Ariba: A Framework for Developing Decentralized Services

Christian Hübsch, Christoph P. Mayer, Oliver P. Waldhorst
Institute of Telematics, Universität Karlsruhe (TH), Germany

{huebsch,mayer,waldhorst}@tm.uka.de

Abstract

Developing new network services in the Internet is complex and costly. This high entrance barrier has prevented new innovation and has stuck the Internet as being mainly client/server-based. End-system based decentralized services are cheaper but more complex than centralized systems. The *ariba* framework aims at easy development of such decentralized services without infrastructure support. In this paper we present the *ariba* framework and show how it can facilitate development of decentralized services through self-organization and its network abstraction.

1 Introduction

Most services in today's Internet are built in a centralized way, even though they are used in a peer-to-peer style communication that would typically be built using a decentralized system. Deploying such a centralized system creates high cost for servers and infrastructure and induces high load on those components. Developing a decentralized system in contrast has a complexity several orders of magnitude higher. In particular, such decentralized approaches need to cope with middleboxes, host mobility, and protocol or device heterogeneity. The Internet today is therefore built around web-browsers and client/server-based schemes. New services are deployed by large companies, backed by a huge amount of servers and infrastructure. Only rarely, decentralized and cost-efficient systems successfully make their way into the market. We believe that minimizing complexity of decentralized service development will foster new innovative services due to lowered entrance cost and complexity.

Using a decentralized approach implemented via *ariba* [2] enables an overlay-based, flexible, and low-cost solution without dedicated infrastructure and reduced bandwidth consumption. *ariba* spans an overlay-based network upon decentralized peers and hides networking issues like middleboxes, host mobility, and heterogeneity to allow for seamless service development using an invariant network abstraction. We give an architectural view of *ariba* in Section 2, and detail on its interfaces in Section 3. Finally, Section 4 concludes and provides an outlook into current development efforts.

2 Ariba Framework

The *ariba* framework has been developed to allow for seamless service development using the concept of *spontaneous virtual networks*, called *SpoVNets* [3]. A SpoVNet is a decentralized overlay-based virtual network that acts as an abstraction to allow for easy service development. An instance of such a SpoVNet is created spontaneously without infrastructure support on a per-application basis. The main components of a SpoVNet instance are the *base-overlay* and the *base-communication* that are provided by the *ariba* component, as shown in Figure 1. First, the base-overlay acts as a basic routing infrastructure—using key-based routing on overlay identifiers—for control and maintenance messages. Below this overlay, the base-communication provides different transport modules for communication (e. g. TCP, UDP, IPv4, IPv6, RFCOMM). Control traffic is routed through the base-overlay to maintain a self-organizing structure comprising all participating overlay nodes in the system. When two such nodes can't communicate directly due to different protocols (e. g. IPv4 vs. IPv6) or network issues like NAT, *ariba* establishes routes through other nodes in the SpoVNet to provide reachability.

