Transparent Conferencing Without Central Control - Demonstration of the DisCo Approach in P2PSIP

Alexander Knauf*, Gabriel Hege*
Thomas C. Schmidt*, Matthias Wählisch†
* HAW-Hamburg, Department Informatik
alexander.knauf@haw-hamburg.de
hege@fhtw-berlin.de, t-schmidt@ieee.org
†FU Berlin, Institut für Informatik, waehlisch@ieee.org

Lothar Grimm‡, Thomas Kluge‡
Peter Pogrzeba§
‡Deutsche Telekom AG, T-Systems
lothar.grimm@t-systems.com, t.kluge@t-systems.com
§Deutsche Telekom AG, Laboratories
peter.pogrzeba@telekom.de

***Abstract*—RELOAD is emerging as a standard from the IETF P2PSIP group that facilitates the design of manifold lightweight applications. P2P group conferencing is an example of particular interest to Internet users. In this demo, we present a RELOAD-based protocol for Distributed Conference Control (DisCo) with SIP, as well as an implementation of DisCo and its underlying RELOAD stack. DisCo transparently addresses a conference through a single SIP URI while splitting its semantic of identifier and locator. Conference members are enabled to select conference controllers based on proximity awareness and to recover from failures of individual resource instances. Conference access control and membership coordination is achieved by a shared resource within the P2PSIP overlay that forms a basic primitive for enabling various coordination and notification schemes among distributed peers.**

***Index Terms*—Group communication, distributed conferencing, RELOAD implementation**

I. Introduction

Session-based group communication such as voice and video conferencing, but also light-weight applications like instant messaging with presence states are continuously driving the Internet development. Since the successful advent of P2P technologies, users push towards operational models that are provider-independent, lightweight services, free of charge and centered at end-devices. Popular communicators such as Skype [1] are built from proprietary models, but the IETF is following up with the near release of its RELOAD [2] protocol.

The emerging RELOAD standard provides base primitives for storing and discovering resources in a structured P2P overlay, including basic authentication and support for clients outside the overlay, but remains completely open with respect to the application type (Usage) and data structure (Kind) employed by users of the overlay. Syntax, protocol operations, and semantic may and need to be defined for the simple use by SIP [3] phones and for any other P2P application based on standards.

In this demo, we present a prototype implementation of DisCo [4], the RELOAD Usage for Distributed Conference Control with SIP and its underlying RELOAD stack. DisCo enables tightly coupled ad-hoc conferences by providing generic methods for transparently distributing conference controllers, i.e., focus nodes, that remain accessible via a single conference URI, are resilient to failures of single focus instances and provide load sharing and proxy functions for DisCo-unaware clients. DisCo is built upon a generic primitive for controlling access to shared resources (ShaRe) [5] in RELOAD that enables the coordination between mutually independent focus peers with authorization stirred by an access control tree.

The implementation of this P2P conferencing protocol requires a full-featured RELOAD stack which we introduce to the community, as well. The stack is augmented by the SIP protocol engine PJSIP [6] that accounts for all standard message handling from clients. Particular focus is donated to supporting mobiles in this .NET solution by integrating handhelds as pure clients and coupling International Mobile Subscriber Identity (IMSI) to SIP URIs.

In summary the demo presents the following contributions:

- A prototype implementation of DisCo and the corresponding RELOAD stack that are each running either on desktop or handheld devices.
- A demo application to emulate multiple RELOAD peers and clients on a single device including a visualization tool build upon the Google Maps API[7].
- A fully decentralized, scalable voice conferencing system with SIP for Windows desktop and mobile devices.

II. Distributed Conference Control in P2PSIP

A. Introduction into RELOAD

RELOAD is designed to support a variety of different applications with similar demands. Each overlay location is assigned specific resource ID and serves as a container for several data elements from different application called RELOAD *Kinds*. A Kind can further consist of a single or multiple element indexed by an continues integer or dictionary key. Applications intending to use RELOAD for data storage need to define a RELOAD *Usage* – a textual specification of requirements by the application, e.g., data structures to be stored and application procedure after data retrieval. In contrast to common P2P overlays, RELOAD is a trustful P2P network using a Central Authority (CA) that authenticates every single peer and client at enrollment. Therefore, each

overlay participant owns one or more public key certificates signed by the CA. Each overlay message and the data it carries is signed with a private key associated with those certificates. The connections between overlay parties are secured using TLS [8]. As a result RELOAD constitutes a trustworthy P2P overlay enabled to serve various applications.

