Appeared in: Proceedings of the 17th UK Teletraffic Symposium, Dublin. 16–18 May 2001.

AGENT-BASED DYNAMIC CONFIGURATION OF DIFFERENTIATED TRAFFIC USING MPLS WITH CR-LDP SIGNALLING

Greg Osinaike*, Rachel Bourne* & Chris Phillips*

Abstract

Advanced network protocols for traffic engineering will enable more intelligent control of network resources and will give network operators the ability to offer quality of service (QoS) guarantees for Internet Protocol (IP) traffic. Our work focuses on using the Multi-Protocol Label Switching (MPLS) protocol together with agent technology to dynamically reconfigure the logical network topology to allow intelligent segregation of different classes of IP traffic. Our agents monitor and report on local network performance, which is used to compute paths that can be optimised with respect to specific constraints such as required bandwidth and maximum packet loss. We have developed an implementation of the MPLS signalling protocol, Constraint-based Routed – Label Distribution Protocol (CR-LDP), used to configure Label Switched Paths (LSPs), that works on top of an existing MPLS forwarding mechanism.

The introduction of agent technology allows decision-making on LSP selection to be based on locally held information. For example, our agent implementation controls the selection, reservation, and configuration of paths. Additionally, an algorithm has been developed that searches for and then reserves the 'cheapest' path through a network. Resources reserved for higher priority traffic are only utilised if no other resources are available. The result is a segregated flow of traffic with the highest priority data using the most 'optimal' resources.

This paper presents an agent-based architecture that has been developed as a test-bed. The test-bed is being used to investigate the system performance of an agent-managed traffic-engineering scheme relative to traditional IP routing for service delivery. The test-bed features and agent architecture are described. The system dynamically configures the network to allow differentiated traffic to travel across different routes.

Introduction

The rise of sophisticated data link and network layer protocols such as ATM, MPLS and IPv6 has enabled more intelligent traffic engineering including connection admission control, dynamic routing of real-time traffic, congestion management through differentiation of services, even responsive pricing regimes to better reflect the real cost of providing different qualities of service (Kelly, 1997). The networks to which these protocols are being applied are highly dynamic, both in terms of the traffic carried and topologically, either through short-term changes due to lines and switches going off-line, or through long-term changes due to the addition and removal of nodes. Moreover, the amount of information available as inputs to the decision-making and control processes is significant, highly changeable and distributed throughout the system. For these reasons, agent technology has been suggested as the best means of making use of the resources available.

*Department of Electronic Engineering
Queen Mary, University of London

Agents are autonomous processes that make decisions based on local information (Jennings & Wooldridge, 1998). This research is aimed at controlling the network effectively using agents that can make decisions to route traffic using criteria different from those of the underlying network protocol. Although agents' actions are limited, giving agents the power to decide which actions to perform and when (e.g., setting up a virtual path) and to set the network parameters for their actions (e.g., what route through the network to take) can lead to more effective use of the network resources and may help fulfil network operators' obligations such as quality-of-service (QoS) guarantees. An agent's decisions are based solely on the information it has to hand, the agent's beliefs, and the decision-making procedures it has been given, i.e., the agent algorithm. The complexity, and hence potential intelligence, of the agent's beliefs and algorithm is limited only by such things as memory capacity and speed (or reactiveness), and the agent designer's ability to enhance the network's overall performance, or a particular aspect of it.

Agent technology also guides the processes by which agents communicate to exchange information, negotiate over shared services, and compete or cooperate to attain their individual and the overall system goals (Weiss (ed.), 1999). As an analogy, consider the signalling layer in an Intelligent Network; through this layer, the nodes send signals about the traffic they wish to transport and hence communicate information between the nodes. In our work, the agents manage higher level decisions such as which routes to construct and when, and when to divert traffic along these routes. It should be realised that other network components can also be considered as agents, e.g., the IP routers themselves; however, the decision-making potential of such agents is limited since they cannot proactively update their own beliefs.

