TeraPaths: End-to-End Network Path QoS Configuration Using Cross-Domain Reservation Negotiation

Bruce Gibbard, Dimitrios Katramatos, Dantong Yu
Brookhaven National Laboratory
{gibbard,dkat,dtyu}@bnl.gov

Shawn McKee
University of Michigan
smckee@umich.edu

Abstract

TeraPaths is a DOE MICS/SciDAC-funded project conceived to address the needs of the high energy and nuclear physics scientific community for effectively protecting data flows of various levels of priority through modern high-speed networks. TeraPaths is rapidly evolving from a last-mile, LAN QoS provider to a distributed end-to-end network path QoS negotiator through multiple administrative domains. Developed as a web service-based software system, TeraPaths automates the establishment of network paths with QoS guarantees between end sites by configuring their corresponding LANs and requesting MPLS paths through WANs on behalf of end users. The primary mechanism for the creation of such paths is the negotiation and placement of advance reservations across all involved domains. This paper describes the status of the project, our experiences so far, as well as the directions of our continued work.

1. Introduction

The TeraPaths [7],[8] project was initiated to explore QoS configuration automation and administration in LANs under the umbrella of the ATLAS experiment [3], specifically, within the USATLAS project [4]. Modern high energy and nuclear physics research places extreme demands even on today's high-speed networks due to the sheer volume of data (petabytes) that needs to be transferred between involved end sites. While not all network flows are of equal priority and/or importance, the default behavior of the network is to treat them as such. Thus, the competition among flows for network bandwidth can cause severe slow downs for all flows, independent of importance, and furthermore cause some applications to fail.

TeraPaths offers the capability to distinguish between various data flows and enable the network to treat each flow differently in a pre-determined way through the use of the differentiated networking services architecture [5]. Within the domain of an end-site's LAN, TeraPaths selectively and/or collectively configures network equipment to dedicate fractions of the site's available bandwidth to qualified data flows, to assure adequate throughput and protect these flows from the disruptive impact they may have upon each other (see Figure 1). At the Brookhaven National Laboratory (BNL), routine Relativistic Heavy Ion Collider (RHIC) [1] production data transfers and Large Hadron Collider (LHC) [2] Monte Carlo challenges between the laboratory and several remote collaborators can disruptively interfere with each other as the aggregate peak network requirement is well beyond the capacity of the BNL network. To ensure that RHIC production data transfers are not affected,

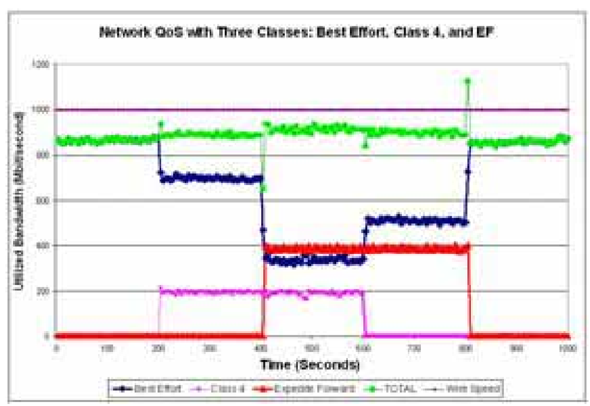

Figure 1: DiffServ-based QoS; data flows assigned to Class 4 and Expedite Forward classes of service remain at predetermined bandwidth levels at the expense of Best Effort traffic.

LHC data transfers need to be constrained to opportunistically utilize available bandwidth.

Offering similar assurances and protection to data flows all the way from their source to their destination is a much harder problem outside the domain of a site's LAN. In this case, source and destination sites need to coordinate their networks to honor a data flow's requirements and arrangements need to be made with one or more WAN providers along at least one path connecting the source and the destination sites to treat that data flow in a compatible manner. Such a capability is essential in the ATLAS multi-tiered distributed data environment (see Figure 2) for the establishment of QoS protected paths between sites belonging to various tiers.

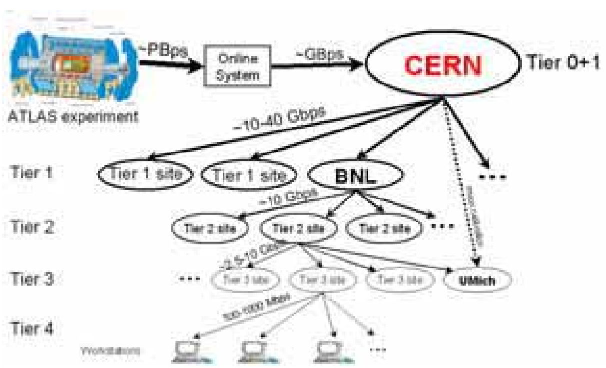

Figure 2: ATLAS data distribution.

