MetaNet: A Distributed Network Management System for Information Delivery and Planning with Quality of Service Guarantees¹

Narayanan Natarajan, Abdelhakim Hafid², Ben Falchuk, Arunendu Roy,
Jorge Pastor, and Young Kim

Telcordia Technologies, Inc., 331 Newman Springs Road, Red Bank, NJ, 07701, USA
ahafid@research.telcordia.com

Abstract. *MetaNet* is a network management system that provides end-to-end communication services with QoS and communication planning support for mission critical applications. It supports immediate and advance resource reservation, and provides information on immediate and future available communication capacity taking into account network configuration and traffic. MetaNet supports these services over multiple heterogeneous networks. The MetaNet consists of a number of *PoPs* (Point of Presence); each PoP is associated with one or more communication networks. PoPs realize the functionality within their networks and coordinate with each other to realize the end-to-end functionality. Associated with each network is an *Adapter* that hides from the PoP the network technology specific QoS model and mechanisms.

1 Introduction

Information delivery and planning support for mission-critical applications with QoS guarantees can be achieved by a three-tier system architecture [1]. The top layer, the *policy layer*, maps mission objectives into resource policies (such as user priorities and resource limits) that guide network resource allocation among competing users. The middle layer, the *information channel control layer*, receives requests for information channels from applications and assigns to the channels QoS, priority, start time, and stop time. It optimizes network resource utilization by making use of policies set by the policy layer and current/future network states reported by the bottom layer, the *resource management layer*. The resource management layer manages network resources spanning multiple heterogeneous networks (ATM, IP, Wireless, etc.), provisions QoS for information channels taking

¹This work was sponsored by the Defense Advanced Research Projects Agency. The content of the information does not necessarily reflect the position or the policy of the Government and no official endorsement should be inferred. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.

Copyright 2000 Telcordia Technologies, Inc. All rights reserved.

² Contact author.

into account resource availability, channel priority, and schedule. It is worth noting that a basic QoS support, by the underlying networks, is assumed; for example, in IP networks diff-serv [2], int-serv [3], and/or MPLS [4] support is assumed.

This paper presents the design and implementation of a distributed system, called *MetaNet* that provides the resource management layer functions outlined above. The client of MetaNet is called Agile Information Controller (AIC). The MetaNet is distinguished from other current-day QoS management systems in the following aspects:

- It provides end-to-end communication services, called *MetaNet channels*, over multiple heterogeneous networks. The end-to-end services may be IP or ATM services.
- It supports multi-dimensional QoS including bandwidth, delay, and loss.
- When it requests a channel, the AIC can specify a QoS region (using a range or a set for each QoS parameter) and MetaNet provisions QoS within the region. This minimizes MetaNet-AIC interactions. This approach is more flexible than primary-alternate QoS descriptors supported in ATM signaling [5].
- MetaNet supports priority management. It preempts lower priority channels to accommodate the QoS of higher priority channels. This is an important requirement in mission critical applications that is not supported in commercial products.
- AIC can request channels to be setup either immediately or at a future time. This feature is again important in mission critical applications. As an additional feature, MetaNet allows AIC to specify a start time range, and selects the start time that meets the channel QoS requirements. This capability is not supported in commercial products.
- To enable AIC to assign “realistic” QoS for channels, MetaNet provides an abstract end-to-end view of the static and dynamic (i.e., over a specific time interval) availability of network resources. By “end-to-end”, it is meant that MetaNet hides details of intranetwork and internetwork topology. Thus, MetaNet supports queries like “What is the maximum capacity possible for a potential channel between Host A and Host B during the time interval t_1 to t_2 ?”. Both individual channel constraints and aggregate network constraints (e.g., maximum capacity possible between two hosts) queries are supported. Static resource constraints are determined by network configuration, such as link capacity limits, and multipoint capabilities. Dynamic constraints are determined by network load. This feature is rather unique to MetaNet and is fundamental to service planning in mission oriented environments.

Design of MetaNet supporting the above features requires solutions to the following challenging problems:

- To provision QoS for channels, MetaNet needs to mimic the QoS provisioning scheme used by each underlying network; this is required if the underlying network technology does not allow route enforcement (e.g., ATM does not allow one to specify an explicit route for SVCs).
- MetaNet needs to incorporate an algorithm that efficiently computes the network state as a function of potential channel end points, channel lifetime, and priority. This algorithm should be scalable to large numbers of networks and

channels. A short description of the algorithm (QoS provisioning algorithm) is presented in Section 3; a more detailed description can be found in [6].

