EMERGE: ESnet/MREN Regional Science Grid Experimental NGI Testbed – Final Report for Northwestern (iCAIR) Subcontract

Designing, Deploying and Testing Differentiated Services on an IP/ATM Regional GigaPoP Network Interoperating with ESnet for Applications in Combustion, Climate and High-Energy Physics

Proposal to DoE 99-10 Next Generation Internet University Network Technology Testbeds

Joe Mambretti
Tom DeFanti
Maxine Brown

(Award # DE-FC02-99ER25408)
For the period August 1, 1999 – July 31, 2001

DOE Patent Clearance Granted
3/26/03
Date

Daniel D Park
(630) 252-2308
E-mail daniel.park@ch.doe.gov
Office of Intellectual Property Law
DOE Chicago Operations Office

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, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness 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. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

Introduction

This document is the final report on the EMERGE Science Grid testbed research project from the perspective of the International Center for Advanced Internet Research (iCAIR) at Northwestern University, which was a subcontractor to this UIC project. This report is a compilation of information gathered from a variety of materials related to this project produced by multiple EMERGE participants, especially those at Electronic Visualization Lab (EVL) at the University of Ilinois at Chicago (UIC), Argonne National Lab and iCAIR. The EMERGE Science Grid project was managed by Tom DeFanti, PI from EVL at UIC.

The basic goal of the EMERGE Science Grid research project was to develop network services that would enable DoE laboratories to collaborate with DoE-funded university research centers utilizing high performance applications, for example, to solve complex problems and to develop tools for collaborative problem solving. In general, this project was established to explore methods for optimally supporting DoE-specific Next Generation Internet (NGI) applications and interoperability among diverse national and regional infrastructures, such as GigaPoPs, regional networks, and national university networks, and ESnet. To accomplish these goals, EMERGE investigated emerging network technologies, architectures and methods and established a testbed to evaluate approaches for implementing an interoperable quality-of-service infrastructure capable of supporting typical DoE-university mission-critical applications development and deployment.

More specifically, the project was established to achieve high performance interoperability by demonstrating the utility a new architectural standard the Differentiated Services (DiffServ), which was being developed when this project began (in 1999) by the Internet Engineering Task Force (IETF). This architectural approach was developed, in part, to address quality of service (QoS) in the context of scalability. The DiffServ architecture was designed with the understanding that as the Internet expanded (in volume of traffic, number of users, number of domains, numer of connections, etc.), a new method was required to differentiate among different types of traffic. The traditional model essentially provides a single shared infrastructure as a common resource. By differentiating among different traffic flows, network resources can be allocated more precisely and effectively. DiffServ enables packets to be marked at an end system. Intervening routers treat marked packets differently, depending on policies implemented in the routers and associated with the markings. For example, the routers may use a separate queue for marked packets, which is then serviced before other queues. If the allocation of priority traffic is managed so that it is not oversubscribed, some degree of QoS can be provided to those flows that are assigned priority traffic.

This project recognized that interoperability of advanced services for these models is important to multiple high-performance scientific applications. Beyond application traffic modeling and requirements definition, the tasks for this project were to a) build the testbed physical infrastructure (by adding components to the existing Metropolitan Research and Education (MREN) network) b) implement DiffServ (by purchasing and installing suitable DiffServ-capable routers) c) control DiffServ (by implementing the Grid Services Package) and d) apply DiffServ (by collecting and distributing application toolkits). In addition, testing and performance experiments were undertaken to demonstrate and prove various extensions and variations of basic methods. The first phase of the project focused on deploying DiffServ on an MREN testbed. In addition, potential for national scalability was examined by conducting experiments across a national academic research network.

Because MREN is connected to STAR TAP, iCAIR was able to conduct a number of experiments which extended the EMERGE testbed to international locations, including the important HEP site of CERN. Therefore, CERN also was able to participate in the EMERGE testbed through international extensions (described in a subsequent section of this document). Also, tests were conducted between MREN and Yokohama, Japan, and between Yokohama, Japan and CERN. Some testing was also conducted with a research center in South Korea, and with the Nanyang Technological Institute through the SingAREN network.

Although many key energy science related sites, nationally and internationally are supported with diverse infrastructure and protocols, enhances to networking infrastructure technology can benefit the DoE community if embedded in common middleware, such as the Grid Services Package, which supports applications through appropriate network resource management. As part of the project, an EMERGE testbed was established to provide a common suite of advanced networking services was established as part of the EMERGE testbed, including the Grid Services Package, which is being increasingly used by DoE laboratories and university applications – nationwide and world-wide. EMERGE demonstrated that DiffServ architecture, if implemented in combination with a suite of other methods and network services, could be a valuable enabling tool for collaborative applications, especially if implemented on links to organizations and within organizations from boundary points to individual labs. The project

also demonstrated that these techniques are scaleable, regionally, nationally and even internationally, as was demonstrated through the iCAIR STAR TAP projects.

This project noted that ESnet-funded university laboratories are likely connect to universities over different types of network infrastructures using various protocols, such as IP/ATM and IP/SONET links. This project was based on an IP/ATM GigaPoP regional network, which is one of the representative models for DoE/University connectivity. One reason that this model proved useful was that it allowed testbed resources to be allocated within an existing infrastructure, through dedicated PVCs. The DiffServ approaches required several major implementation considerations, among them intradomain implementations and interdomain consideration. Interdomain considerations are a particularly QoS challenging issue.

The EMERGE project examined DiffServ implementations within the context of a variety of other network services, capabilities and protocols, especially, advanced Grid Services. Consequently, the project implemented capabilities for: access control (identification, authorization, authentication, and resource utilization); directory services via the Lightweight Directory Access Protocol (LDAP)[1]; delivery of multimedia data through sequence numbering, time stamping, and contents identification using Real-Time Transport Protocol (RTP); and Real-Time Control Protocol (RTCP) to control RTP data transfers; and network management including instrumentation. This project focused con facilitating advanced data flows insufficiently served by a best-efforts only network: extremely large computed datasets, ultra-high resolution rendered imagery, and real-time unicast/multicast DV (including implementations of the 1394 (Firewire) protocol encapsulated within IP).

iCAIR extended these basic concepts to include considerations of introducing more flexibility in provisioning at the network edge by implementing host-based packet policy making (DiffServ servers) at the network edge. In addition, iCAIR extended these basic concepts to an IPv6 testbed. In conducting these experiments, iCAIR was fortunate to have the assistance of Brian Carpenter, who was in residence at there at the time. Now a Research Fellow with IBM, he was at CERN for almost 20 years, for five years, Chair of the IETF Architecture Committee and he was, and still is Co-Chair of the IETF DiffServ Committee. During the course of this project Brian taught a course on DiffServ at the Northwestern CS department, and many students of that course participated in experiments. Other participants included, from iCAIR, Joe Mambretti (also MREN), Jim Chen, Jeremy Weinberger, Dan Weaver, Tim Ward, Rute Sophia, from IBM, Doug Freimuth, Ashish Mehra, Dinesh Verma, Jim Kelly, from ANL Linda Winkler (also MREN), Alain Roy, Sander Volker, from CERN, Joop Joosten, Paolo Moroni, Tiziana Ferarri, from UPenn Roch Guerin, Wael Ashwami (supported in part by NSF contract #ANI99-06855), as well as multiple teams from EVL at UIC led by Tom DeFanti, and from the Singapore, Japan, and Korea advanced networking community.

The core proposal partners included the Electronic Visualization Laboratory (EVL) at the University of Illinois at Chicago (UIC), the National Center for Supercomputing Applications (NCSA) at the University of Illinois at Urbana-Champaign, and the International Center for Advanced Internet Research (iCAIR) at Northwestern University, with the assistance of the Metropolitan Research and Education Network (MREN)[2]. (MREN was the first GigaPoP, and is now the largest and the most successful model on which others are based.) The research center partners were the University of Chicago ASCI FLASH Center, the University of Wisconsin Engine Research Center, the University of Illinois at Chicago Electronic Visualization Laboratory, the University of Illinois at Urbana-Champaign ASCI CSAR Center, and the iCAIR Center at Northwestern University.

Applications and Network Performance

It has often been noted that QoS is measured differently by the application and the networking communities. For example, application QoS is measured in terms of parameters which are important to end delivered objectives, while network performance is measured in terms of parameters which are meaningful to system design, configuration, operation and maintenance. Network performance parameters may influence application QoS. However, the exact characteristics involved are usually obscure. Also, required application QoS QoS parameters almost never directly translates to objective network performance measures. As a result, developing Grid services are important to both domains. Application-oriented QoS parameters include reliability, real-time and interactive sensitivities, security and traffic burstiness. Network performance parameters include residual error rates (data lost, corruption, out-of-sequence

[1] LDAP allows a collection of different distributed directories to function as a single integrated directory service.

[2] These partners have been leaders in advancing high-performance applications and networking services for over seven years now, producing the MREN (the model for university GigaPoPs), the I-WAY (the first large-scale interagency network cooperative effort), and the Grid (the DoE/NSF/DARPA/NASA advanced middleware project). NCSA also provides support for high-performance networking under NSF's NLANR project.

and duplication), end-to-end delay, jitter or delay variation, mean throughput and peak throughput. High reliability can be translated into low residual error rates. High real-time and interactive sensitivities can be translated into low delay and jitter. Low traffic burstiness (streaming traffic) could imply that mean and peak throughputs are similar. High traffic burstiness could be supported in two ways: peak throughput allocation, or mean throughput allocation with optional dynamic throughput adaptation for better channel utilization.

