

RATES: A Server for MPLS Traffic Engineering

P. Aukia, M. Kodialam, P. V. Koppol, T. V. Lakshman, H. Sarin, B. Suter

Bell Laboratories
Lucent Technologies
101 Crawfords Corner Road
Holmdel, NJ 07733, USA

{aukia,muralik,pramod,lakshman,hsarin,suter}@bell-labs.com

Abstract—It has been suggested that one of the most significant reasons for Multi-Protocol Label Switched (MPLS) network deployment is network traffic engineering. The goal of traffic engineering is to make best use of the network infrastructure and this is facilitated by the explicit routing feature of MPLS which allows the potential addressing of many shortcomings associated with current IP routing schemes. This paper describes a software system called Routing and Traffic Engineering Server (RATES) developed for MPLS traffic engineering. It also describes some new routing ideas incorporated in RATES for MPLS explicit path selection. The RATES implementation consists of a policy and flow database, a browser based interface for policy definition and entering resource provisioning requests, and a COPS (Common Open Policy Service) server-client implementation for communicating paths and resource information to edge routers. RATES also uses the OSPF topology database for dynamically obtaining link-state information. RATES can set-up bandwidth guaranteed label switched paths (LSPs) between specified ingress-egress pairs. The path selection for LSPs is based on a new “minimum-interference” routing algorithm aimed at making the best use of network infrastructure in an on-line environment where LSP requests arrive one-by-one with no a priori information about future requests. Though developed for an MPLS application, the RATES implementation has many similarities in components to an intra-domain differentiated services bandwidth broker.

I. INTRODUCTION

The explicit routing feature of MPLS was introduced to address the shortcomings associated with current IP routing schemes, which are hampered by the requirement to forward packets based only on destination addresses along shortest paths computed using mostly static and traffic characteristic independent link metrics. While this shortest path routing is sufficient for achieving connectivity, it does not always make good use of available network resources and is not satisfactory from a traffic engineering point of view. A prime problem is that some links on the shortest path between certain ingress-egress pairs may get congested while links on possible alternate paths remain free. Even in the best-effort model, this means that avail-

able network resources are not being used well resulting in higher delays, and that there is potential for providing better quality of service with the same network infrastructure.

In MPLS networks, when bandwidth guaranteed LSPs are set-up, shortest path routing with fixed link metrics can cause LSP set-up requests to be rejected even though these requests may have been admissible using a different routing scheme. Therefore routing schemes that can make better use of network infrastructure are needed. This better use of network infrastructure while maintaining quality-of-service guarantees is the primary objective of traffic engineering. MPLS networks’ ability to control the path from ingress node to egress node so as to optimize utilization of network resources and enhance performance is regarded as a primary reason justifying the use of MPLS [1].

Traffic engineering is usually associated with offline routing. In offline routing, it is assumed that all tunnels or LSPs that are to be routed and their resource requests are known at the time that routing is done. The objective is to route all these requests while minimizing total network resource usage cost. The advantage of offline routing is that routing can be very efficient since all requests are known at the time of routing. However, in practice, it is likely that new requests need to be set-up after an initial batch of requests have been set-up or the resource requirements of existing requests may change over time. Attempts to route these new requests can be made using the same offline algorithm on the residual network or using a different on-line algorithm. In either case, it may be difficult to accommodate these new requests since in the offline framework the initial routing’s objective was to minimize total resource usage with no consideration regarding accommodation of future requests. This problem of accommodating new requests in the offline framework can be avoided if it is possible to reroute existing connections to accommodate newly arriving requests. However, this is often infeasible because it may be necessary to reroute a large number of existing LSPs to accommodate the newly arrived set-up request. Rerouting of large numbers of existing LSPs to accommodate newly arriving LSPs is operationally unde-

sirable. Hence, even when a batch of LSPs are initially routed, as long as future arrivals can occur and rerouting of a significant number of existing LSPs is not allowed, it is necessary to route using an “on-line” traffic engineering objective for all requests that are routed.

On-line routing, to be practical, can only depend on information obtainable from routing protocols, and in particular link state routing protocols such as OSPF [2] or ISIS with appropriate extensions for dissemination of QoS parameters. The basic idea in link state protocols is that each node within a network sends out information about its links and associated attributes to all other nodes. This mechanism causes each node in a network to have a complete knowledge of network topology. This topology could then be used by individual nodes to make local decisions such as deciding on next hop for a given IP destination. In a LSP set-up scenario, the link state database could be used to compute an explicit path from an ingress node to an egress node. This explicit path can then be installed in the network using a signaling mechanism such as RSVP. Since it is difficult to obtain measured delay or buffer usage from routing protocols, the LSP path selection must not depend on these. RATES uses only link state and bandwidth usage information in its path selection.

