Designing Hypergraph Layouts to GMPLS Routing Strategies

Jean-Claude Bermond\textsuperscript{1}, David Coudert\textsuperscript{1}, Joanna Moulierac\textsuperscript{1}, Stéphane Pérennes\textsuperscript{1}, Ignasi Sau\textsuperscript{1,2}, and Fernando Solano Donado\textsuperscript{3}

\textsuperscript{1} Mascotte joint project, I3S(CNRS-UNS) INRIA, Sophia-Antipolis, France
\textsuperscript{2} Applied Mathematics IV Department of UPC, Barcelona, Spain
\textsuperscript{3} Institute of Telecommunications, Warsaw University of Technology, Poland

Abstract. All-Optical Label Switching (AOLS) is a new technology that performs packet forwarding without any Optical-Electrical-Optical (OEO) conversions. In this paper, we study the problem of routing a set of requests in AOLS networks using GMPLS technology, with the aim of minimizing the number of labels required to ensure the forwarding.

We first formalize the problem by associating to each routing strategy a logical hypergraph whose hyperarcs are dipaths of the physical graph, called tunnels in GMPLS terminology. Such a hypergraph is called a hypergraph layout, to which we assign a cost function given by its physical length plus the total number of hops traveled by the traffic. Minimizing the cost of the design of an AOLS network can then be expressed as finding a minimum cost hypergraph layout.

We prove hardness results for the problem, namely for general directed networks we prove that it is \textit{NP}-hard to find a $C\log n$-approximation, where $C$ is a a positive constant and $n$ is the number of nodes of the network. For symmetric directed networks, we prove that the problem is \textit{APX}-hard. These hardness results hold even is the traffic instance is a partial broadcast. On the other hand, we provide an $O(\log n)$-approximation algorithm to the problem for a general symmetric network. Finally, we focus on the case where the physical network is a path, providing a polynomial-time dynamic programming algorithm for a bounded number of sources, thus extending the algorithm given in [1] for a single source.

1 Introduction

All-Optical Label Switching (AOLS) \cite{9} is an approach to route packets transparently and all-optically, thus allowing a speed-up of the forwarding. This very promising technology for the future Internet applications also brings new constraints and new problems. Indeed, since the forwarding functions are implemented directly at the optical domain, a specific correlator (device) is needed

\footnote{This work has been partly funded by the European project IST FET AEOLUS and the project “Optimization Models for NGI Core Network” (Polish Ministry of Science and Higher Education, grant N517 397334).}
for each optical label processed in the node. Therefore, it is of major impor-
tance to reduce the number of employed correlators in every node, implying a
reduction in the number of labels (as referred in the rest of the paper) that are
going to be used by the traffic. Due to its flexibility as a control plane and to the
fact that it handles traffic forwarding, the Generic MultiProtocol Label Switch-
ing (GMPLS) is the most promising protocol to be applied in AOLS-driven
networks.

In GMPLS, traffic is forwarded through logical connections called Label
Switched Paths (LSPs). When GMPLS is used with packet-based network, pack-
quets are associated to LSPs by means of a label, or tag, placed on top of the header
of the packet. In this way, routers - called Label Switched Routers (LSRs) - can
distinguish and forward packets.

The GMPLS standards allow packets to carry a set of labels in their header,
conforming a stack of labels. Even though a packet may contain more than one
label, LSRs must only read the first (or top) label in the stack in order to take
forwarding decisions. This helps to reduce both the number of labels that need to
be maintained on the core LSRs and the complexity of managing data forwarding
across the backbone.

Stacking labels and label processing, in general, are standardized by the fol-
lowing set of operations that an LSR can perform over a given stack of labels:

- **SWAP**: replace the label at the top by a new one,
- **PUSH**: replace the label at the top by a new one and then push one or more
  onto the stack, and
- **POP**: remove the label at top in the label stack.

The labels stored in the forwarding table are significant only locally at the node
and swapped all along the LSP (see Fig. 1).

Solutions deployed by GMPLS for reducing the number of labels are label
merging [4, 11, 13] (not discussed here) and label stacking [12, 15]. With label
stacking, when two or more LSPs follow the same set of links, they can be
routed together “inside” a higher-level LSP, henceforth a *tunnel*. In order to
setup a tunnel, multiple labels are placed in the packet’s header.

Fig. 1 represents the general operations needed to configure a tunnel with the
use of label stacking. At the entrance of the tunnel, λ PUSH are performed in
order to route the λ units of traffic through the tunnel. Then, only one operation
(either a SWAP or a POP at the end of the tunnel) is performed in all the nodes
along the tunnel, regardless of λ. In this figure, a stack of size 2 is used to route
the λ LSPs in one tunnel from node A to node E. The top label l is swapped and
replaced at each hop: by l₁ at node B, by l₂ at node C, and is finally popped at
node D. The λ units of traffic, at the exit of the tunnel at node E can end or
follow different paths according to their bottom label kᵢ, for all i ∈ {1, 2, ..., w}
in the stack.

A consequence of the way in which the GMPLS operations can be configured
at LSRs is the following: traffic can enter in any node of a tunnel but can exit
in only one point, the last node of the tunnel. In other words, when some traffic
is carried by a tunnel, it follows the tunnel until its end.