Architectural and Technology Context

There are many approaches and proposals for implementing per-flow QoS control and management, each of which uses certain metrics related to either performance or microeconomics. Some comprehensive approaches involve interactions between the QoS/policy server and the DiffServ Internet core. Nevertheless, the QoS primitives between applications and the QoS/policy server are ultimately manageable because they are somewhat limited. They include:

- application class
- flow specification
- resource request (bit rate, buffer)
- resource allocation
- admission control
- flow policing
- priority
- packet marking policy
- service level agreement
- others

Many types of meaningful QoS primitives can be generated. The encoding scheme for each primitive has to account for existing QoS methods and standards that involve that primitive. By incorporating these QoS primitives in the QoS signaling and control protocol, it is possible to deliver comprehensive tools by which it is possible to test and evaluate various QoS control and management methods in the edge network. This capability would enable edge devices to support network-based dynamically adjusted applications, which would provide enhanced digital communication services. One approach to implementing this concept could be to presume various mechanisms for explicit application programming interfaces and signaling. However, in addition, other forms of signaling could be independent of specific applications, such as application attribute signaling as a part of a session-initiation protocol. Applications could have a capability for explicitly requesting resources based on requirements. Furthermore, they should be able to receive back-signals related to potentials for resource fulfillment networks, including quality and priority of service as well as related resource management. Fulfillment of such requests almost always requires dynamic inter-signaling interchange.

Within the IETF, there is no standard mechanism for this type of integrated signaling to network resources. However, there is an emerging architectural framework developing under the term "middleware." In December 1998, a workshop was organized and sponsored by Cisco, iCAIR, IBM, and the National Science Foundation (NSF). The MCS Division of ANL also assisting in organizing this event. The goal of the workshop was to identify existing middleware services that could be leveraged for new capabilities as well as identifying additional middleware services requiring research and development. This workshop resulted in an IETF RFC - 2768. It notes the state of middleware and its components, including APIs, authentication, authorization, and accounting (AAA) issues, policy framework, directories, resource management, networked information discovery and retrieval services, quality of service, security, and operational tools.

RFC 2768 sets forth a description of "Middleware, which can be viewed as a reusable, expandable set of services and functions that are commonly needed by many applications to function well in a networked environment." This definition can be expanded to include some types of persistent services, such as those within device operating systems, distributed operating environments, the network infrastructure, and transient capabilities (e.g., run time support and libraries) required to support client software on systems and hosts. A middleware framework can be comprised of a suite of integrated components, including signaling methods, access/admission controls, and a series of defined services and related resources, management of service levels and priority attributes, scheduling, a Service-Level-Agreement (SLA) functions, a feedback mechanism for notifying applications or systems when performance is below the SLA specification or when an application violates the SLA. Any such mechanism

implies capabilities for: 1) an interaction with some type of policy implementation and enforcement, 2) dynamic assessment of available network resources, 3) policy monitoring, 4) service guarantees, 5) conflict resolution ,and 6) restitution for lack of performance.

APIs comprise an area that the IETF has rarely addressed, but that are increasingly important. A range of different types of APIs are important to emerging Internet environments. RFC 2768 highlights environmental discovery interfaces, eg, discovering hardware resources, network status and capabilities, data sets, applications, remote services, or user information, remote execution interfaces, data management interfaces, and process management interfaces. Many of these middleware concepts were implemented in the context of this project by the Grid Services Package (described in the next section). However, supplemental techniques, such as host-based, policy-driven, network resource allocation (described in this document) were also used.

Other IETF architectural contexts for the EMERGE project include RSVP and RFCs 2768, 2474, 2475, 2598 (EF), IntServ, including signaling, resource reservation, and path determination. Various policy implementations techniques through flow control mechanisms, especially queuing implementations were examined during this project. Also explored were various prototype policy information bases, implemented to translate application-specific understandings of precise conditioning indicated by DiffServ code points, e.g., a specific set of queue and threshold settings). Queuing implementations included different techniques for router based classification, priority queuing, class-based queuing, with all flows equally weighted and with identically characterized packets grouped with single flows, including distinctions between those for microflows at ingress points and aggregate flows within the core of the network.

The project examined techniques for defining packets within classes based on various criteria e.g., access control, ingress, and linking those classes to specific queues. In some tests, classes were assigned various bandwidths, weight, queue parameters such as limits, etc. Some of the behaviors that were examined include performance under various conditions, effects of highly dynamic allocations, adaptive shaping, behavior under large aggregation (scalability), insurance to avoid starved flows. Other tests examined the impact of policy control of access to DiffServ under the various LAN conditions. These mechanisms included those for standard and substandard conditions (e.g., out-of-profile flows), and for fault conditions -- for various levels of service. Although the question of the possibility of standard methods for back-signaling was examined, it was done only in the context of standard network engineering. Another issue examined was granularity of potential adjustment for specifically defined classes.

This project assisted in determining optimal means for DiffServ applications management, including mechanisms for network back-signaling and maintaining state conditions at the network edge. The project included determining the best mechanisms that adjust EF policing at ingress points, and implementing specific parameters that guarantee specific levels of quality with set boundaries, considering that the nature of IP traffic behavior. DiffServ monitoring, measurement, analysis, and instrumentation techniques are still evolving, especially with regard to precise quantitative and qualitative evaluations. However, this project successfully determined a variety of productive architectural directions for such measures.

Development and Deployment of Network Middleware to Participating University Laboratories - Grid Services Package

EMERGE leveraged the Grid and Globus efforts supported by the nationwide DoE/DARPA/NSF/NASA funding of ANL, ISI, NCSA, EVL, LBNL and others. The scope of EMERGE efforts under 99-10 involved developing and deploying a DiffServ network that supports the Grid. This project was established, in part, with a broad goal of defining and implementing an Integrated Grid Architecture for advanced network-enabled applications. As has been demonstrated by multiple activities and projects, advanced NGI applications require more from a network than simple data transport; they require a range of Grid services (see Figure 1) that allow the network and other resources to be treated as an integrated whole. These services include authentication, information, resource management, instrumentation, communication, fault detection, and data access. Meanwhile, distributed application development and runtime environments such as Globus have been steadily maturing with a growing number of application projects evaluating, adopting, and helping to improve the Globus middleware services and API's. Increasingly, these distributed applications are requiring guaranteed service levels from the end-to-end system, which from the applications point of view is represented by the Globus toolkits. In order to deliver these guarantees, it seemed necessary for the network quality of service to be integrated into the Globus middleware system.

Motivated by these issues, this project designed, developed, and deployed at each participating MREN institution a Grid Services Package, comprising Grid services deemed useful for DoE NGI applications. An initial Grid Services Package was constructed from end-systems services based on those provided by the Globus toolkit, including authentication, access-control-list-based authorization, LDAP-based resource allocation and characterization, fault detection, and resource management. These particular services are chosen because they were already well developed, proved their utility in a wide range of application projects, and were clearly required by multiple proposed DoE NGI application projects.

The Grid Services Package was then be further developed in a series of stages, with new services being introduced incrementally in response to user demand and as developed in other activities (e.g., in the DoE China Clipper project and in proposed DoE NGI Technology projects). Two early priorities for new capabilities were instrumentation and differentiated services resource management, with the goal of enabling as soon as possible both end-to-end scheduling of networks, computers, and other resources and verification of requested quality of service properties.

This architecture promotes the development of high-performance, reliable, network-aware applications and the sharing of code across disciplines by the definition of a layered architecture comprising four principal components:

- At the *Grid Fabric* level, primitive mechanisms provide support for high-speed network I/O, differentiated services, instrumentation, etc.
- At the *Grid Services* level, a suite of Grid-aware services implement basic mechanisms such as authentication, authorization, resource location, resource allocation, and event services.
- At the *Application Toolkit* level, toolkits provide more specialized services for various applications classes: e.g., data-intensive, remote visualization, distributed computing, collaboration, problem solving environments.
- Finally, specific *grid-aware applications* are implemented in terms of various Grid Services and Application Toolkit components.

Previous experience developing successful Grid services (e.g., Globus) [Foster98a, Foster99b], and working on substantial Grid applications [Foster99a, Brunett], indicated that the definition of such an Integrated Grid Architecture is essential if the scientific community is to adopt and profit from next generation internet environments.

The development and deployment of this Grid Services Package benefited applications projects operating on MREN (and DoE) resources. Broader benefits were contributions to the emerging national and international Grid infrastructures. Multiple Grid programs have now adopted major Grid services and these activities continue to contribute to DoE high performance applications activity.

