Tools for Application-tailored Network Engineering

Helge Backhaus, Roland Bless, Denis Martin, Martin Röhricht, Hans Wippel, Martina Zitterbart
Institute of Telematics, Karlsruhe Institute of Technology (KIT)
{backhaus, bless, martin, roehricht, wippel, zitterbart}@kit.edu

I. Introduction

In contrast to today's one-size-fits-all Internet architecture, we consider a pluralist Future Internet scenario within the G-Lab project. In this scenario Service Providers (SPs) construct their own *application-tailored* networks to deliver their service to end-customers. Such networks are designed exclusively for certain applications or use cases. Thus, the structure of the network and the protocols running inside it are specifically tailored to the SP's target service or application. To this end, we consider *network virtualization* as an enabler for the deployment and isolation of such specialized networks. In addition, application-tailored protocols are rapidly created by utilizing *protocol composition*. In this Future Internet scenario, SPs create their own network topologies, compose their specialized protocols, and then deploy, run, and manage their networks. End-systems of customers then dynamically connect to the SP's network on-demand when they access the corresponding service. In this process, the end-system attaches to the SP's virtual network and downloads specialized protocols if necessary. In addition, a general-purpose networking API delivers high-level abstractions to networking that no longer bind applications to specific protocol families – which allows any application to easily access any specialized network.

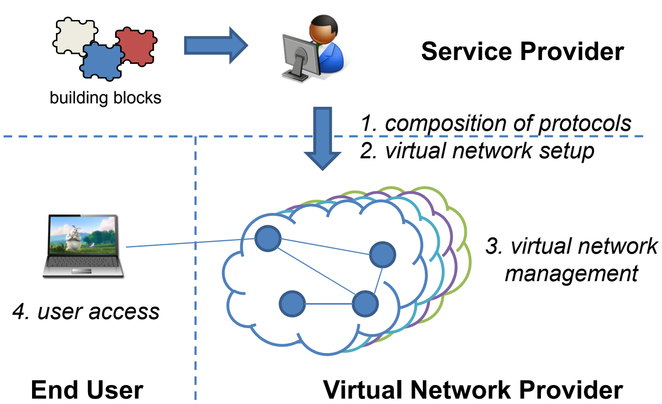

Fig. 1. Application-tailored network engineering

In this talk, we give an overview of selected, complementing work that we and other partners conducted within the G-Lab project as first steps towards such application-tailored network engineering. This work is described from the perspective of three main stakeholders (Figure 1): Service Providers (SPs) design networks tailored to their applications or services. Virtual Network Providers (VNPs) operate the underlying network infrastructure and provide SPs with virtual networks. Finally, end-users access an SP's service and thus use the respective network. After the overview, we give an outlook on open challenges and future steps.

II. Service Provider Perspective

The SP's process for creating application-tailored networks comprises the definition of topology and capacity as well as the design of a protocol family running within the network. Protocols tailored to a specific use case have the advantage to work more efficiently in that particular use case than generic protocols. For instance, special distribution patterns such as multicast can be used to optimize network traffic. By using application-level knowledge, additional optimization can be achieved by making more intelligent decisions, e. g., regarding packet drops on buffer overflows, selective retransmissions, or even in-network transcoding. Protocol composition approaches aim at reducing time, costs, and errors when constructing such optimized protocols. Within G-Lab, both run-time and design-time approaches were investigated [1]–[3]. Both types can benefit from each other, thereby allowing to find the optimal balance between performance optimization and run-time flexibility. Flexible run-time frameworks such as NENA [3] then allow for an easy deployment, the selection of the designed protocols, and a transparent usage by existing applications (see Section IV).

III. Virtual Network Provider Perspective

Virtual Network Providers operate the virtual network infrastructure and provide means to create virtual networks and manage their network resources.

Since design, creation, and deployment of virtual networks plays an important role in our Future Internet vision, appropriate tool and protocol support is necessary. With ToMaTo [4], a powerful tool for designing topologies for virtual networks exists. In order to complement this tool with an on-demand signaling solution for an automated setup of virtual nodes and virtual links, the Virtual Link Setup Protocol (VLSP) [5] can be used. VLSP is designed as a modular extension to the standardized state-of-the-art signaling protocol suite NSIS. This signaling protocol is used to combine an authenticated and on-demand setup of virtual links with the establishment of Quality-of-Service guarantees in the underlying substrate.

Network nodes potentially maintain connections to a multitude of virtualized application-tailored networks at the same time. All these networks might be based on different network architectures and operate individual protocols. Furthermore, these network connections are not static. Instead, connections can be added and removed at runtime. Management of node resources across such changing networks and architectures is a key challenge in this context. Our goal is to manage node resources that are used for active connections and protocols as well as to configure and to adapt them to application requirements. To this end, we use an hierarchical management system [6] on network nodes. On the lowest layer, the protocol-specific management is included in every managed protocol and possesses detailed knowledge about the protocol. On the next higher layer, a network-specific management with detailed network knowledge is used for every active connection to a virtual network. On the highest layer, the node-wide management possesses coarse-grained knowledge about the entire node. On every layer management information is gathered, aggregated, and passed on to the next higher layer. Additionally, the node's resources are managed in resource pools and distributed from higher layers down to lower layers.