- Another major challenge is the definition of the abstract end-to-end network view that MetaNet provides to AIC and an algorithm for processing queries on end-to-end resource availability. The AIC-MetaNet interface is described in Section 2 and the algorithm is introduced in section 3; a more detailed description of the algorithm can be found in [6].
- To cope with multiple heterogeneous networks, the MetaNet architecture should clearly separate technology specific part from technology independent part. A two-tier architecture adhering to this principle is described in Section 2.
- *Internetwork QoS signaling*: A signaling protocol across diverse interconnected networks is needed to realize the end-to-end functionality of MetaNet. More specifically, such a protocol should allow for the support of all features of MetaNet in terms of priority management, advance reservation of resources, and current/future network state computations. None of the existing signaling protocols for ATM and IP networks satisfies these requirements. In Section 2, we briefly present a QoS signaling protocol that supports these requirements.

2 MetaNet Functional Architecture

One of the key challenges in MetaNet design is that MetaNet should be able to support seamlessly network heterogeneity and scalability. To satisfy these requirements, we developed a two-tier MetaNet system architecture (see Figure 1).

Points of Presence (PoP)

Each PoP is associated with one or more networks. Each PoP realizes its portion of the MetaNet functionality within the associated network(s) and coordinates with other PoPs to realize the end-to-end functionality across all networks under the purview of MetaNet. Inter-PoP interactions are supported by an internetwork QoS signaling protocol supporting multi-dimensional QoS, priority based channel preemption, and resource constraints queries and responses.

Adapter

Each adapter is associated with one network. The adapter hides the network technology (e.g., ATM, IP, tactical networks) and QoS mechanisms, including QoS model and signaling protocols, from the PoP while providing a number of primitives that allow the PoP to access the functionality provided by the associated network in a generic manner.

2.1 MetaNet Channel Management and QoS Model

MetaNet provides functions for the management of point-to-point and point-to-multipoint channels. This includes channel setup, release, modify QoS, and modify topology (i.e., add/remove sinks) functions. MetaNet supports both immediate and advance reservation of resources for MetaNet channels.

Fig. 1. MetaNet Architecture

When defining the QoS model to be used in MetaNet, the challenge we faced is that such a model should allow for multi-dimensional QoS specification (a) using generic QoS metrics that are not unique to any network technology; (b) for any type of traffic (e.g., video, WWW); and (c) in terms of QoS regions to minimize the interactions/negotiation between MetaNet and its clients. (a) and (b) are required in order to support the heterogeneity in terms of networks and traffic assumed by MetaNet. (c) is required to provide AIC with the flexibility in specifying QoS (e.g., bandwidth ranges) via a single request and thus, optimizing the response time.

The QoS parameters that are supported by the MetaNet QoS model are *bandwidth*, *delay* and *loss ratio*. These QoS parameters are defined for an unidirectional MetaNet channel originating from the source network interface to the sink network interface traversing the communication networks under the MetaNet. For each MetaNet channel, AIC specifies a range for bandwidth, delay and loss ratio parameters. Of the three QoS parameters, the bandwidth parameter is mandatory while the other two QoS parameters may be left unspecified in a request to indicate a “don’t care” or “not applicable” value.

Three classes are identified for the bandwidth parameter specification: constant bandwidth, variable bandwidth, and adaptive bandwidth.

Constant Bandwidth

Constant bandwidth is specified for traffic that is delay sensitive and has constant requirements over time. It is specified as a range that consists of the worst acceptable value (minimum range value) and the preferred value (maximum range value).

Variable Bandwidth

Variable bandwidth is specified for traffic that is delay sensitive and has variable requirements over time (e.g., MPEG-2 Video). The bandwidth specification consists of a set of the peak bandwidth and the average bandwidth. This is very useful when AIC wants to setup a channel to transmit video, for example; if AIC wants (in decreasing order) to send video as a MJPEG, MPEG-2, or H.261 depending on the availability of network resources, then it may specify three pairs of (peak bandwidth, average bandwidth) in the request (a pair per video traffic type). Upon receipt of this request, MetaNet will try to setup a MetaNet channel to transmit MJPEG; if this is not possible, it will try to setup a channel to transmit MPEG-2; if this is not possible, it will try to setup a channel for H.261. This type of negotiation is not supported by any of the existing QoS models. For instance, ATM UNI [5] allows for the specification of only ATM traffic descriptor & the Minimum acceptable ATM traffic descriptor or ATM traffic descriptor & alternative ATM traffic descriptor. In the video example described above, two ATM SETUP requests may be required. Using MetaNet, one needs only one request.