System Integration

When this project began, the Globus project was in the process of building DiffServ and advanced resource reservation capabilities into the Globus Services Package. NCSA had been working with the Globus team to "harden" and deploy Globus software at NASA and NCSA Alliance sites. With this project, the collaboration was expanded to include the incorporation of DiffServ and the transformation of the Globus software suite into an integrated Grid Services Package for DoE and university sites. The NCSA effort and ANL's Globus team worked on the development and deployment of Grid Services Package at the selected DoE and university sites. During the duration of this project, these researchers worked to develop, harden and deploy the Grid Services Package to enable DiffServ applications and provide additional needed middleware for authentication, encryption, and scheduling. These goals here were to:

Extend information service to represent DiffServ classes and availability
Develop gateway from DiffServ managers to information service
Publish this information
Package and deploy "GRAM" supporting management of DiffServ premium service applications
Develop Software Regression Testing Suite for the Grid Services Package
Deploy and support Grid Services Package v2.0

This effort included the installation, and management of Globus servers at the sites, a testbed-wide integrated Globus system, and the use of public key infrastructure (PKI)-based Globus security systems. [Foster98b] The team also worked with target application projects at the sites to guide the use of the Globus/DiffServ API and perform end-to-end performance evaluation. A suite of services was offered to the target application groups that lead to a nationwide offering. The Grid Services Package was designed to simplify installation at the five sites, and included service prerequisites, installation procedures, documentation and training, addressing site-specific integration concerns, and capabilities for common testbed services.

The Grid Services Package is based on the Globus Toolkit. Globus assumes a set of basic services such as Secure Socket Layer (SSL), Secure SHell (SSH), and LDAP. These were packaged together with Globus to provide a uniform set of installation procedures and scripts. The packaging effort also provided administration documentation for the overall system as well as for individual components. Activities were undertaken to validate Grid Services Package releases through its implementation in conjunction with tesbed experiments. One of the issues that arose was implementing the Grid Services Package in the context of local services and policies, and integrating Grid Services with those services and policies. As the Globus team developed new services to support advanced reservation and DiffServ, specified as Grid Services (Middleware) in Figure 1, they were made them available to the EMERGE testbed.

All of these efforts were based on the specifications provided by various IETF initiatives, such as Policy Framework, DiffServ, and the emerging IETF method of access control called Authentication, Authorization, Accounting (AAA). Implementations through specific Bandwidth Brokers may differ in that separate models may be implemented on different infrastructures for experimental purposes. The DiffServ implementations have been based on concepts of a defined Service Level Specification (SLS), initially statically engineered, two service levels, premium and best effort, point-to-point, static inter-domain provisioning, and simple local domain management.

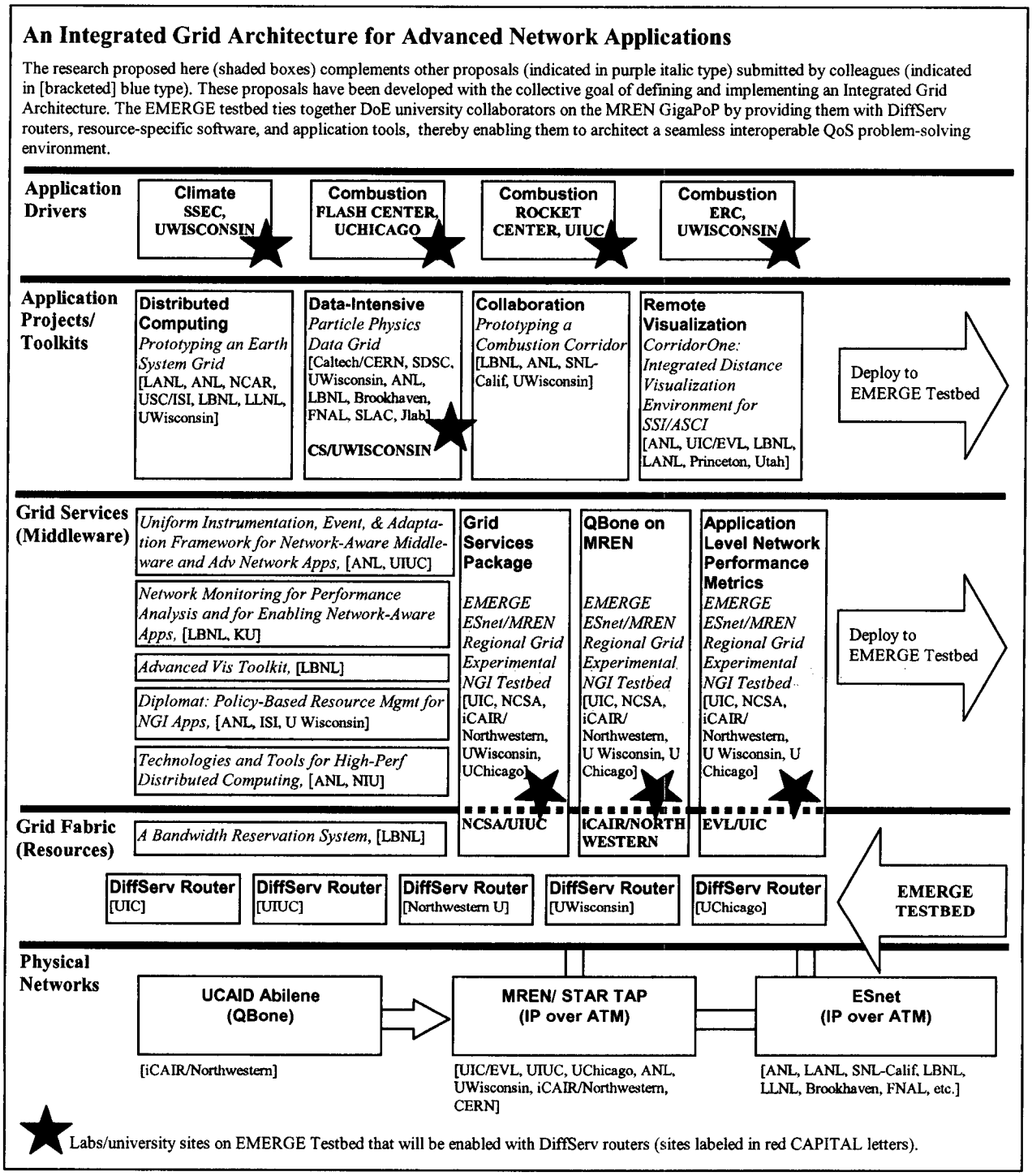


Figure 1: An Integrated Grid Architecture for Advanced Network Applications

EMERGE DiffServ Network Testbed Concepts

Traditionally, Internet services provide all traffic with the same level of performance (a best-effort service, BE). This approach leads to suboptimal performance as Internet services are scaled, or even if there is simply local congestion, because all packets are treated identically. When he was a member of the LBNL Network Research group, Van Jacobson developed a variety of concept to address this issue. For example, he developed software for individually marking packets so that they could be identified as requiring priority treatments. This approach allowed for the implementation within routers of capabilities for policy decision making, treating different packets, or classes of packets, individually depending on set conditions, eg, priorities. He also developed concepts of class-based queuing. Subsequently, these concepts were successful demonstrated through digital video experiments between LBNL and ANL. Van Jacobson's concepts lead to the development of the DiffServ architectural standard. [Van Jacobson] [Nichols]

The IEFT DiffServ architecture offers a framework within which it is possible to define and implement a range of network services that are differentiated on the basis of precise measures of performance. DiffServ makes it possible to request a specific performance level on a packet by packet basis, by marking the DS field of each packet with a specific value. This value specifies the Per-Hop Behavior (PHB) to be allotted to the packet within a network. Typically, in a pre-implementation process, a profile is negotiated (policing profile) describing the rate at which traffic can be submitted at each service level. Packets submitted in excess of this profile may not be allotted the service level requested. A salient feature of DiffServ is its scalability, which allows it to be deployed in very large networks. This scalability is achieved by forcing as much complexity out of the core of the network into edge devices that process lower volumes of traffic and lesser numbers of flows, and offering services for aggregated traffic rather than on a per-micro-flow basis. [Blake]