Applying agent technology to the rapidly developing communications industry is an area that has received a lot of attention from the applied artificial intelligence (AI) community (Hayzelden & Bigham (eds.), 1999; Hayzelden & Bourne (eds.), 2000). Agent-based techniques have been applied to provide intelligent control and management in a variety of settings and at varying points in communication systems (Appleby & Steward, 1994; Arvidsson et al., 1999; Bigham et al., 2000).

The dynamic configuration capabilities of Multi-Protocol Label Switching (MPLS) offer the chance to use agents to respond both reactively and proactively to changes in network topology, traffic conditions and even QoS requirements. MPLS provides a mechanism through which the flexibility of IP routing can be enhanced using a facility to set up virtual paths for certain classes of traffic. Packets can be re-directed onto the next node on their route without having to travel up to the IP layer in the stack. This saves on processing costs at the IP layer and provides a mechanism by which layer-3 least-cost routing algorithms can be bypassed, allowing traffic to be transported across the network according to other criteria.

The aim of this research is to demonstrate the advantages that may be gained through applying agent technology to networks equipped with the MPLS forwarding mechanism and the CR-LDP signalling capability. Through the use of different classes of service, we show that the agents in the network can negotiate and successfully route the prioritised traffic both rapidly and effectively, and respond intelligently to changes in traffic that may limit their ability to handle real-time traffic. To this end, this paper reports on work in progress on the development of a test-bed that will be used to investigate the efficacy of agent-based control of a MPLS aware network and that has the flexibility to use multiple criteria to route traffic.

Test-Bed Components

The test-bed builds on a Linux-based MPLS forwarding mechanism developed by (Fraser, 2000). This is installed in the kernel of each Linux switch so that traditional IP routing can be bypassed by

MPLS packets assigned to specific Forward Equivalence Classes (FECs). The MPLS forwarding mechanism is configured using the Constraint-based – Routing Label Distribution Protocol (CR-LDP); this protocol is controlled by a layer separate to and above the IP routing layer, developed as part of this project. The CR-LDP layer is instructed by the agent layer; the agent layer is a separate process that determines the routes, or Labelled Switched Paths (LSPs), that are set up using CR-LDP. Successful set up of these routes leads to the update of MPLS forwarding tables. An overview of these components is shown in Figure 1.

