennifer Rexford, *Hosting Virtual Networks on Commodity Hardware* Technical Report GT-CS-07-10, January 2008. http://www.cs.princeton.edu/~jrex/papers/trellis07.pdf

[62] Andrew Whitaker, Marianne Shaw and Steven D. Gribble, *Denali: Lightweight Virtual Machines for Distributed and Networked Applications*, University of Washington Technical Report 02-02-01, http://denali.cs.washington.edu/pubs/distpubs/papers/denali_usenix2002.pdf

[63] VMWare Server, http://www.vmware.com/products/server/

[64] Kernel-Based Virtual Machines, http://kvm.qumranet.com/kvmwiki

[65] Linus-Vserver, http://linux-vserver.org/

[66] Poul-Henning Kamp and Robert N. M. Watson, *Jails: Confining the omnipotent root*, Proc. of SANE 2000, 2000. http://phk.freebsd.dk/pubs/sane2000-jail.pdf

[67] Solaris Containers, http://www.sun.com/bigadmin/content/zones/

[68] OpenVZ, http://wiki.openvz.org/Main_Page

[69] Virtual Tunnels, http://vtun.sourceforge.net/

[70] D. Farinacci, T. Li, S. Hanks, D. Meyer and P. Traina, *Generic Routing Encapsulation (GRE)*, RFC 2784, March 2000, http://www.rfc-editor.org/rfc/rfc2784.txt

[71] D. Oppenheimer, B. Chun, D. Patterson, Alex C. Snoeren and Amin Vahdat, *Service Placement in a Shared Wide-Area Platform*, Proc. of USENIX Annual Technical Conference, pp.273-288, 2006. http://www.usenix.org/event/usenix06/tech/full_papers/oppenheimer/oppenheimer.pdf

[72] CoMon, A Monitoring Infrastructure for PlanetLab, http://comon.cs.princeton.edu/

[73] Planetary scale Event Propagation and Routing, http://psepr.org/

[74] Stork, http://www.cs.arizona.edu/stork/index.html

[75] Mark Huang, Andy Bavier and L. Peterson, *PlanetFlow: Maintaining Accountability for Network Services*, ACM SIGOPS Operating Systems Review, Vol. 40, Issue 1, 2006. http://www.planet-lab.org/taxonomy/term/30

[76] L. Peterson, Andy Bavier, Marc E. Fiuczynski and Steve Muir, *Experiences Building PlanetLab*, Proc. USENIX OSDI '06, pp. 351-366, 2006.

[77] Andy Bavier, Mic Bowman, Brent Chun, David Culler, Scott Karlin, Steve Muir, Larry Peterson, Timothy Roscoe, Tammo Spalink and Mike Wawrzoniak, *Operating System Support for Planetary-Scale Network Services*, Proc. of 1st NSDI, pp. 253-266, 2004.

[78] Mike Hibler, Robert Ricci, Leigh Stoller, Jonathon Duerig, Shashi Guruprasad, Tim Stack, Kirk Webb and Jay Lepreau, *Large-scale Virtualization in the Emulab Network Testbed*, Proc. of Usenix, pp. 113-128, 2008. http://www.usenix.org/events/usenix08/tech/hibler.html

[79] GENI Planning Group, *GENI Facility Design*, March 2007, http://geni.net/GDD/GDD-07-44.pdf

[80] R. Ricci, C. Alfeld, and J. Lepreau, *A Solver for the Network Testbed Mapping Problem*, ACM CCR, Vol. 33, Issue 2, pp. 65-81, 2003.

[81] METIS, Family of Multilevel Partitioning Algorithms, http://glaros.dtc.umn.edu/gkhome/views/metis

[82] G. Even, J. Naor, S. Rao and B. Schieber, *Fast Approximate Graph Partitioning Algorithms*, SIAM J. of Computing, Vol. 28. No. 6, pp. 2187-2214, 1999.

[83] G. Even, J. Naor, S. Rao and B. Schieber, *Divide-and-Conquer Approximation Algorithms via Spreading Metrics*, Journal of the ACM, Vol. 47, No. 4, pp. 585-616, 2000.

[84] Yi Wang, Eric Keller, Brian Biskeborn, J. Merwe and J. Rexford, *Virtual Routers on the Move: Live Router Migration as a Network Management Primitive*, ACM SIGCOMM Computer Communication Review, Vol. 38, No. 4, pp. 231-242, October 2008.