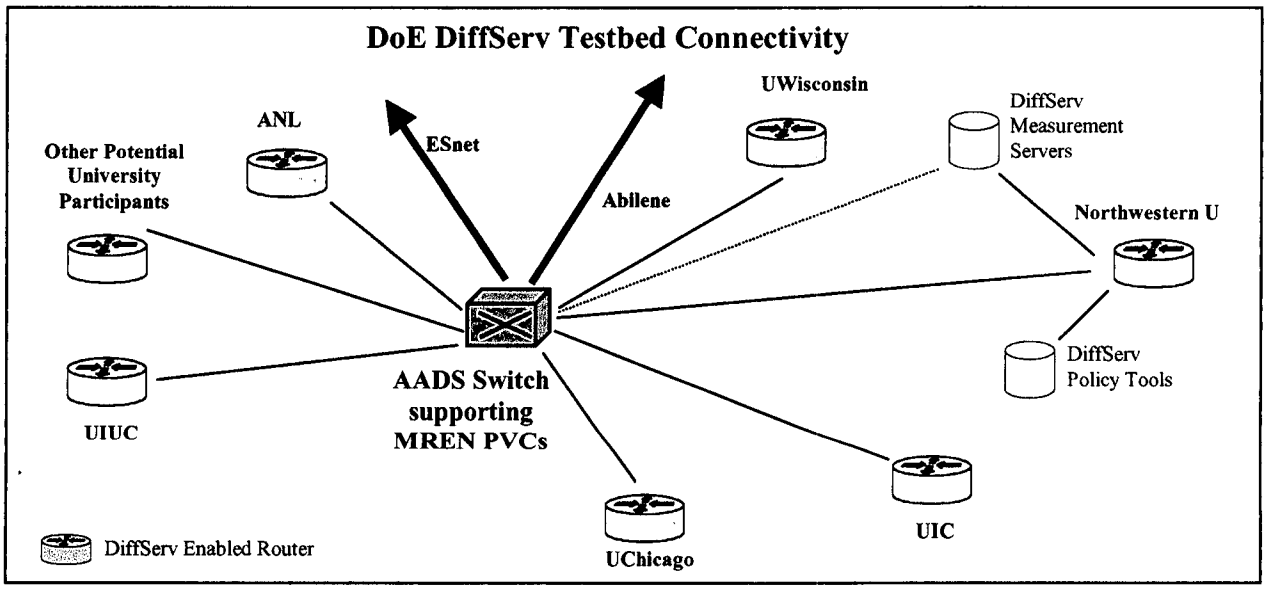


Figure 2: Initial EMERGE Testbed Design

The DiffServ edge router implementations were made possible by implementing Cisco edge routers with beta DiffServ implementations (initially using Cisco IOS 12.0(5)XE2. These routers were used to request at ingress points (e.g., via CAR as a traffic conditioner) packet classification and packet marking (and policing, via drops) for specified flows. For egress point bandwidth allocations were governed via WFQ. CAR is used to set DSCPs in routers. These settings were linked to policy mechanisms. Some experiments were run using priority queueing. The Grid Services Package was provided with an interface to the routers in order to find out available priority bandwidth, allocate it, and do advance reservations. GARA was used as a mechanism for admission control, resource monitoring, scheduling and other management along with router configuration. These mechanisms also supported deployment of advanced DiffServ technology across autonomous networks both when the priority flow represents a small fraction of the available capability and when the priority flow is a significant fraction of the available capability.

EMERGE: DiffServ on an IP/ATM Regional GigaPoP Network

Another important component to the testbed was the transit links, which were based on ATM PVCs.EMERGE provisioned ATM PVCs and PVPs among participating MREN sites to provide controlled test conditions for DiffServ middleware. EMERGE worked with ESnet and and national academic research network DiffServ teams to extend these services to the National Labs and other DoE-funded universities.

Initial GARNET Testbed

The GARNET testbed was developed by ANL in order to implement the first proof of concept for the design concepts behind the Science Grid. These design concepts were then incorporated into the EMERGE testbed. The basic initial configuration for the GARNET testbed is illustrated in Figure 3. The EMERGE iCAIR-ANL/GARNET infrastructure is illustrated in Figures 4 and 5.

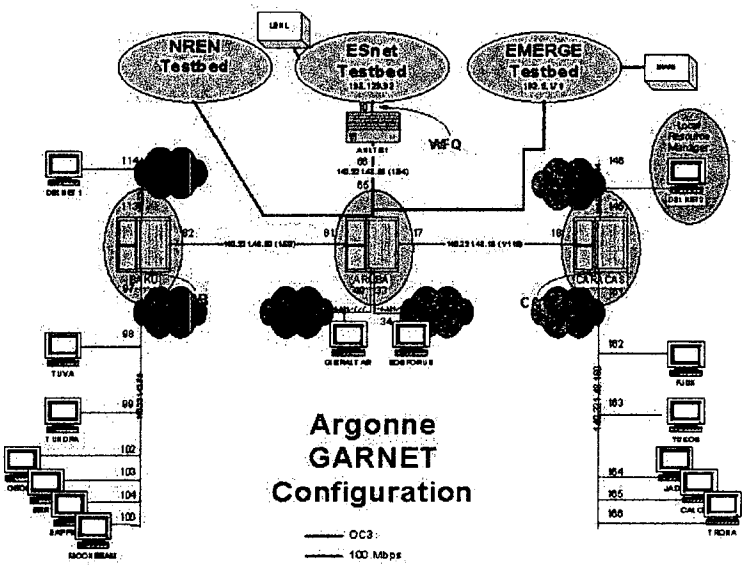


Figure 3: Initial GARNET Configuration

According to CISCO documentation (IP to ATM CoS, per-VC WFQ and CBWFQ CISCO feature specification) for per-VC CBWFQ, the total amount of bandwidth available to service classes was at maximum 75% of the bandwidth allocated for the VC because of ATM cell and other overhead. Consequently, the maximum capacity within the 5Mbps VC was measured at approximately 3.75-4.11 Mbps of data. In general, the bandwidth allocation for specific links was predictable and achievable. This bandwidth became the baseline for various BE/EF experiments.

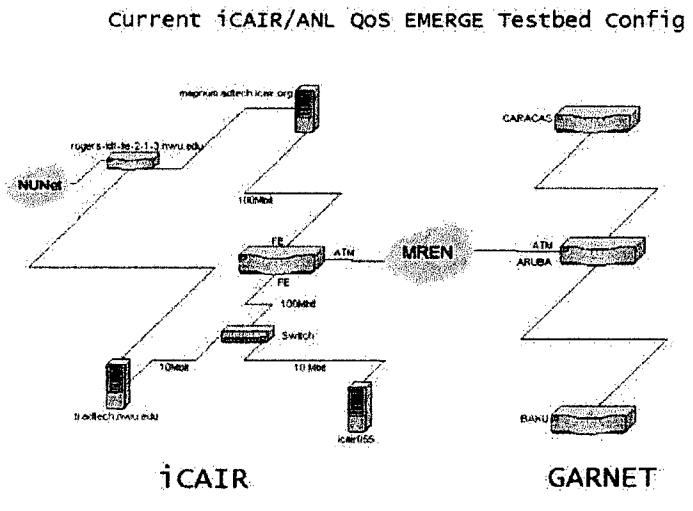


Figure 4: Initial iCAIR – GARNET Configuration

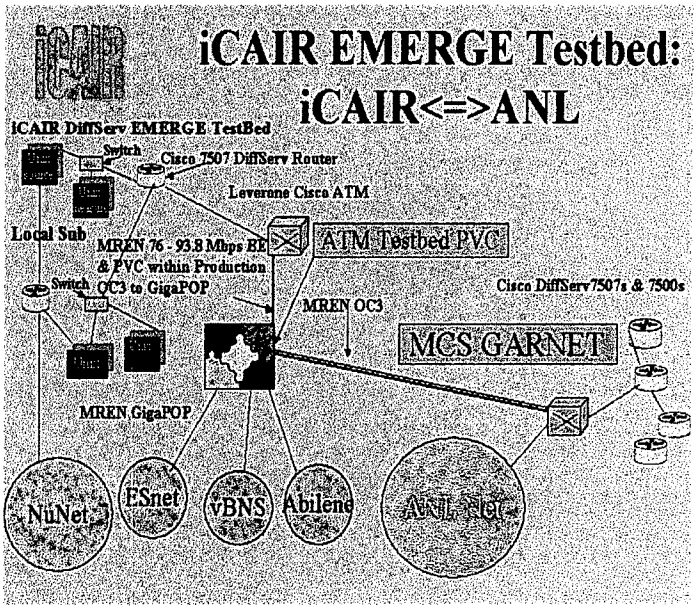


Figure 5: Initial iCAIR – GARNET Configuration

Network DiffServ Architecture: Contrasting MREN and ESnet

At the time of this project, MREN and ESnet had almost identical infrastructures, best-effort IPv4 over ATM over SONET. The underlying regional (MREN, ref Fig 6) and national (ESnet) ATM infrastructures have been provided by both Cisco and Fore state-of-the-art switches, with non-ATM (SDH and Ethernet) infrastructure also accessible.

Figure 6: MREN in the Midwest.

MREN, the Metropolitan Research and Education Network, [http://www.mren.org] is a high-performance infrastructure backbone, primarily OC-3c (155 Mbps). Regionally, MREN interconnects networks from multiple carriers throughout the upper Midwest, as shown in Figure 5. In the Chicago metro area, a series of local loops built originally on Ameritech Public Data Network (APDN). This SONET infrastructure allows MREN to minimize latency, while redundancy and reliability is provided through core switch components. This topology provides full cell switching connectivity between the sites without the delay of additional hops through routers. The core switched service is provided by four Lucent Ascend switches, with highly redundant components, and allows for extremely large volumes of traffic, large-scale expandability, and numerous broadband interconnection links. Uptime for MREN has been 99.999%. Some MREN organizations also provide for redundancy through secondary links; others are planning to do so. The MREN model has always provided an easy path for additional sites (those of existing or new organizations) to acquire economical high-speed access to any site on the network simply — by connecting a local loop to the APDN infrastructure, which allows for immediate high-speed switching capability among all connected sites. MREN provides essentially an L2 transit service, implemented with a full mesh of Permanent Virtual Paths (PVPs). For the early stages of DiffServ proof-of-concept models, dedicated PVCs have provided a useful means of traffic segmentation and experimentation. DiffServ-enabled routers were placed on any MREN campus and linked with dedicated PVCs to provide for experimentation for this project. MREN is planning to migrate to optically based services, based on lightpath provisioning.

