Design and Empirical Analysis of MPLS Technologies for the NGI
FY99 Project Plan

Principle Investigators:

Mark Carson
NIST
Bldg 820 Rm 453
Gaithersburg, MD 20899-8920
Voice: 1-301-975-3694
Facs: 1-301-590-0932
Email: carson@nist.gov

Doug Montgomery
NIST
Bldg 820 Rm 457
Gaithersburg, MD 20899-8920
Voice: 1-301-975-3630
Facs: 1-301-590-0932
Email: dougm@nist.gov

Research Objectives:

A primary goal of the Next Generation Internet (NGI) initiative is the research and development of technologies capable of providing selectable, controlled Qualities of Service (QoS) to advanced applications. QoS in the NGI context is much more involved than the simple provision of very high bandwidths. QoS architectures and protocols must provide users, and network providers the capability to control the allocation of expensive network resources (e.g., link bandwidth, packet processing, charged circuits) in ways that are useful to application designers and amenable to network operators (e.g. for purposes of performance, security, business practices). The Internet research and development community has been working on the design of scalable and functional QoS architectures for several years.

Recently, there has been a great deal of interest in *Muti-Protocol Label Switching* [4, 6] (MPLS) as a technology that may enable support of QoS in large scale internets. MPLS attempts to functionally separate the computation of routes in a network from the actual forwarding process. MPLS establishes *Label Switched Paths* (LSPs) through a network or domain and performs forwarding solely based upon virtual circuit-style label swapping along this pre-established path. This basic paradigm of label switching is interesting because it holds the promise of addressing several of the major challenges facing the NGI, including:

- Functionality – label switching provides new functions that were either unavailable or inefficient with conventional hop-by-hop routing. The ability to do explicit path routing could enable network providers to support traffic engineering [8, 12], and could provide leverage in the development of scalable QoS routing algorithms [1] that compute paths based application requirements. MPLS can also be used to address scaling and implementation issues in Virtual Private Networks [15].
- Flexibility – by separating route computation from forwarding, MPLS allows evolution of routing algorithms and protocols at the edge of large networks without impacting the behavior of switches in the core of the network. Given the relative immaturity of QoS / constraint based routing technology, we expect much research, innovation, and change in this area in the future.

While MPLS holds potential as an enabling technology for QoS control in NGI networks, there is much research, development and analysis that must be completed for this technology to mature. Many issues remain to be resolved in the design of MPLS signaling protocols [9], the integration of MPLS signaling with IP routing [7] and IP QoS signaling [5,16], and the integration of MPLS with advanced link technologies such as ATM[10] and WDM. To date almost all of the design and development of MPLS has been driven by a handful of large corporations in the IETF. To date there are no generally available, non-proprietary implementations of MPLS for use by the research community. Likewise, there has been little independent analysis of the real scaling implications of using MPLS technology in large networks. The only published studies [2] have been trivial in their scope and scale.

The objective of this research is to aggressively pursue the design and analysis of MPLS protocols early in their development phase. Our approach is to (1) promote and conduct testing of MPLS technology through rapid prototyping, pilot deployment testing and empirical measurement; to (2) analyze, through large scale detailed simulations, the applicability and scalability of MPLS in today's internets; and to (3) design and test new QoS routing algorithms that exploit the new functionality provide by MPLS forwarding.

By producing public domain, open prototypes of MPLS technology and through the development of publicly available large-scale simulation capabilities, we hope to foster greater involvement of the network research community in the definition of MPLS.

Technical Approach:

NIST Switch Platform for MPLS Research:

NIST Switch[1] is an open (FreeBSD and Linux), public domain prototype of a label switch router (LSR), designed to serve as a platform for research in MPLS protocol design and to serve as a tool for early pilot deployment and empirical testing of label switching in research networks. In our initial efforts, we have produced a prototype of label switched forwarding mechanisms, including support for label aware class based QoS and a prototype of simple extensions to RSVP to support label distribution. The initial limited prototype will be made available to the research community in Q299.

