A Virtual and Distributed Control Layer with Proximity Awareness for Group Conferencing in P2PSIP [*]

Alexander Knauf[1] Gabriel Hege[1] Thomas C. Schmidt[1] Matthias Wählisch[2]
[1]HAW Hamburg, Dept. Informatik, Berliner Tor 7, D–20099 Hamburg, Germany
[2]Freie Universität Berlin, Inst. für Informatik, Takustr. 9, D–14195 Berlin, Germany
alexander.knauf@haw-hamburg.de hege@fhtw-berlin.de {t.schmidt,waehlisch}@ieee.org

ABSTRACT

There is an increasing demand to access voice or video group conferences without the burden of a dedicated infrastructure, but at any place and in an ad hoc fashion. Corresponding solutions require a lightweight, fully distributed cooperation among parties that share and manage the conference in an efficient, self-adaptive way. The technology framework of P2PSIP can be seen as a promising starting point to meet these objectives. In this paper, we make several contributions towards such a distributed, virtualized control layer based on P2PSIP that seamlessly scales and adapts to the user needs. We propose a P2P-signaling protocol scheme for a distributed conference control with SIP, that splits the semantic of Identifier and Locator of a SIP conference URI in a standard-compliant manner. This protocol scheme serves as further basis for a virtualization in RELOAD. We further design and evaluate a self-organizing communication layer that provides load sharing and churn resilience with proximity-awareness. Finally, we address key aspects of security and trust, as well as compatibility for conference unaware clients.

Categories and Subject Descriptors

C.2.2 [**Network Protocols**]: Applications—*SIP*; C.2.4 [**Distributed Systems**]: Distributed applications—*Conferencing*

General Terms

Scalability, Reliability, Security

Keywords

Overlay virtualization, ID locator split, tightly coupled SIP conferencing, distributed conference control

[*]This work is supported by the German Ministry for Research and Education within the projects *Mindstone* (http://mindstone.hylos.org) and H∀Mcast (http://hamcast.realmv6.org).

1. INTRODUCTION

Voice and Video over IP (VVoIP) conference applications follow a trend to become independent tools driven by end users, since the capabilities at end systems (CPU, Memory) and the connectivity to broadband Internet are increasing continuously. They not only offer an alternative to traditional telephony, but liberate users from provider-bound infrastructure at common service charges. These changes are even more visible in the mobile domain with its spread of intelligent smartphones, and a foreseeable decline of operator control of end systems. In addition to traditional telecommunication services, VVoIP deployments open the realm to richer and more flexible use cases such as ad-hoc multi-party conversations of variable sizes.

Popular lightweight group communicators such as Skype [1] are built from proprietary models and protocols, while multiuser multimedia conferencing systems based on the Session Initiation Protocol (SIP) standard [2] are mainly deployed on dedicated server systems. Recent IETF activities, though, emerge to a new, infrastructureless session management using P2PSIP overlays and a control layer that converges to the form of REsource LOcation And Discovery (RELOAD) [3]. Conferencing solutions built by SIP and non-SIP means are still almost exclusively constructed with the help of one central conference controller per session, which — in a P2P setup — severely limits scalability and reliability of the application. Distributed conference session management has not yet been taken up by the P2PSIP community.

In this paper, we propose a virtual and distributed conference management architecture and a protocol that operate in a P2P ad-hoc mode independent of infrastructure. By separating the locator from the identifier of the conference controller, the *focus* [4] or conference Unified Resource Identifier (URI), we show how the multi-party session management can be distributed among multiple peers. We introduce a simple routing scheme that transparently guides conference signaling through the focus cloud, but still requires a globally routable physical focus instance for an initial conference contact. To overcome the dependence on individual peers, we virtualize the focus addressing within RELOAD. The conference URI, which commonly provides global routability to a dedicated focus, is published on the P2PSIP overlay network as a key to several end system devices. Further on, the transparent routing is transfered to operate on a proximity-aware overlay identifier space and gives rise to a self-adaptive tuning of the mutual communication flows.

The overall result of this work is a decentralized serverless system for distributed conference management with SIP coined DisCo. It solves the open problem of organizing conferences in a spontaneous, scalable and robust way based on the emerging standards of P2PSIP technologies. Evaluations reveal that the use of proximity-aware identifiers in an adaptive routing lead to a seamless self-organization with efficient neighborhood selection in our solution. These mechanisms are also designed as base implementations for a distributed media mixing which scales up to a large number of participants, and remains reliable against node departure or failures.

The remainder of this paper is organized as follows. Section 2 presents an overview of background technologies and related work on the subject, followed by a discussion of the distributed conferencing problem and its requirements. Our core distribution mechanisms for a SIP conference focus are outlined and evaluated in Section 3. Section 4 is dedicated to the conference virtualization in P2PSIP and its adoption of the core distribution scheme; RELOAD usages and kinds are defined here along with the self-organizing procedures, authentication and trust aspects and a protocol evaluation. Finally, Section 5 is dedicated to conclusions and an outlook.

