

*Virtual Network On Demand: Dedicating Network
Resources to Distributed Scientific Workflows*

Dimitrios Katramatos, Sushant Sharma, Dantong Yu

The Fifth International Workshop on Data Intensive Distributed Computing (DIDC 2012)
Delft, The Netherlands
June 18-22, 2012

February 2012

Computational Science Center

Brookhaven National Laboratory

**U.S. Department of Energy
Office of Science**

Office of Advanced Scientific Computing Research

Notice: This manuscript has been authored by employees of Brookhaven Science Associates, LLC under Contract No. DE-AC02-98CH10886 with the U.S. Department of Energy. The publisher by accepting the manuscript for publication acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

This preprint is intended for publication in a journal or proceedings. Since changes may be made before publication, it may not be cited or reproduced without the author's permission.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or any third party's use or the results of such use of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof or its contractors or subcontractors. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Virtual Network On Demand: Dedicating Network Resources to Distributed Scientific Workflows

Dimitrios Katramatos
Brookhaven National
Laboratory
Upton, NY 11973
dkat@bnl.gov

Sushant Sharma
Brookhaven National
Laboratory
Upton, NY 11973
sushant@bnl.gov

Dantong Yu
Brookhaven National
Laboratory
Upton, NY 11973
dtyu@bnl.gov

ABSTRACT

The VNOD project aims to build an on-demand network virtualization infrastructure that can deliver the unprecedented networking performance and quality of service required by modern, distributed, data-intensive applications utilized by user communities. This infrastructure leverages technologies both recently deployed in production and currently under development to dynamically provision network resources interconnecting multiple end sites hosting storage and computing resources. VNOD provides an environment that facilitates the establishment and management of virtual topologies and at the same time offers a platform for co-scheduling end-site resources with local and wide-area network resources using different algorithms and optimization objectives.

Keywords

network, virtualization, co-scheduling

1. INTRODUCTION

In the world of modern data-intensive scientific computing efficient and predictable data movement between globally distributed compute and storage sites is key to successfully executing the workflows necessary for a community of users to accomplish their goals. For example, in communities such as the high-energy and nuclear physics, genomics, and climate modeling, to name a few, teams of scientists around the globe need to routinely share petabytes of experimental data for processing, analysis, and visualization. Transferring and processing data at such a scale imposes heavy requirements on system capabilities, especially on the network, while coordination is also central to fair and effective sharing of resources.

The behavior of the network has therefore a major effect on the performance and efficiency of applications, and as a result, of their research communities. Recent advances in networking technology have enabled several research and

development efforts that focus on the allocation of network resources for specific activities. New networking infrastructure in major research and education (R&E) networks, albeit on an experimental or near-production mode, allows middleware to provide users/applications with the ability to reserve fractions of the resources of a network domain for dedicated, scheduled use. ESnet [7] and Internet2 [14] support dynamically established circuits through their backbone R&E networks, using the On-demand Secure Circuits and Advance Reservation System (OSCARs) [19] software. Europe's GÉANT [10] has similar capabilities through their AutoBAHN project [8]. Brookhaven National Lab's (BNL) TeraPaths [16, 17], and the recently funded End-Site Control Plane Service (ESCPS) [27] projects focus on creating on-demand end-to-end (host-to-host) virtual paths with guaranteed bandwidth by acquiring and extending such dynamic circuits into end-site local area networks (LANs) and managing the bandwidth they provide. Distributed monitoring infrastructures, such as perSONAR [26], facilitate sharing domain information across participating network domains. The StorNet project [28] successfully demonstrated how network resources can be co-scheduled with storage resources to achieve predictable and reliable data transfers. The VNOD project builds on these developments aiming to develop new capabilities as part of creating a scalable and reliable network virtualization system, a next generation middleware system that will accommodate the networking needs of user/application communities with applications requiring access to widely distributed sets of resources.

The thus far developed infrastructure and middleware can be used to construct end-to-end paths between eligible sites, however, such paths can only be established between one pair of endpoints at a time and, furthermore, require detailed knowledge of the available services along a path - information which most users cannot find easily. Accommodating the needs of an application community requires establishing sets of paths as directly related entities serving a common purpose. Such paths must be collectively managed and monitored. Providing the required performance levels, fault tolerance, and recovery are key elements as minor disturbances on even a single path may adversely affect the whole community. With this perspective in mind, VNOD integrates the functionality of the emerging network resource reservation and distributed monitoring services to provide Virtual Network domains (ViNets), logical network constructs that dedicate network resources to user/application communities.

The remainder of this paper is organized as follows: in section 2, we provide some background information for the projects whose functionality is necessary for VNOD. In section 3 we present the architecture of the VNOD system, while in section 4 we focus on the aspect of resource co-scheduling. In section 5 we describe the functionality of our early prototype implementation. In section 6, we discuss works with similar goals as ours. Finally, in section 7 we summarize and discuss future work directions.