The LAN managing capabilities as well as the LAN/WAN/LAN coordination capabilities offered by TeraPaths can be integrated into the infrastructure of grid computing systems to enable the scheduling of network resources along with CPU and storage resources and help increase their overall performance and efficiency.

2. End-to-end: crossing multiple domains

In the general case, setting up the QoS for a path between a source and a destination site requires the coordinated configuration of the network devices of all domains along that path. We place a major distinction between the end-site LANs and the one or more WANs connecting these LANs. While the end-site LANs are under the direct control of a configuring system (in our case, we assume that all end-sites are under the control of TeraPaths), WANs have their own admission control system for any traffic other than best effort. Thus, an end-to-end setup entails ensuring the compatibility of the end-site LANs and interfacing with one or more WAN admission systems. Typical method for coordinating across domains is through the use of reservations. We consider three models for carrying out an end-to-end setup: the star model, the daisy chain model, and a star/daisy chain hybrid model (see Figure 3). Although the three models differ in the way the reservation requests propagate, ultimately all sites need to agree for the setup to go forward.

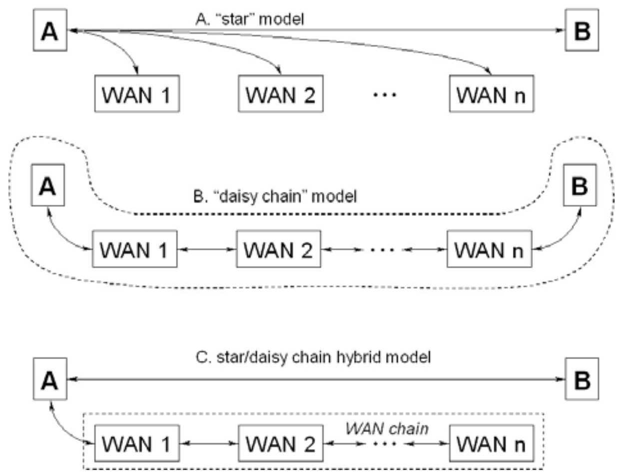

Figure 3: The three end-to-end setup models.

Star Model (figure 2A): In this model, the initiating end-site is the coordinator for the path setup. It communicates with the other end-site and with each involved WAN provider site individually and negotiates a path setup both in terms of start/end time and type of service. If all involved domains can offer compatible reservations, the coordinator activates the path setup.

Daisy Chain Model (figure 2B): Here, the initiating site sends a reservation request to the first WAN provider site in a chain of WAN domains ending at the destination site. Each site in the chain receives, acts upon the request, sends back a reply, and forwards the request to the next site (alternatively, the forwarding of the request by a site can be contingent to the success of the reservation at that site).

Star/Daisy Chain Hybrid Model (figure 2C): The third model is a combination of the previous two. In this model, the initiating site coordinates with the destination site and with *one* WAN site. This WAN site uses the daisy chain model to setup access up to, but excluding, the destination site (i.e. through the WAN chain).

While the end result is the same in all three models, there are other implications that need to be considered. The star model is, in essence, centralized control. Although the number of sites that need to coordinate is not anticipated to be so large as to cause scalability problems, following this model means that any end-site needs to know the topology of the network between itself and any destination site so that it can determine which WAN providers to contact. On the positive side, requests can be sent out in parallel. Also, if the involved sites provide some detail about their reservations, it can be easier for the coordinating site to determine if a request will go through all the way, and if not, what modifications will be necessary to make it succeed. Finally, the end-site software can be made to support any number of APIs needed for talking to various WAN providers.

The daisy chain model is fully distributed and only requires that each site is aware of its neighbors. However, this model also requires an interface specification supported by all involved sites so that all necessary information can be forwarded to each site in the path and ultimately to the destination site.

The hybrid model is, at this point in time, the most feasible one. In this model, the initiating site is again the coordinator, but only communicates with the destination site and one WAN provider. In this way, the end-site LANs are directly configured and there is no need for the coordinating site to know the wide area topology. This task is delegated to the first WAN provider of a WAN chain, who is best suited to make arrangements for a data flow with its peering networks. This is the default model currently supported by TeraPaths. However, any of the three models can be supported with the addition of the necessary code modules. Furthermore, TeraPaths can also select which WAN chain to contact, depending not only on the destination of a data flow, but also on other selection criteria, as will be described later in this paper.

3. The TeraPaths architecture

TeraPaths is designed around the DiffServ architecture [5]. Of the QoS supporting architectures offered by modern networking hardware, DiffServ is the most scalable. With this architecture, traffic needs to be conditioned (policed/shaped) only at the network boundary. Up to 64 traffic categories - classes - are supported, using six bits of the Type of Service (ToS) byte, known as DSCP bits. Treatment of data is determined on a per-packet basis. In contrast, the IntServ architecture [13] (RSVP protocol [14]) determines treatment on a per-flow basis and thus requires the maintenance of flow information in all involved network devices.