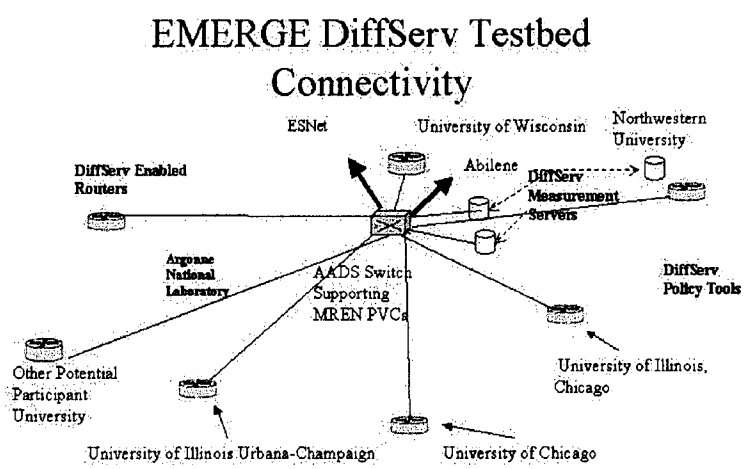
Enabling DiffServ at MREN University Sites

Participating universities provisioned campus network resources for application needs, including some portion of campus access bandwidth. An initial implementation was established at ANL within a laboratory testbed (GARNET, Figures 3,4 and 5). This testbed established the basic architectural and experimental model. Earlier ESnet DiffServ experiments were undertaken in conjunction with ANL and LBNL. Next, an ATM GigaPoP testbed was established over MREN, exemplifying a prototypical regional network. On some campus, including Northwestern, this was accomplished via a dedicated campus PVC (implemented with a campus OC-12 SONET ring connecting the Chicago and Evanston campuses) linking a campus level deployment of a DiffServ router, for example a Cisco 7507 deployed in one of iCAIR's laboratories. The router was configured for CBWFQ on the output interface. This DiffServ router was linked to MREN via a PVC. The core servers for these experiments were at iCAIR's research and development lab at Northwestern University at 1890 Maple in Evanston, Illinois, (within the Evanston-Northwestern Research Park), which is connected to MREN with OC-3 ATM links. iCAIR also has access to the larger national research network community community and to all major agency networks through the MREN GigaPoP infrastructure. It has direct international connectivity through STAR TAP. In addition, iCAIR conducted some experiments using its Advanced Internet Services Facility in the western suburbs and to an experimental supercomputer based in Evanston, both based on SP supercomputers (now replaced by Grid clusters).

MREN, through iCAIR and ANL, designed a "cloud" that was a type of network within a network, with its own services, policies, admission control, accounting, etc., which was segmented for test DoE applications, linked to DiffServ implementations. Interconnections among MREN organizations under this type of implementation could be viewed as a type of regional bilateral Service Level Agreement (SLA), including traffic conditioning agreements. The initial model was based on one emerging from current DiffServ initiatives. For MREN, national research network and international research network application testing, iCAIR deployed application servers that provided controls for simultaneous sessions, set to reflect network configuration and administrative policies. An LDAP server was used as the policy server. This technique served as a realistic generator of latency sensitive differentiated services traffic.

Support for processing of RTCP feedback was implemented, which was used to perform dynamic coarse-grained adjustment of application content.

A number of UNIX workstations and NT systems were used in project activities, including UNIX DifServ servers with specialized TCP stacks. To support experiments to extend the EMERGE concept to edge routers, iCAIR deployed RS/6000s running AIX, and Linus servers implemented with routing software that had the capability of marking IP packets according to any defined DiffServ behavior. This capability was provided in the form of a kernel extension coupled with an application interface and a policy control agent. The policy control agent retrieved QoS policy information from a LDAP-based policy repository. Advanced-networks-based DoE applications can exploit such a capability either directly via the application interface or indirectly using the policy control agent. The policy APIs provide facilities to inform applications of changes in the levels of provisioning for DiffServ classes, to enable applications to react to changes in network conditions. One advantage of this approach is that pre-compiled applications can be integrated into Grid fabrics.



QoS Support Deployment

DiffServ within the EMERGE testbed was implemented incrementally. The 1st increment supported multiple traffic classes with relative precedence (in terms of delay or reliability priority or a combination of both). The simplest case is two traffic classes with two precedence or priority levels. The second developmental increment experiments supported multiple traffic classes with quantitative probabilistic (soft and hard) assurances. Another developmental increment supported multiple traffic flows with quantitative probabilistic assurances.

Figure 7: EMERGE and Measurements

Performance Controls

The degree of resolution for network performance is proportional to the complexity of network controls. Network controls can be classified as either reactive or preventive, and by the lengths of control intervals. Preventive controls take actions to prevent congestion from occurring and reactive controls take actions to recover from congestion once it occurs. For this project more focus was placed on provisioning of reactive controls rather than preventive controls. .Short time-scale controls include priority, traffic shaping and scheduling controls. Medium time-scale controls include admission and load balancing controls. Long time-scale control include resource provisioning. It is expected that the provisioning of short time-scale controls precede that of the long time-scale controls. The network monitoring and control to enable the corresponding DiffServ capability include 1) Multiple Traffic Classes with Relative Precedence – a) Relative Priority Control (in terms of delay or reliability priority or a combination of both) b) Optional per-class monitoring, Multiple Traffic Classes with Quantitative Probabilistic Assurance (soft and hard) - Per-class: a) Monitoring b) Congestion Control c) Resource Allocation Control d) Admission Control e) Usage Control or Shaping, Multiple Traffic Flows with Quantitative Probabilistic Assurance (soft and hard) Per-flow: a) Monitoring b) Congestion Control c) Resource Allocation Control d) Admission Control e) Traffic Shaping.

t is notable that related to these efforts that some investigators examined the influence of a variety of other protocols on overall performance, such as TCP windows sizes, kernel tunings, striped TCP, specialized TCP stacks, eg, iCAIR experiment with a specialized TCP stack developed by the Watson research center. Another example, is the research at ANL Early ANL experiments suggested that test flows required a 10% overprovisioning in order to obtain expected overall performance. To determine the reasons for this, they investigated a variety of components that

contributed to overall results. For example they noted that it was not clear whether TCP's flow and congestion control mechanisms optimally with the mechanisms used for end-to-end QoS. Consequently, they (in particular Sander Volker) analyzed whether existing differentiated services mechanisms could be used with standard TCP implementations, or new versions of TCP were required. ANL performed careful evaluations of high-bandwidth TCP performance to determine the best way to best configure DiffServ for high quality application services .

iCAIR EMERGE Testbed Performance Measurement

Network performance management has become an integral component of the overall network management architecture, which includes configuration, fault, security and high-availability management. These issues are even more crucial for DiffServ networks, which require corresponding network management applications and common management services. Consequently, performance measurement is a key issue for Diffserv networks. To assist in optimizing performance, iCAIR undertook performance measurements for tests across MREN, national research networks and international networks. These experiments were undertaken to provide for appropriate performance measurements, analysis and reporting for the EMERGE testbed as well as to provide the basis for additional performance optimization. They were designed to determine optimal throughput, eg, determining premium bandwidth configuration to optimize applications, latency, eg, RTT to be a good measure of latency at first, avoiding the need to deal with clock synchronization, source of overhead eg, determining how overhead is proportioned among protocol, OS, interface, buffer, and issues related to applications. Experiments conducted to study the effect of various DiffServ options on actual performance behavior—Advanced Forwarding[3] (AF) as well as Expedited Forwarding[4] (EF). These measurements were supplemented by passive Real-Time Flow Measurement (RTFM) network probes. UNIX workstations were used for these probes. The information gathered from these probes and the end-to-end RTCP feedback were correlated to monitor end-to-end network performance and to identify network congestion.

Instrumentation

Network measurement tools can be categorized as active or passive. Active measurement tools are those that actively generate packets and send them to some destination, and then measure attributes of the flow they create. Passive measurement tools attempt to measure aspects of network performance without affecting the network.For instrumentation, iCAIR obtained several measurement tools and analysis packages of both active and passive variety useful for these tests:

- Netperf—traffic generator, which can generate UDP and TCP traffic. An AIX build was provisioned for this measurement.
- udp_gen—traffic generator used at the U. of Michigan for a QoS project; can generate up to 100Mbps of UDP traffic. It is less configurable than Netperf, but is useful to compare with other test results. We have AIX/Solaris builds.
- Ttcp—traffic generator similar to Netperf/udp_gen used at ANL for GARA testing.
- Netflow/Netramet—passive tools to dump and analyze a variety of router information.These are very configurable and already used extensively on the vBNS.
- Surveyor—measurement of one way delay and packet loss, was implemented on a number of EMERGE links, but was not as useful as other tools. Although iCAIR utilized some Surveyor data, because of its limitations, it was not considered a major instrument.

Initially, a plan was developed to use active tools (Netperf, udp_gen) to test links with a server dedicated to testing. These initial determined generally whether the link is behaving as expected. A passive measurement tool (Netflow) was set up on each router to examine what kinds of packets were actually leaving the routers, and thus indicate a) which side of a link had a problem and b) what the problem was.

Specific Packet Generation Tests

First performance was measured with best effort (BE) UDP traffic only (to obtain a baseline):

- Single BE flow that conformed to the bandwidth of the test application server.
- Single BE flow that exceeded the bandwidth of the test application server.

[3] AF is a techinique for implementing DiffServ, basically addressing TCP.
[4] EF is a related techinique for implementing differentiated services.

- Multiple BE flows that conformed to or exceeded the test application server bandwidth.