2. BACKGROUND

The VNOD project relies on the technology and know-how developed by the TeraPaths [1], OSCARS [19], and the more recent StorNet [28] and ARCHSTONE [29] projects, as well as under development by the ESCPS project. In this section we provide a brief description of each project and the capabilities that VNOD integrates.

2.1 TeraPaths

The TeraPaths system establishes virtual paths by directly configuring the networking hardware of end-site LANs and by interfacing with the OSCARS software to acquire dynamic circuits through the WAN domains that interconnect the end-sites. Within LANs, TeraPaths uses DiffServ-based QoS [?] to protect and regulate individual end-to-end flows. To carry these already conditioned flows through WAN domains, TeraPaths uses Policy-Based Routing (PBR) to steer the flows in dynamic circuits (Layer 2) provisioned by OSCARS. The TeraPaths project has demonstrated the feasibility, effectiveness, advantages, and disadvantages of using end-to-end virtual paths with guaranteed QoS to reserve/schedule network resources for exclusive use and specific time periods. The capability to establish point-to-point virtual paths is fundamental for VNOD.

2.2 OSCARS

The On-demand Secure Circuit Advance Reservation System (OSCARS) is a project, initiated by ESnet and with the collaboration of Internet2. A major achievement of the project is the development and standardization of the Inter-Domain Controller (IDC) protocol. An OSCARS instance dynamically provisions secure, guaranteed bandwidth circuits within a network domain it controls and can communicate through the IDC protocol with the OSCARS instances controlling other network domains, or, for that matter, instances of other controllers that use the IDC protocol, to propagate the provisioning of circuit segments in these domains. OSCARS initially provided guaranteed bandwidth circuits within ESnet in the form of MPLS tunnels (layer 3). Through the collaboration between ESnet and Internet2, the system evolved into a more general Inter-Domain Controller which, except for the MPLS tunnels within ESnet, provides guaranteed bandwidth layer 2 circuits within and between ESnet's Science Data Network (SDN) and Internet2's Dynamic Circuit Network (DCN).

OSCARS exposes a web services API through which it is possible to create and manage circuit reservations between specific network endpoints. TeraPaths (and ESCPS) utilize this API to acquire circuits as segments of end-to-end paths between end-sites. The latest version of the software follows a fully modular architecture allowing for easy addition/modification of components and features.

2.3 ESCPS

The End-Site Control Plane Service project is developing the next generation virtual path creation middleware layer. ESCPS inherits the functionality of TeraPaths and the additional API and co-scheduling functionality that were developed for StorNet in a strengthened and mature code base. Furthermore, ESCPS incorporates new technologies for establishing virtual paths within an end-site's LAN. Notably, ESCPS can extend a WAN circuit all the way to a host by using a site-specific OSCARS instance to establish a circuit segment within the site's LAN and a host-based software agent to attach this circuit to a virtual network interface on a host.

2.4 StorNet

The StorNet project [28] has developed a framework to co-schedule storage and network resources and provide predictable and reliable data transfers. Through this framework, BeStMan [?], an implementation of the SRM standard [?], interfaces with TeraPaths (which in turn interfaces with OSCARS for provisioning WAN circuits) to acquire bandwidth guarantees for its planned data transfers. Major contributions of this project were the BeStMan/TeraPaths API and the functionality that was added to BeStMan and TeraPaths to co-schedule storage and network resources by coordinating the reservations of the multiple instances of BeStMan, TeraPaths and OSCARS involved in a data transfer between two end-sites.

2.5 ARCHSTONE

The Advanced Resource Computation for Hybrid Service and TOpology NETworks project [29] is developing technologies that enable resource computation and provisioning across next generation multi-layer network architectures. Core component in the ARCHSTONE software is the Multi-layer/Multi-dimensional Topology Computation Element (MX-TCE). This component can perform sophisticated path and topology computations. Given suitable input data, MX-TCE can provide answers to "what is possible/available?" questions from clients seeking to reserve bandwidth on paths or topologies. ARCHSTONE is an extension to OSCARS and utilizes an extension to the IDC protocol to enable client communication with the MX-TCE component.

3. SYSTEM ARCHITECTURE

The VNOD system creates virtual network topologies (ViNets) interconnecting the resources of end-sites (see figure 1). Each virtual topology comprises a set of end-to-end virtual paths. The resources at the endpoints of each path "think" that they are a single hop away from each other, while at the same time the bandwidth between them is guaranteed to be up to the level reserved for each path.

The architecture of VNOD is shown in figure 2. On top of the physical network hardware there is a layer of middleware, comprising the TeraPaths/ESCPS and OSCARS with ARCHSTONE extensions systems. The functionality of these systems constitutes a virtual end-to-end path layer. On top of this layer sits, in turn, the functionality of VNOD, a virtual networking layer.