2. DISTRIBUTED CONFERENCING: PROBLEM STATEMENT & RELATED WORK

2.1 Traditional Conferencing

Three models of multi-party communication have been defined in the discussion process at the IETF. The *loosely coupled* model does not provide a signaling relationship between conference participants. Membership is achieved by joining multicast groups and control information are learned out of band or from the application transport protocol (e.g., RTCP [5]). In a *fully distributed* model, each participant somehow manages a signaling dialog to all other remote participants. Finally, in the *tightly coupled* model signaling relationships are established between participants and one central point of control, that negotiates media parameters to establish media sessions. In SIP, this central point of control is called the *focus* of a conference [6]. It is identified and located by a conference-specific SIP URI that must be globally unique and routable. The first two models are not further defined, leaving details and complexity to further specifications. In the tightly coupled approach, a conference-specific URI will be obtained by querying a dedicated conferencing server. This allocates and publishes a conference URI (e.g., *sip:meeting@muppets.com*) and instantiates its corresponding focus . The focus then serves as interface towards SIP user agents that are interested in joining the multimedia session. In addition to media negotiations, a conference focus may comprise presence and conference state [7] notification services. The focus also enforces a predefined conference policy (e.g., permitted participants) and controls the media mixing components.

2.2 Peer-to-Peer SIP Overview

The P2PSIP working group is dedicated to provide a virtualized communication infrastructure for IP-based session services. It decided to rely on a structured peer-to-peer approach. Structured P2P systems are based on Distributed Hash Table (DHT) algorithms that can provide resource location and storage in an application layer overlay network. The overlay routing and data storage efforts are equally distributed among the participating peers and scales up to a very high number of joining nodes. The benefits of DHTs originate form its performance properties of typically $O(log(N))$ routing hops on average, and for a requirement of $O(log(N))$ routing table entries per node, where N is the number of overlay members. DHTs such as Chord, Pastry, CAN, and Kademlia [8, 9, 10, 11] have proved for their distributed, robust and scalable characteristics and now experience a wider deployment in various file sharing applications (e.g., BitTorrent [12]).

Proximity Aware Overlays.

Overlay network identifier are typically generated from hash-functions (e.g., SHA1 in Chord) for maintaining a uniform flat address space. These IDs normally do not have any relation to relative network positions of a nodes in the underlay. Numerical neighbors in the overlay can be physically far apart. Improved structuring of P2P overlays [13, 14] therefore may account for proximity information. One class of approaches is built from landmarks. To determine the relative network position p of a node, the round-trip times (RTT) are measured against a fixed set of well known landmarks $l_0, l_1, .., l_n$. These measurement results will be ordered according to the landmark index with the result of a *landmark vector* $< l_1, l_2, .., l_3 >$. Thereafter, the entire address space will be divide into equally sized *regions*. The definition of a region depends on the DHT and its address structure in use. The ring-type address space like in Chord can be cut into equal slices; subtrees in Pastry can define a region or an n-dimensional space in CAN. Each landmark vector permutation produces exactly one related region. A node then joins the overlay at a 'random' point in the region, that belongs to its landmark vector permutation. A node may then be assigned to a relative position according to its overlay ID, since every node has constructed its ID using the same calculations. A disadvantage of this ID construction is caused by an uneven population of the address space. This may cause peers to become responsible for much larger address ranges than others. Load-balancing algorithms can handle this problem by relocating responsibilities for overlay spaces to less loaded peers.

Other approaches construct mappings between overlay IDs and position information that are stored in the overlay. Assuming the relative position p for node-ID ID_n, then $p' = hash(ID_n)$ is an overlay identifier, for position p of node n. Node n will then be mapped with p' into the overlay region, stored on the node that is responsible for this address range. The nodes n_1 and n_2 are close to each other if the difference of $|p_1 - p_2|$ is low, with p_1 and p_2 were retrieved by a lookup operation on ${p_1}'$ and ${p_2}'$.

P2PSIP approaches.

Traditional SIP-oriented service architectures depend on proxy servers that assist in call routing, user location, NAT and Firewall traversal, as well as additional functionalities. This orientation on static infrastructure limits deployment and motivates approaches to relocate the proxy roles into a P2P overlay for SIP sessions. Peers query the overlay by using a P2P signaling protocol, and may contact an address of the desired user agent without further hindrance. Core procedures for call establishment should still be achieved by using standard SIP mechanisms. K. Singh *et al.* [15] and D.