The TeraPaths software configures and manages LAN QoS paths from end hosts to border routers. Each such path can dedicate a percentage of the available site bandwidth to its assigned data traffic. Distinction between data packets is done by means of their DSCP markings. TeraPaths controls which data traffic goes into each of the configured classes at the data flow level (a data flow is defined by a source and a destination IP address and port number pairs). Access to QoS paths is further controlled by advance reservations.

Going beyond the limits of a site's LAN, as is necessary for establishing an end-to-end path setup, requires coordination with the TeraPaths controller at the target site and arrangements for honoring a data flow's requirements through a chain of one or more WAN domains. TeraPaths invokes the appropriate WAN provider's interface to make such arrangements. Typically, the WAN provider will create a suitable MPLS [6] tunnel through the network, which, if necessary, will extend through other WANs down the chain according to the cooperation agreement between peering WANs.

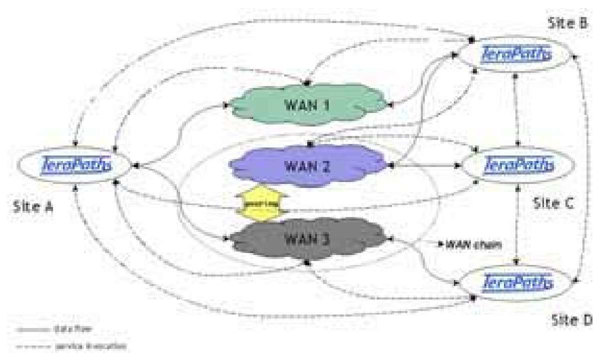

Figure 4: Conceptual view of the network.

Figure 4 shows the conceptual view of the network that TeraPaths utilizes for data transfers. End sites (A, B, C, and D) run the TeraPaths software for configuring and controlling their LANs. Communication between each pair of sites is done through one or more alternative WAN routes. WANs may have peering points, however, when following the daisy chain or the hybrid model, inter-WAN route setup is not in the list of TeraPaths responsibilities. End-to-end route setup for a data flow entails first setting up the LAN of the source site, then arranging for an MPLS tunnel through a WAN, and finally setting up the LAN of the destination site, again through remote invocation of that site's TeraPaths software.

Figure 5 offers a more detailed view of the TeraPaths architecture, based on the hybrid model and realized with the help of web services. Without loss of generality, it is assumed that site A initiates a data transfer to site B through a WAN chain. Each site runs its own instance of the TeraPaths system. The system is comprised of a set of core services that cooperate with the necessary databases to provide user Authentication, Authorization, and Accounting (AAA), advance reservation scheduling, negotiate remote requests, and distribute network configuration commands to management nodes. Management nodes are hosts that supervise network devices (routers, switches) and are responsible for their configuration and monitoring. The configuration is performed through the invocation of a subset of hardware controlling services available at each management node. This subset of services offers a layer of abstraction between configuring requests and hardware configuration and invokes, in turn, suitable hardware drivers to "speak" the actual hardware language (e.g. Cisco IOS commands) and setup the necessary network equipment accordingly.

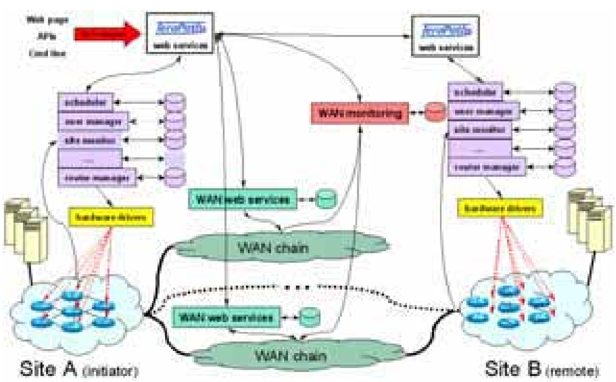

Figure 5: The TeraPaths architecture.

The initiating site is responsible for negotiating and arranging an end-to-end LAN QoS/MPLS path from the source host at site A through a WAN chain to the destination host at site B. If there are more than one WAN chains that can be used for reaching B, then one of them is selected based on preference criteria (e.g. anticipated availability based on monitoring data).