Five notable aspects of VNOD are the following:

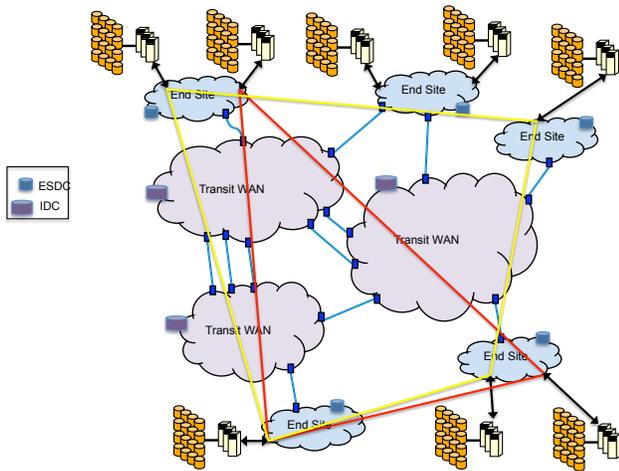


Figure 1: Virtual networking layer.

- The virtual paths that comprise a ViNet are truly “end-to-end”. This follows from the functionality of TeraPaths/ESCPS upon which VNOD is based. Such paths are established between end-site machines and extend from network card to network card. As such, there are one or more campus network devices between each machine and the network devices of the regional or wide-area network provider(s) that connect each end-site with the rest of the world. While the term “end-to-end” is routinely used to qualify connections, in most - if not in all - cases, it is stated or implied that the machines are on special networks or are directly connected to the devices of the WAN.
- VNOD virtualization results from direct or indirect network connection configuration. This is a decisive difference between our approach and overlay networks, which rely on virtual devices - software objects that use the best-effort Internet. Configuring real, guaranteed network services to prioritize, protect, and regulate data flows is the only effective method to provide QoS guarantees because the implementation is at the network level.
- VNOD utilizes “profiles” to describe a virtual topology interconnecting a set of end-sites. Such topologies are simple, since they’re based on virtual paths, and do not reflect the actual physical topology that will have to be configured (see figure 1). A profile serves as a template that can be combined with sets of bandwidth requests between end-sites. This constitutes a scheduling problem that VNOD needs to find solutions for. Each solution is a set of network bandwidth reservations that can be passed to the underlying virtual end-to-end path layer for implementation.
- The project’s goal, at least at this stage, is to support data intensive/real-time users of R&E networks, who constitute a relatively small and manageable set compared to the Internet as a whole. Therefore, VNOD does not encounter the scaling and management issues that a system intended for the Internet at large must address.

- The design of VNOD follows a distributed system approach. A VNOD instance may be responsible for one or more end-sites and WAN domains, and a ViNet may be collectively established and managed by the coordinated actions of multiple instances. However, in a deployment where scalability is not an issue, it is easy to resort to a centralized approach by delegating the responsibility for the scheduling of paths through all participating domains and end-sites to a single VNOD instance.

The VNOD architecture comprises middleware components that interface with the virtual end-to-end path creation and management services and a set of distributed auxiliary services (e.g. service discovery, topology data, monitoring, and user community membership) to dynamically create distributed application ViNets. This architecture is depicted in figure 2. The VNOD middleware system has two major components: the Resource Scheduler (RS) and the Virtual Network Domain Controller (VNDC). The RS accepts resource requests submitted in the form of application resource requirement profiles. Subsequently, the RS consults service discovery servers and gathers site resource availability information and physical network topology information. Then, it executes a scheduling algorithm to select a ViNet topology and generate appropriate resource reservation sets that meet the submitted requests. If a solution is found, the RS passes the generated ViNet creation information to its local VNDC to implement. The RS also reports back with status information for user requests (success or failure). A local RS coordinates with remote RS instances, as necessary, during information gathering and ViNet creation. VNOD integrates the requested resources within a ViNet, thus creating a virtual “container” for the application (see Figure 3), i.e., a dedicated environment for the application that runs protected from resource contention from other applications. Establishing, managing, and monitoring status of ViNets is the job of the VNDCs. End-site domain controllers, such as TeraPaths or ESCPS instances, already have the capabilities to configure and manage individual end-to-end QoS paths between specific source and destination endpoints (configuration of transit domain segments takes place indirectly through interfacing with OSCARS). The VNDCs use these capabilities to establish ViNets. The VNDCs act as clients to the underlying end-site control systems and, through their APIs, have access to a set of primitive operations to bring up, tear down, and manage end-to-end virtual paths. Each such path can be associated with one or more individual data flows.