In building a software system that uses link state information to compute LSP routes, two distinct paradigms are of prime concern: a centralized route server paradigm where each ingress node solicits a route from a well known route server, and a completely distributed paradigm where each ingress node computes a route using the link state database by itself. Each of these paradigms has its merits and demerits as are discussed later in this paper. RATES is a software system built using the centralized paradigm. Note however that the LSP path selection algorithms in RATES do not use any information that requires a centralized implementation. Only *aggregated and not per-path* information is used in path computation, and obtaining this information in a non-centralized manner is feasible using proposed traffic engineering extensions to routing protocols [3], [4]

MPLS and Bandwidth Guaranteed LSPs

In MPLS [5] packets are encapsulated, at ingress points, with labels that are then used to forward the packets along label switched paths (LSPs). Service providers can use bandwidth guaranteed LSPs as components of an IP Virtual Private Network (VPN) service with the bandwidth guarantees used to satisfy customer service-level agreements (SLAs). These LSPs can be thought of as virtual traffic trunks that carry flow aggregates generated by classifying the packets arriving at the edge or ingress routers of an MPLS network into “forwarding equivalence classes”

(FECs) [5], [6]. The classification into FECs is done using packet filters that examine header fields such as source address, destination address, type-of-service bits, etc. The filter rules determining the FECs can be established in a variety of ways such as by downloading from a policy or route server, or by interaction with routing protocols. The purpose of classifying packets into FECs is to enable the service provider to traffic engineer the network and route each FEC in a specified manner. This is done by mapping arriving packets belonging to an FEC to one of the LSPs associated with the FEC. Before mapping packets onto an LSP, the LSP is set-up, along an explicit route if specified, using a signaling protocol permitting resource reservation such as RSVP, or CR-LDP (Constrained Routing Label Distribution Protocol). RSVP extensions to support explicit routing by incorporating an EXPLICIT_ROUTE object into RSVP Path messages have been proposed in [7].

In this paper, we mostly consider only the setting up and routing of LSPs with bandwidth guarantees (which we merely refer to as LSPs in the rest of the paper). This does not mean that service level agreements (SLAs) cannot incorporate other metrics such as delay and losses. We concentrate on bandwidth routing because we think that the most common traffic engineering usage of LSPs will be to set-up bandwidth guaranteed paths. If QoS constraints such as delays and losses are to be incorporated in SLAs, the most practical way of handling this, given the traffic descriptor and SLA, is to convert such an SLA into an effective bandwidth requirement for the LSPs (with the queueing delays and losses primarily restricted to the network edges) which can then be routed through the MPLS network as a constant-bit-rate stream encountering only negligible or predictable queueing delays in the MPLS core network. Routing taking delay and loss metrics directly into account is difficult computationally and requires information difficult to acquire from the network such as nodal load versus delay characteristics. Even though RATES only uses routing with bandwidth constraints, it is possible to incorporate various policy, hop-count and delay constraints within the bandwidth-routing framework.

II. ARCHITECTURE CONSIDERATIONS AND DESIGN CHOICES

The following are the main architecture/design concerns and decisions.

1. *Centralized vs. Distributed Implementation:* Even though the algorithms used in RATES use only information obtainable distributedly via extensions to routing protocols [3], [4], these extensions are still under

- development and do not have a large installed base. Until the completion of standards, a distributed implementation is not necessarily desirable for the heterogeneous network scenario that we consider. This is a prime motivation for adopting a centralized approach in RATES.
2. *Obtaining link state information in a centralized paradigm:* There are at least two possible approaches for obtaining link state information from the network. One is to use a standard protocol such as SNMP to retrieve necessary information. However, this is viable only if standardized MIBs exist for QoS related link and nodal attributes. The other approach is to peer as a link state protocol peer with some node in the network and get the link state data as part of the protocol operation. If the link state protocol has traffic engineering extensions as suggested in [3], [4], then all topology related data needed by path selection algorithms is readily available from the link state database. In the absence of these extensions, for example in a network running OSPF/IS-IS, topology is available but not the link resource usage information. In RATES, given the premise that traffic engineering extensions are not widely deployed and that SNMP MIBs for QoS attributes are still not standardized, OSPF peering is the mechanism to obtain topology information and binary link and node states (up/down). In addition, to obtain QoS related static link and nodal attributes, RATES has a graphical user interface which the network administrator can use to provide necessary parameters such as link bandwidth, and preferences or constraints. Since RATES is responsible for all bandwidth allocation on links, it can keep track of reserved and available bandwidth on each link in its database.
 3. *LSP Route Computation:* Computation of a route for a new demand can be triggered either by a request from an ingress node to RATES or by the network administrator using the RATES graphical user interface. As explained before, the LSP path selection must be done using an on-line routing algorithm and it cannot be assumed that no future requests will arrive (as in offline algorithms).
 4. *Use knowledge of ingress-egress points of LSPs:* Even though future demands may be completely unknown, the routers where LSPs can potentially originate and terminate are known since these are the network's edge routers. The LSP path selection algorithm must be able to use any available knowledge regarding ingress-egress pairs and must not always assume that every router can potentially be an ingress and egress point (though this may be the case sometimes).
 5. *Good re-routing performance upon link failure:* This is clearly an important performance metric for path selection. When a link fails, it must be possible to find alternate routes for as many LSPs as possible. If before failure, certain ingress-egress pairs have no residual capacity available between them then re-routing LSPs between these pairs after link failure is not possible.
 6. *Policies:* Routing policies need to be established for using the LSP routes computed and set-up in the network for data forwarding. The policies could be in the form of packet classifiers that filter IP packets and redirect them into LSP tunnels bypassing the routing table lookup at the ingress router, or could take the form of routing table based policies which result in routing tables that use LSP tunnels as their next hop. Sending all transit data going out of the routing domain from a given BGP NEXTHop could be sent through a LSP set-up between an ingress router and the router identified by the BGP NEXTHop. RATES provides a graphical user interface for the network administrator to specify these policies.
 7. *Installing an LSP route in the network:* Once RATES computes a route for an LSP demand, it must be installed in the network. Again, there are at least two possible choices. One is hop by hop provisioning of each device on the route as is done for PVCs in ATM networks. Another is similar to soft PVCs in ATM networks where the route server communicates with the source of the route and spawns off signaling from the source to the destination for route setup. In keeping with the requirement of being able to operate on heterogeneous networks, RATES uses the second approach where it communicates only with the ingress router in order to set-up an LSP route. No standards exist for this mechanism. Since LSP route can be viewed as a policy for routing as administered by the network operator, RATES views this communication as a policy decision and therefore uses the COPS [8] (Common Open Policy Service) protocol currently under proposal in IETF for managing policy related state between a Policy Decision Point (PDP, policy server) and a Policy Enforcement Point (PEP, policy client). Necessary extensions to the COPS protocol for this purpose have been defined and are discussed later in this paper. An alternative to COPS would have been the use of SNMP. However, most vendors and network administrators seem to prefer SNMP for predominantly read operations and for vari-

