Selecting and Composing Requirement Aware Protocol Graphs with SONATE

Daniel Günther, Dennis Schwerdel, Abbas Siddiqui, Rahamatullah Khondoker, Bernd Reuther, Paul Müller
University of Kaiserslautern
Germany
{guenther, schwerdel, siddiqui, khondoker, reuther, pmueller}@informatik.uni-kl.de

I. Introduction

Nowadays, Internet applications have reached such a high complexity that the network stack concept developed decades ago is not suitable anymore. In the beginning, it was basically a simple data network. Today, many activities have moved onto the Internet, such as communication (e.g., VoIP), entertainment (e.g., podcast, webradio, TV) and education (e.g., e-learning with live streams). The layered model of the Internet has problems adapting to these varied use cases. Some of the problems encountered are increasing complexity which pushes functionality between layers, cross-layer concerns such as addressing and security, and matching mechanisms to the appropriate layer. The outlined problem will be handled by scientific Future Internet research. One approach is the Service-Oriented Network Architecture SONATE [6].

SONATE was developed in the context of, and bridges, the G-Lab projects[1]. The demonstration presents several scenarios to show how SONATE makes use of different protocol graphs based on different requirements. The next section describes how SONATE selects and composes requirement aware protocol graphs. Section III describes the proposed demonstration which uses the SONATE Framework.

II. The Service-Oriented Network Architecture

One approach of Future Internet research is to build protocol graphs from a given set of functionalities which are called functional blocks. To realize and evaluate this approach, a service-oriented network architecture was developed, namely SONATE. It increases network flexibility by providing requirement specific protocol graphs instead of today's static network stacks.

Flexible protocol graphs are made up of functional blocks. Each block implements a certain network functionality like encryption, loss reduction, or compression. From a pool of available functional blocks, appropriate blocks are selected by looking at certain criteria, including application requirements, and then connected via their ports [1] to create a requirement and network specific protocol graph. To reduce the naturally occurring complexity in creating an entire protocol graph, we use a template-based functional composition (TFC) approach.

The idea of the TFC approach is to split the functional composition process among different time-phases, i.e., design-time, deployment-time, and run-time, so that the time consuming activity is performed at the least time-critical phases, e.g., design-time. Potentially less time consuming activities are performed at run-time. The most time consuming activities are the selection of functionalities, but not actual functional blocks, and placing them in appropriate order in addition to connecting them so that they can interact with each other. To utilize the less time critical phases and still provide enough flexibility, TFC utilizes place-holders to represent the functional blocks which will be selected at run-time.

TFC may produce more than one protocol graph which fulfills the application's requirements. Selecting which of these protocol graphs to use is a Multi-Criteria Decision Analysis (MCDA) problem which we solve for this demonstration using an adaptation of the Analytic Hierarchy Process (AHP) [4]. A description language [4], [5] is used to describe the application requirements and functional block capabilities. These descriptions are necessary for the selection and composition process.

As a part of SIG FUNCOMP, a special interest group for functional composition in the German-Lab project, we created an interface called GAPI: G-Lab Application-to-Network Interface [2] which an application uses to specify its requirements to SONATE.

III. Demonstration

In our demonstration Firefox will retrieve different kinds of data, each with different requirements, and be used to show the impact of requirement-specific protocol graphs on the user experience. There are also servers which are accessible via different networks, which demonstrate adaption to different network capabilities. We have extended a Firefox plug in, which was developed in cooperation with project partners, to interconnect Firefox with the GAPI to communicate with the components in the SONATE Framework. In general, Firefox sends the requirements via the GAPI to SONATE. SONATE uses service selection and composition to select a requirement and offer appropriate protocol graph. We encode application requirements in URLs and communicate with different servers. With the help of the virtualization and emulation capabilities of ToMaTo, which was also developed as part of the German-Lab project, we realize different network link characteristics between our clients and servers.

[1]The German-Lab project and the German-Lab-Deep project are both funded by the German Federal Ministry of Education and Research (BMBF) under the grant numbers 01BK0808 (German-Lab) and 01BK0901 (German-Lab-Deep)

The utilized protocol graphs will be visualized to show decision processes within SONATE. The proposed demonstration consists of the following three scenarios:

Scenario 1: Network Aware Image Adaptation Some network applications such as image and video transfer are adaptable to the network environment. This scenario transfers a large resolution image over low as well as high data rate connections. The protocol graph will automatically adapt the quality of the image to the available data rate. This allows for low data rate devices to quickly load the (quality reduced) image. This class of applications has two main requirements: reliable transmission and adaptation mechanism.

Scenario 2: Additional Encryption Many applications require the use of some encryption mechanism to transport sensitive information over the network. This scenario assumes the application wants to transfer a large file over the network.

Such an application has two main requirements for its network communication: reliable transmission and encryption. The application may also specify the level of encryption desired, providing the trade-off between security and computation requirements.

Scenario 3: Error Correction Adaptation Network applications which require reliable transmission may also have other requirements such as latency. The loss-rate on a network connection can have a strong affect on the impact of a loss reduction mechanism [3]. For example, when a low-latency, reliable connection is required over a high-datarate connection with little loss, it may be better to use a mechanism such as forward error correction in addition to an automatic repeat request method, as many errors can be corrected without resorting to the additional latency required by retransmission.

It does not matter to the application which mechanisms are used to fulfill its requirements, instead SONATE can automatically choose an efficient method to achieve the desired results. The amount of forward error correction applied can also depend on the actual loss rate during the communication. This scenario shows how an error correction method can be adapted according to the network loss rate.

References

[1] D. Schwerdel, D. Günther, M. R. Khondoker, and P. Müller. A building block interaction model for flexible future internet architectures. In *7th Euro-NF Conference on Next Generation Internet, Kaiserslautern, Germany*, 2011.

[2] F. Liers, T. Volkert, D. Martin, H. Backhaus, H. Wippel adn E. M. Veith, A. Siddiqui, and R. Khondoker. Gapi: A g-lab application-to-network interface. In *Proceedings of the 11th Wuerzburg Workshop on IP: Joint ITG and Euro-NF Workshop on Visions of Future Generation Networks (EuroView 2011)*, 2011.

[3] Günther, D., Veith, E. MSP, and Müller, P. Extracting decision criteria by modeling retransmission and forward error correction in a future internet architecture. In *37th Euromicro Conference on Software Engineering and Advanced Applications (SEAA2011), Oulu, Finnland, 30.08.-02.09.2011*, September 2011.

[4] R. Khondoker, B. Reuther, D. Schwerdel, A. Siddiqui, and P. Müller. Describing and selecting communication services in a service oriented network architecture. In *Proceedings of the 2010 ITU-T Kleidoscope: Beyond the Internet? - Innovations for Future Networks and Services, Pune, India*, 2010.

[5] R. Khondoker, E. M. Veith, and P. Müller. A description language for communication services of future network architectures. In *Proceedings of the 2011 International Conference on the Network of the Future, Paris, France*, 2011.

[6] Reuther, Bernd and Müller, Paul. Future internet architecture - a service oriented approach. *In IT - Information Technology*, 50(6), 2008.