Establishing a ViNet requires establishing multiple virtual paths interconnecting the host nodes of endpoints. Multiple VNDCs coordinate to establish ViNets and maintain coherent information for their ViNets (in a centralized deployment, this is accomplished by a single VNDC). The tasks required to establish and manage the individual virtual paths composing a ViNet are distributed, as allowed by access policies, among end-site controller instances at participating sites; this greatly reduces the required response time. VNDCs are also responsible for monitoring the status of a ViNet throughout its lifetime. After creation, they register the ViNet with monitoring services, periodically checking and logging status information, and listening for notifica-

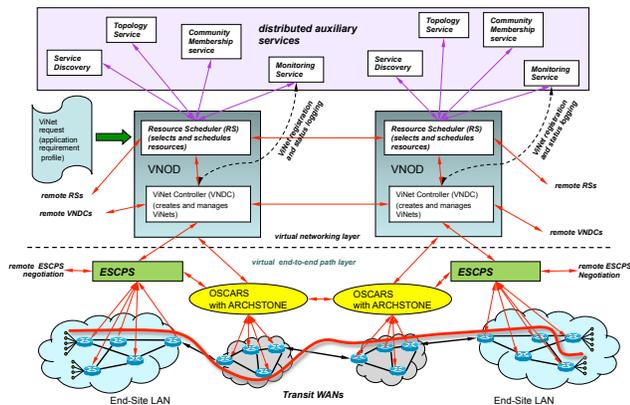


Figure 2: The architecture of VNOD.

tions of error conditions. The VNOD approach is particularly well-suited to “tie” together a number of sites included in a high-level workflow involving moving and processing data. The motivation is to provide guaranteed network resources so that distribution of data from storage to processing sites and/or forwarding of data between processing sites can be performed at rates sufficient to, e.g., ensure maximum utilization of computing resources during the time window these resources are available to a user community, or sustain smooth frame rates in a distributed visualization pipeline. The network virtualization process uses as basis an application profile or directives from a workflow system as a description of resource requirements. These requirements are taken into account along with the resource availability of the involved network domains to decide which end-to-end connections are necessary, and whether they can be established so as to satisfy the QoS and time requirements (further negotiation between the requestor and the virtualization system may be necessary if the request cannot be satisfied). This process is essentially a scheduling decision resulting in a set of point-to-point virtual paths managed as one entity, i.e., they need to be timely established and managed together as a set. The scheduling problem is much more complex than when attempting to establish individual end-to-end paths independently since the number of domains is larger than 2 and the resource availability is quite different given that all virtual paths may have to co-exist during the same or overlapping periods. Depending on the number of involved sites, resource requirements and availability, and policies, end-site domain controllers may establish a fully or partially connected mesh topology. The actual implementation of the necessary virtual paths depends on the capabilities, configuration, and preferences of these underlying controllers (TeraPaths, ESCPS, OSCARS instances that are in control of the network domains involved).

3.1 Auxiliary Services

To support a distributed (or centralized) deployment, VNOD system instances (or instance) need access to information about other instances, the location and capabilities of end-site and WAN domain controllers, network topologies, and users. It is also useful to have access to - and also provide - monitoring information concerning virtual paths and dynamic circuits. Such information is provided by a set of

auxiliary services which include the following functionalities:

- Service discovery, a directory of VNOD instances and domain controllers
- Topology database, that can provide descriptions of physical network topologies, especially circuit endpoints associated with end-sites
- Monitoring services for inquiring and uploading information about paths, ViNets, circuits, performance data, etc.
- Membership services for user/application communities such as Virtual Organizations (VOs)

Developing such a support infrastructure is not within the scope of VNOD especially because there are systems in use and/or under development that can perform such duties, notably perfSONAR [26] and E-Center [?]. We believe that perfSONAR services such as the Lookup Service (LS), Topology Service (TS), Measurement Point Service (MP) and Measurement Archive Service (MA), with suitable extensions if necessary, can cover the needs of VNOD. The community membership service is essentially a distributed database of trusted entities that can reserve resources and are allowed to use ViNets. For this service, we also leverage existing services such as the Virtual Organization Management System (VOMS) [?], which was used for similar purposes in TeraPaths.

4. RESOURCE CO-SCHEDULING

The StorNet project [28] was the first to investigate how co-scheduling of storage and network bandwidth could take place in a real production environment where the new network resource reservation capabilities are offered. The goal was to increase the reliability and predictability of data transfers and maximize the benefit of network reservations while minimizing resource wastage. Without network bandwidth reservation the performance of a data transfer is subjected to random congestion conditions in the network and is essentially unpredictable: the same amount of data that could take a few minutes or hours to transfer at certain times could take days at other times. Without co-scheduling, however, network bandwidth could be reserved manually but may never be utilized fully because the storage systems could be heavily loaded and/or incapable of maintaining the transfer rates that the reserved network supports. We therefore developed an algorithm for co-scheduling storage and network resources for “flexible” data transfer requests between pairs of end-sites. A flexible request is defined as a triple {earliest possible start time, deadline, data volume} in contrast to a “fixed” request with specific start time, end time, bandwidth to reserve. The algorithm coordinates the reservations of BeStMan, TeraPaths, and OSCARS to provide end-to-end paths for data transfers performed by BeStMan. Each individual system advertises their resource availability with a Bandwidth Availability Graph (BAG), which is a step function representing the available bandwidth vs. time, in an “anonymous” fashion, i.e., without making public any individual reservation details. The original flexible request is then fitted or modified to fit into the overall BAG resulting from the intersection of individual BAGs. The number

