The Resource Reservation Protocol for New Network Concepts

Sebastian Meier, Joachim Scharf and Sebastian Gunreben

Institute of Communication Networks and Computer Engineering (IKR), University of Stuttgart
Pfaffenwaldring 47, 70569 Stuttgart, Germany

The substantial evaluation of new network concepts requires real protocol implementations. We advocate the usage of a common and standard compliant GMPLS control plane framework as a basis for custom extensions and modifications. Our contribution to this framework is a RSVP-TE implementation. This paper presents the feature list, the basic architecture and possible application areas of this implementation.

Introduction

Today's transport networks are often based on different technologies, e.g. due to heterogeneous requirements in access and core networks. In former days, such different network technologies required separate and potentially non-compatible control planes. However, the operation of multiple control planes is neither simple nor cost efficient.

The Generalized Multi-Protocol Label Switching (GMPLS, [1]) framework enables control of transport networks irrespective of differing technologies and has therefore gained a lot of attention in the last years. It relies on a set of interacting protocols and elements as shown in Figure 1. The Internet Engineering Task Force (IETF) specifies most of these protocols and elements.

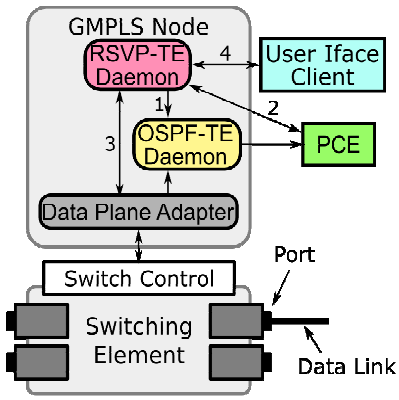

Figure 1. GMPLS Node: Components and Interfaces

The Open Shortest Path First protocol with Traffic Engineering and GMPLS extensions (OSPF-TE, [2]) enables network topology discovery and distribution. OSPF-TE provides topology information to a Path Computation Element (PCE, [3]), which performs constraint based path computation for traffic engineered paths (TE-Paths). The Resource Reservation Protocol with Traffic Engineering and GMPLS extensions (RSVP-TE, [4]) enables TE-Path maintenance consisting of setup, management and teardown. This requires access to the underlying switching element for bandwidth reservation and forwarding table configuration. The DPA provides a technology independent interface for this task. The IETF neither standardizes the DPA nor its interfaces, which might be device specific or rather generic, e.g. in case of OpenFlow [5].

Resource Reservation Protocol

For TE-Path setup, a sender as shown in Figure 2 sends a RSVP-TE *PATH* message towards a receiver. A PCE determines the path of this message through the network (source routing). The message announces the desired quality of service parameters, data plane interfaces and the encoding of the traffic transported by the TE-Path. Upon reception, each node along the path checks whether it can fulfill the request and forwards the message to the next hop. Once the *PATH* message arrives at the receiver, it triggers the creation of a *RESV* message, which is sent back towards the sender. The *RESV* message triggers the resource reservation in each node and contains the labels for sending traffic along the TE-Path. Once the *RESV* message arrives at the sender, TE-Path setup is complete.

As RSVP-TE relies on soft state, periodic *PATH* message retransmissions ensure soft state refreshment in each node and thus maintain TE-Paths. Changes in the content of the periodically send *PATH* messages allow rerouting, bandwidth adjustment and further modifications of TE-Paths. *PATHTEAR* messages trigger explicit TE-Path teardown. Alternatively, stopping the periodic state refresh will result in TE-Path removal.