IV. End-User Perspective

Applications of end-users access services provided by application-tailored networks via a *name-based application interface*. Connections to the appropriate virtual network and the download of necessary protocols are handled by the end-user's system.

The application interface uses (1) a globally unique naming scheme to specify content, hosts, or services, (2) interface primitives to establish communication associations as well as to send/receive data, and (3) a specification of transport service requirements without selecting a specific protocol [7], [8]. In contrast to today, names should not only identify hosts, but also services and content in general. For this reason, a globally unique naming scheme based on Uniform Resource Identifiers (URIs) is used. At the application level no network specific addresses or locators are used. Applications only work on globally unique, network and architecture independent names to identify services or content. Alongside each request specific application requirements are passed down to the network stack as well. These aim at describing different QoS and QoE properties expected from the ensuing communication. The requests can also contain additional information specifying the type of data an application is about to send, similar to a MIME type. Communication properties are represented in an XML-based RDF format, so that they are easy to extend and modify. An overview can be found in [9]. The goal of such an API is to allow for choosing appropriate network protocols and to simplify communication.

End-user applications access content or services via this API. The end-user's system must determine the network in which the content or service is located. Currently, the name space of the given URI is used to determine networks that are capable of handling requests on this URI. If the node is not connected to such a network, the node has to perform the end-user node access process [10]. During this process the node needs to select one of the available networks and then creates a virtual link to it. Additionally, it retrieves any specialized protocols that are used in the network from public repositories via a basic network. As a final step, the protocols are executed and the actual communication between the end-user application and the requested content or service takes place.

V. Outlook

The research described here aims to provide first steps towards a Future Internet scenario where Service Providers rapidly design, deploy, and operate their application-tailored networks. However, since networking becomes more and more heterogeneous and ubiquitous, network interoperability issues must be considered already during design-time of a network. For instance, custom smart-home networks should be able to interact with other networks and devices in order to allow ubiquitous control over a smart-home. Resource critical sensor systems that should allow queries from any other networked device are another example. Thus, future work needs to investigate how protocol composition can support interoperability of completely different networks. In addition, it remains to be investigated to what extent virtualization is available on resource-constrained devices and thus can be used in such cases.

References

[1] B. Reuther and D. Henrici, "A model for service-oriented communication systems", *Journal of Systems Architecture*, vol. 54, no. 6, pp. 594–606, 2008.

[2] F. Liers, T. Volkert, and A. Mitschele-Thiel, "Forwarding on Gates: A clean-slate Future Internet Approach within the G-Lab project", in *9th Würzburg Workshop on IP: Joint ITG and EuroNF Workshop "Visions of Future Generation Networks" (EuroView2009)*, Würzburg, Jul. 2009.

[3] D. Martin, L. Völker, and M. Zitterbart, "A Flexible Framework for Future Internet Design, Assessment, and Operation", *Computer Networks*, vol. 55, no. 4, pp. 910–918, Mar. 2011.

[4] D. Schwerdel, D. Hock, D. Günther, B. Reuther, P. Tran-Gia, and P. Müller, "ToMaTo – a network experimentation tool", in *Proceedings of the 7th International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TridentCom 2011)*, Apr. 2011.

[5] R. Bless, M. Röhricht, and C. Werle, "Authenticated Quality-of-Service Signaling for Virtual Networks", *Journal of Communications*, vol. 7, no. 1, pp. 17–27, Jan. 2012.

[6] H. Wippel, T. Gamer, C. Faller, and M. Zitterbart, "Hierarchical Node Management in the Future Internet", in *Proc. of 4th International Workshop on the Network of the Future (FutureNet IV)*. IEEE, Jun. 2011.

[7] D. Martin, H. Wippel, and H. Backhaus, "A Future-Proof Application-to-Network Interface", in *Proc. of the International Conference on the Network of the Future (NoF 2011)*. Paris, France: IFIP/IEEE, Nov. 2011.

[8] F. Liers *et al.*, "GAPI: A G-Lab Application-to-Network Interface", in *Proceedings of the 11th Würzburg Workshop on IP: Joint ITG and Euro-NF Workshop "Visions of Future Generation Networks" (EuroView2011)*, Würzburg, Germany, Aug. 2011.

[9] H. Backhaus, "Towards a Property and Requirement-based Application Interface for Future Networks", in *10th Würzburg Workshop on IP: Joint ITG, ITC, and Euro-NF Workshop "Visions of Future Generation Networks" (EuroView2010)*, Würzburg, Germany, Aug. 2010.

[10] H. Wippel and O. Hanka, "End User Node Access to Application-Tailored Future Networks", in *International Conference on Computer Communication Networks (ICCCN 2012)*. Munich, Germany: IEEE, Aug. 2012.