ous reasons, including security, discourage the use of SNMP for setting MIB data in devices. Also, RATES needs the ingress routers to send requests to it and also to report to it the status of route install operations. With SNMP these can be realized either by polling or by SNMP traps. Neither of these are satisfactory because polling is expensive and traps are unreliable. COPS uses TCP for transport from which it naturally inherits flow and congestion control as well as some notion of a session between a COPS client and server. Also, efforts are under way in IETF to reuse information in MIBs in the context of COPS. These reasons justify the choice of COPS over SNMP for RATES.

8. *Installing packet classifiers in the network:* RATES uses COPS to provision the packet classification policies at the ingress routers. SNMP could be used for this purpose as well given that both require extensions to the protocol that need to be implemented at the ingress router. Given that RATES needs COPS extensions to be implemented at the ingress routers for LSP set-up, provision of classifiers has been bundled into COPS extensions. This has the additional advantage that as further services are defined in terms of classifying data into LSP tunnels, extensions to only one protocol are needed.

9. *Directory vs. Relational database:* RATES uses a relational database as its information store. Although directories have been hailed as preferable repositories in policy servers, a preliminary examination of the data model used in RATES suggests that the repository we use should be efficient for both data access and updates. While directories are very efficient for access, a relational database seems to be better for frequent updates. For this reason, RATES uses a relational database. Use of a hierarchical data model or use of directories is an issue for future work.

10. *Scale:* RATES operates on a network within a single OSPF area. The prime reasons for this choice are: 1) With multiple areas, peering at just one node gives a link state database with abstract information for areas that are not directly connected. Even though it is possible to summarize resource usage parameters for hierarchical path selection, for traffic engineering it seems best to optimize resource usage within a single area with no loss of information due to summarization of topology and usage information. 2) MPLS concepts such as label stacking that are needed in the context of LSP setup in a hierarchical network are not yet deployed significantly. As MPLS deployments mature and support label stacking and as network utilization issues in a hierarchical network are

better understood, RATES will incorporate the ability to work in a hierarchical structure. The path selection algorithms used in RATES can provide information useful for aggregation of QoS parameters for use in hierarchical networks. Another important aspect of scalability is the manner in which RATES reacts to abnormal and highly frequent changes in topology in the network. When LSP routes are affected by topology fluctuations, RATES attempts to reroute these LSPs along alternate routes. However, abnormal conditions could cause RATES to hit overload conditions. RATES has builtin dampening mechanisms to deal with such overload conditions.

11. *LSP restoration options:* Multiple levels of rerouting are possible within the RATES framework, each with increasing scope, increasing reaction time and increasing bandwidth utilization and efficiency. Besides the link level protection schemes like SONET APS which are beyond the control of RATES, paths can be protected by preestablished backup paths or through rerouting in the case of failure. Backup paths can have a full associated bandwidth reservation or a reservation that is shared or partially shared with other paths. Establishing shared paths requires further extensions to the MPLS signaling protocol to allow a group of labels from unrelated sessions to share the same bandwidth reservation. Rerouting with established backup paths will take effect as soon as the tunnel ingress router has become aware of the failure of the primary path, either through routing protocols, signaling protocol or other means. The switch-over decision is local to the ingress router and does not need any interaction with RATES. Refer [9] for a detailed description of the RATES algorithm for restoration.

12. *Network re-optimization:* Even though it is generally discouraged to reroute existing MPLS tunnels, a situation may occur in a degenerate scenario where the network is in a very suboptimal state through a sequence of tunnel creations and deletions (possibly due to abnormal changes in network topology), that the network administrator may decide to reroute some tunnels even without network failures. In this case the reroute should use a “make before break” approach where the new path is set up before the old one is removed. RATES has the necessary infrastructure to support this functionality.