Bryan *et al* [16] presented two different approaches, using a Chord overlay network for replacing the SIP client-server infrastructure. Both are using SIP messages within their P2P signaling protocol which are routed throughout the overlay. For example, sending REGISTER requests is mapped to the meaning of a DHT *join* message. Note that the semantics of these SIP messages are either changed or extended by new extension-header fields. Fessi *et al.* [17] presented a hybrid model, connecting a user agent to a dedicated SIP server, and likewise to a P2P SIP overlay. In this way, the authors gain the benefits of the traditional SIP of low signaling latencies and a trustworthy instance for security considerations. In the case of a SIP server failure, a user agent may regain connectivity by the P2P SIP overlay. To provide backwards compatibility, a *CoSIP* proxy server is proposed as gateway from SIP to the P2P protocol.

The necessity for an P2PSIP storage and lookup service overlay was adopted by the IETF. The P2PSIP working group is now standardizing a signaling protocol for REsource LOcation and Discovery (RELOAD) [3]. The intention is to establish a P2P overlay network based on a improved Chord DHT, providing a storage and resource location platform for different kinds of data. It is firstly designed for a usage for SIP [18], but can be extended for new kinds with similar requirements. We use this flexibility to define a new Usage for RELOAD for a virtual and distributed conference, mapping contact data and positioning informations into the overlay.

2.3 Related Decentralized Approaches

Several approaches have already dealt with the problem space of distributed conferences. S. Romano *et al.* [19] presented a framework that allows to receive information about conferences from various distributed conference servers. Therefore, they foresee signaling relations between multiple instances of dedicated centralized conferencing servers. A user agent can query its local conference server about multi-party sessions running at remote XCON Servers. The local conference server then requests the remote server for the required parameters to participate and passes them to the requesting client. An approach for a P2P SIP conference construction were developed by K. Tirasoontorn *et al.* [20]. The conference URI, created by a *Conference Factory* placed on a dedicated Server, will be announced in an P2P overlay network including the required media and contact information to join the conference. The user agent that stored the conference information in the overlay, is responsible to perform conference operations. It remains as a single contact point to manage the multi-party conversation. Y. Cho *et al* [21] presented a distributed architecture for signaling and media mixing. In this hierarchical approach, a dedicated *primary focus* server schedules conference participation requests among a set of *regional focus* servers. The latter are responsible to include the new participants and grant their access to the provided media data. The encoding effort thereby will be distributed onto several devices providing large-scale multimedia conferences.

2.4 Problems and Requirements for Virtualized Distributed Conferences

The traditional way to manage a multi-party conference is to create a central point of control. Thereby SIP correctly identifies a conference as a single logical entity and maps its identity to single point of control. The centrality of the latter is limited by scalability and stability, and conceptually forms the main problem for any approach of distributing a conference architecture [22]. This conference controller in SIP is called *focus* and plays the role of an interface to participants, serves as negotiator for media parameters, and often provides conference state notification services. The focus is identified and located by a Globally unique Routable User agent URI (GRUU). Each request of a callee will be routed to the physical device behind this address. This results in a *single point of failure* problem, as the conference breaks down with failures in this device or its connections. In a P2P scenario, the reliability of the conference controlling node cannot be guaranteed and may cause a complete failure of the conference on regular departure. Apart from signaling, the chain of decoding, mixing and again encoding of media data demands high computational effort. Therefore, common solutions for multiuser voice and video conferencing are placed at dedicated server systems. They are capable to reliably serve a fixed amount of media streams at limited numbers (video solutions are typically designed for about 20 participants). Common end user systems are only able to handle a fraction of this amount due to computing efforts. Deployed P2P-streaming (e.g., Zattoo [23]) solutions challenge the possibilities of using the end-user systems for distributed media streaming or mixing in pure audio. Currently, many approaches apparently remain at a borderline quality, but provisioning of reliable media streams will be soon enabled by the continuous dissemination of high speed Internet connections in home networks and the rising computational power of consumer computer.

From this perspective, we follow the need to design a distributed conferencing scheme in a P2P fashion as a future standard-based solution for fully distributed voice and video conferences. Therefore we define a set of requirements to be met by our distributed conferencing protocol DisCo:

- **Ad-hoc conference creation** Any user agent implementing the conferencing scheme, must be able to create a multi-party session at any time. The creation must be independent from a server infrastructure.

- **Splitting the central conference control** The conference focus must be divisible into several independent end systems. The split of the focus must thereby be transparently achieved with respect to standard-compliant SIP implementations and should appear as one single entity. The focus distribution should be activated prior to a focus peer management resource exhaustion. Any party should be enabled to discover other potential focus peers within among active members.

- **Robustness against focus failure** It must be possible to *re-arrange* (not to re-create) a conference, as one or more controlling peers fail, and thus to increase the reliability as compared with centralized solutions.