Adaptive Bandwidth

Adaptive bandwidth is specified for traffic that is neither delay nor loss sensitive and that requires only a minimum bandwidth. For this class, bandwidth specification consists of the worst acceptable value (minimum range value) and the preferred value (maximum range value).

2.2 Abstract Network View and Resource Constraint Queries

To perform its resource planning function, the AIC needs an abstract view of end-to-end network resource availability. For example, AIC needs information about the maximum capacity possible, over a given time interval, for several potential MetaNet channels; e.g., “What is the maximum capacity possible for a potential channel between Host A and Host B during the time interval t1 to t2 without any preemption and with preemption of all channels with priority smaller than priority p1?”. Using this information along with the mission objectives, AIC should be able to determine the channels (and their attributes in terms of QoS, priority, start time and stop time) to be setup.

Providing AIC with information, for example, about the maximum available capacity of potential MetaNet channels is not enough. AIC also needs to know the impact on the capacity available to one channel if another channel is setup with a specific capacity, i.e., bandwidth tradeoffs among potential channels. The challenge is the definition of the necessary information that MetaNet should provide to AIC without incurring an unacceptable overhead and/or violating the MetaNet philosophy of hiding the underlying network details from AIC.

Our solution, that satisfies these requirements, is based on the concept of *MetaNet Link*, defined below, and providing the maximum channel capacity and the maximum aggregated capacity of MetaNet links of interest including information on their dependencies.

MetaNet Link

It is defined as a virtual link between two hosts. It is an end-to-end concept and it does not represent the internal structure of networks under MetaNet. If MetaNet is used to manage only WAN connectivity, routers in LANs can be considered as hosts.

Maximum Channel Capacity

Maximum channel capacity of a MetaNet Link is the maximum capacity that can be requested for a single MetaNet channel between the hosts on the two ends of the link.

Maximum Aggregate Capacity

Maximum aggregate capacity of a MetaNet Link is a ceiling on the sum of the capacities of all potential MetaNet channels that can be setup between the hosts on the two ends of the link.

Dependency

It is a binary relationship defined on MetaNet Links as follows. Two MetaNet Links are dependent on each other if a potential MetaNet channel over one link may compete for common resources with a potential MetaNet channel over the other link. Determination of this relationship is based on intra-network and inter-network resource allocation (including routing) algorithms and policies used within MetaNet.

Using these concepts, MetaNet supports queries on specific MetaNet links, including their static and dynamic capacity over a specific time interval and at a specific level of priority, and dependency relationships.

2.3 Inter-Pop Signaling

PoPs realize the functionality locally within their associated communication networks, and interact/collaborate with each other to realize the end-to-end MetaNet functionality. The set of these interactions is called *inter-PoP signaling*.

In defining the inter-PoP signaling, a number of challenging issues arise: (a) the protocol should not be technology dependent; and (b) the protocol should allow for transmission of information necessary to support MetaNet's unique features such as advance reservation, priority management and static/dynamic resource availability computations. The existing signaling protocols do not tackle any of these issues. In this section, we briefly outline the salient features of our inter-PoP signaling by describing the inter-PoP interactions involved in setting up a MetaNet channel.

When a PoP receives the setup request (from AIC), it determines the internetwork route for the channel (source routing), determines the availability of its local network resources, possibly after determining the set of lower-priority channels to preempt, "conditionally" commits the local resources to the new channel, and propagates the setup request to the next hop-PoP, including the route information. Each PoP repeats this step.

The PoPs involved in a MetaNet channel setup effectively commit the network resources for the channel only when all PoPs determine that resources are available in every network traversed by the channel. This occurs in the second phase of the

setup where each PoP, starting from the destination PoP, sends to the previous-hop PoP a “commit” message. Note that when a PoP determines the set of low priority channels to preempt, it includes this information in the setup request that it sends to the next hop-PoP. When, the latter processes the request, it first “conditionally” preempts these channels and then determines local resource availability. If it determines that it has to preempt some additional low priority channels, it adds them to the list of preempted channels, and propagates the updated list to the next hop-PoP. In this manner, we minimize the number of channels preempted by the PoPs in order to accommodate the new channel.

Note that the signaling protocol we developed supports the setup, release, and QoS modification of point-to-point and point-to-multipoint channel and topology modification (i.e., add/remove sink) of point-to-multipoint MetaNet channels.