While this first release of NIST Switch will provide the research community with the first publicly available MPLS research prototype, there is much research and development of this platform that must be completed. In particular, one focus of our future work will be the research, development and testing of advanced label distribution protocols (LDPs), and the integration of LDPs with IP routing and QoS signaling protocols. The design of LDPs is currently a hotly debated topic within the IETF community and a critical component in the future applicability and scalability of MPLS technology.

A second focus of our future prototyping work will be research in the use of MPLS technology in the implementation of end-to-end QoS services. Of particular interest is the use of label switching mechanisms to support QoS service architectures such as differentiated services, the ability to use MPLS mechanisms to support aggregation and scaling of QoS services, and the effects of heterogeneous QoS architectures on end-to-end services.

A third area of future work on our research platform is the integration of MPLS forwarding and signaling with advanced link layer technologies. Our first target, will be to expand the platform to support label switched integration with ATM[10]. Once complete, we will conduct experimental deployment and testing in wide area research networks such as CAIRN. We will seek to team with interested application and network researchers to experiment with the use of MPLS QoS to support high performance applications.

Distributed Test Tools for QoS Signaling and Routing Protocols:

Just providing research prototypes is not enough to enable and drive rigorous testing of MPLS technology in research and pilot networks. In order to enable serious testing of such systems automated test tools must be developed. NIST has done some initial research and development of a multiparty distributed test tool that addresses both functional and performance testing of IP QoS signaling protocols. Our Distributed Internet Protocol and Performance (DIPPER) Test System is being designed to support both controlled, laboratory based testing and live, deployment testing of topologically sensitive protocols. The system allows slave testers (STs) to be deployed on multiple hosts that are topologically distributed around the system(s) under test. Test scripts, written in an extension of TCL, can be downloaded and executed on the distributed system of STs. Example test scripts would include ones for generation of significant signaling load and state information in a pilot deployment of QoS routers.

As part of this research we will expand the focus of this effort to specificly address the testing of MPLS routing and signaling protocols. We will produce an expanded prototype of the DIPPER system and make it publicly available to the NGI network research community. The combination of the NIST Switch prototype and DIPPER extended to test MPLS will provide a toolkit for testbed and pilot deployment experimentation with MPLS.

Large Scale Simulation of MPLS Technology:

While prototyping helps us to understand the behavior of these protocols in detail, it will be infeasible to investigate critical scaling and stability issues of MPLS technology through actual testbed deployments. In order to study and understand issues related with large scale networks like loop prevention, traffic aggregation and scalability, we will also develop detailed packet level simulations of MPLS protocols and their interactions with IP routing and other signaling protocols. In order to achieve significant scale, we will work with parallel simulators such as the Dartmouth Scalable Simulation Framework (DaSSF) [13,14] running on multi-processors such as the SGI Origin. Such environments are capable of supporting packet level simulations approaching 100,000 nodes in reasonable execution times.

We will extend the IP routing / signaling models (i.e., OSPF and BGP4) available with DaSSF to include detailed models of MPLS forwarding semantics, label distribution protocols, and interfaces to IP routing and signaling. The focus of this work will be to quantify the impact MPLS technology would have on the stability and routing performance of large scale internets. We will use these large scale models to quantitatively compare alternatives for open issues in LDPs design, to characterize the scaling properties of MPLS based QoS services, and to analyze and tune our own algorithms for label aware QoS routing.

Our simulation results and protocol models will be made publicly available to the research community.

QoS Routing Research:

The area in which MPLS may have the greatest impact is in enabling new approaches to QoS routing. The label swapping paradigm, with its support for explicit routing and the separation of route computation and forwarding, provides a foundation for algorithmic approaches to QoS routing that are not feasible in traditional hop-by-hop routing. In our initial research into label-mediated QoS routing, we have been studying two-staged segmented routing algorithms similar to those used in routing FPGA designs [3]. Our initial approaches may be summarized as follows: In general, a label designates a path or tree segment ("stick") in a network with certain QoS characteristics. These "sticks" may be preallocated or (particularly for RSVP-created paths) made as needed. The set of sticks will in general form a covering forest for the network graph.