- **Availability of a conference** To provide accessibility to a distributed conference, it must be announced on a stable platform. For this purpose, a well-defined conference data structure must be stored redundant in a P2P network, that allows to resolve a conference URI, that points to several independent conference managers as entry points.

- **Proximity aware participation** The proposed conference signaling topology should serve road map for the transfered media data. The media processing peers should be arranged. New participants should be able to select the physically closest focus peers, to minimize signaling and data transfer delays.
- **Security and Privacy** A distributed conference must ensure that only authorized participants can attend the conference. Also needs to be ensured that only determined user can change and manage a conference state.
- **Backwards compatibility** A virtualized and distributed conference must be accessible by client implementations that do not support our DisCo Usage.

3. DISTRIBUTING A FOCUS WITH SIP

3.1 Protocol Scheme

The first step for designing a distributed conference is to separate the central control of the focus at the SIP layer. A conference URI refers per se to a dedicated focus. Our Scalable Distributed CONference (SDCON) [24] approach splits the meaning of the conference URI into identifier and locater. This is achieved by introducing a source routing approach, which transparently forwards data among conference controllers that share a common conference URI. The *focus service* of a conference is distributed among several participating user agents supporting the SDCON scheme. This leads to two classes of focus. First, the *primary focus*, which initially arranged and managed a multi-party conference. Second, the *secondary focus*, which is a participating user agent requested by the primary focus to become part of the distributed conference controllers. There is no functional difference between primary and secondary focus. Participants can have a signaling relation to either a primary or secondary focus. Both provide the same conferencing operations and notification services based on the same predefined policies. However, the conference URI is bound to the primary focus. We propose the virtualization of the conference identifier in section 4. This allows to completely decouple the conference URI from dedicated peers.

Conference initiation, control, and management is performed by the participating user agents adapting to the size of a dynamically growing conference. Therefore, SDCON defines a focus discovery procedure, call delegation, and state synchronization mechanisms. As the primary focus delegates a call to a secondary focus, it also transfers the used SIP Call-ID and session identifier. Using this information, a secondary focus is able to seamlessly send a re-invitation to the transfered user agent and negotiate new media parameters. To implement the source routing, the secondary focus inserts a *Record-Route* header field carrying its Globally unique Routable User agent URI (GRUU). Further signaling is thus routed to the secondary focus. An example of the SIP re-invite request is shown below:

```
INVITE sip:elmo@sesamestreet.com SIP/2.0
Call-ID: 0818@141.22.26.55
CSeq: 1 INVITE
From: <sip:puppets.meet@conf.muppets.com>;tag=134652
To: <sip:elmo@sesamestreet.com>;tag=643684
...
```

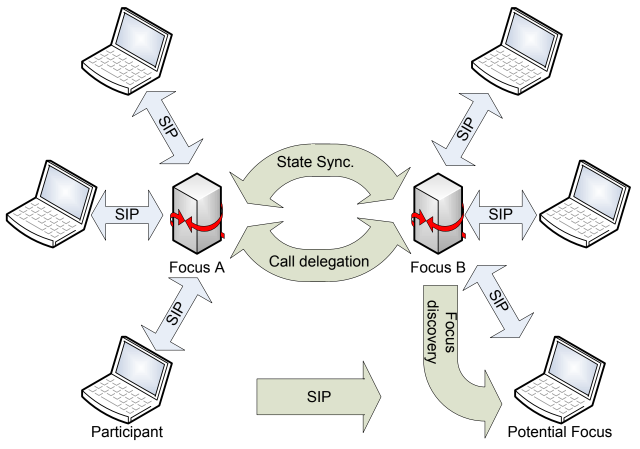


Figure 1: A distribute conference control scenario

```
Contact: <sip:puppets.meet@conf.muppets.com>;isfocus
Record-Route: <sip:kermit@sesamestreet.com>
...
```

The Record-Route header is usually added by SIP proxies to force further requests in a SIP dialog to be routed via these entities [2]. In the example above, the secondary focus *kermit* adds its own SIP URI into the Record-Route header and forces the re-invited user *elmo* to send subsequent SIP requests via him. Those source-routed requests to secondary focus peers are intercepted by them and processed. Only the focus peers are aware of the distributed fashion of conference control. Participants do not recognize the ID/Locator split, thus, the compatibility to SIP standard compliant implementations is achieved.