13. *Interface with other services:* RATES provides open and extensible interfaces to enable future connectivity with value add services such as billing and also with legacy network management applications

that support provisioning of service level agreements (SLAs) among other features. Due to its modular architecture and the use of CORBA, new modules can be written to interact with the data model, without requiring any changes to the other modules.¹

III. ON-LINE ROUTING: CURRENT STATE OF THE ART

We define the residual bandwidth along a link to be the difference between the bandwidth of the link and the sum of the LSP demands that are routed on that link. First note that a new LSP can be routed along a given link only if the residual bandwidth on that link exceeds the bandwidth requested by the new LSP. These links will be referred to as feasible links (with respect to the given LSP demand). The network consisting of all the routers and just the feasible links will be referred to as the feasible network. Therefore, when performing the routing we can restrict our attention to paths in the feasible network. Note that if the ingress and the egress routers are disconnected in the feasible network then there is no path that has the desired bandwidth and the LSP request is rejected. The objective of any routing scheme is to reject as few demands as possible.

The most commonly used algorithm for routing LSPs is the min-hop algorithm [7]. In this algorithm, the path from the ingress to the egress with the least number of feasible links is chosen. This algorithm though simple uses the same information that the proposed new algorithm uses. Its performance in terms of efficient network usage can be easily improved upon with a little more computation. With the rapid rise in processor speeds and with traffic trunks (virtual leased lines) with expected holding times in the order of hours, days or even months, it is justifiable to trade-off increased computation for more efficient network usage. The min-hop algorithm does not take information on ingress-egress pairs into account nor does it adapt routing to increase chances of successful re-routing upon link failure.

A min-hop like algorithm which attempts to load-balance the network traffic is proposed in [10]. This widest-shortest path algorithm (WSP) finds a feasible min-hop path between ingress and egress such that the chosen min-hop path has the maximum residual path bottleneck link capacity. This algorithm too does not take ingress-egress information into account. With every node being assumed to be a potential ingress and egress point, a widest path algorithm without a min-hop restriction does not work well since long paths which increase network usage get chosen. Hence, a widest-shortest hop path heuristic

¹Currently there is work in progress for replacing part of the user interface module by a machine to machine interface towards an IP telephony signaling gateway.

performs better. However, if the ingress-egress pairs are known then an algorithm which picks paths longer than min-hop works well provided it avoids unnecessarily loading certain “critical” links as we shall see later.

More sophisticated algorithms in addition use the residual bandwidth on the link to influence the weight of the link and the shortest path is chosen with respect to these dynamically changing weights [11]. Since the weights are chosen to increase with link load, the idea is to not use up link capacity completely if alternate lower loaded paths are available. This has the tendency to make capacity available for future demands. Nevertheless, this algorithm is also oblivious to information regarding ingress-egress pairs and therefore can pick long paths to defer loading on links which may not be important to satisfy future demands.

In [12], a mathematical programming formulation is presented for on-line routing taking ingress-egress pairs into consideration. The case of both split and unsplit routing of bandwidth requests was presented. In RATES, instead of solving a linear or integer program, the approach is to route using a shortest path computation on an appropriately weighted graph. This is described in the next section.

IV. KEY IDEAS FOR MINIMAL INTERFERENCE ROUTING

In this section, we give an informal description of the key ideas pertaining to “minimum interference” routing. A formal mathematical description, details of the algorithm, performance studies, and also a proof NP-hardness of this formulation of the MPLS routing problem is in [12], [13].

Illustrative Example

As seen in the previous section, existing schemes take into account the topology of the network and the residual capacities on the links, but do not take into account the location of the ingress/egress routers. These routers serve as sources and destination of future traffic. If routing is done oblivious to the location of these sources and destinations of traffic then we may “interfere” with the routing of some future demands. We illustrate this with a simple example. Consider the network shown in Figure 1. There are three potential source destination pairs, (S_1, D_1) , (S_2, D_2) , (S_3, D_3) . Assume that all links have a residual bandwidth of 1 unit. We now have a request for an LSP between S_3 and D_3 with a bandwidth request of 1 unit. If min-hop routing is used, the route will be 1-7-8-5. Note that this route blocks the paths between S_2 and D_2 as well as S_1 and D_1 . In this example it is better to pick route 1-2-3-4-5 even though the path is longer.

Interference

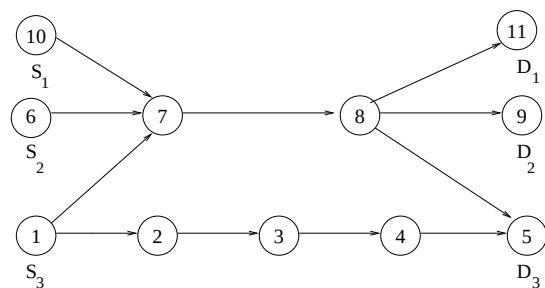


Fig. 1. Illustrative Example

The key idea is to pick paths that do not interfere too much with potential future LSP set-up requests (demands) between other source destination pairs. We first have to make this concept of interference more concrete. Consider the maximum flow (maxflow) value [14] v_1 between a given ingress-egress pair (S_1, D_1) . This maxflow value is an upper bound on the total amount of bandwidth that can be routed between that ingress-egress pair (S_1, D_1) . Note that maxflow value v_1 decreases whenever a bandwidth demand is routed between S_1 and D_1 . Note that the value of v_1 can also decrease when an LSP is routed between some other ingress-egress pair. We define the amount of interference on a particular ingress-egress pair say, (S_1, D_2) due to routing an LSP between some other ingress-egress pair as the decrease in the value of v_1 .