After obtaining baseline data, including the maximum amount of BE traffic that the test application server can handle without congestion, combinations of marked premium and unmarked BE UDP flows were transmitted:

- Premium flow that exactly conformed to the bandwidth allotted to premium traffic.
- Same as above, with a BE flow occupying the remaining bandwidth.
- Premium flow that exceeded the premium bandwidth allotted to it.
- Same as above, with a BE flow occupying the remaining bandwidth.
- Premium flow that used half of the premium bandwidth.
- Same as above, with a BE flow occupying the remaining bandwidth.
- Equally sized premium and BE flows that together exceeded available bandwidth on the test application server.

In the above cases, "premium flow" indicates a flow of marked packets, which can be treated preferentially in cases where conflict arises with best effort traffic. These test cases tested various scenarios that can arise on a QoS-enabled network where marked and unmarked packets may or may not get congested. Some of the above tests were performed repeatedly, varying CBWFQ settings (the assigned weight, bandwidth, and maximum queue length) on the routers to determine how each factor affects characteristics of the link. Also tested was CBWFQ in conjunction with WRED rather than tail-drop to see how this affected congestion scenarios.

Data Analysis

Different approaches to data analysis were undertaken with various tools. In investigating these tools, some consideration was given to the performance "tax" of measurements (eg, with Netflow). Netperf generates its own summary data of bandwidth, latency, and CPU usage. Udp_gen generates traffic continuously until it is shut off, which can then be plotted to see bandwidth used over time for each flow. Some collections of tools (eg, Netromet) are useful for analyzing otherwise bulky Netflow data as well. The data was used to create graphs and comparisons (eg, plotting bandwidth against time and latency against time for scenarios with congestion and baseline scenarios. Initially, the plan was to set quantitative predictions related to traffic flows, graph results, highlight unexplained phenomena and problems, and then investigate them with sophisticated measurement analysis and other experiments. However, it the course of the experiments, it became clear that some attention also had to be focused on router and host configurations. Cisco for example provides a very wide spectrum of tools for both enterprise and providers with multiple types of functions and capabilities. Many initial tasks related to determining an appropriate router tool set, configuration, and implementation strategy. These tasks resulted in the need to define a number of Cisco configuration formulas as a foundation to experiments. As a cooperative project with the IBM Watson Research Center, iCAIR also conducted experiments related to adding additional functionality to the network through the implementation of DiffServ enabled host servers. These servers provided mechanisms for allowing precompiled applications to be integrated into Grid fabrics and yet also benefit from enhanced QoS mechanisms. As a contribution to this set of experiments, IBM provided an experimental architectural implementation (Figure 8).

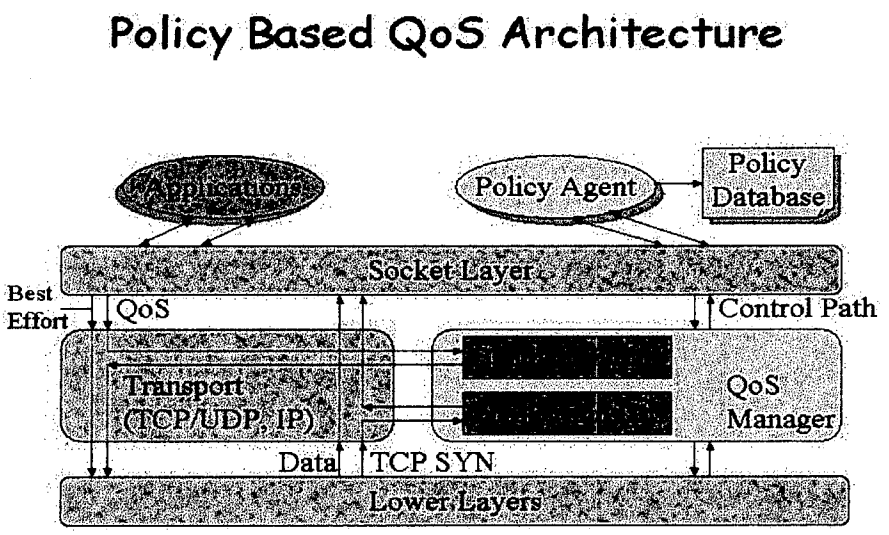


Figure 8: Policy Based QoS

EMERGE, Abilene and QBone

To extend the concepts and architecture explored in this project to universities nation-wide, a number of activities were undertaken to examine the potential for migrating the EMERGE techniques, including Grid serevices, to include interoperability with Abilene QBone DiffServ effort to allow for connectivity to universities on that network and therefore promote interoperability among site at ESnet, MREN, and Abilene. The Qbone project was directed at creating a national interdomain DiffServ testbed to provide the higher education community with end-to-end services over Abilene in support of emerging advanced networking applications. However, because Abilene is an IP over SONET infrastructure, it required special consideration when implementing DiffServ. A key consideration was that in Abilene, for example, there is no easy way to segment traffic, in the same way that EMERGE used PVCs. In large part, the Abilene core infrastructure relies on over-provisioning to achieve QoS. The Qbone architecture has tried to address methods for what the project terms 'virtual wires' while the EMERGE project was able to easily provision these dedicated paths. In part, for purposes of experimentation, this situation was resolved through implementations that considered the highly over provisioned Abilene backbone as an equivalent of a dedicated QoS link because the results, from a packet flow perspective, were "virtually" identical.

When this project was initiated, this effort among MREN/STAR TAP connected-universities formed the largest group of QoS testbed developers. A number of MREN organizations participated in the initiative, including one coordinating the core Bandwidth Broker project, which focused on issues related to providing interdomain QoS. EMERGE participants explored the engineering, behavior and policy consequences of DiffServ running over MREN. These activities extended EMERGE techniques across the national fabrics. This set of activities was intended to be more than a migration of techniques and technologies. This project also examined techniques not being considered by other initiatives, such as alternative DiffServ implementation techniques, especially other Bandwidth Broker models. Interoperability, however, will be a key development requirement. (The QBone Interoperability Group (QIG) first met on December 1, 1998 at Northwestern University for a initial meeting hosted by iCAIR. On January 26, 1999, a larger group met at MCNC in Research Triangle Park, NC to focus on initial testbed deployment issues. QIG included participation by vBNS, Abilene, ESnet, NREN, CA*net2, SURFnet, TransPAC, MREN, NYSERNET, NCNI, Texas GigaPoP, and numerous universities and labs (see Figure 9).

Bandwidth Broker models, built on the system model of computer resource brokers for shared components, have been proposed and explored as intermediaries to dynamically negotiate end-to-end QoS parameters and resource scheduling in accordance with specific time variety requirements such as latency and jitter. However, these models have proven to be extremely challenging technically, complex to administer and costly in implementations.

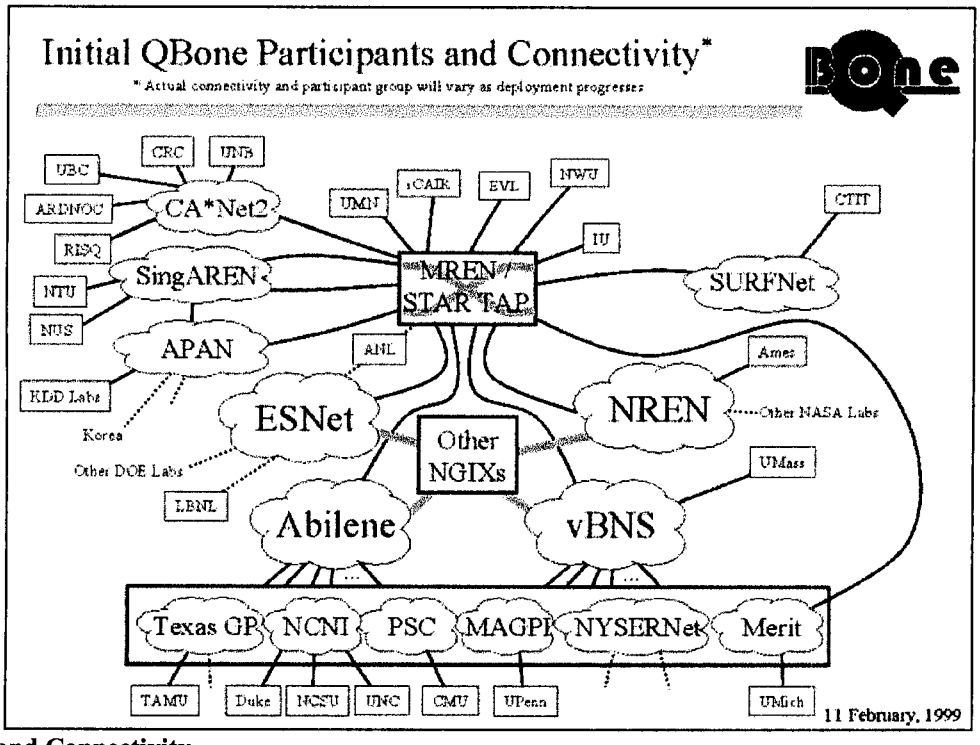
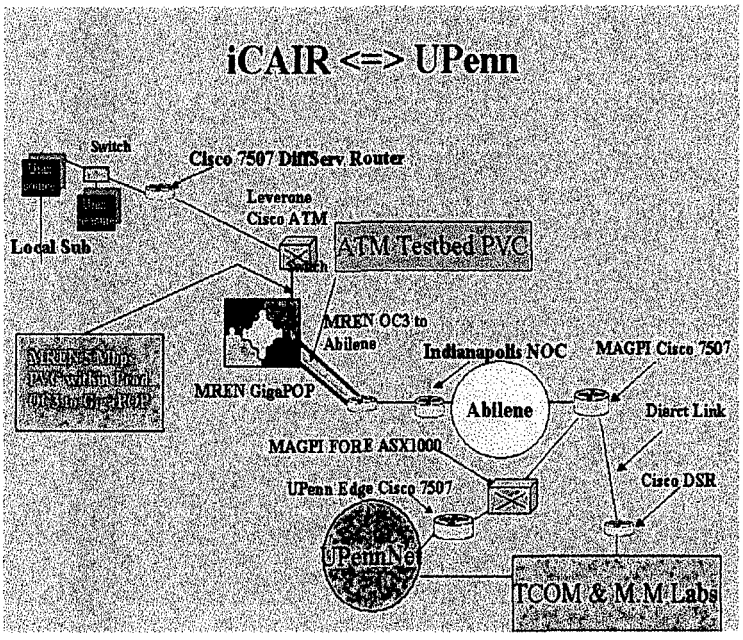