3 Resource Management Algorithms

This section presents a short description of two key resource management algorithms that have been implemented in the MetaNet prototype: QoS provisioning algorithm and bandwidth computation algorithm. More details about these algorithms including experiment results on the performance and scalability of the algorithms can be found in [6].

The QoS provisioning algorithm allocates resources to channels taking into account QoS, channel start time, channel end time, and priority. In general, the problem of QoS provisioning is difficult due mainly to the multi-dimensional nature of QoS constraints, i.e., bandwidth, delay, and loss. Multiple constraints often make QoS provisioning intractable; for example, QoS provisioning to accommodate a channel with two independent constraints is NP-complete [7]. MetaNet adds complexity to this difficult problem by allowing the specification of start time, stop time, and priority for channels. To accommodate a channel, the QoS provisioning algorithm should preempt low-priority channels only if it cannot accommodate the channel otherwise. Our goal is to define and implement an algorithm that can be used by a PoP to allocate resources for a requested channel satisfying the channel QoS, time and priority constraints while remaining tractable. This algorithm achieves reduction in computational complexity by decomposing the process of QoS provisioning into multiple steps. First, the algorithm determines the network links that satisfy the bandwidth and loss requirements of the channel over the required time interval. Then, it computes the shortest path in terms of delay. If no such path can be found, the algorithm repeats the same process but considering, this time, the network links that satisfy the bandwidth and loss requirements assuming that all channels with lower priority than the new channel are preempted. With this approach, we transform the NP-complete problem to a problem that can be solved by Dijkstra’s algorithm [8]. Such a transformation comes with a penalty: instead of computing a path that optimizes every QoS parameter (e.g., a path with the maximum available bandwidth and the smallest delay that satisfies the requested QoS), we compute a path that optimizes only one QoS parameter and satisfies the other parameters (e.g., a path, if it exists, with the smallest delay that satisfies the requested bandwidth). The most related work on this topic was done by Wang et al.

[9]. However, there are important differences, which distinguish our work from [9]. The problem solved in [9] is limited to the determination of a path that satisfies the bandwidth and delay requirements; it does not consider the connection start time and stop time. More importantly, if no path is found, the algorithms in [9] generate a rejection. On the contrary, our algorithm considers preemption of lower-priority channels to accommodate the new channel.

A PoP uses the QoS provisioning algorithm to check the availability of resources to setup a MetaNet channel within the associated network: It determines a path, if it exists, that satisfies the requested bandwidth and delay. Then, it communicates, via the adapter, with the underlying network to setup the path/connection. The details of this process depends on the type of the underlying network associated with the PoP. In the following, we briefly present these details in the case of ATM network and MPLS-capable IP network [4].

ATM Network:

The PoP mimics the QoS provisioning scheme (i.e., determining a path that satisfies the requested QoS), used by ATM network; this is required because ATM network does not allow route enforcement (SVCs). With this approach, we make sure that the PoP's decisions (accept/reject) match ATM network decisions.

MPLS-Based IP Network:

The PoP can use the QoS provisioning scheme, described above, to compute a path that satisfies the requested QoS. Then, it enforces the setup of this path by setting the corresponding Label switched Path (LSP)/tunnel in the ingress router of the MPLS-capable IP network. The resources reservations and label distributions can be performed using the extended version of RSVP [13].

The second algorithm, *bandwidth computation* algorithm, computes the availability of bandwidth between two hosts that may be in different networks. Two kinds of bandwidth measures are computed: *maximum channel capacity* and *maximum aggregate capacity* (see section 2.2 for definitions) between any two hosts. MetaNet clients may request either static or dynamic (i.e., over a specific time interval) bandwidth measures. Further, the clients may request these measures either without any preemption or with preemption up to a specific priority level. The main problem with the computation of such bandwidth measures is scalability. A simple approach is for one PoP to act as the coordinator. This PoP solicits from every PoP (including self) the static/dynamic state of the latter's associated network (e.g., network topology, maximum capacity of links, available capacity of links). The coordinator then constructs the end-to-end network state by merging the states of all the networks; then it uses Dijkstra's algorithm [8] to compute the maximum channel capacity and Ford-Fulkerson's algorithm [8] to compute the maximum aggregate capacity. However, this approach is neither scalable (with the size of the networks) nor practical (e.g., a network may not be willing to divulge its topology/configuration to other networks because of security policies). Our goal is to define and implement an approach that improves the scalability of this process and that requires exchange of only aggregate network information between PoPs without loss of precision in bandwidth computation. A novel aspect of our approach is that PoPs exchange only the *necessary information* that is required for the specific bandwidth computation requested by the client. No other (either abstract or raw)

information about the associated networks is exchanged. For example, to compute the maximum channel capacity, each PoP uses Dijkstra's algorithm to compute the maximum channel capacity between every pair of its border nodes or between the source/destination and every border node (in the case of source/destination PoP). Then, the source PoP constructs an abstract global network view (based on the information produced by the other PoPs) that is represented as a graph where the nodes are either border nodes or hosts. Finally, The source PoP uses Dijkstra's algorithm on the constructed graph to compute the maximum channel capacity between the hosts. A more detailed description of the algorithm can be found in [6].