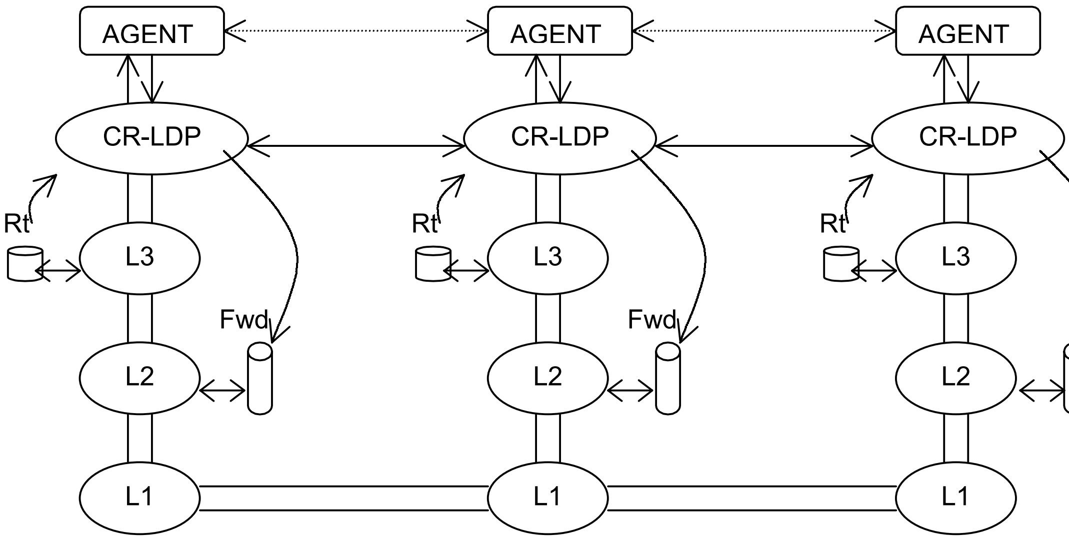

Figure 1. MPLS Aware Network using CR-LDP with Agent Layer

<u>MPLS Forwarding</u>

The MPLS protocol, as defined in (Rosen, Viswanathan & Callon, 2001), consists of a forwarding mechanism which forwards packets through the MPLS domain and a signalling component for configuration of the network. MPLS works in conjunction with routers within the domain called Label Switched Routers (LSRs). A packet enters the MPLS domain via an ingress LSR. Here the packet is assigned to a FEC, which is a grouping of packets with some common properties and defines how packets are to be treated within the MPLS domain. The assignment of packets to FECs can be based on any criteria including protocol, source address, size, etc.

Once a packet has been assigned to the correct FEC, an identifier called a Label is used to identify which FEC a packet belongs to. The packet is subsequently forwarded by LSRs based on its label. A LSR receiving a labelled packet performs a lookup of its forwarding table to determine the next hop for the packet. The forwarding table maps incoming labelled packets on an interface with an outbound interface for the packet. The forwarding table has only one entry for each labelled packet.

Packets not matching any FECs are not lost but simply routed via conventional IP. LSRs will usually be able to forward MPLS and conventional IP packets, allowing the MPLS domain to coexist with traditional IP routing.

CR-LDP Signalling

The CR-LDP protocol, as defined in (Andersson et al, 2001; Jamoussi, (ed.) 2001), enables LSRs to set up LSPs using a source-routing mechanism. When a LSR wishes to initiate or change a LSP for a FEC, all LSRs along the new (and old) route must reconfigure their own MPLS forwarding table to take account of the new LSP. For the purposes of this paper, only a subset of the CR-LDP and LDP signals were implemented, namely Label Request, Label Mapping, Label Release and Notification. In particular, the Hello procedures for discovering other LSRs were not implemented since it is assumed that the MPLS domain is already established.

Agent Layer

The agent layer is composed of one (autonomous) agent per network node or LSR. Each agent uses its own resource management algorithms and its current beliefs about the network to make decisions about how and when to override the fixed IP routing. The agent's beliefs are updated periodically through direct agent-to-agent communication (though this could be implemented through the use of new Type Length Values or TLVs) and through monitoring the state of the local switch. The agent has various responsibilities and these are split between four modules as shown in Figure 2.