Figure 9: Initial QBone Participants and Connectivity

iCAIR –UPenn Experiments

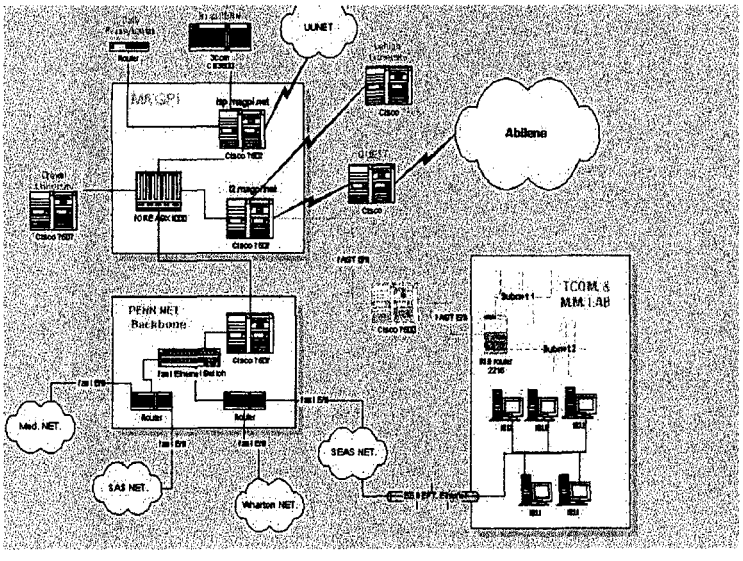


The initial design and development of DiffServ components, processes, access policy, policy enforcement, and monitoring techniques were direct extensions of the EMERGE implementations. The first experiments with Abilene related to considerations of QoS services on the Abilene backbone. The subsequent activities depended on DiffServ implementations at regional GigaPOPs, especially for granulated examinations of flow behavior and policing mechanisms, through specifications for types, classes and levels of service. One of the first partner sites external to EMERGE was UPenn, through the Magpi GigaPOP, where a DiffServ router was provisioned, as part of the Internet2 QBone initiative.

Figure 10: iCAIR-UPenn DiffServ Testbed

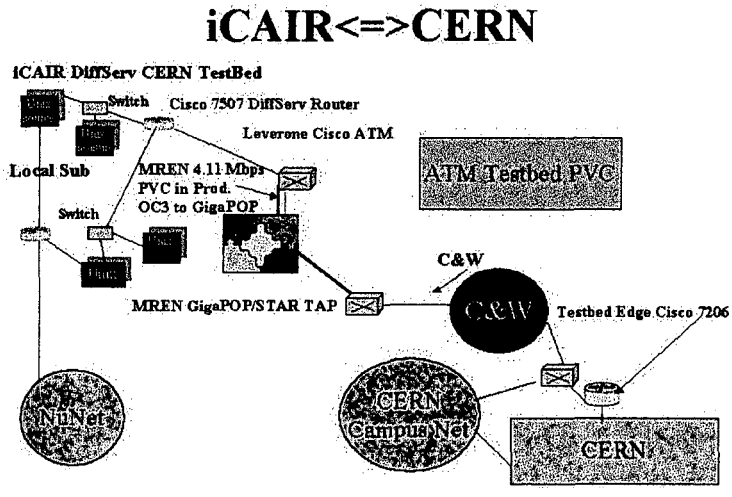
A principal goal of QBone, EMERGE, and related iCAIR initiatives was to develop and implement DiffServ models and to conduct performance testing with key real-time latency intolerant applications and by leveraging these efforts adapt the best results. (iCAIR is also the lead institution for a project that is creating a national digital video network testbed). These applications must be able to signal their requirements, and have networks understand and interpret those requirements as well as translate them into consistent resources — while allowing for a certain amount of dynamic adjustment in resource allocations depending on variations in requirement/resource ratios over time. This second requirement is particularly important for non-premium service applications, for which it is important to develop a secondary control channel to continually monitor and adjust performance.

Figure 11: UPenn Networks with Links to Test Labs



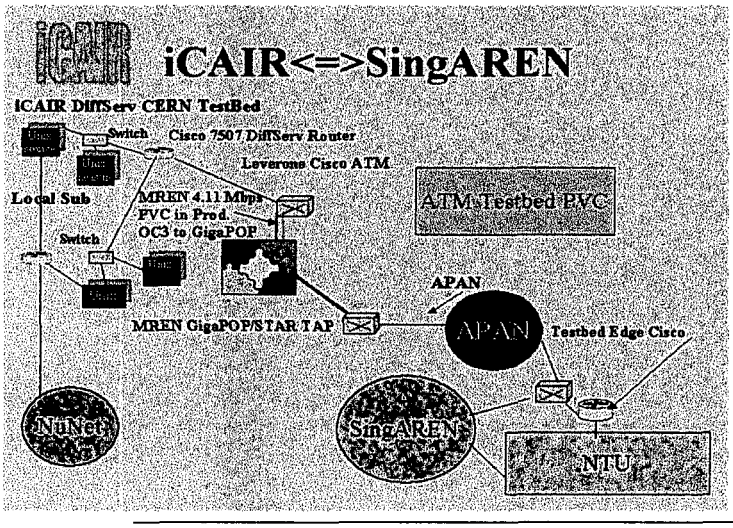
EMERGE and International DiffServ Experiments

To provide for end-to-end, high performance, high quality service for science applications across national and international infrastructures, it is necessary to develop, test, and provide for early deployment of processes and functions for a range of network services. Such services should rely on multiple management policy options that allow for DiffServ categories and distributed governance and resource allocation mechanisms across multiple domains. Prior to the EMERGE project, iCAIR assisted in organizing an international consortium established world-wide DiffServ research project to allow for implementation of international QoS services. This consortium a) investigated the potential for creating a global differentiated services testbed, including those that interlinked NRNin several countries to allow for experimentation with a wide-range of key QoS issues b) designed international testbeds and c) undertook several experiments. This project allowed for these experiments to continue by extending the EMERGE testbed to international locations - almost all under the auspices of the STAR TAP project. These activities developed several international DiffServs testbeds, to address a wide-range of key QoS issues and performance testing across the globe. These testbeds demonstrated the potential for providing for end-to-end high quality service across national and international infrastructures, by developing, testing, and providing for early deployment of processes and functions for a range of network services, including management policy options for DiffServ categories and distributed governance and resource allocation mechanisms across multiple domains.



CERN Experiments (Figure 12)

The first such extension was established with CERN, first through a Cable and Wireless trans-Atlantic circuit and then with a KPM/Qwest circuit. Initially the connection was direct, then it was established via STAR TAP. The experiments were run on a Cisco 7507 running 12.0.5(XE2), - IOS RSP Software (RSP-PV-M), Experimental Version 12.0(20000119:015853) [rolsen-conn_isp 154]. It is notable that with these international testbed also, some of the initial issues related to router configurations. For example, in the initial implementation there was a problem with counter mismatches. This problem was identified and resolved by a member of the TF-TANT testbed researchers, who noted that packet counters splayed with sh pol ... or with sh int fair, only count packets during congestion and that if the tx is not congested, then WFQ is not instantiated. As a result some EF packets were being transmitted but not counted by WFQ. The suggested, and implemented, initial testing methodology was to run tests with UDP constantly congesting the link (e.g. by transmitting at a rate which slightly exceeded the line capacity) and afterward initiating the EF streams (i.e. when congestion was already generated). This method made sure that WFQ was always active and counters registered all of the EF packets.



Experiments with SingAREN (Figure 13)

Similarly, an experimental testbed was established with Nanyang Technological University (NTU) in Singapore through the SingAREN network, via STAR TAP. Various DiffServ experiments were successfully conducted on this fabric using the EMERGE techniques.

