A Service-oriented Approach towards the Design of a Future Internet Architecture

Paul Mueller, Bernd Reuther, and Markus Hillenbrand
University of Kaiserslautern, 67653 Kaiserslautern, Germany
Integrated Communication Systems (ICSY) Lab (http://www.icsy.de)
{pmueller; reuther; hillenbr}@informatik.uni-kl.de

Overview

Driven by the demands of ever emerging applications and the capabilities of new communication networks, the Internet has become an architectural patchwork resulting in increasing complexity and unpredictable vulnerabilities. This patchwork is the result of layer violations (e.g. cross-layer design), sub-layer proliferation (e.g. MPLS at layer 2.5, IPsec at layer 3.5, and TLS at layer 4.5), feature interaction (resulting in complexity), and erosion of the end-to-end model (middle-boxes, such as firewalls, NATs, proxies, caches, etc.). This erosion of the formerly clearly layered architecture is not just because of too much functionality, but also because of lots of implicit dependencies, i.e. tight coupling of functionalities, which results in an ossified Internet.

The problems mentioned above lead us to a new design approach for a future open network architecture based on orchestrating loosely coupled services. By defining open, standardized and generic service interfaces it is possible to decouple logic from implementation and enable simple integration of new technologies and thus the implementation of future complex applications. Through the orchestration of services the architecture can adapt to changing requirements on demand. Moreover, the approach presents new design concepts for the data, control and management planes and may also introduce new communication paradigms. Therefore we will evaluate requirements for a new architecture and propose, develop and demonstrate new ideas.

Relevant Work and its Impact

In our approach of a "Service-oriented Network Architecture" (SONATE) we consider the Internet as a largely distributed software system. Therefore we address a new inter-network architecture by using a software engineering methodology. One of the most promising methodologies within software engineering today is the Service-oriented Architecture (SOA) paradigm, especially for the development of distributed software systems[1].

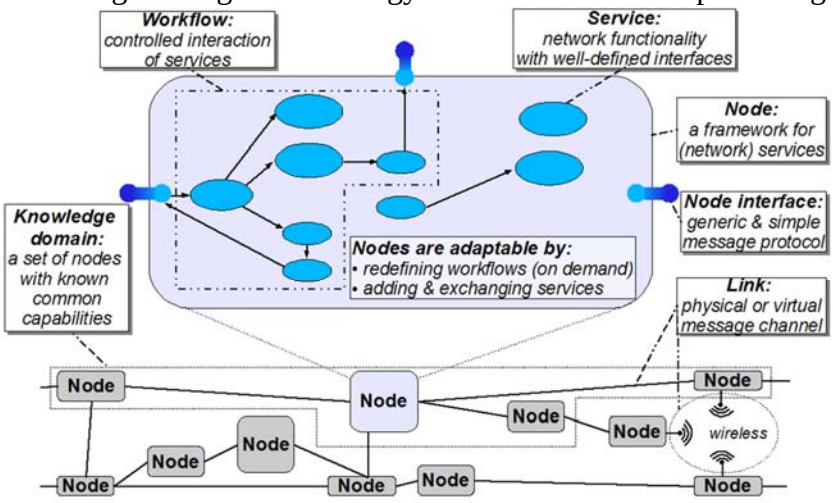

Figure 1: Basic concept of the SONATE approach

For SONATE we assume that a network is made up of interconnected nodes, whereby linked nodes use a common generic protocol. A node may implement network functionality on different levels. There may be nodes implementing low level transport functionality only, some nodes may represent a member of a peer-to-peer overlay network, while others might implement a broad range of application related functionality. In this model applications are also just nodes.

[1] Note: We consider SOA as a paradigm, and thus it does not rely on a specific technology (e.g. Web Services) and is not limited to specific application areas (e.g. enterprise information integration).

The essential building blocks of the architecture are services within the nodes (see Figure 1). Services offer self-contained functionality and must have well-defined interfaces, which hide internal mechanisms and data structures. The granularity of services should be similar to micro-protocols. For example, there may be a service for "ensuring reliable transmission" which can be implemented by different retransmission strategies. Complex functionality is provided by several interacting services (an orchestrated "communication workflow"). Moreover, it should be possible to map traditional mechanisms from TCP/IP onto this approach. Thus SONATE can also be seen as an evolutionary approach, even if it seems to be a "clean slate" approach at first glance.

Research issues within our approach are to find concepts and techniques enabling SOA principles on a network level. This includes the identification of fine-grained services with proper (explicit) interfaces and data types, so that the dependencies between services are minimized. Also a framework for managing services and enabling service interactions is required. Such a framework represents a node as shown in Figure 1. Another goal is the definition of common practices for service design; for example to enable services to be developed incrementally or adapt to capabilities of other services. Further a concept of knowledge domains should be defined (see Figure 1), to describe explicitly which nodes have common capabilities. For example having a priori knowledge which nodes can handle the same address types may improve efficiency of communication.

Our first approach is to build an experimental infrastructure for services on the network level. This infrastructure will provide mechanisms to keep states of services on a per-flow basis. A notification broker enables communication among services and there are mechanisms to manage sequential and parallel processing of services. All kinds of data sent over a network are represented as a stream of messages, which are simple "type-length-value" (TLV) structures in a first approach. This enables (nearly) arbitrary ordering of data structures and thus can be extended easily.

Using services instead of protocol layers mean replacing standardised technologies by standardised interfaces. Such interfaces can evolve much easier than protocols and thus are much more suitable to support evolutionary processes than changing protocols. The impact of such a service based architecture, which is open by design, is a seamless integration of all different communication entities. That mean the network, the communication workflow and the applications can be seen as one integral system which can incorporate new technologies and support future complex applications by design. Services on different nodes communicate by some kind of message exchange, which may result in a new communication paradigm. In contrast to packet headers messages are sent only when needed, may contain varying data, and may travel on different paths. By building an experimental infrastructure for services on a network level we aim to demonstrate the potentials of SONATE and will incrementally propose and develop these concepts.

Related Work

Our "Service-oriented Network Architecture" (SONATE) approach is influenced by several publications in the last years. As the most important can be seen the "***Role Based Architecture***" (RBA) approach from Braden et. al. (2002), the "***Service Integration, controL and Optimisation***" (SILO) approach from Dutta et.al. (2007), and the "***Recursive Network Architecture"*** (RNA) approach from Touch et.al. (2006).