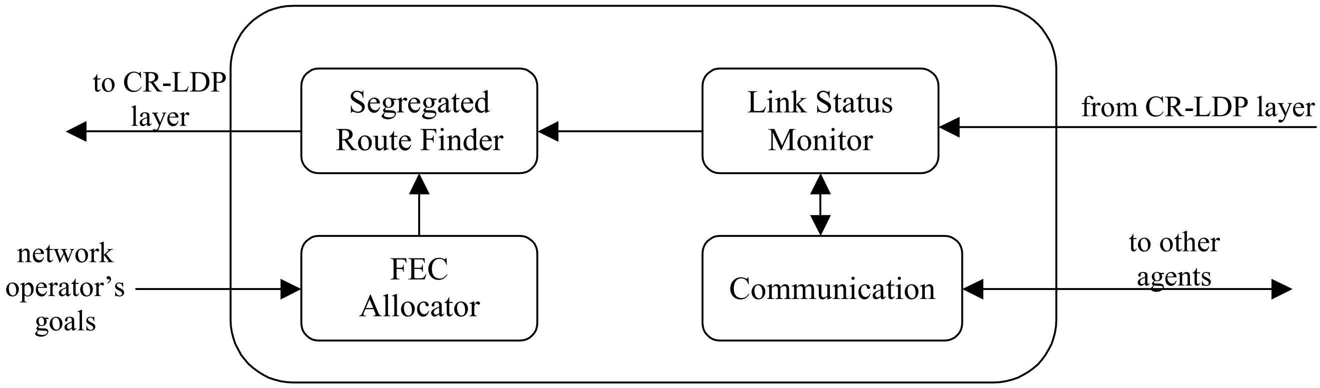

Figure 2. Agent Functions

In this preliminary work, we focused mainly on the agent situated at the ingress node of a MPLS aware network, and assume that all traffic is destined for the egress node. In this scenario, the other agents act purely as monitors of their own resources (e.g., link capacity and utilisation) and communicate only with the ingress node agent. More generally, all agents in the network may act both as ingress and monitor agents; however, we leave this aspect for future work.

The ingress node agent's goal is to provide routes for each FEC based on its beliefs about the network state. More specifically, given a number of FECs, it attempts to segregate the traffic based on their priorities, assigning the most favourable route to the highest priority traffic. This could be useful, for example, if it were desirable to divert best-effort traffic (e.g., ftp) away from highly utilised routes to give higher priority traffic (e.g., VoIP) a better QoS. In principle, different FECs may have different metrics by which they measure the relative desirability of routes; in this case, the route finding algorithm will use the particular metric selected for that FEC.

The agent maintains its beliefs about the state of the network within its Link Status Monitor. These beliefs are formed through monitoring and updating its own metrics, and receiving reports on those of other agents via its Communication module. Should it become aware of significant changes in its own or the network state, the Segregated Route Finder module is activated.

The FEC Allocator & Labeller module is responsible for deciding to which packets MPLS forwarding will be applied and assigning their FECs and labels. In this paper, we do not consider how this is determined, though clearly it depends on the network operator's requirements and QoS commitments; should these change, the Segregated Route Finder module will again be activated.

Segregated Route Finder Algorithm

The algorithm used to discover the segregated routes is based on a uniform cost search (Russell & Norvig, 1995), for which the first route discovered is guaranteed to be optimal. The inputs to the algorithm are the FEC that requires routing, along with its priority; higher priority existing LSPs currently assigned to FECs; and the agent's beliefs in the form of the link status table for the metric to be optimised. Pseudo-code for the algorithm is given in Figure 3.

```
Inputs:     FEC priority, higher priority FEC routes, link table for metric.
Output:     An optimally segregated path for the FEC.

   1. Construct paths from start node to next reachable nodes and insert
      into prioritised queue.
   2. While queue is not empty:
         a. Select and remove first path from queue.
         b. If this path leads to destination, return path.
         c. Else extend path from end node to next reachable nodes and
            insert into prioritised queue.
   3. If queue is empty then no path exists.
```

Figure 3. Agent Segregated Route Finding Algorithm

The algorithm works by primarily preferring paths that are not already in use by higher priority FECs, and only secondarily optimising for the metric. To achieve this, the agent maintains an additional link status table that records the highest priority FEC currently using each link. The head of the prioritised queue is always the least 'cost' path to its end node; therefore, if this is the destination node, it is the optimal path according to our criteria.

Note that, routing a FEC may cause lower priority FECs to require re-routing; however, higher priority routes remain unchanged. Thus, after a new LSP has been found for some FEC, the algorithm must be re-applied to lower priority FECs, in priority order. The CR-LDP layer is activated when changes to the LSPs have occurred.

Results

The completed integrated system was tested for various network topologies and different path costs (arbitrary costs were assigned to links for the purposes of validation). The agent-based system successfully segregates traffic over the MPLS network and allows reservation of network resources, taking account of path costs. Figure 4 shows a sample network of 4 nodes and links costs. Three FECs were defined (in order of priority) for HTTP, TELNET and SMTP packets. The system successfully found and set up three routes from Node 1 to Node 4. For HTTP ($1 \rightarrow 2 \rightarrow 4$), for TELNET ($1 \rightarrow 2 \rightarrow 3 \rightarrow 4$) and for SMTP ($1 \rightarrow 2 \rightarrow 3 \rightarrow 4$). Note that the two lowest priority FECs travel along the same route and all FECs travel along $1 \rightarrow 2$, through necessity.