The most related work to this bandwidth computation problem is the state aggregation scheme used in ATM PNNI [10]. However, there are important differences, which distinguish our work from PNNI. First, in PNNI, the aggregation of network information does not take into account the priority or time schedule of channels/connections. Our approach uses aggregation of network information taking into account priority and time schedule; in fact, our algorithms produce aggregate network information assuming that all channels that have priority lower than a given priority are preempted. Second, in PNNI, each network aggregates and propagates the network state information. On the contrary, our approach requires each network to aggregate and propagate only the necessary information (i.e., a subset of the network state) to support bandwidth computation. Third, in PNNI, the aggregation of network information produces imprecise information. Our approach uses lossless information aggregation.

4 MetaNet Prototype Software Architecture

We have implemented a prototype MetaNet system based on the concepts described in Section 2. Figure 2 illustrates the software architecture of the prototype system. In the prototype architecture, the PoP component described in Section 2 is divided into two parts: one *MetaNet PoP*, and one or more *Network PoPs*. The MetaNet PoP receives client requests. Running under the control of the MetaNet PoP is one or more Network PoPs, one Network PoP for each network under the PoP. For each network under the PoP, there exists an *Adapter* that shields from the PoP technology specific QoS mechanisms and interfaces of the associated network.

The Network PoP is the core component of the MetaNet prototype; it contains four components that interact with each other. The *Internetwork Routing* module computes internetwork routes for MetaNet channels. In the current prototype, routing is topology based; topology changes are exchanged between peer routing modules (see Figure 2). QoS routing is a topic of ongoing research. The Network State Maintainer module maintains the network state needed to support MetaNet functionality. It supports sophisticated state filtering functions.

Thus, for example, it provides an operation for retrieving the state of a network path (sequence of inter-switch links) for a specific time interval considering only channels below a specific priority level. Such state filters simplify the QoS provisioning algorithm used by the Network PoP. The Network Channel QoS Manager module provisions QoS for the channels in the local network. The MetaNet Channel Manager module implements the inter-PoP signaling protocol.

Fig. 2. MetaNet Prototype Software Architecture

Each Adapter provides to the corresponding Network PoP the configuration of the underlying network, including its topology, link capacities, and QoS policies such as bandwidth partitioning among multiple traffic classes. Further, it provides primitives for setting up channels in the local network hiding the technology specific signaling mechanisms and management interfaces.

All interactions between AIC, MetaNet PoP, network PoPs, and adapters occur through CORBA interfaces.

4.1 ATM Adapter

To illustrate how an Adapter hides technology specific details from the PoP, we describe, in this section, the implementation of the ATM network adapter in our prototype.

The ATM Adapter consists of three modules: *Network Mapper*, *Network Channel Scheduler*, and *Channel Policy Server*. See Figure 3. The Network Mapper autodiscovers from the ATM switches (via their SNMP MIBs) the network topology and link capacity information and presents this information to the Network PoP via a CORBA interface.

The adapter views the QoS and scheduling information on channels provisioned by the PoP as policy information. This information is stored in the Channel Policy Server, which is implemented using an LDAP (Light Weight Directory Access Protocol) server [11]. When the Network PoP provisions a channel, it communicates the channel QoS and schedule information to the Adapter using a CORBA interface provided by the Network Channel Scheduler module. This module then stores this information in the policy server using the LDAP API.

Fig. 3. ATM Network Adapter Architecture

The Network Channel Scheduler periodically accesses the policy server looking for channels that should be started or released. If such channels exist, the scheduler sets up or releases the channels in the network by invoking the Network Mapper via a CORBA interface. The Network Mapper realizes the setup/release of channels by creating/deleting ATM SoftPVCs using SNMP primitives [12]. Note that the Network Channel Scheduler looks at the start time and stop time of a channel. Based on that information, it will either store it in its cache or send it to the policy server for storage and retrieval depending on how far in time it has to perform the setup/release action. The requests in the cache are the ones that will be scheduled for action in the near term.