Minimum Interference Paths

With interference defined as above, we can think of a minimum interference path for an LSP between say, (S_1, D_1) as that explicit route which maximizes the minimum maxflow between all other ingress-egress pairs. Intuitively, this can be thought of as a choice of path between (S_1, D_1) that maximizes the minimum open capacity between every other ingress-egress pair. Though this formulation has intuitive appeal it has the drawback that it is the minimal maxflow that impacts the routing irrespective of values of the other maxflows. Hence, another objective might be to pick a path that maximizes a weighted sum of the maxflows between every other ingress-egress pair. Either of these formulations, not only makes capacity available for the uncertain but possible arrival of future demands but also makes capacity available for re-routing of LSPs in case of link failures.

Critical Links

To progress from the notion of minimum interference paths to a viable routing algorithm that uses familiar max-flow and shortest path algorithms we need the notion of “critical links”. These are links with the property that whenever an LSP is routed over those links the maxflow values of one or more ingress-egress pairs decreases. The algorithm to determine critical links with a reasonable amount of

computation is given in [13].

Path Selection by Shortest Path Computation

Once the critical links are identified, we would like to avoid routing LSPs on critical links to the extent possible. Also, we would like to utilize the well-used Dijkstra or Bellman-Ford algorithms to compute the actual explicit route. We do this by generating a weighted graph where the critical links have weights that are an increasing function of their “criticality”. The increasing weight function is picked to defer loading of critical links whenever possible. The actual explicit route is calculated using a shortest path computation as in other routing schemes. See [13] for details of the routing algorithm including link weight computation, and for performance studies.

V. RATES

A major requirement determining the software architecture is that RATES should work within a heterogeneous network with equipment from multiple vendors. The only requirement on the devices is that they should be MPLS and RSVP capable (with MPLS extensions).

A. High level design

RATES has been implemented in Java and C/C++ in a UNIX environment. It consists of the following major modules

- Explicit route computation
- COPS server
- Network Topology and State Discovery
- Dispatcher
- Graphical user interface
- An open application programming interface (API)
- Data repository
- A message bus connecting the above modules

Figure 2 shows a high level view of RATES and its interaction with the network. We give a brief overview of each of the modules and also of the relevant aspects of the edge and core network elements. Each RATES module is implemented as a UNIX process, using predominantly Java and C/C++. Communication between the modules is achieved using a message bus and is completely event driven. The dispatcher has the responsibility for scheduling and dispatching events.

A.1 Graphical User Interface

A server based approach for traffic engineering provides easy and controlled management access for the provisioning and monitoring purposes. The RATES system provides a graphical console that has the following salient features.

- definition of constraints
- monitoring of network topology

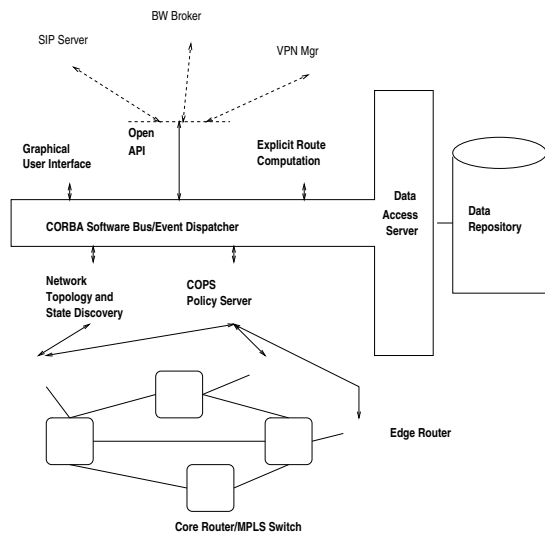


Fig. 2. Block diagram of RATES application

- provisioning of LSPs and filters with facilities to specify constraints such as those suggested in [1]
- monitoring network utilization
- alerts on certain anomalous events

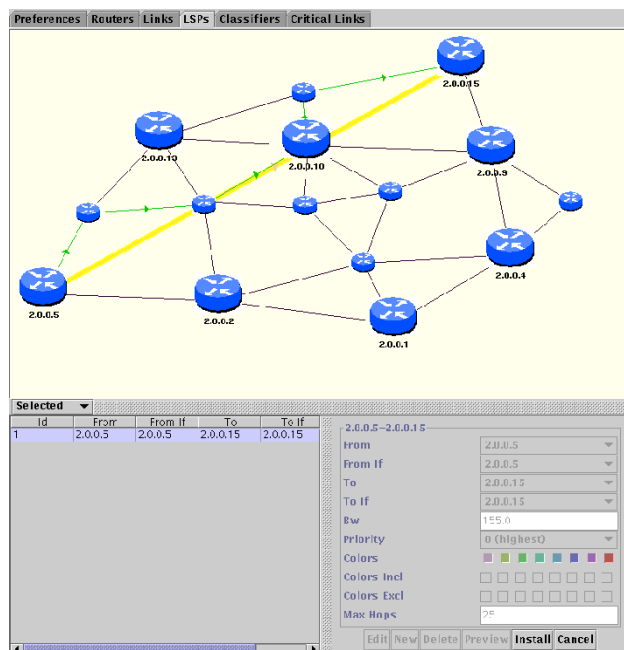


Fig. 3. LSP demand editor

Figure 3 shows a screenshot of the LSP demand editor pane of the RATES graphical management console. The top window shows the current topology overlaid with the current path for any highlighted tunnel demand from the list below.