B. The DisCo Approach in P2PSIP

The distributed conference control (DisCo) scheme defines a usage in RELOAD to enable group conferences in a tightly coupled model with SIP [3]. This usage provides self-organizing and scalable signaling for allowing RELOAD peers, clients and plain SIP user agents to participate in a managed P2P conference. DisCo defines the following functions:

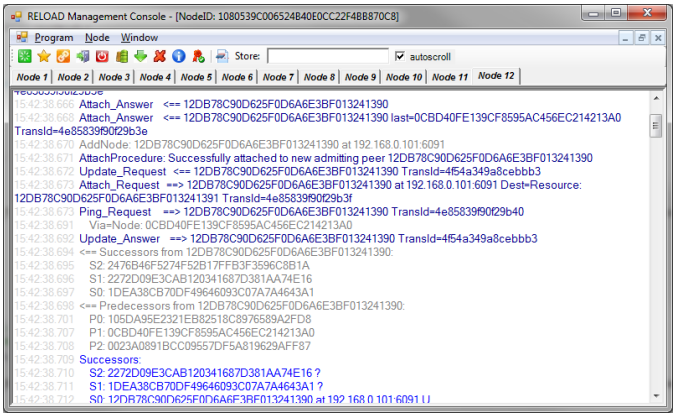
- A SIP protocol scheme for distributed conference control that allows access to a variable number of focus points by (source-)routing to one conference URI.
- A RELOAD Usage and definition of a corresponding conferencing Kind.
- Mechanisms for conference synchronization and call delegation that account for load sharing and failure recovery.
- Mechanisms for proximity-aware routing within a conference based on landmark identifiers.
- An XML event package for distributed conferences that synchronizes conference state using a vector clock.

The term distributed conferencing in this context refers to a multiparty conversation in a tightly coupled model in which the point of control (i.e., the focus) is identified by a unique URI, but the focus service is located at many independent entities. Multiple SIP user agents uniformly control and manage a multiparty session. A DisCo data structure stores the mapping of a single conference URI to multiple conference controllers in a RELOAD overlay network and thereby separates the conference identifier from focus instantiations.

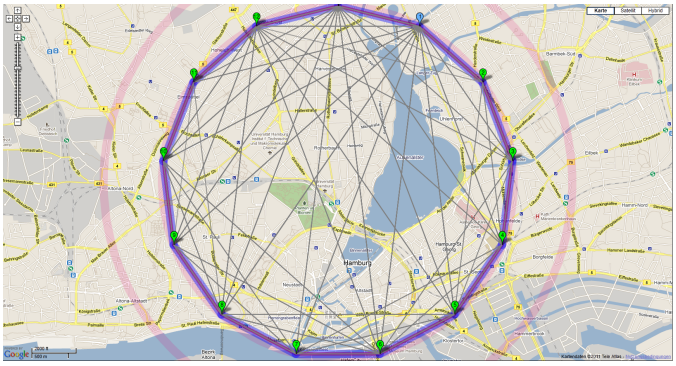
Authorized controllers of a conference are permitted to register their mapping in the DisCo data structure autonomously. Thus, the DisCo data structure represents a co-managed Resource in RELOAD. To provide trusted and secure access to a co-managed Resource, DisCo uses the definitions for Shared Resources (ShaRe) [5]. Access to a DisCo-Registration is controlled by an *Access Control List* that provides a list of RELOAD users allowed to register as focus to a conference. The list can be extended by authorized focus peers by storing new access control list items referring to further conference participants allowed to register as potential focus peers.

Delay and jitter are critical issues in multimedia communications. The proposed conferencing scheme supports self-organizing mechanisms to build an optimized interconnecting graph between conference participants and their responsible conference controllers. Conference members will be enabled to select the closest available focus with respect to delay or jitter. Previous evaluations [9] have shown that this auto-selective algorithm approaches topologies that are close to optimal with respect to the chosen metric.

DisCo extends conference control mechanisms to provide a consistent and reliable conferencing environment. Controlling



(a) RELOAD emulator



(b) Overlay Visualization

Fig. 1. RELOAD Demo application: Overlay Emulation (a) and Visualization (b) Tools

peers maintain a consistent view of the entire conference state. The multiparty system can be re-structured based on call delegation operations. Additional, a balancing mechanism at the controlling peers allows to uniformly distribute the load of serving the conference participants.