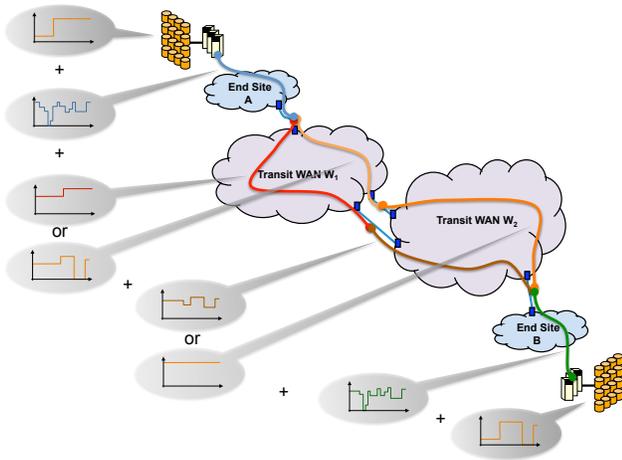


Figure 3: Co-scheduling with BAGs along a path.

of possible solutions depends on the flexibility of the original request and if the exact request cannot be satisfied, a solution is picked based on preference (e.g., shortest duration or earliest start time). Combining this distributed reservation negotiation algorithm with the earlier developed Bandwidth Allocation and Circuit Assignment (BACA) algorithm [?] pursues a balance between minimizing the number of circuits required to service multiple end-site reservations between the same pair of end-sites (by consolidating circuit reservations) and maximizing the request acceptance rate of the system.

In VNOD the number of end-sites, and thus individual domain systems involved in establishing a ViNet, can be larger than 2. This raises the question of what interconnecting topology should be used, based not only on application requirements but also on resource availability, cost, performance and other constraints. Also, the framework has to be able to handle the co-scheduling of multiple requests from multiple sites at a time. As a first step, we extended the early StorNet work with the Resource Reservation Algorithm (RRA) that handles multiple requests between a pair of sites [?]. We further developed the Resource Reservation and Path Construction (RRPC) algorithm that can handle multiple requests and also select among candidate WAN paths [?]. The path selection is achievable through our collaboration with the ARCHSTONE project. The ARCHSTONE extension to OSCARS can provide, given the endpoints of a WAN path and a time frame, a number of alternative paths and the bandwidth availability for each one during that desired time period. Figure 3 demonstrates the co-scheduling workflow between two end-sites. This workflow can be considered in conjunction with end-sites storage systems or just for the network. Each participating system provides its bandwidth availability for the same given time period (for storage systems, the bandwidth availability expresses the achievable transfer rate which is analogous to the bandwidth availability of the network). The intersection of all BAGs along each alternative path represents the availability of bandwidth for that path, including end-site storage systems if so desired. Next, the algorithm attempts to fit the given set of requests, modifying them appropri-

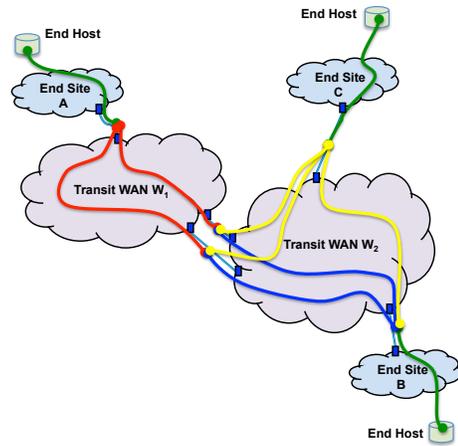


Figure 4: Co-scheduling problem with multiple end-sites and multiple paths.

ately within their flexibility limits, with the goal to satisfy as many as possible. The algorithm will terminate with a path solution that is found to satisfy the largest subset of the set of requests.

As in the case of StorNet, however, this algorithm is applicable to simple point-to-point topologies where data needs to be transferred from one end site to another. There are several challenging problems that still need to be solved when co-scheduling is to be performed for multiple pairs of end-sites. For multiple end-point scenarios, the existing point-to-point solutions will most probably be inefficient. As an example, figure 4 shows a simple topology that includes three end-sites (A, B, and C). Let's assume that a user needs to transmit data between the (A, B), (B, C) and (A, C) pairs. By naively using the point-to-point approach, the RRPC algorithm would have to be executed three times (once for each pair) and the availability re-assessed after each run. Because some network segments between these end-sites may be shared, the pairs scheduled first may get hold of all the available bandwidth of a segment leaving nothing for the remaining pairs. A scheduling algorithm that can jointly schedule all three pairs is clearly a better choice and will perform better than the naive sequential approach. The difficulty, however, lies with the bandwidth availability information that is made available to this algorithm. For RRPC, the availability is provided on a per-path basis since only transfers between a single pair of sites are considered. If there are common path segments when multiple pairs are considered, intersecting the BAGs of all paths would guarantee a correct solution, but such an approach would be draconian in the sense that there could be several other feasible solutions that would be ignored. We are currently implementing a new algorithm that can jointly schedule multiple end-site pairs. For this algorithm ARCHSTONE will provide the bandwidth availability of "service topologies" with a per-hop or per-segment availability. It will be, therefore, possible to take into account the effect that allocating bandwidth for a request will have to the scheduling of the remaining requests of the submitted set. This could lead to modifying requests, within their flexibility limits, in different