If the requested bandwidth can be reserved at the specified time locally (at site A), the system proceeds to request an MPLS tunnel through the WAN chain and a compatible QoS path through the LAN of site B. This is done by remotely invoking the corresponding interface of the WAN provider and the TeraPaths instance of site B and request appropriate reservations. TeraPaths supports temporary reservations that expire after a short period of time. WAN providers may or may not support such a kind of reservations. Only when all three reservations can be obtained does the system proceed to actually put the reservations in place (commit) so that the end-to-end QoS path configuration can be guaranteed at the requested time. If one or more reservations cannot be obtained, the system cancels whatever successful reservations were placed before the failed one and attempts to find a set of reservations as close as possible to the initial request and present the user with a "counter offer". The user can accept the counter offer or try a different request. An alternative approach, possible only if the involved sites expose details about their already in-place reservations, is to first send out the reservation requests in parallel and if any one fails then obtain, again in parallel, the schedule of current reservations of each involved site and search for a slot that is as close as possible to the initial request to present as a counter offer. Yet another alternative, if the reservation schedules are not available, is to have the initiating end-site attempt variations of the original request until one goes through or a trial and/or time limit is reached.

TeraPaths can be invoked using a web interface, which graphically displays the available bandwidth classes and the existing reservations and facilitates the placement of new reservations. If the source site and destination site is given first, the web interface can aggregate and display the existing reservations of both end-sites so that reservation request is not guaranteed to succeed only at the WAN (the displayed free slots are free at both end-sites). Alternatively, the services can be invoked through Application Programming Interfaces (APIs), further enabling the use of Command Line Interfaces (CLIs) and direct TeraPaths invocation from within applications.

Properly configured, TeraPaths can achieve a partitioning of a site's available bandwidth in statically and/or dynamically allocated slots, to accommodate the needs of a large number of data flows. Each slot, according to its type, is assigned to a class of service from the set of services classes pre-configured within the LAN perimeter. All LAN hardware knows how to treat packets belonging to each such service class; however, the actual policing/shaping of flows is done at the first piece of equipment encountered when leaving the source host. The network configuration module of TeraPaths can automatically reconfigure the entire network or part of the network that is under its control and modify the role and bandwidth assignments of service classes. This is, however, an infrequent administrative task, as the combination of statically and dynamically allocated bandwidth slots can satisfy a wide array of flow requirements.

Figure 6 shows an example qualitative partitioning scheme of a site's available bandwidth. Dedicated static classes are policed on an aggregate bandwidth basis. That is, while a single flow can utilize all of the pre-determined bandwidth, any additional flow will cause the bandwidth to be equally shared among all flows assigned to the same class. Shared static classes are policed on an aggregate and a per-flow bandwidth basis. In this case, the bandwidth allocated for a class is further divided to a number of sub-slots defined by the specified per-flow bandwidth fraction. While the number of flows is less or equal to the number of sub-slots, each flow receives exactly one bandwidth fraction. If the number of flows exceeds the number of available sub-slots the total allocated bandwidth will still be observed by reducing the bandwidth fraction accordingly. Finally, shared dynamic classes are classes selected for their widely recognizable DSCP markings, e.g. the Expedite Forward (EF) class, and thus bound to be honored even by older generation equipment.

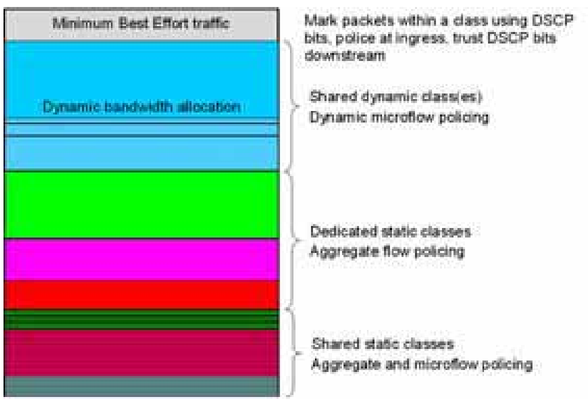

Figure 6: TeraPaths bandwidth partitioning scheme.

These classes are assigned a portion of a site's bandwidth which can be further distributed to a number of data flows dynamically, i.e. the per-flow bandwidth is not pre-determined but allocated to flows according to requests as site policy permits. Summarizing, site bandwidth partitioning under TeraPaths provides the mechanism to satisfy a variety of bandwidth allocation policies. Thus, frequent, high-priority flows can have their own, dedicated, class; groups of hosts can share a dedicated class without affecting other traffic; flows that require small bandwidth amounts can be funneled into the same shared static class, thus reducing the number of in-use classes (recall that a total of only 64 classes with corresponding DSCP markings is possible); finally, shared dynamic classes can be utilized to cover the needs of flows that cannot be otherwise satisfied. It should be pointed out that bandwidth partitioning occurs only when privileged flows are present. In absence of such flows, the network resumes best effort behavior. Nevertheless, there should always be a minimal fraction of bandwidth allocated for class 0 (best effort) so that common traffic can always proceed through the network.