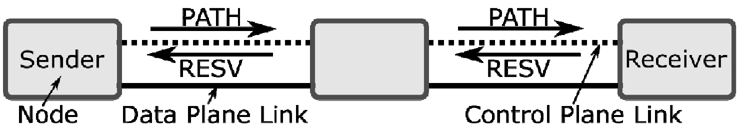

Figure 2. RSVP-TE Signaling

Towards a Standard Compliant GMPLS Implementation

Our major contribution to a GMPLS control plane implementation is a RSVP-TE implementation including the most important TE and GMPLS extensions as specified in RFC 2205, 3209 and 3473. This includes the separation of control plane – which is used for signaling – and data plane. As depicted in Figure 1, our implementation features slim interfaces towards the OSPF-TE daemon (1), the PCE (2) and the DPA (3). This modular architecture allows simple substitution of these components by other implementations. Especially the generic DPA supports arbitrary technologies and strictly separates the protocol implementation from device specific code. Technology or device specific modules that are attached to the DPA control underlying switching elements. For increased portability, we implemented the RSVP-TE daemon in Java. Configuration files allow configuration and parameterization of the daemon. For instance, adjustments to RSVP-TE parameters such as timeouts are possible. Furthermore, it is possible to enable additional protocol extensions, e.g. overhead reduction capabilities as specified in RFC 2961. For control and management, a user interface (4) allows setup, modification and teardown of TE-Paths. Additionally, the user interface provides an overview of TE-Paths, which are maintained by a RSVP-TE node.

New Network Concepts

Besides the aforementioned common control plane tasks, new network concepts impose additional requirements on a GMPLS control plane. We shortly introduce two research topics, which require new or modified control plane functionalities and may benefit from a standard compliant GMPLS control plane implementation.

Bypassing

With respect to performance and energy consumption in a multilayer network, the most efficient way to switch traffic is in the lowest layer (e.g. an optical layer). However this paradigm contradicts resource savings due to traffic grooming, which itself requires

switching in the upper layers. For such grooming the node has to handle a specific amount of transit traffic in the upper layer.
In case of variable traffic volumes between nodes, the amount of transit traffic and therewith the optimum switching configuration (lower vs. upper layer) might change over the time. The idea of (optical) bypassing is to disburden a node from too much transit traffic in the upper layer by establishing a bypass in a lower layer. However it is also possible to teardown low utilized paths by relocating their traffic. This process potentially generates more transit traffic.
An implementation of such a bypassing mechanism would heavily use features of the GMPLS control plane and especially RSVP-TE (e.g. for creating the bypass) but also needs some extensions. For example there is currently no element with the necessary information for making decisions about the bypasses.

Virtualization

In future networks, network virtualization enables multiple virtual nodes on a common physical node sharing the link resources in between. For management of these virtual networks, protocols should be aware of virtualized resources and automatically establish these virtual nodes and links.
As GMPLS already supports forward adjacencies, only minor extensions to GMPLS are necessary to enable creation and management of virtual links. These adjacencies are re-announced by OSPF-TE and appear as any other link in the network topology. In contrast to virtual links, support for virtual nodes requires several extensions to GMPLS. Regarding RSVP-TE, signaling needs extensions for virtual node parameterization, creation, relocation and teardown.

Conclusion and Outlook

In this paper we presented the architecture and features of our RSVP-TE implementation for usage in a GMPLS framework. We outlined possible extensions to that framework for evaluation of new network concepts. Our next steps encompass implementation of a DPA module for control of commercial router hardware. In addition, we plan to carry out interoperability tests as well as performance evaluation.

Acknowledgements

This work was partially funded by the Federal Ministry of Education and Research of the Federal Republic of Germany (Förderkennzeichen 01BK0918, G-Lab).

References

[1] E. Mannie, “Generalized Multi-Protocol Label Switching (GMPLS) Architecture”, Request for Comments: 3945, 2004

[2] K. Kompella and Y. Rekhter, “OSPF Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)”, Request for Comments: 4203, 2005

[3] A. Farrel and J.-P. Vasseur and J. Ash, “A Path Computation Element (PCE)-Based Architecture”, Request for Comments: 4655, 2006

[4] L. Berger, “Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions”, Request for Comments: 3473, 2003

[5] N. McKeown and T. Anderson and H. Balakrishnan and G. Parulkar and L. Peterson and J. Rexford and S. Shenker and J. Turner, “OpenFlow: Enabling Innovation in Campus Networks”, 2008