III. Demo Application

The demo presents our DisCo prototype application that implements the specifications for Distributed Conferencing including a advanced implementation of the RELOAD base specification [2]. The RELOAD stack is implemented in C# and runs likely on Windows desktop or mobile devices. The desktop variant can be executed by a GUI application that allows the emulation of multiple peers on a single device as shown in figure 1 (a). The secondary figure 1(b) shows the overlay visualization tool. It displays all participants of an RELOAD overlay and arranges them according to their overlay address in a ring representation.

A distributed conference is build upon the emulated RELOAD overlay. The conference participants are each running on a separate device and are joining to the overlay network. DisCo desktop applications thereby serve as focus peers to the conference each enabled to handle signaling and media relations for their connected conference participants. The desktop clients register themselves as potential focus

peers in the overlay with their contact and relative position in the network. These coordinates vector are emulated for demonstration purposes, but are enable for real estimated metrics, e.g., landmark vectors.

The mobile application is running on Windows Mobile 6.x devices and uses the same RELOAD stack implementation as the desktop variant. Only a few preprocessor directives are needed for the compilation and deployment on handhelds. The DisCo specification is partly implemented as RELOAD *Usage* on top of the stack and partly implemented as SIP VoIP application (see figure 2). The Usage implementation realizes the lookup functionality for resolving the conference URI of a multiparty session registered in the RELOAD overlay. On resolving a conference URI, the RELOAD Usage returns a list of contacts and relative coordinates vectors to potential focus peers via IPC to the SIP client. The SIP application then chooses one of these focus peers by reference to the coordinates vector that is the relatively closest to its own previously determined vector. Finally, the mobile applications joins the VoIP conference via SIP signaling to establish the media connections.

IV. Demo Setup

We provide the following material for our demo:

- Up to three notebooks to emulate a P2P overlay network
- Up to three mobile devices running the VoIP Clients
- If needed, a switch or wireless access point

The following environment is needed to setup the demo for Distributed Conference Control:

- Internet access via LAN/WLAN for at minimal three devices (ideally) with public IP-addresses.
- Multiple socket for at minimal three devices
- Table with a length of about 2-3 meters

The following equipment is nice to have for the demo setup:

- Pin board to attach flyers and posters
- Greater screen for demonstration proposes

The setup period is expected to be below one hour.

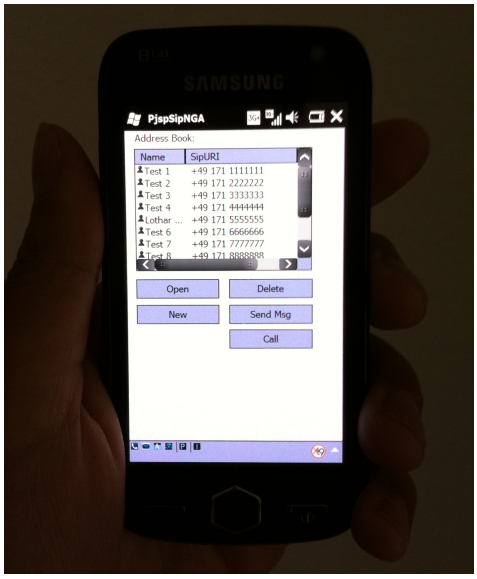


Fig. 2. Mobile application: VoIP client

References

[1] "The Skype homepage," http://www.skype.com, 2009.

[2] C. Jennings, B. Lowekamp, E. Rescorla, S. Baset, and H. Schulzrinne, "REsource LOcation And Discovery (RELOAD) Base Protocol," IETF, Internet-Draft – work in progress 13, March 2011.

[3] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol," IETF, RFC 3261, June 2002.

[4] A. Knauf, G. Hege, T. Schmidt, and M. Waehlisch, "A RELOAD Usage for Distributed Conference Control (DisCo)," IETF, Internet-Draft – work in progress 02, March 2011.

[5] ——, "A Usage for Shared Resources in RELOAD (ShaRe)," IETF, Internet-Draft – work in progress 00, March 2011.

[6] "PJSIP Stack," http://www.pjsip.org/, 2011.

[7] "Google Maps API," http://code.google.com/intl/de-DE/apis/maps/, 2011.

[8] T. Dierks and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2," IETF, RFC 5246, August 2008.

[9] A. Knauf, G. Hege, T. C. Schmidt, and M. Wählisch, "A Virtual and Distributed Control Layer with Proximity Awareness for Group Conferencing in P2PSIP," in *Proc. of IPTComm 2010*, ser. Digital Library. New York: ACM, August 2010, pp. 122–133.