Routing at the ingress LSR then consists of selecting the appropriate "bundle" of one or more of these sticks, and applying the corresponding label stack to the packets. Then, through the rest of the net, the labels dictate both the route taken and the traffic handling characteristics applied. Desirable routes are those with a small bundle size, and hence small label stack (preferably one only).

Determining good routes requires two steps: the *a priori* allocation of useful sticks, and the on-demand concatenation of these into appropriate bundles. From a graphic-theoretic viewpoint, the questions of interest then become how to allocate a labeled covering forest and how to manage it to achieve the desired routing goals. Our current best heuristic is a form of randomized greedy length-bounded augmenting path algorithm.

Even assuming QoS routing within a single MPLS domain can be done properly, how can such routing be handled end-to-end? Here there are two central issues. The first is one of coherence or QoS translation: given a union of domains, each with different QoS characteristics stated in different terms, what can be said about the overall QoS available? The second is really one of optimization in the face of limited information: given a route through several domains, each with its own cost structure for obtaining "better" QoS, what combination should be selected end-to-end to achieve the overall desired QoS at the minimum possible cost? We intend to devise and test various solutions to these problems, both through simulation and through implementation on the NIST Switch platform

References:

1. *"NIST Switch: A Platform for Research on Quality of Service Routing"*, Mark Carson, SPIE 1998 Conference on Voice, Video, and Data Communications – November 1998 Boston MA.
2. *"MPLS Routing Dynamics"*, Siamack Ayandeh, Yanhe Fan, 3/15/98.
3. *``On Two-Step Routing for FPGAs''*, G. Lemieux, S. Brown, D. Vranesic, Proc. ISPD-97 ACM, April, 1997. (See also W. Moore, W. Luk, ed. *Oxford 1991 International Workshop on Field Programmable Logic and Applications*).
4. "*A Framework for Multiprotocol Label Switching*", Ross Callon, George Swallow, N. Feldman, A. Viswanathan, P. Doolan, A. Fredette, 11/26/1997.
5. "*Use of Label Switching With RSVP*", Bruce Davie, Y Rekhter, A. Viswanathan, S. Blake, Vijay Srinivasan, E. Rosen, 03/12/1998.
6. "*Multiprotocol Label Switching Architecture*", Ross Callon, A. Viswanathan, E. Rosen, 02/01/1999.
7. "*Carrying Label Information in BGP-4*", Y Rekhter, E. Rosen, 08/05/1998.
8. "*Requirements for Traffic Engineering Over MPLS*", Michael O'Dell, Joseph Malcolm, Johnson Agogbua, Daniel Awduche, Jim McManus, 08/28/1998.
9. "*LDP Specification", Bob Thomas*, N. Feldman, P. Doolan, Loa Andersson, A. Fredette, 02/01/1999.
10. "*MPLS using ATM VC Switching*", Keith McCloghrie, Bruce Davie, George Swallow, Y Rekhter,
11. "*Extensions to RSVP for LSP Tunnels*", Der-Hwa Gan, Tony Li, George Swallow, Lou Berger, Vijay Srinivasan, Daniel Awduche, 11/19/1998.
12. "*Constraint-Based LSP Setup using LDP*", Ross Callon, Andy Malis, Joel Halpern, Juha Heinanen, N. Feldman, Eric Gray, Kenneth Sundell, P. Doolan, Loa Andersson, A. Fredette, Pasi Vaananen, Tom Worster, Bilel Jamoussi, Liwen Wu, Muckai Girish, Timothy Kilty, Ram Dantu, 01/27/1999.
13. "*Modeling the Global Internet*". James Cowie, David M. Nicol and Andy T. Ogielski. Computing in Science & Engineering, January 1999.
14. "*Efficient Large-scale Process-oriented Parallel Simulations*."K. Perumalla and R. Fujimoto. Proceedings of the Winter Simulation Conference (WSC'98).
15. "*CPE based VPNs, using MPLS*", Tony Li, 3/15/98
16. "*MPLS Extensions for Differential Services*", Liwen Wu, Pasi Vaananen, 11/15/98.