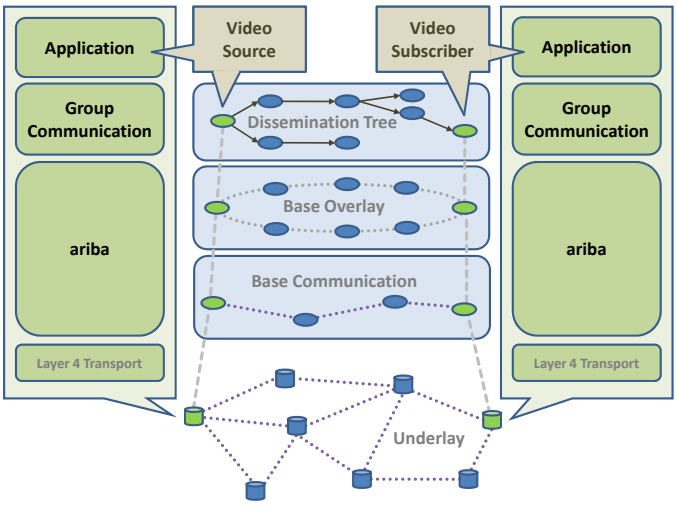

Figure 1: Architectural view of *ariba*

Two main concepts foster the easy service development in *ariba*: *NodeID*s and *LinkID*s. The *ariba* framework incorporates an identifier/locator-split: Nodes are addressed using NodeIDs that are invariant to mobility, NAT, or multi-homing. A NodeID maps to a set of locators that describe physical network reachability of that node in terms of layer 2/3/4 protocols. In *ariba* a service that builds up communication with another node requests establishment of a *virtual link* that is identified through a *LinkID*. Again, the LinkID is invariant to mobility, NAT, or multi-homing. Each node maintains a set of local locators (i. e. layer 2/3/4 addresses) that is called *endpoint-descriptor*. Creation of a link is transparent to the service. A handshake over the base-overlay between the two nodes exchanges their endpoint-descriptors in the first place. Using all available communication modules both nodes try to establish communication links from both directions. *ariba* negotiates these communication links and finally establishes a single virtual link that is identified through its LinkID. Such a virtual link is built up by *ariba* using one or several transport connections and relay nodes are used in case the two nodes have no direct network reachability due to middleboxes (e. g. NAT), or protocol heterogeneity (e. g. IPv4/IPv6).

3 Service development with Ariba

In this Section we given an overview on service development with the *ariba* API. *ariba* exposes two interfaces to the developer, as shown in Figure 2: (1) node-specific interface, and (2) communication-specific interface. The node-specific interface defines functionality used for creating and joining the SpoVNet instance while the communication-specific interface provides establishment of virtual links and message sending. Several services can be bound to the communication-specific part of the interface, each with its own service-specific ID—the ServiceID—used for multiplexing. SpoVNet provides a set of services that are ready to use, e. g. an application-layer multicast service for group communication. Service instances are bound to the communication-specific interface of *ariba* using a unique ServiceID for multiplexing. *ariba* has been designed to operate in an event-driven way, preventing complex threading issues for service developers. Services exclusively operate on virtual links created through the communication-specific interface. Using event-functions services are notified of communication-specific and node-specific events and may react accordingly.

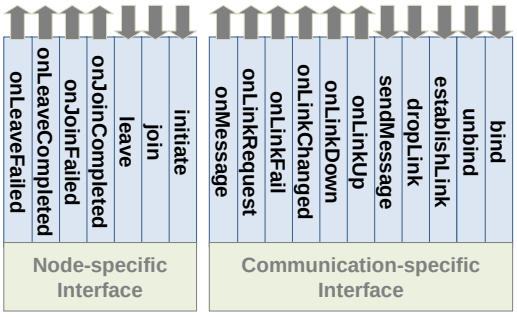

Figure 2: Developer view of *ariba* interfaces

4 Conclusion and Outlook

ariba has been developed to allow for easy service development in a distributed manner. Our goal is to foster development of usually infrastructure-hungry services in a cost-efficient and distributed setting. The current open-source implementation of *ariba*[1] runs on Linux flavors and Maemo-based Nokia handhelds. We are in the progress of porting *ariba* to OpenWRT for use on infrastructure WLAN routers. Incorporation of routers into overlay-based services—as described in [1]—is clearly a goal we are approaching.

Acknowledgment

This work was partially funded as part of the *Spontaneous Virtual Networks* (*SpoVNet*) project by the Landesstiftung Baden-Württemberg within the BW-FIT program and as part of the Young Investigator Group *Controlling Heterogeneous and Dynamic Mobile Grid and Peer-to-Peer Systems* (*CoMoGriP*) by the *Concept for the Future* of Karlsruhe Institute of Technology (*KIT*) within the framework of the German Excellence Initiative.

References

[1] Bruce Davie and Jan Medved. A programmable overlay router for service provider innovation. In *Proceedings of PRESTO Workshop at ACM SIGCOMM*, Barcelona, Spain, August 2009.

[2] Christian Hübsch, Christoph P. Mayer, Sebastian Mies, Roland Bless, Oliver P. Waldhorst, and Martina Zitterbart. Reconnecting the internet with ariba: Self-organizing provisioning of end-to-end connectivity in heterogeneous networks. In *Proceedings of ACM SIGCOMM*, Barcelona, Spain, August 2009. Demo.

[3] O. Waldhorst, C. Blankenhorn, D. Haage, R. Holz, G. Koch, B. Koldehofe, F. Lampi, C. Mayer, and S. Mies. Spontaneous Virtual Networks: On the Road towards the Internet's Next Generation. *it — Information Technology Special Issue on Next Generation Internet*, 50(6), December 2008.

[1] Available at www.ariba-underlay.org