B. Application Programming Interface

The RATES open API exposes a set of core services such as LSP setup to external devices and modules. Examples of external devices include a SIP server for IP telephony trunk reservation [15], a diff-serv bandwidth broker for trunk reservation or any other management entity that may require setup of fixed bandwidth LSPs. An external module enabling novel policies and services such as LSP setup with temporal constraints is an example of an external software module that can enhance the functionality of RATES using this open API. This API is implemented as a library that interfaces directly with the message bus.

B.1 Explicit Route Computation

The routing module takes network state, policies, constraints and user requests into account in its explicit route computation. The routing algorithm is on-line and based on the “minimum interference” idea as described earlier. The route computation is fast enough to be used for re-computation of routes upon link failures. The route computation time is expected to be generally smaller than the time taken for the link state database to stabilize after link failures. The route computation takes into account set-up and holding priorities.

The RATES architecture permits the easy addition of modules to enhance route computation functionality. One option is the addition of a module to convert user specified SLAs (which may include requirements such as loss and delays) into an effective bandwidth requirement. Another option is a module for traffic engineering of elastic traffic such as the sizing of an aggregate traffic trunk for carrying TCP traffic with a minimum service assurance. A third option is the routing of traffic with bandwidth guarantees under the “hose” traffic model where only the aggregate traffic from a customer to a network ingress is known, and the fractions of the incoming traffic destined to the different network egresses is not known. Refer [16] for a traffic engineering scheme applicable in this context. Another important option is the incorporation of modules to provide multi-service and cross-layer functionality. Examples are the addition of path computation algorithms for wavelength routing in optical networks and algorithms for joint cross-layer traffic management in IP over WDM networks.

B.2 Restoration of LSPs

One option for restoration of LSPs is to let RATES detect link failures (via changes in its topology database) and then explicitly reroute LSPs which were traversing failed links. As shown in the performance studies in [13], the path selection algorithms in RATES increase the chances

of successful rerouting upon link failure.

Recently, proposals [17], [18], [19] have been made to incorporate fast restoration mechanisms in MPLS. These restoration mechanisms allow backup paths, onto which traffic can be quickly redirected upon failure detection, to be setup simultaneously with the active path thus ensuring that an LSP if set-up is quickly restorable upon failure. RATES can incorporate restoration mechanisms by using a restoration-capable on-line routing algorithm described in [9]. The incorporation of restoration leads to a new QoS routing problem where it is necessary to dynamically route both an active path and a backup path in order to satisfy a request to set-up a restorable bandwidth guaranteed LSP. Simultaneous routing of both paths ensures that sufficient resources will be available upon failure for successful LSP restoration. Simultaneously established backup paths also aid fast restoration by elimination of path computation and path set-up signaling delays.

The primary restoration goals for RATES are either complete path protection or path protection under single link failure. The latter option permits more efficient backup bandwidth usage. For protecting against single link failures, note that two LSPs whose active paths are completely link disjoint can share backup links. If they are partially link disjoint then they can share some backup links. The objective of the algorithm in [9] is to exploit this sharing in order to reduce the total amount of bandwidth consumed by the LSPs. The amount of sharing that can be achieved in the backup paths is a function of the information available to the routing algorithm. The algorithm developed for RATES path restoration needs to know only what fraction of each link's bandwidth is currently being used by primary paths, and is currently being used by backup paths. This information is obtainable by appropriate routing protocol extensions. Performance studies of this algorithm are in [9].

B.3 Data Repository

The RATES server application is based on a CORBA based software bus for communication among the modules and for access to shared data such as network state, user demands and route decisions. The data repository provides a view of a shared object pool to the software modules. Each class of objects, like network node, network link, user demand or routed demand has an associated state-machine where each transition is associated with the pairs of input event and source module as well as output event and sink module. This state machine is enforced by the data repository access server for each instance of an object. The data repository uses a relational database to implement a persistent storage of the objects used by the

application.

B.4 COPS Policy Server

Any policy decision entered by the user or generated by the traffic engineering server application, needs to be communicated to the appropriate network elements. In the RATES framework, this function is carried out using the COPS (Common Open Policy Service) protocol [8]. The base protocol defines very little application specific semantics, but allows for the definition of extensions. The base protocol allows to open a session with a specific capability set (client type), monitoring and restoring this session. COPS supports client requests, server replies and server asynchronous updates.

The RATES server uses the client specific extension for intra-domain traffic engineering [20]. It allows the server to install or uninstall policy objects like tunnels and filters on the client, either unsolicited or responding to a request by the client. Filters can be associated with one or more tunnels which allows for rapid and autonomous restoration scenarios in the network. Using a pre-established backup tunnel, the edge router can respond to failures of the primary tunnel without delay and even if the connectivity to the server should be disrupted. The client also reports to the server on the state of each tunnel. To be fully functional only the edge router need to support the specific COPS extension and functionality.