Figure 1 shows the main functionalities supported by SDCON user agents. The focus peers maintain signaling relations mainly by two message flows: the *State synchronization* messages and the *Call delegation* request messages. Call delegations occur when a focus is fully booked and needs to refer additional calls to less loaded focus peers. This is realized by sending standard SIP compliant REFER requests. Plain calls that address the conference URI are routed to the primary focus. Call delegation, thus, will mainly be performed for secondary focus peers. Synchronization messages are sent on change of state in any single focus entity, e.g., announcing the arrival of a new participant. These messages have to reach every controller to keep a consistent view on the conference. Synchronizations are sent within SIP NOTIFY messages carrying an XML document defined by the Event Package for Conference State [7], which is extended for multi-focus demands. The additional elements include information about each focus capacities, list of the participants that are connected to it, and shows the signaling relation to other focus peers. The capacity information is used to prevent a call delegation to an already busy focus. In the case that the synchronization process has not been completed while a call delegation is performed, each focus peer can use SIP 4xx response messages types [2] to advertise its status as busy. Another function consists in the ability to discover focuses capabilities among participating peers [24]. The *focus discovery* procedure is initiated before a focus reaches its threshold for serving new clients.

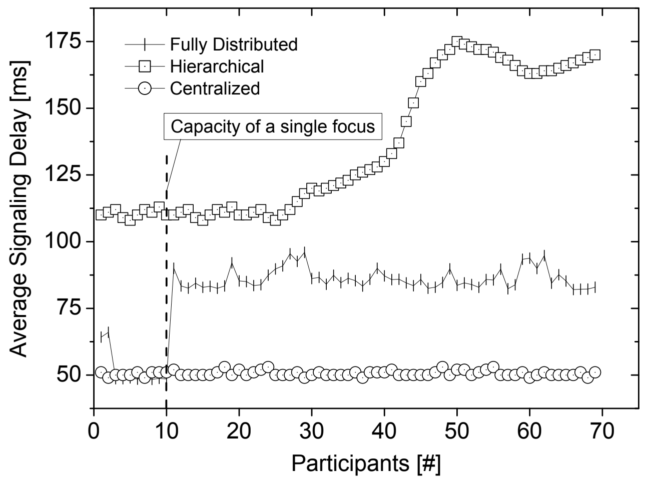


Figure 2: Time to completion of an INVITE request for a newly arriving peer

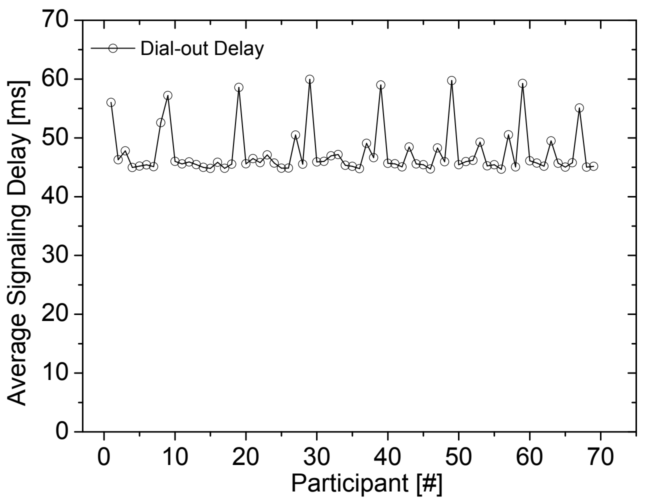


Figure 3: Third-party participation via REFER requests

3.2 Evaluation

To validate the operation and test the scalability of SDCON signaling, we implemented a prototype application and performed experimental measurements. The prototype is based on the NIST Jain SIP stack [25], which represents the reference implementation for Java. All measurements were performed in emulation mode. A minimal SIP proxy implementation was executed on a Pentium D 2*2.80 GHz 2 with 2 GB RAM. The emulated participants and conference focus peers have been executed on an Intel(R) Xeon(R) CPU 16*2.33 GHz with 16 GB RAM. The capacity of a single focus was fixed to 10 conference members. Each measurement result presents the average signaling delay of 50 independent runs.

Figure 2 presents the average signaling delay to participate a conference, i.e., sending SIP INVITE requests towards the conference URI. It compares our fully distributed conference management with a centralized [4] and hierarchical [21] approach. The later implements a recursive call delegation starting from the primary focus along the focus servers. For small conferences, where all parties can be served from a single focus, our results agree with delays of a centralized approach. The redistribution of the focus attachment in our scheme causes one additional REFER message and thus slightly doubles the signaling times. Apart from this delay enhancement, the distributed conferencing admits almost constant delays, in contrast to the hierarchical scheme. The latter experiences increasing delays of approximately linear scale with growing conference size.

The signaling delay for a third-party invitation is presented in Figure 3. In this scenario, each recently joined conference member initiates a third-party request to its related conference focus peer by sending corresponding SIP REFER messages. The measurements follow our previous observations. Most third-party participations are handled in a constant signaling delay around 45 ms. Delay peaks reflect overloaded focus peers with a maximum of 10 conference members. On reaching more than 10 members, a focus initiates the focus discovery procedure and delegates further participation requests to the new capable focus peer.