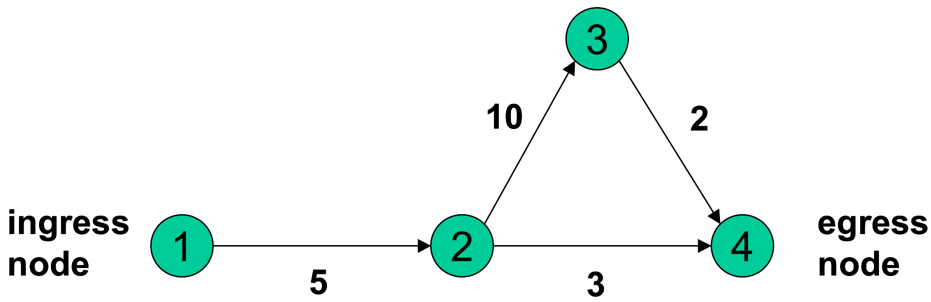

Figure 4: Sample Network with Link Costs

The agent-based system learns the network topology only through information it receives from the other (in this case) monitor agents. Thus changes to network topology would automatically be taken into account in the MPLS routing.

Conclusion

We have presented the details of an agent-based system that can perform traffic engineering using MPLS forwarding and CR-LDP signalling. The system segregates traffic for FECs where possible, according to their priority and the relevant metric. The system has been fully developed as a test-bed.

In future work, the test-bed will be used to investigate the system performance of the agent-managed traffic-engineering scheme relative to traditional IP routing for service delivery. Performance metrics based on packet latency and average packet loss will be used, and other suitable metrics may be considered. The performance loss in delivering lower priority packets will also be used to examine how concessions for one class of traffic affect the other, lower priority classes.

References

Andersson, L, Doolan, P., Feldman, N., Fredette, A. and Thomas, B (2001) LDP Specification, RFC 3036, www.ietf.org.

Appleby, S. and Steward, S. (1994) Mobile software agents for control in telecommunications networks. *BT Technology Journal*, **12:**104–113.

Arvidsson, A., Jennings, B., Angelin, L. and Svensson, M. (1999) On the use of Agent Technology for IN Load Control. In: *Proceedings of 16th International Teletraffic Congress*.

Bigham, J., Cuthbert, L.G., Hayzelden, A.L.G. & Luo, Z. (2000) Agent Interaction for Network Resource Management. *Interoperable Communication Networks*, **2(1)**.

Fraser, K. (2000) MPLS on Linux (version 2.2). Cambridge University. www.cl.cam.ac.uk/Research/SRG/netos.html/netx/index.html

Hayzelden, A.L.G. & Bigham, J. (eds.) (1999) *Software Agents for Future Communication Systems.* Springer-Verlag.

Hayzelden, A.L.G. & Bourne, R.A. (eds.) (2000) *Agent Technology for Communication Infrastructures*. J. Wiley.

Jamoussi, B. (ed.) (2001) Constraint-Based LSP Setup using LDP. Internet-Draft http://search.ietf.org/internet-drafts/draft-ietf-mpls-cr-ldp-05.txt

Jennings, N.R. & Wooldridge, M.R. (1998) *Agent Technology Foundations, Applications and Markets*. Springer-Verlag.

Kelly, F. P. (1997) Charging and Rate Control for Elastic Traffic. *European Transactions on Communications*, **8:**3337.

Rosen, E, Viswanathan, A. and Callon, R. (2001) Multiprotocol Label Switching Architecture, RFC 3031, www.ietf.org

Russell, S. and Norvig, P. (1995) *Artificial Intelligence: A Modern Approach*. Prentice Hall.

Weiss, G. (ed.) (1999) *Multiagent Systems*. MIT Press. Cambridge, MA.