Conclusions

The EMERGE Science Grid testbed research project demonstrated that it is possible enhance research collaborations, that utilize high performance applications, between DOE labs and DoE-funded university research centers through specialized techniques to establish high quality network services, including those based on middleware such as Globus. The EMERGE Science Grid project primarily explored methods for optimally supporting DoE-specific Next Generation Internet (NGI) applications and interoperability among diverse national and regional infrastructures, such as GigaPoPs, regional networks, and national university networks, and ESnet. The EMERGE project evaluated approaches to implementing an interoperable quality-of-service infrastructure based on "Differentiated Services" – DiffServ, an architecture developed, in part, to address quality of service (QoS) in the context of scalability. By differentiating among different traffic flows, network resources can be allocated more precisely and effectively. This project a) built a testbed physical infrastructure (by adding components to the existing Metropolitan Research and Education (MREN) network) b) implemented DiffServ (by purchasing and installing suitable DiffServ-capable routers) c) established mechanisms to control DiffServ (by implementing the Grid Services Package) and d) applied DiffServ (by collecting and distributing application toolkits). In addition, testing and performance experiments were undertaken to demonstrate and prove various extensions and variations of basic methods. This project also explored new techniques for host based policy QoS, examined mechanisms for extending DiffServ to IPv6, and extended the testbed to several international sites. Because MREN is connected to STAR TAP, iCAIR was able to conduct a number of experiments which extended the EMERGE testbed to international locations, including the important HEP site of CERN, as well as Japan, Singapore, and Korea. Some of the architectural approaches used in this project are now being used to develop new methods of managing resource allocations on advanced optical networks.

Additional Areas of Investigation

Further areas of investigation include additional exploration and experiments related to the following topics:

a) Extensions of DiffServ implementations among multiple levels of network domains

b) Bandwidth Broker implementations, which remain a complex challenging issue

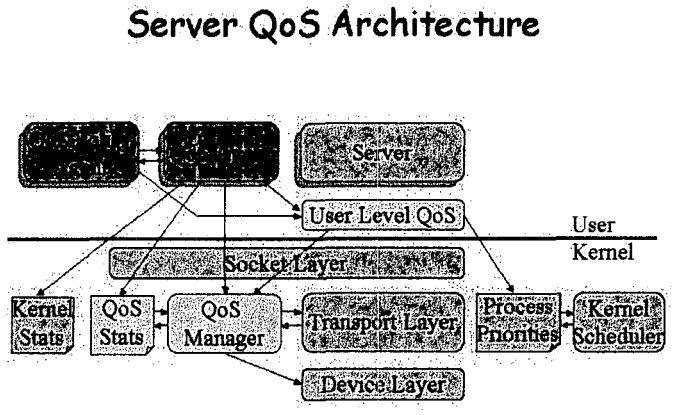
c) Additional integration of QoS concepts into advanced middleware such as Globus Services

d) Additional investigation of these concepts with high performance application behavior

e) Host based extensions to DiffServ implementations. Below is a conceptual design of one such extension (Figure 14).

f) Extensions of these concepts to all optical networks that support wavepath-based services

Figure 14: Server Based QoS Architecture



Bibliography

[Blake] S. Blake, et al., IETF Differentiated Services Framework v01, October 1998, http://www.ietf.org/internet-drafts/draft-ietf-diffserv-framework-01.txt

[Brunett] Sharon Brunett, Karl Czajkowski, Steven Fitzgerald, Ian Foster, Andrew Johnson, Carl Kesselman, Jason Leigh, and Steven Tuecke. Application experiences with the Globus toolkit. In Proc. 7th IEEE Symp. on High Performance Distributed Computing, pages 81-89. 1998.

[Czajkowski] K. Czajkowski, I. Foster, N. Karonis, C. Kesselman, N. Karonis, S. Martin, W. Smith, S. Tuecke, "A Resource Management Architecture for Metacomputing Systems," Proc. IPPS/SPDP '98 Workshop on Job Scheduling Strategies for Parallel Processing, 1998.

[DVC] Data and Visualization Corridors, Report on the 1998 Data and Visualization Corridor (DVC) Workshop Series, Technical Report CACR-164, published by California Institute of Technology, November 1998, http://www.cacr.caltech.edu/publications/DVC

[Foster98a] I. Foster and C. Kesselman, The Globus Project: A Status Report, Proceedings of the Heterogeneous Computing Workshop, IEEE Press, 4-18, 1998.

[Foster98b] I. Foster and C. Kesselman and G. Tsudik and S. Tuecke, A Security Architecture for Computational Grids, ACM Conference on Computers and Security, 83-91, ACM Press, 1998.

[Foster99a] I. Foster and C. Kesselman, editors. The Grid: Blueprint for a Future Computing Infrastructure. Morgan Kaufmann Publishers, 1999.

[Foster99b] I. Foster and C. Kesselman. Globus: A Toolkit-Based Grid Architecture. In The Grid: Blueprint for a Future Computing Infrastructure, pages 259-278. Morgan Kaufmann Publishers, 1999.

[Johnston98] W.Johnston, G. Jin, C. Larsen, J. Lee, G. Hoo, M. Thompson, and B. Tierney (LBNL) and J. Terdiman (Kaiser Permanente Division of Research), "Real-Time Generation and Cataloguing of Large Data-Objects in Widely Distributed Environments," (invited paper), International Journal of Digital Libraries: Special Issue on Digital Libraries in Medicine, May, 1998. [http://www itg.lbl.gov/~johnston/papers.html]

[Johnston99] Gary Hoo, William Johnston, Ian Foster, Alain Roy, "QoS as Middleware: Bandwidth Brokering System Design," HPDC '99, (draft document v0.72, February 19, 1999)

[Lee] C. Lee, C. Kesselman, J. Stepanek, R. Lindell, S. Hwang, B. Scott Michel, J. Bannister, I. Foster, A. Roy, "The Quality of Service Component for the Globus Metacomputing System," Proc. IWQoS '98, 1998, pp. 140-142.

[Leigh97] Leigh, J., Johnson, A., DeFanti, T., CAVERN: A Distributed Architecture for Supporting Scalable Persistence and Interoperability in Collaborative Virtual Environments. In Virtual Reality: Research, Development and Applications, Vol 2.2, December 1997 (1996), Pp 217-237.

[Leigh99] Leigh, J. Johnson, A. DeFanti, T., Brown, M., et al. A Review of Tele-Immersive Applications in the CAVE Research Network, Proc. IEEE Virtual Reality 1999, Houston, Texas, Mar 14 - Mar 17, 1999.

[Nichols] K. Nichols, V. Jacobson, L. Zhang, "A Two-bit Differentiated Services Architecture for the Internet", Internet Draft <draft-nichols-diff-svc>, <http://www-nrg.ee.lbl.gov/papers/2bitarch.pdf>, November 1997.

[Stevens] R. Stevens and T.A. DeFanti, "Tele-Immersion and Collaborative Virtual Environments," The Grid: Blueprint for a New Computing Infrastructure, I. Foster and C. Kesselman (eds.), Morgan Kaufmann Publishers, 1999, pp. 131-158.

[Van Jacobson] Van Jacobson, "An Architecture for Differentiated Services", Talk at IRTF End-to-end WG, ftp://ftp.ee.lbl.gov/talks/vj-e2e-jul97.pdf

Relevant IETF Documents

F. Baker, K. Chan, A. Smith, Management Information Base for the Differentiated Services Architecture draft-ietf-diffserv-mib-16.txt

Y. Bernet, S. Blake, D. Grossman, A. Smith, An Informal Management Model for Diffserv Routers, draft-ietf-diffserv-model-06.txt

N. Seddigh, B. Nandy, J. Heinanen, An Assured Rate Per-Domain Behavior for Differentiated Services, draft-ietf-diffserv-pdb-ar-01.txt>

D. Grossman, New Terminology and Clarifications for Diffserv, draft-ietf-diffserv-new-terms-08.txt

M. Fine, K. McCloghrie, B. Davie, A. Charny, J.C.R. Bennett, K. Benson, J.Y. Le Boudec, W. Courtney, S. Davari, V. Firoiu, D. Stiliadis, An Expedited Forwarding PHB (Per-Hop Behavior), RFC 3246 (Obsoletes: 2598)

J. Seligson, K. Chan, S. Hahn, C. Bell, A. Smith, F. Reichmeyer, Differentiated Services Quality of Service Policy Information Base, draft-ietf-diffserv-pib-06.txt

K. Nichols, S. Blake, F. Baker, D. Black, Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers, RFC 2474 (Obsoletes: 1455, 1349)

S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, An Architecture for Differentiated Services RFC 2475

J. Heinanen, F. Baker, W. Weiss, J. Wroclawski, Assured Forwarding PHB Group
RFC 2597

D. Black, Differentiated Services and Tunnels, RFC 2983

K. Nichols, B. Carpenter, Definition of Differentiated Services Per Domain Behaviors
and Rules for their Specification, RFC 3086

D. Black, S. Brim, B. Carpenter, F. Le Faucheur, Per Hop Behavior Identification Codes, RFC 3140 (Obsoletes: 2836)

B. Davie, A. Charny, J.C.R. Bennett, K. Benson, J.Y. Le Boudec, W. Courtney, S. Davari, V. Firoiu, D. Stiliadis An Expedited Forwarding PHB (Per-Hop Behavior)
RFC 3246 (Obsoletes: 2598)

A. Charny, J.C.R. Bennett, K. Benson, J.Y. Le Boudec, A. Chiu, W. Courtney, S. Davari, V. Firoiu, C. Kalmanek, K.K. Ramakrishnan, Supplemental Information for the New Definition of the EF PHB (Expedited Forwarding Per-Hop Behavior), RFC 3247

G. Armitage, B. Carpenter, A. Casati, J. Crowcroft, J. Halpern, B. Kumar, J. Schnizlein,
A Delay Bound alternative revision of RFC 2598, RFC 3248