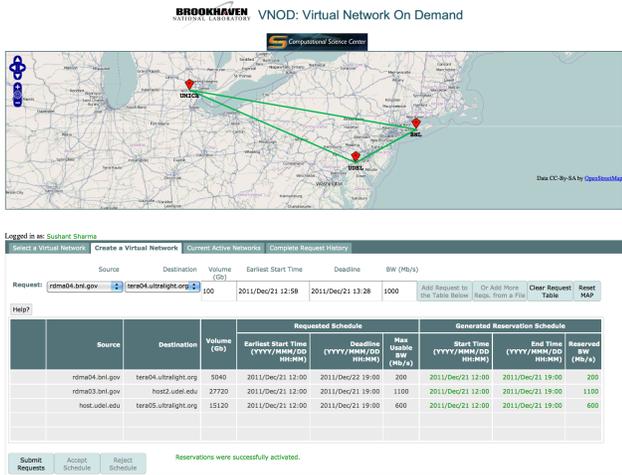


Figure 5: The VNOP front end prototype.

ways that require less overall bandwidth. Also, alternative paths could be chosen when a common segment does not have enough bandwidth to accommodate requests between multiple end-site pairs.

5. EARLY PROTOTYPE

The ultimate goal of the VNOP project is to develop a virtual networking framework and the necessary tools to facilitate the establishment of virtual networks. As a first step, we have developed an early prototype that takes advantage of the TeraPaths testbed. This testbed currently includes dedicated subnets with dynamic circuit interconnection capability at Brookhaven National Laboratory, the University of Michigan, and the University of Delaware. The aim of this prototype is to enable end users to submit requests for and manage virtual networks. Figure 5 shows an example of such a ViNet request comprising three flexible reservation requests. The front-end collects and forwards the user requests to an intelligence layer. This layer contacts the necessary domain controllers to collect the resource availability within each involved networking domain. The submitted requests along with the resource availability are then forwarded to a co-scheduling engine. The co-scheduling engine runs a co-scheduling algorithm and constructs a feasible schedule for reserving resources. Next, this reservation schedule is passed to each domain controller which in turn creates and activates the necessary reservations according to the schedule. To facilitate ViNet utilization, our prototype allows individual users to save templates of virtual topologies for future use. A user can select a saved template and modify some information, if necessary, to re-establish a ViNet with minimal effort. The ViNet request will be, of course, scheduled from scratch given the known resource availability at that time. The prototype also provides functionality to cancel any pending or existing ViNets.

6. RELATED WORK

Virtualization provides a “level of indirection” [16] between applications and shared infrastructure. While various virtual machine technologies have appeared (VMware, User-Mode Linux and Xen), virtual networking is also receiving much attention. Projects involving network virtualization

includes VNET [17], Virtual Networking on Overlay Infrastructures (Violin) [18], Virtual Service Grid (VSG) [20], and Resilient Overlay Network (RON) [22], among others [21, 23].

VNET and Violin use virtual IP networks for virtual machine networking. Violin, for instance, involves middleware that allows a virtual distributed environment based on a shared infrastructure such as the Grid or PlanetLab. Violin daemons create a user-level overlay network that serves as a virtual network. Virtual machines inside the virtual network utilize standard IP services. Below the virtual network, daemons emulate such services with application level methods such as UDP tunneling [16]. In VNET, network virtualization is not fully implemented at the user level. Host kernel-level devices are used to tunnel network traffic. VNET has sophisticated topology adaptation capability [19]. The virtual network topology “adapts” to the virtual machine networking patterns that applications follow [16].

Virtual Service Grid (VSG) [20] is a system architecture, middleware, and replication management strategy based on the virtual service concept. It provides location, replication, and fault transparency to users accessing high-end servers. VSG is deployed on a wide area Internet testbed (built using the Legion system) for performance evaluation.

RON [22] is another well known network overlay system. It is a simulated computing network built on top of the existing Internet (the substrate of an overlay network). RON consists of many end hosts in the application layer functioning as “network routers and/or switches”. These end hosts are interconnected by logical network links functioning as network physical or data links. RON moves routing control from the routers towards end hosts that are plugged into the networks to be overlaid. The overlaying end hosts simulate router and switch functionality, and act as proxy nodes through which data traffic is forwarded. QoS performance, in term of bandwidth and jitter, within RON is limited by both host performance and the underlying best effort Internet. In contrast to RON, VNOP directly interacts with the network control plane. VNOP implements a set of network device drivers to allow it to interact with the network infrastructure to do provisioning. In fact, VNOP will provide a reliable substrate with QoS guarantees for network overlay systems, such as RON, for their time-critical data traffic.

