Mediation between Service and Network Composition

Abbas Ali Siddiqui*, Christian Henke†, Daniel Günther*, Paul Müller*, Thomas Magedanz† * University of Kaiserslautern, Postbox 3049 67653 Kaiserslautern, Germany
Email: {siddiqui,guenther,pmueller}@informatik.uni-kl.de †Technical University Berlin, Straße des 17. Juni 135, 10623 Berlin, Germany
Email: {c.henke,tm}@tu-berlin.de

I. Introduction

In todays Internet the network stack is divided into distinct layers which can be implemented by different protocols. Each layer offers a service to directly adjacent layers. Although this crisp and robust design has proven its advantages (functionality scoping, stability) there are also some disadvantages of this architecture. Protocols on different layers implement the same functionality (e.g. IP and TCP Checksum), the physical layer is not aware of the application and cannot adapt error correction or coding (e.g. for multimedia over wireless). Besides this, there are also issues like cyberspace tussles [10], the increasing mobility of the end-hosts and the ossification of the Internet due to the increasing complexity of the protocol interdependencies that lead to some new architecture proposals for a Future Internet. One proposal is functional composition which decomposes the functionalities of the network stack in different functional blocks. These functional blocks are loosely coupled and provide means to exchange information between functionalities of different levels. Many projects (e.g. ANA [4], RBA [3], 4WARD [1], AutoI [5], Self-Net [2], Net-Silo [7], RNA [8], Network Service Architecture [6], SONATE [11]) have addressed this approach from different angles to find a best solution for a flexible future Internet architecture which can cope with the requirements of futuristic trends. A functional composition architecture leads to a two-layer architecture - the applications running on top at the service layer and the network which is composed based on the application requirements. This separation is still valuable because an application designer should not know and compose the network functional blocks by himself but explicitly state the abstract requirements of the application, e.g. encryption and QoS (maximum delay, maximum loss). Nevertheless, there should be a feedback of the network if these requirements can be met or not, so that the application itself can also adapt to the network (like encryption on service level, use another media encoding, or choosing another destination which may be more reachable). For this purpose we propose in this paper a cross layer mediator that negotiates and exchanges information between the two layers. We will explain the main concepts and advantages of this cross-layer mediator.

II. Mediation

Some mediation techniques have been proposed by different researchers e.g. Protocol Negotiation [12], Dynamic and adaptive service brokering [9]) or within different scope (e.g. among web-services). We propose to use a mediation approach for negotiation between service and network layer in a functional composition approach which increases the flexibility of the architecture because it provides feedback from the both layers and enables adaptation of the network based on the application and vice versa. The following text will describe the main aspects of the mediation approach.

A. Why Mediation is Required

Contrary to the current Internet, functional composition makes the architecture flexible enough to accommodate a variety of network services with minimal management or configuration effort because functional blocks are loosely coupled. Furthermore functional composition allows the network to be dynamically and automatically composed based on the application specific requirements. Nevertheless, the exchange of requirements can vary in complexity. A scenario could be that an application selects pre-composed functionality and sends a request stating its requirements. The composition process will then be executed at the network layer without any further feedback from the application. In a complex scenario a network composition process may not provide the exact functionality requested by the application instead it tries to negotiate with the application some of the requirements parameters, upon which the application can adapt (e.g. multimedia codec in case of a lossy network). The network may then provide functionality which is a superset or a subset of the requested functionality.

B. Challenges of Mediation

Mediation between service and network layer has its own challenges which we will further investigate in our work.

1) Placement of Functionality: As mentioned above mediation will be done between two layers, both of which have a set of functionality. It is likely that some of the functionality will be available at both layers. A decision

must be taken from which level this functionality should be used - if this resolution will not be performed then it may cause redundancy of a functionality, mediation can assist to resolve this issue. Different policies could be presented for a solution such as service layer searches for available functionality within a network layer and in case the service layer also posses the functionality it can decide whether to ask for the functionality or to negotiate with the network layer to select a better option.

2) Conflict between functionality: Mediation will play an important role when there is a conflict among network functionality requested by an application, or between selected services at the service layer and requested functionality at the network layer. Some of these cases, where conflict can appear, are listed below.