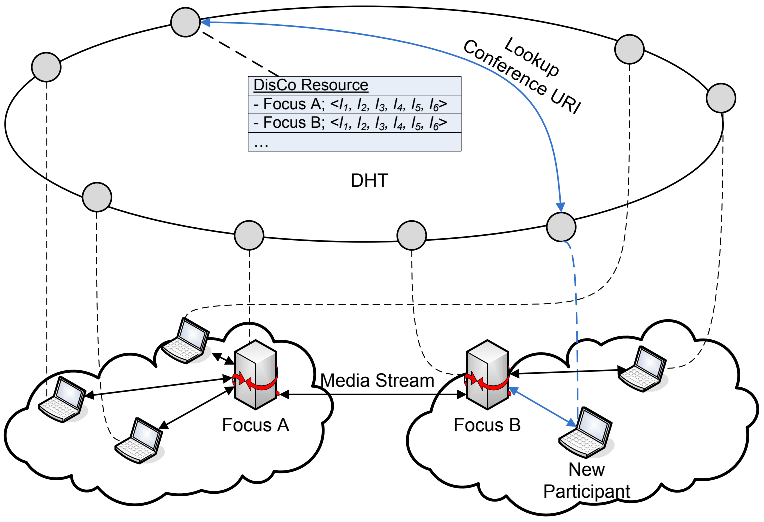
4. VIRTUALIZED CONFERENCE CONTROL WITH P2PSIP

The aim of virtualizing the conference is to separate its logical ID from any physical instance. The P2PSIP overlay RELOAD [3] facilitates a corresponding mapping and lookup of currently available conference management peers. By using P2PSIP with RELOAD we gain the benefits of an open and extensible signaling protocol that provides solutions for common problems in traditional SIP and P2P systems.

RELOAD serves as a P2P service platform providing a message transport protocol, data storage and lookup functionalities, as well as connection establishment for different types of applications. Since connectivity of many peers in an overlay may be limited by NATs or firewalls, Interactive Connectivity Establishment (ICE) [26] is supported for NAT and firewall traversal. RELOAD also provides a security framework based on public/private-key certificates to establish trust relations and message authentication.

Overlay messages are designed with a simple and lightweight forwarding header reducing forwarding effort and increasing the routing performance. A noteworthy feature of RELOAD is that the overlay algorithm to be used is not fixed, but left to the implementation. However, the current version of the RELOAD draft foresees a deployment on an improved Chord distributed hash table (DHT). To support different applications, RELOAD allows for the specification of new *Usages.* A Usage defines the data structures (*kinds*) to be stored, the corresponding data identifier (*kind-ID*), *access control* rules to those resources and how the resources' overlay IDs are to be formed.

Our concept of a virtual and distributed conference control uses these RELOAD benefits to provide a reliable, flexible and scalable conferencing service in a P2P fashion. We define a RELOAD Usage for separating the conference URI from any specific focus entity and map it to the set of participants that act as a focus instance. The proposed RELOAD data structure provides network positioning information to enable a proximity based focus selection. Based on this kind definition, our Distributed Conference Usage (*DisCo*)



DisCo Resource
- Focus A; <l_1, l_2, l_3, l_4, l_5, l_6>
- Focus B; <l_1, l_2, l_3, l_4, l_5, l_6>
…
Lookup Conference URI
DHT
Media Stream
Focus A
Focus B
New Participant

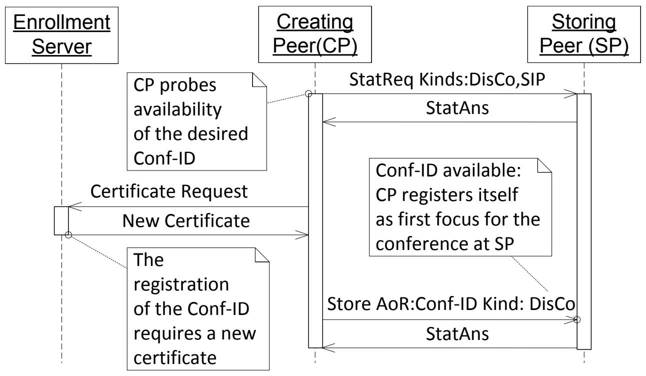
Figure 4: Discovery of a secondary focus using proximity information

[27] allows for the ad hoc creation of multimedia conferences without a dedicated server infrastructure. Conference signaling is performed using the call delegation and synchronization mechanisms as described in the previous section.