All the above overlay networks [17, 18, 20, 22] construct logical networks over the best effort transport networks. Since resource availability over best effort networks is unpredictable, QoS guarantees in existing virtual network infrastructures is hard to achieve. Our goal in this paper is to propose an architecture for virtual networks that can provide hard QoS guarantees.

There are several large WANs (including ESnet and Internet2) that are moving towards service-oriented networks. Such networks can dynamically provision virtual circuits with guaranteed QoS. This capability provides the foundation for the WAN virtualized network architecture that we propose, i.e., VNOP. VNOP interacts directly with the domain controllers of the WANs to provision end-to-end paths crossing multiple domains. The routing/switching decision

is distributed into each involved ISP, which has its own intra- and inter-domain protocols. Such distribution of capabilities makes the VNOD virtualization system, distributed, scalable, and reliable. Any failures that affect some WAN can transparently be recovered by the associated domain controller before the end applications notice the performance degradation or service outage.

There are virtual networks built on MPLS transport networks that allow traffic engineering and scalable QoS management [21]. Resource management receives special emphasis in such research efforts as the VSNM (Virtual Network based Service of Network Management) architecture [21] and VNARMS (Virtual Network based Autonomic Network Resource Control and Management System) [23]. MPLS traffic engineering (TE) software [?] by CISCO enables traffic engineering only on MPLS enabled networks. It uses constraint-based-routing and uses RSVP to establish MPLS tunnels across the backbone. In contrast to MPLS-TE, our goal is to build an architecture that can work with multiple WANs employing heterogenous QoS enabling technologies. OpenFlow [18] is another recent effort that proposes to develop a common interface to update routing tables within routers and switches from different vendors. Such an interface will be helpful for wider adoption of VNOD among sites that deploy network devices (routers, switches, etc.) from different vendors. Flowvisor [23] is another project that uses OpenFlow as the underlying mechanism to reserve the desired resources (e.g., bandwidth) within networks. However, the bandwidth reservations in their deployed system had to be made manually via request submissions to a network administrator. This is one of the drawbacks that we aim to remove via our proposed VNOD architecture. Resource reservation within VNOD is done via negotiation of available resources within different participating domains and without any manual intervention.

Two architectures for virtual networks are presented in [?] and [?]. However, the concept of virtual networks in [?] and [?] is different from the VNOD concept in this paper. Virtual networks in [?] and [?] cannot be used for large scale data transfers between end sites as our VNOD architecture aims to do. End nodes of the virtual networks in [?] and [?] lie within the physical network (e.g., within the WAN) and may not be the part of end sites. End users cannot control the physical end nodes that constitute the virtual network. The end goal of the virtual networks in [?] and [?] is not the large scale data transfer between multiple end sites.

7. SUMMARY AND FUTURE WORK

We presented the architecture, foundations, internal logic, and prototype implementation of a virtual networking infrastructure with hard QoS guarantees. The VNOD infrastructure radically differs from all known network virtualization systems in that it supports hard QoS guarantees enforced by timely, reservation-driven network device configuration modifications. VNOD leverages recent network resource reservation and co-scheduling technology developed or under development under the TeraPaths, ESCPS, StorNet, OSCARS and ARCHSTONE projects. This technology currently targets leading R&E networks, such as ESnet and Internet2, and end-sites interconnected by them and is not available for the general Internet. VNOD enables the selec-

tion and establishment of virtual topologies based on end-to-end virtual paths and is capable to co-schedule storage and network resources between multiple end-sites and WAN domains. Key aspect of co-scheduling is the concept of intersecting bandwidth availability graphs to express the overall availability along a path and the notion of flexible requests which allow the system to “negotiate” how requests can be implemented with reservations. The VNOD front-end allows users to handle virtual network creation and management and facilitates repetitive creation tasks through the storage of virtual network templates.

Our future work plans focus on further research and development of co-scheduling algorithms and on the prototype framework and tools. As mentioned in section 4, we are currently working on an algorithm that jointly schedules multiple requests from multiple end-sites while also selecting among alternative interconnecting paths through the WAN. This algorithm is meant to address a typically anticipated situation where a group of users wants to perform multiple data transfers between multiple end sites as part of a workflow. Such a workflow could include, e.g., transfer of data sets from their original locations to locations with compute resources for processing, then transfer of processed data to other locations for visualization. However, more research is needed since even in the simplest case the scheduling problems are NP-hard [22] and we plan to develop and evaluate different heuristics and solution search strategies. Additionally, there are many other scenarios possible (e.g., multicast, broadcast) for which co-scheduling algorithms do not currently exist. Furthermore, one cannot assume that the resource availability from all involved networking domains and systems can always be obtained easily. Some systems may be incapable of providing such information, or may restrict making such information public due to security reasons. We therefore need to be able to accommodate scenarios where not all availability information along a path is known by making the scheduling algorithm intelligent enough to detect such scenarios and do its best. One of our main goal in developing the VNOD architecture is to provide the capability to easily plug-in different scheduling algorithms depending on the particular scenario.