In a heterogeneous network, there may be certain edge nodes that do not support the COPS client functionality. At least two alternative approaches can be used to make such devices controllable by RATES. One is to install a translation module that translates COPS requests and responses to the specific control mechanism understood by the device. A second approach is to incorporate additional modules into RATES that understand the message bus interaction and the device specific control mechanism.

B.5 Network Topology and State Discovery

The RATES server needs to have a view of the network topology and also the state of links and nodes. Such information can be obtained by the server in a number of ways. For example, the topology could be fed into the server statically using a configuration file. The dynamic state of links and nodes can then be obtained using traditional network monitoring approaches such as SNMP traps or polling. However, traps are unreliable and polling can be an expensive operation. An alternative approach is to let RATES auto-discover the network topology and keep track of the network topology state by peering as a link state protocol entity. In an IP network, the link state protocol used could be either OSPF or IS-IS. In a different net-

work, such as an optical network, a similar method could be standardized.

In RATES, the network topology and state discovery module is a plug in module that could potentially implement any of the above approaches. The current RATES system uses an OSPF protocol entity for topology and state discovery. OSPF provides link-state information and this is used by RATES to auto-discover topology and react to topology changes.

B.6 Edge Router

Edge routers are designated ingress and egress points of traffic into the traffic engineering domain. An edge router needs to support origination and termination of MPLS tunnels including the signaling protocol sessions. The most likely signaling protocol for this purpose is RSVP with specific extensions for MPLS [7]. To assign incoming traffic to MPLS tunnels, it needs to support filtering, e.g. at the packet or route level. To communicate with the traffic engineering server, each edge router which needs to originate any MPLS tunnels, needs to support the COPS protocol with the specific extension.

B.7 Core Router/MPLS Switch

The only requirement for the core router is that it implements MPLS switching and the MPLS signaling protocol being used. To ensure bandwidth guarantees, it needs to support a resource reservation scheme both in the signaling and forwarding path. The diff-serv framework [21] offering a small set of classes with associated rate guarantees and frame-relay style contracts, is the most likely contender for this purpose.

B.8 RATES Capability Extensions

Though RATES is primarily an MPLS traffic engineering tool, an approach essentially similar to routing of bandwidth guaranteed LSPs can be used for routing of dynamic wavelength path set-up in optical networks. A wavelength can be thought of as the outermost (non-stackable) label in the MPLS label stack. These functional similarities between setting-up wavelength switched paths and setting-up MPLS label-switched paths have been pointed out in [22] as a basis for integrating the optical layer control plane and Multi Protocol Label Switched (MPLS) control plane (as also the fact that integration permits more efficient network resource allocation). Given these similarities and the possible standardization effort toward integrated control protocols [22], RATES can be evolved into a multi-service cross-layer service provisioning and traffic management tool. This involves the addition of optical-domain specific

constraints into RATES routing module, routing using network information at different (optical, IP) layers, and the incorporation of other topology discovery modules (using different routing protocols such as IS-IS and methods for auto-discovery of optical network topology).

VI. CONCLUSION

We described a system for MPLS traffic engineering in an on-line setting where future requests for bandwidth guaranteed ingress-egress paths can be accommodated without re-arranging existing paths. Ideas for routing that permits best usage of the network infrastructure in this on-line scenario were described. The presented schemes also are capable of protecting specified "privileged" ingress-egress pairs so that LSP requests between these pairs is less likely to experience blocking due to lack of available bandwidth. Although these ideas were presented for use in the MPLS context, they can be used for bandwidth provisioning in a differentiated services network. RATES has components with functions similar to those needed in a bandwidth broker and is extensible for use as an intra-domain bandwidth broker.

VII. ACKNOWLEDGEMENTS

We would like to thank Zbigniew Bogdanowicz for implementing part of the algorithms used in RATES and Ping Pan for working on the COPS extensions and client implementation.

REFERENCES

- [1] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus, "Requirements for traffic engineering over MPLS," Request for Comments (Informational) 2702, Internet Engineering Task Force, Sept. 1999.
- [2] J. Moy, "OSPF version 2," Request for Comments 2178, Internet Engineering Task Force, July 1997.
- [3] D. Katz and D. Yeung, "Traffic engineering extensions to OSPF," Internet Draft, Internet Engineering Task Force, Apr. 1999, Work in progress.
- [4] H. Smit and T. Li, "IS-IS extensions for traffic engineering," Internet Draft, Internet Engineering Task Force, June 1999, Work in progress.
- [5] R. Callon, A. Viswanathan, and E. Rosen, "Multiprotocol label switching architecture," Internet Draft, Internet Engineering Task Force, Apr. 1999, Work in progress.
- [6] R. Callon, N. Feldman, A. Fredette, G. Swallow, and A. Viswanathan, "A framework for multiprotocol label switching," Internet Draft, Internet Engineering Task Force, June 1999, Work in progress.
- [7] D. Awduche, L. Berger, D. Gan, T. Li, G. Swallow, and V. Srinivasan, "Extensions to RSVP for LSP tunnels," Internet Draft, Internet Engineering Task Force, Mar. 1999, Work in progress.
- [8] J. Boyle, R. Cohen, D. Durham, S. Herzog, R. Rajan, and A. Sastri, "The COPS (common open policy service) protocol," Internet Draft, Internet Engineering Task Force, Feb. 1999, Work in progress.