4.1 Distributing a Focus in RELOAD

DisCo defines a distributed SIP conferencing Usage that publishes all available entry points to a conference in a P2P fashion. The inter-focus SIP signaling is performed using the SDCON protocol scheme presented in the previous section. This keeps the conference state in sync and performs load balancing whenever focus peers are reaching their service threshold for hosting clients. The DisCo Usage allows a SIP user agent to create a tightly coupled conference in P2P fashion, without assistance of a dedicated conference server. Figure 5 displays the procedure of how to register a distributed conference in a RELOAD instance. The creating peer (CP) of a conference generates the desired conference URI (Conf-ID) and first probes whether this address is available. This is performed by using the RELOAD StatReq message which is routed to the storing peer (SP) responsible for the overlay ID. Overlay storage is organized according to keys obtained by hashing the conference URIs. The corresponding StatAns messages contains all meta data about the RELOAD *resources* already stored at this *resource-id*. If no other DisCo or SIP registrations for the selected Conf-ID exist, CP can proceed by querying the enrollment server of this RELOAD instance to obtain a new certificate created for the conference URI. Using this security certificate, CP then creates a DisCo *kind* data structure that comprises tuples of two types of information. At first the address where a joining peer can contact the CP to join the conference, at second a coordinate vector that encodes the relative position of the CP within the underlying network. Using the RELOAD Store operation, CP registers the conference in the overlay.

The distributed conference registration will be treated as a RELOAD *resource* of *Kind* DisCo maintained by the storing peer. The RELOAD overlay itself acts as a registrar and establishes direct transport connections traversing NATs and firewalls.



Enrollment Server
Creating Peer(CP)
Storing Peer (SP)
CP probes availability of the desired Conf-ID
StatReq Kinds:DisCo,SIP
StatAns
Conf-ID available: CP registers itself as first focus for the conference at SP
Certificate Request
New Certificate
The registration of the Conf-ID requires a new certificate
Store AoR:Conf-ID Kind: DisCo
StatAns

Figure 5: Creation of a distributed conference

DisCo-enabled peers intending to participate in the conference need to look up the hash of the conference URI as displayed in figure 4. They retrieve the DisCo conference resources, i.e., a RELOAD *dictionary* data structure in which each single dictionary entry points to a distributed conference focus. In a RELOAD *dictionary* data model, each value stored is indexed by a *key*. Using this index scheme, a focus peer can explicitly update its own contact and coordinates information maintaining its own overlay ID as dictionary key. The contact information of the conference focus can be of two different types, an Address-of-Record or a RELOAD overlay ID. In the first case, if the retrieved Address-of-Record (AOR) is a GRUU, the participating peer simply establishes a regular SIP session by sending a SIP INVITE request towards the announced contact. Otherwise the received AOR is registered with the standard SIP Usage for RELOAD and must be resolved following the SIP Usage protocol. If the retrieved contact is a RELOAD overlay ID, a participating peer needs to perform a RELOAD *appattach* request to establish a direct connection to the remote overlay peer. This request will be routed along the overlay with ICE parameters and defines the desired application protocol

```
enum {sip_focus_uri (1), sip_focus_node_id (2)
} SipDistConfRegistrationtType;
struct {
  opaque  coordinate<0..2^16-1>
  select (SipDistConfRegistration.type) {
      case sip_focus_uri:
        opaque  uri<0..2^16-1>
      case sip_focus_node_id:
        Destination destination_list<0..2^16-1>
  }
} SipDistConfRegistrationData
struct {
    SipDistConfRegistrationType     type;
    uint16                          length;
    SipDistConfRegistrationData     data;
} SipDistConfRegistration
```

Figure 6: Proposed RELOAD data structure for a distributed conferencing kind

as SIP. After the *appattach* request has succeeded, an ordinary SIP session will be build upon the newly created transport connection. A new conference member can advertise its focus ability by adding an *allow event* to the multi-focus conference state event package in the INVITE request.

Each contact in the data structure is complemented by coordinate values that indicate the relative position of the peer within the underlying network. Based on this information, a joining peer may choose the focus from the dictionary entries that is closest according to the proximity selection mechanism explained in the following section.

After a *DisCo*-enabled peer has established a SIP session by sending an INVITE, it is free to decide on advertising its own capacities. To do so, it registers as a *potential focus* to the conference storing its contact and network positioning information within the same *DisCo* resource. Focus functions will be activated either by a new joining peer that chooses this potential focus as (nearest) entry point, or by the focus *discovery procedure* explained in section 3. As a potential focus is requested by a user agent to participate via SIP signaling, it first accepts the call and establishes the requested media sessions to this client. Afterwards, the active focus will advertise its new status to all other active peers managing the conference. It subscribes its related focus to the extendedconference event package while transmitting its focus capacities and contact and media information of the new participants. The request focus will interpret this message as indication for a new user agent acting in the role of a focus and notifies all remote conference controller about this change of state. It further responds with a SIP SUBSCRIBE request to the new focus, transmitting the conference state XML document. This finishes the focus acceptance and the potential focus is a known active focus. All focus peers take the same authorities and responsibilities to manage the distributed conference as the initial focus.