Network monitoring is not required but, nevertheless, valuable as a means of reliably determining if and how well QoS paths are working. Furthermore, although monitoring information is not directly necessary for carrying out QoS path setup requests, in the future, monitoring information will be utilized in an MPLS-capable version of TeraPaths (see Figure 7) that will also provide route selection options within a site's complex LAN, based on desired criteria (e.g. bottleneck avoidance). For the TeraPaths project, network monitoring is conducted by the SLAC IEPM project [10].

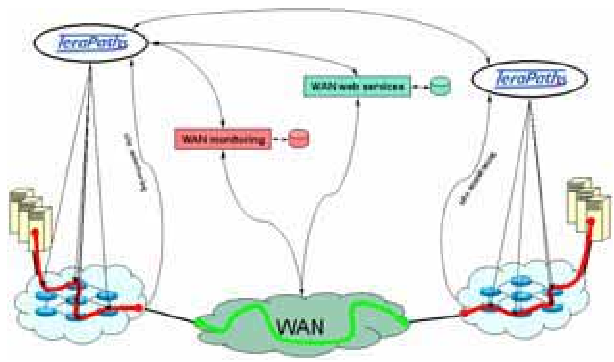

Figure 7: Route selection within end-site LANs.

4. Implementation and tests

The TeraPaths software is being developed using the proven underlying network communications technology of web services. The web services technology is secure, reliable, freely available, and permits the designer to specify the services offered by each administrative domain without specifying how they will be implemented. The BNL implementation of TeraPaths uses Java-based web services and a MySQL database to program Cisco routers. However, end-users can only see the fact that BNL provides a service that permits them to negotiate for a fraction of bandwidth, at a particular time and for a specific duration. For TeraPaths to be easily adopted by other end-users and deployed at their sites, we only use freely available software (e.g. Java and GlassFish or Jboss application servers) and standard distribution techniques (.war files). TeraPaths is designed to request guaranteed bandwidth WAN paths (e.g. MPLS tunnels) by invoking the web services of a WAN provider. To deal with any number of different WAN provider web service interface implementations, as well as possible compatibility issues, TeraPaths uses façade objects in the form of proxy server modules. Each such module, one per provider, wraps the services the corresponding provider makes available and exposes a common interface compatible to the core TeraPaths modules requirements.

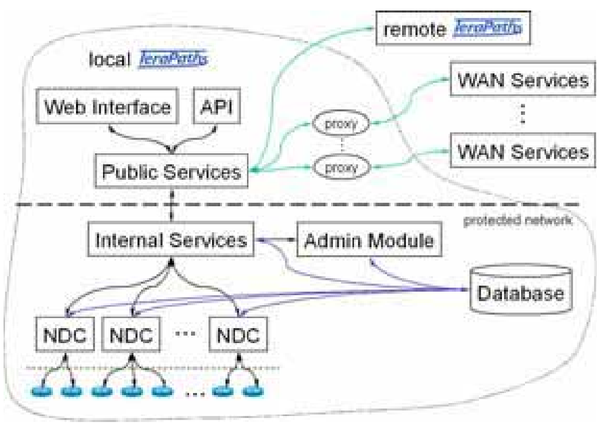

Figure 8: The web services architecture of TeraPaths.

The current implementation of the system is realized with three layers of cooperating web services, a web interface, a set of WAN proxy servers, and a relational database (see Figure 8). At the lowest-level layer are the Network Device Controller (NDC) modules. Each network device is assigned to one NDC, and there can be one or more NDCs at each site. Each NDC dynamically loads the appropriate device driver code from the database. This feature allows replacing/loading new modules at practically any time without having to restart the system. Most of the functionality of the system lies with the internal services layer. A subset of this functionality is restricted for use only by a system administrator and can be used for collectively configuring all network devices of a site. The public services layer implements the public interface of TeraPaths. The web interface, API, and remote TeraPaths invoke public services to carry out operations. The public services layer is also responsible for cooperating with available WAN web services through the proxy server modules. These modules, although part of a site's TeraPaths installation, are independent and, like device drivers, can also be dynamically deployed without affecting other service modules. For security reasons, the NDCs, the internal services, the administrative interface, and the database reside in a protected network segment and only the public service layer is allowed to cross into the protected area by invoking the internal services. The internal services can be further configured to accept invocations only from known hosts running public services.

For testing purposes, we put together a fully featured testbed (see Figure 9) using the same Cisco hardware as in the BNL production network. This testbed allows for all kinds of experiments without the risk of adversely affecting the production network because it is isolated. The testbed represents two end-sites connected with a dedicated link. The TeraPaths software was initially developed and tested on this testbed's private network.