8. ACKNOWLEDGMENT

This work is supported by the Director, Office of Science, Office of Advanced Scientific Computing Research, of the U.S. Department of Energy under Contract No. DE-AC02-98CH10886

9. REFERENCES

- [1] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, and G. Swallow “RSVP-TE: Extensions to RSVP for LSP tunnels,” *RFC 3209*, 2001.
- [2] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, “An architecture for differentiated service,” *RFC 2475*, 1998.
- [3] R. Braden, D. Clark, and S. Shenker, “Integrated services in the Internet architecture: An overview,” *RFC 1633*, 1994.
- [4] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, “Resource reservation protocol (RSVP),” *RFC 2205*, 1997.

- [5] C. Chekuri and S. Khanna, "A PTAS for the multiple knapsack problem," In *Proc. ACM-SIAM SODA*, pp. 213–222, Philadelphia, PA, USA, 2000.
- [6] A.R. Curtis, J.C. Mogul, J. Tourrilhes, P. Yalagandula, P. Sharma, and S. Banerjee, "DevoFlow: scaling flow management for high-performance networks," In *Proc. ACM SIGCOMM*, Toronto, Canada, August 15–19, 2011.
- [7] "Energy sciences network," <http://www.es.net/>
- [8] "AutoBAHN: AUTOMated Bandwidth Allocation across Heterogeneous Networks" <http://www.geant2.net/server/show/ConWebDoc.2544>
- [9] "EsNet Topology," <http://www.es.net/network/>
- [10] "The GÉANT2 network," <http://www.geant2.net/>
- [11] "GridFTP Documentation," <http://globus.org/tool-kit/docs/3.2/g-ridftp/>
- [12] J. Gu, D. Katramatos, X. Liu, V. Natarajan, A. Shoshani, A. Sim, D. Yu, S. Bradley, and S. McKee, "StorNet: Co-scheduling of end-to-end bandwidth reservation on storage and network systems for high-performance data transfers," in *Proc. IEEE INFOCOM, Workshop on High-Speed Networks*, Shanghai, China, April 10–15, 2011.
- [13] R.A. Guerin and A. Orda, "Networks with advance reservations: The routing perspective," in *Proc. IEEE INFOCOM*, pp. 118–127, Tel-Aviv, Israel, March 26–30, 2000
- [14] "Internet 2," <http://www.internet2.edu/>
- [15] "Internet 2–Topology," <http://www.internet2.edu/observatory/archive/datacollections.html#topology>
- [16] The TeraPaths Project: <http://www.terapaths.org>
- [17] D. Katramatos, D. Yu, K. Shroff, S. McKee, and T. Robertazzi, "TeraPaths: end-to-end network resource scheduling in high-impact network domains," *International Journal On Advances in Internet Technology*, vol 3, no. 1–2, pp. 104–117, 2010.
- [18] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol 38, no. 2, pp. 69–74, March 2008.
- [19] "OSCARS: On-Demand Secure Circuits and Advance Reservation System," <http://code.google.com/p/oscars-ids/>
- [20] K. Rajah, S. Ranka, and Y. Xia, "Advance reservation and scheduling for bulk transfers in research networks," to appear *IEEE Transactions on Parallel and Distributed Systems*.
- [21] S. Sharma, D. Gillies, and W. Feng, "On the goodput of TCP NewReno in mobile networks," In *Proc. International Conference on Computer Communications and Networks (ICCCN)*, Zurich, Switzerland, August 2–5, 2010.
- [22] S. Sharma, D. Katramatos, and D. Yu, "End-to-end network QoS via scheduling of flexible resource reservation requests," To appear *ACM/IEEE International Conference for High Performance Computing Networking Storage and Analysis (Supercomputing)*, Seattle, WA, November, 12–18, 2011.
- [23] R. Sherwood, G. Gibb, K. Yap, G. Appenzeller, M. Casado, N. McKeown, and G. Parulkar, "Can the production network be the testbed?," In *Proc. USENIX OSDI*, Vancouver, Canada, October 4–6, 2010.
- [24] O. Younis and S. Fahmy, "Constraint-based routing in the internet: Basic principles and recent research," *IEEE Communications Surveys & Tutorials*, vol. 5, no. 1, pp. 2–13, 2003.
- [25] The Lambda Station project: <http://www.lambdastation.org>
- [26] PerfSONAR: PERFORMANCE Service Oriented Network monitoring ARchitecture: <http://www.perfsonar.net>
- [27] The End Site Control Plane Service (ESCPS) project: <https://plone3.fnal.gov/P0/ESCPS>
- [28] StorNet: Co-scheduling Network and Storage with TeraPaths and SRM: <https://sdm.lbl.gov/twiki/bin/view/Projects/StorNet/WebHome>
- [29] ARCHSTONE: Advanced Resource Computation for Hybrid Service and TOPOLOGY NETWORKS <http://archstone.east.isi.edu>