5 Conclusion

This paper described the functional and software architecture of MetaNet, a network resource management system that provides comprehensive QoS management capabilities for mission critical military applications. It supports current and advance resource reservation for communication channels spanning multiple heterogeneous networks, priority based resource preemption, and queries on static and dynamic end-to-end network resource constraints. MetaNet consists of two major functional components: PoPs and Adapters. Each PoP manages one or more networks, provides MetaNet functions within its domain, and cooperates with other PoPs to support end-to-end services. Adapters hide network specific QoS mechanisms from the PoPs. A prototype based on this architecture has been implemented. See [6] for a

description of the resource management algorithms used in this prototype. While this prototype serves as an initial proof-of-concept system, there are some issues that merit further investigation:

Integrated experiments combining AIC and MetaNet are needed to verify whether the MetaNet Link dependency notion is sufficient to support AIC's resource planning functions. If the internetwork topology is not rich enough to support significant route diversity, the dependency information will not be very useful, as each link would then be dependent on other links. In such cases, it may be beneficial for MetaNet to provide to AIC a finer view of the network state. For example, given a potential new channel between two hosts, the MetaNet can return the set of MetaNet channels whose routes intersect the route of the new channel. This will enable AIC to make resource tradeoffs among competing channels.

When MetaNet cannot accommodate a new channel within the requested QoS region, we believe that providing options to AIC (e.g., different start time or different QoS) will enable AIC to make better resource planning decisions. Extending MetaNet with such options introduces new challenges in the design of MetaNet resource allocation algorithms and the QoS signaling protocol.

Currently, we are investigating the use of traffic-engineering techniques, (e.g., MPLS traffic trunks [4] and ATM with VPCs and VCCs) to optimize the setup of MetaNet channels in terms of signaling and traffic multiplexing. For example, instead of setting one VCC (resp. Label Switched Path: LSP) per MetaNet channel, we can setup only one VP (resp. LSP) to transmit the multiplexed traffic of a bundle of MetaNet channels. Note that the MetaNet channels in the bundle are not required to have the same source and destination; VPs that transmit the traffic of the bundle of MetaNet channels will be setup between network devices that are traversed by the bundle of MetaNet channels. The determination of the bundle of the MetaNet channels and the values of the attributes of the VPs (resp. LSPs), such as end points, bandwidth, setup time and priority, depend on many factors including the attributes of MetaNet channels, the state of the network and the aimed trade-off optimization/overhead. Note that with MPLS, MetaNet has more control since it can enforce routes for LSPs (explicit routes are provided to the ingress routers); this is not true with ATM's SVCs.

References

- 1) DARPA BAA 98-26, AGILE Information Control Environment, June 1998.
- 2) R. Braden, D. Clark and S. Shenker, Integrated Services in the Internet Architecture: an Overview, RFC 1633, 1994
- 3) D. Black et al., An Architecture for Differentiated Services, RFC 2475, Dec. 1998
- 4) E. Rosen, A. Viswanathan, and R. Callon, Multiprotocol Label Switching Architecture, Internet draft <draft-ietf-mpls-arch-06.txt> Aug. 1999
- 5) ATM User-Network Interface (UNI) Signaling Specification, Version 4, af-sig-0061.000, ATM Forum
- 6) A. Hafid, N. Natarajan, B. Falchuk, A. Roy, J. Pastor and Y. Kim, Network Resource Management Algorithms for Information Delivery and Planning with Quality of Service Guarantees, Proc. of IEEE MILCOM'00, October 2000

- 7) M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, NY, W.H. Freeman and Co., 1979.
- 8) R. Ahuja, T. Magnanti, and J. Orlin. *Network Flows: Theory, Algorithms and Applications*. Prentice Hall, 1993.
- 9) Wang and J. Crowcroft, QoS Routing for Supporting Resource Reservation, *IEEE Journal on Selected Areas in Communications*, September 1996.
- 10) ATM Forum, Private Network Network Interface (PNNI), v1.0 specification, May 1996.
- 11) The OpenLDAP Project, The OpenLDAP Foundation, 1998-1999.
- 12) ForeRunner ATM Switch Configuration Manual Software version 6.0, 1995-1999, Fore Systems, Inc.
- 13) B. Davie et al., Use of Label Switching with RSVP, IETF MPLS Working Group, draft-ietf-mpls-rsvp-00.txt, 1998