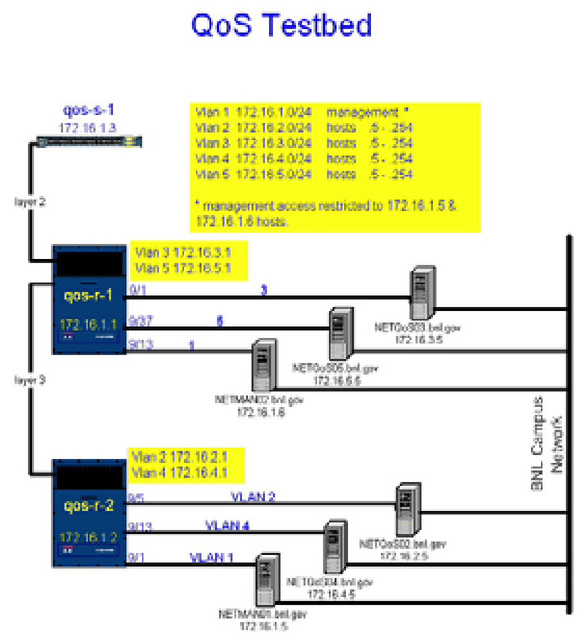

Figure 9: The TeraPaths testbed at BNL.

The upper half of Figure 10 displays a simple test bed experiment. Two iperf streams initially share the bandwidth of the gigabit link between the two test bed routers (approximately 60MB/s each stream). While a TeraPaths reservation for the Class 2 iperf stream is active, the bandwidth that stream occupies falls to the reserved 30MB/s, conceding the rest 90MB/s to the competing iperf stream. The lower half of Figure 10 depicts a more complex experiment (non-testbed, using BNL's actual production network) demonstrated at SuperComputing 2005. Here, two bbcp disk-to-disk data transfers, one at 200Mb/s and one at 400Mb/s, are protected from background competing iperf traffic through an ES-net MPLS tunnel. Only the iperf, best effort, traffic gets affected by the bbcp transfers which do not interfere with each other and maintain constant, pre-determined bandwidth throughout each transfer cycle. This experiment demonstrates the feasibility of the desired end-to-end operation of TeraPaths.

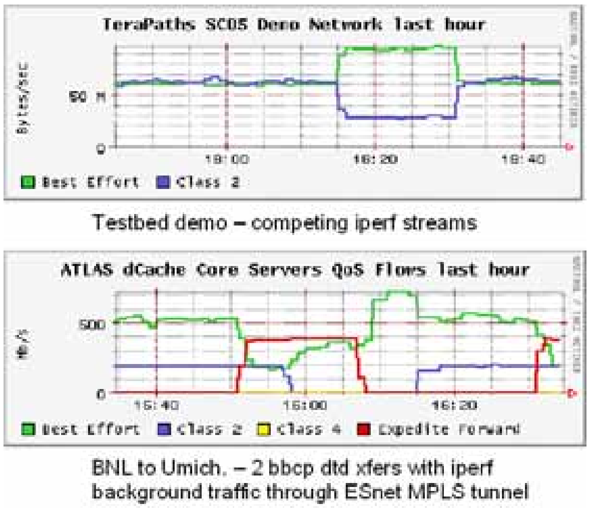

Figure 10: Initial experiments for the validation of the TeraPaths system.

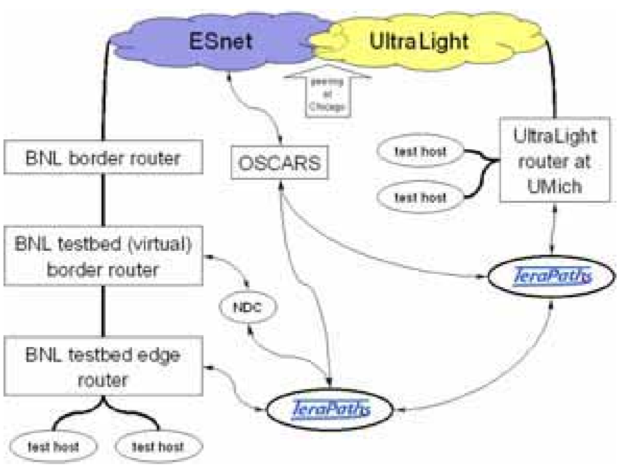

Figure 11: The new TeraPaths testbed – BNL to UMich.

Having verified the effectiveness of the LAN QoS/MPLS tunnel/LAN QoS setup in practice, we proceeded, in cooperation with the University of Michigan, to put together a testbed capable of end-to-end testing. This testbed involves two TeraPaths end-sites, one at BNL and one at the University of Michigan, connected through the ESnet and UltraLight networks (see Figure 11). WAN MPLS tunnel requests are directed to ESnet's OSCARS service [9]. The original BNL testbed was modified to now represent a single end-site (see Figure 12), while a second end-site was put together at the University of Michigan. The BNL border router was set to trust the traffic from the testbed while the original testbed's second router was set to play the role of the border router and thus accept the necessary configuration commands from its NDC. The configuration of the virtual border router is identical to that needed for the actual border router, however, possible errors encountered during testing cannot affect the actual border router, which is critical for regular site operations.