- When two functionalities can not work properly together and one functionality must be excluded (e.g. an answering machine service and a forwarding service)
- When some functionality has been added by the service layer which forbids functionality at network layer to work properly (e.g. an encryption service at service layer and a transcoding service in the network).

3) Common Understanding for Negotiation: Mediation requires a common negotiation scheme between two layers. It helps to exchange understandable messages to come to a decision via negotiation. Messages with a generic nature could be proposed here to exploit its ability to be adaptable to various conditions without having any drastic changes.

C. Explicit and Implicit Negotiation

To perform a mediation process it is required to have some kind of a negotiation. Two kinds of negotiation are being suggested here.

- Implicit
- Explicit

In implicit negotiation, a functional composition process will not further communicate to the service layer and decide on its previous experience for the requested functionality or pre-composed services will be used for the purpose. In case the same requirements are posed which have previously been negotiated then the previously composed services will be reused rather than commencing the renegotiation, as shown in (Fig. 1). An implicit negotiation is efficient as no time will be invested for a negotiation.

On the other hand in an explicit negotiation a network composition process can be asked for further functionalities from the service layer when requirements have not been priorly negotiated or the network state changes. The negotiation process results in a set of suitable services and functional blocks that will be selected and composed in a work-flow. The disadvantage of an explicit negotiation is that it requires additional overhead to make a decision.

D. Advantages and Disadvantages of Mediation

A mediation process has advantages and disadvantages, some of them are listed below.

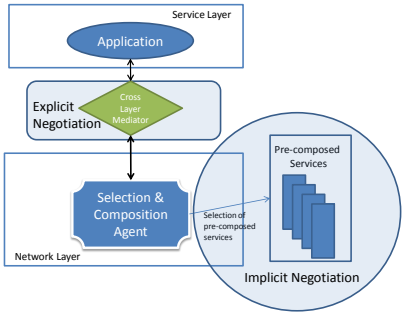

Fig. 1. Explicit and Implicit Negotiation

- In-dependency: A mediation helps two layers (i.e. service and network layer) work independently.
- Flexibility: A mediation helps an architecture to exploit the flexibility of a composition process. Without having a negotiation it is likely that application requirements will remain unsatisfied.
- Help in making a decision: Via mediation a negotiation can be performed to achieve a suitable outcome for the application.
- Performance Drain: Additional overhead in terms of negotiation has an impact of on overall performance of the architecture.

III. Conclusion

In this paper a mediation process has been proposed which provides more flexibility in a functional composition architecture. Instead of only following a top down approach where the application tells the network its requirements, the network and application can interact to find a suitable solution. In a functional composition approach, certain application level services are likely to move down to the network level. Mediation help determine where functionalities should be executed.

References

[1] 4WARD EU Project. http://www.4ward-project.eu/
[2] SelfNet Project. http://www.ict-selfnet.eu
[3] Robert Braden, Ted Faber, and Mark Handley. From protocol stack to protocol heap: Role-based architecture (2003).
[4] Autonomic Network Architecture (ANA). http://www.ana-project.org
[5] Autonomic Internet. http://ist-autoi.eu/autoi/index.php.
[6] Sivakumar Ganapathy and Tilman Wolf. Design of a network service architecture (August 2007).
[7] R. Dutta, G.N. Rouskas, I. Baldine, A. Bragg, and D. Stevenson. The Silo architecture for services integration, control, and optimization for the future internet (June 2007).
[8] Venkata Pingali, Joseph D. Touch, Yu-Shun Wang. A recursive network architecture (2006).
[9] Howard Foster, Arun Mukhija, David S. Rosenblum and Sebatian Uchitel. A Model-Driven Approach to Dynamic and Adaptive Service Brokering using Modes.
[10] David D. Clark, Karen R. Sollins, John Wroclawski, Robert Braden. Tussle in Cyberspace: Defining Tomorrows Internet
[11] Paul Mueller, Bernd Reuther. Future Internet Architecture - A Service Oriented Approach, it - Information Technology, Jahrgang 50 (2008)
[12] Bryan Ford, Janardhan Iyengar. Efficient Cross-Layer Negotiation. ACM Hot Topics in Networks VIII, New York, October 2009