- [9] M Kodialam and T. V. Lakshman, "Dynamic routing of bandwidth guaranteed tunnels with restoration," in *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, Mar. 2000.
- [10] R. Guérin, A. Orda, and D. Williams, "QoS routing mechanisms and OSPF extensions," in *Proc. of Global Internet (Globecom)*, Phoenix, Arizona, Nov. 1997.
- [11] S. Plotkin, "Competitive routing of virtual circuits in atm networks," *IEEE J. Selected Areas in Comm.*, pp. 1128–1136, 1995.
- [12] M. Kodialam and T. V. Lakshman, "On-line routing of guaranteed bandwidth tunnels," in *Seventh IFIP Workshop on Performance Modelling and Evaluation of ATM/IP Networks*, June 1999.
- [13] M Kodialam and T. V. Lakshman, "Minimum interference routing with applications to MPLS traffic engineering," in *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, Mar. 2000.
- [14] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin, *Network Flows: Theory, Algorithms and Applications*, Prentice Hall, 1993.
- [15] M. Gibson, "The management of MPLS LSPs for scalable QoS service provision," Internet Draft, Internet Engineering Task Force, Oct. 1999, Work in progress.
- [16] M. Kodialam, T. V. Lakshman, and S. Sengupta, "Multicast routing with bandwidth guarantees: A new approach using multicast network flow," preprint, Nov. 1999.
- [17] D. Haskin and R. Krishnan, "A method for setting an alternative label switched paths to handle fast reroute," Internet Draft, Internet Engineering Task Force, June 1999, Work in progress.
- [18] D. Haskin and R. Krishnan, "Extensions to RSVP to handle establishment of alternate label-switched paths for fast re-route," Internet Draft, Internet Engineering Task Force, June 1999, Work in progress.
- [19] S. Makam, V. Sharma, K. Owens, and C. Huang, "Protection/restoration of MPLS networks," Internet Draft, Internet Engineering Task Force, Nov. 1999, Work in progress.
- [20] B. Suter and P. Pan, "COPS extension for intra-domain traffic engineering," Internet Draft, Internet Engineering Task Force, June 1999, Work in progress.
- [21] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated service," Request for Comments (Proposed Standard) 2475, Internet Engineering Task Force, Dec. 1998.
- [22] D. Awduche, Y. Rekhter, J. Drake, and R. Coltun, "Multi-protocol lambda switching: Combining MPLS traffic engineering control with optical crossconnects," Internet Draft, Internet Engineering Task Force, Nov. 1999, Work in progress.

Petri Aukia received an M.Sc. degree from Helsinki University of Technology, Helsinki, Finland and Institut Eurécom, Sophia-Antipolis, France in 1998. He is currently at the High Speed Networks Research Dept. at Lucent Technologies, Bell Labs. His recent research interest include traffic engineering and network security.

Murali Kodialam received his Ph.D. in Operations Research from MIT in 1991. He has since been working in Bell Labs. His research interests are in the areas of resource allocation in communication systems and performance analysis.

Pramod V. N. Koppol received the Ph.D degree in Computer Science from North Carolina State University, Raleigh, in 1996. His interests are in software architectures for net-

working devices and systems and in software development issues in signaling, routing and management protocols used in ATM and IP networks. He is a Distinguished Member of Technical Staff in the Multi-Service Optical Networking Department at Lucent - Bell Labs.

T. V. Lakshman received the Ph.D degree in Computer Science from the University of Maryland, College Park, in 1986. Prior to that he received a Master's degree from the Dept. of Physics, Indian Institute of Science, Bangalore, India. From 1986 to 1995 he was at Bellcore where he was most recently a Senior Research Scientist and Technical Project Manager in the Information Networking Research Laboratory. He is currently a Distinguished Member of Technical Staff in the High Speed Networks Research Department at Bell Labs. His recent research has been in issues related to traffic characterization and provision of quality of service, architectures and algorithms for gigabit IP routers, end-to-end flow control in high-speed networks, traffic shaping and policing, switch scheduling, routing in MPLS and optical networks. Dr. Lakshman is a co-recipient of the 1995 ACM Sigmetrics/Performance Conference Outstanding Paper Award, and the IEEE Communications Society 1999 Fred. W. Ellersick Prize Paper Award. He is an Editor of the IEEE/ACM Transactions on Networking.

Helena Sarin is a member of technical staff in the R&D Group of the Lucent Technologies' Cable Communications business unit. She is interested in issues related to distributed objects in the form of standard models such as CORBA, COM, Java, and the Web. She earned an M.S. degree in system analysis from Moscow Institute of Civil Engineering.

Bernhard Suter received a degree in Communications Systems Engineering from Swiss Federal Institute of Technology, Lausanne, Switzerland and Institut Eurécom, Sophia-Antipolis, France in 1996. He is currently a member of the High Speed Networks Research Dept. at Lucent Bell Labs in Holmdel. His recent interests have included traffic management and system architectures for high speed IP routers.