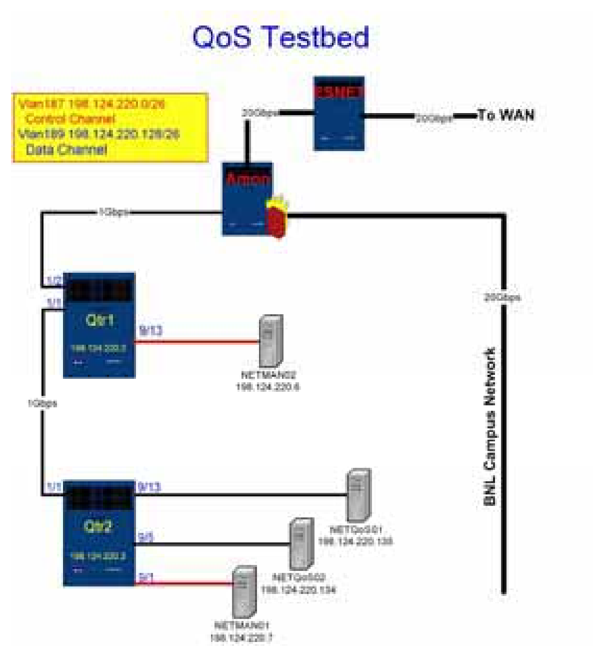

Figure 12: Details of the BNL-side TeraPaths testbed.

5. Related work

The Bandwidth Allocation and Reservation (BAR) project [11] within the EGEE JRA4 activity is an attempt to provide users with a simple software interface and an implementation to reserve a guaranteed bandwidth service. BAR supports two services; the Guaranteed Deadline File Transfer (GDFT) service with varying instantaneous but guaranteed average bandwidth over a period of time, and the Virtual Leased Line (VLL) service with guaranteed constant bandwidth. BAR relies on the concept of Network Service Access Points (NSAP), local and remote, for the relaying the necessary configuration directives to network devices. BAR follows a top-down approach to the reservation of bandwidth proposing definitions for the necessary interfaces to lower level services that configure the network.

The Network Resource Scheduling System [12] is a simple implementation of DiffServ on Cisco routers, for data transfers between two end-sites. NRS performs admission control of data flows into the Expedite Forward class of service. This class is assigned a maximum total bandwidth, fractions of which can be in turn dynamically allocated to flows subject to the availability of bandwidth at the time a request is placed.

Due to the common goal to provide end-to-end guaranteed bandwidth network paths, TeraPaths has common ground with both the BAR and NRS projects (e.g. use of similar reservation negotiation models, use of DiffServ). However, TeraPaths is essentially a bottom-up approach, combining the capabilities of BAR and NRS in a practical system, which is currently undergoing testing. TeraPaths was initially conceived as a last-mile solution, similarly to NRS, however, it is a highly sophisticated solution utilizing the full spectrum of capabilities of the DiffServ architecture and offers advance reservations. The system has already proven its effectiveness in configuring QoS within end-site LANs and admitting data flows in the configured classes of service. The extension of TeraPaths to negotiate reservations between end-sites under its control and also with WAN providers, results in a true, practical, end-to-end network path QoS negotiating and configuring system. Finally, through the façade design, the TeraPaths system makes certain that cooperation between TeraPaths and any WAN provider will take place regardless of what API the provider supports (as long as that API includes a core of basic operations). Although a standard universal interface is certainly desirable for guaranteed interoperability, it is not clear if and when such standardization will be realized. TeraPaths deals with this matter through the use of service modules specifically designed to interface with WAN services on a one-on-one basis similarly to network device drivers.

The Lambda Station [15] is a project that addresses the problem of forwarding data flows through local area networks to selected network connections. Under Lambda Station, designated data flows can be directed to high bandwidth WAN paths (e.g. optical paths) while common traffic follows default forwarding to conventional WAN paths. Lambda Station performs selective routing through a LAN and as such is complementary to TeraPaths, which selectively prioritizes data flows using the DiffServ architecture.

6. Conclusions and future work

TeraPaths demonstrates that the combination of LAN QoS techniques, based on the DiffServ architecture, combined with WAN MPLS tunnels is a feasible and reliable approach to providing end-to-end, dedicated bandwidth paths to data flows in a demanding distributed computational environment such as the environment needed for high energy and nuclear physics research. TeraPaths offers a flexible way to partition a site's available bandwidth into pre-determined bandwidth slots to protect various data flows from competing against each other.

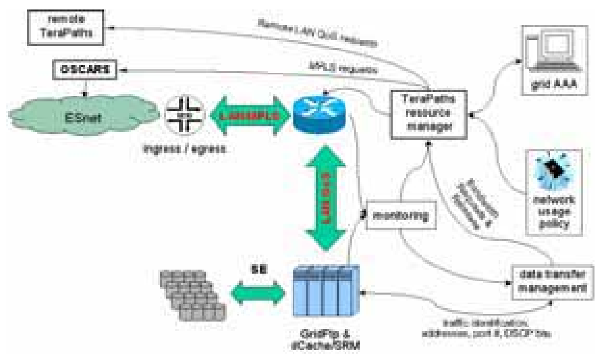

Figure 13: Integration of TeraPaths into BNL's production network.

A series of experiments at BNL, using both a testbed and the production network, indicate that LAN QoS does not impact the overall network utilization. No overall network performance deterioration was observed while QoS policies were active. Furthermore, across domain tests have demonstrated the effectiveness of combining LAN DiffServ- based QoS with WAN MPLS tunnels for configuring end-to-end protected network paths. Having extended the TeraPaths software to negotiate reservations across multiple domains and created a testbed suitable for end-to-end tests we are currently testing the fully automated end-to-end setup process. We are, further, studying the compatibility of the protected network paths with the regular network traffic and the impact on overall network performance across all networks involved in such paths. We are also in the process of widening the deployment of the TeraPaths system to higher and lower tier (at least down to tier 2) sites. The wider deployment will put the software under scrutiny and enable us to fix errors, obtain user feedback, and work towards robust, production quality software.

The modularity and extensibility of the TeraPaths software design, the administrative capabilities of the NDC layer, and the dedicated testbed equipment make TeraPaths an ideal platform for the pursuit of a wide range of research and development goals.

We plan to integrate TeraPaths into BNL's production network and grid environment (see Figure 13), support Virtual Organizations (VOs), and utilize monitoring information for status control. To widen the usability of TeraPaths across end-sites with widely varying QoS configurations we plan to add the capability of automatically synchronizing site configurations, within acceptable limits, and the capability of using different (but compatible bandwidth-wise) classes of services at each site by remarking incoming packets at the border router of the destination network.

We further plan to go beyond the DiffServ architecture and explore the use of MPLS technology (see Figure 7). More specifically, MPLS-supported policy/constrain-based routing may improve network utilization in complex LANs by taking advantage of available monitoring information to identify congested paths and directing flows to alternative routes. Combining DiffServ and MPLS technologies within a LAN in a hybrid scheme may further increase utilization/prevent congestion while reducing the volume of state information that each network device participating in a MPLS tunnel needs to maintain. Thus, we plan to study network configurations where flows are policed/shaped with DiffServ at the entry routers and then grouped and funneled through appropriately defined MPLS tunnels to the border routers. Another aspect of using MPLS is the pursuit of seamless MPLS tunnels for flow groups in a real end-to-end configuration, i.e. all the way from the source host to the destination host. Current MPLS implementations do not provide adequate security to be trusted by providers so that seamless tunnels across domains can be established, nor is it possible to support advance reservations in a pure MPLS environment. It is therefore worth exploring ways to augment the capabilities of MPLS devices, in manners similar to the current TeraPaths design, e.g. by associating with each device a controlling node that provides the additionally required functionality.

7. References

[1] The Relativistic Heavy Ion Collider (RHIC) at BNL: http://www.bnl.gov/RHIC/

[2] The Large Hadron Collider (LHC): http://lhc.web.cern.ch/lhc/

[3] The ATLAS experiment: http://atlas.web.cern.ch/Atlas/index.html

[4] The USATLAS project: http://www.usatlas.bnl.gov/

[5] An architecture for differentiated services, RFC 2475

[6] Multiprotocol label switching architecture, RFC 3031

[7] The TeraPaths project: http://www.usatlas.bnl.gov/twiki/bin/view/Projects/TeraPaths

[8] S. Bradley, F. Burstein, L. Cottrell, B. Gibbard, D. Katramatos, Y. Li, S. McKee, R. Popescu, D. Stampf, D. Yu. "TeraPaths: A QoS-Enabled Collaborative Data Sharing Infrastructure for Peta-scale Computing Research." Computing in High Energy and Nuclear Physics (CHEP 2006), T.I.F.R., Mumbai, India, Feb 13-17, 2006

[9] The ESnet OSCARS project: http://www.es.net/oscars/

[10] The IEPM project: http://www-iepm.slac.stanford.edu

[11] The BAR project: http://egee-jra4.web.cern.ch/

[12] The NRS project: http://www.cs.ucl.ac.uk/staff/S.Bhatti/grs/

[13] Integrated Services in the Internet Architecture: an Overview, RFC 1633

[14] Resource ReSerVation Protocol (RSVP), RFC 2205

[15] Lambda Station: http://www.lambdastation.org/