The definition of the distributed conference registration kind is shown in figure 6. Every focus peer is allowed to store or update mapping bindings using its node-id as the dictionary key. The mappings stored can be of two varieties corresponding to the types allowed in the SIP-REGISTRATION: The first type *sip_focus_uri* contains the Address-of-Record of a focus peer and the second *sip_focus_node_id* returns

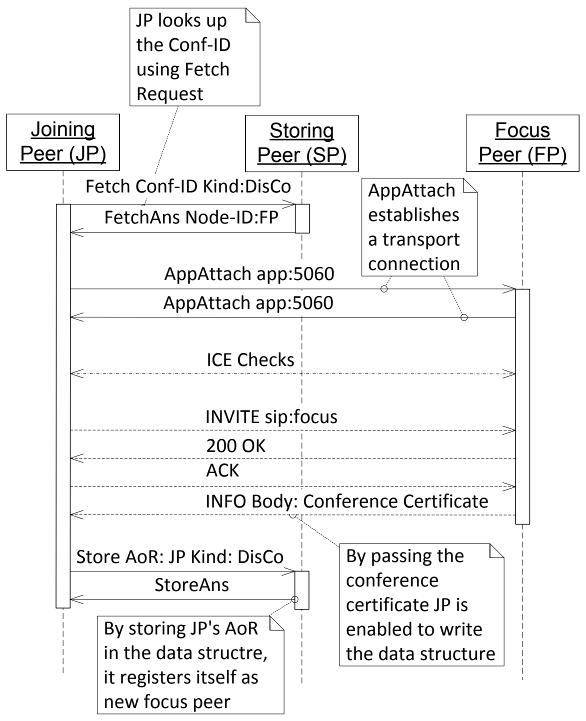

JP looks up the Conf-ID using Fetch Request
Joining Peer (JP)
Storing Peer (SP)
Focus Peer (FP)
Fetch Conf-ID Kind:DisCo
FetchAns Node-ID:FP
AppAttach establishes a transport connection
AppAttach app:5060
AppAttach app:5060
ICE Checks
INVITE sip:focus
200 OK
ACK
INFO Body: Conference Certificate
Store AoR: JP Kind: DisCo
StoreAns
By passing the conference certificate JP is enabled to write the data structure
By storing JP's AoR in the data structre, it registers itself as new focus peer

Figure 7: Joining a distributed conference and advertising focus abilities

the a RELOAD *destination list* containing overlay node-IDs. The destination list feature in RELOAD is used, to enable a requesting peer to perform a recursive overlay source routing. We define for the DisCo Usage, that the accompanying coordinates value belongs to the final target of the destination list. If storing an AoR, the related coordinates value must define the relative position of the AoR location. The coordinates value is stored as an opaque string containing the relative network defining a *landmark vector*. A *landmark vector* represents a set of Round-Trip-Times (RTT) measurements against well-known landmarks. A more detailed explanation follows in the next section. We use is explicit coordinate value, because it can not be assumed that used overlay algorithm in an RELOAD P2PSIP instance supports proximity awareness. The proposed Chord overlay in the RELOAD base definition for example, does not support proximity information.

4.2 Self Organization with Proximity-aware Load Sharing

The DisCo conference construction is performed using relative network position information. Each joining participant chooses its closest focus, and every new peer managing parts of the conference establishes an SDCON relation to its nearest active focus node. A benefit of this proximity peer selection arises from an optimized mesh build-up causing short signaling paths by default. The single steps to joining a virtual and distributed conference are the following as displayed in figure 7:

1. **Determining coordinates:** Before a peer joins the multiparty conversation, it determines RTTs against a set of

stable Internet hosts $l_1, l_2, .., l_n$ serving as landmarks. The measurement results are ordered along a landmark index that is equal for all parties and focus peers. Ordered in this manner, the measurement results in milliseconds comma-separated define our *landmark vector* representing a peers relation network position in an n-dimensional Cartesian space with n is the number of landmarks (e.g. $< 311, 87, 42, 137, 228, 75\ , .., 55 >$). We thereby follow the landmarking approach from Ratnasamy *et al.* for proximity-aware server selection [13] without the explicit binning of peers whose landmark vectors equal each other. It just serves as am abstract descriptor for a peer's relative position in the network and is not used to identify a peer.

2. **DisCo data structure retrieval:** To obtain alle available focus peers for a conference the joining peer (JP) achieves a RELOAD *fetch* request that is routed to the storing peer thats own the resource-id for the hashed Conference URI. It thereby in the sets the *kind* value in the request to the DisCo kind-id querying for the complete conference dictionary.

3. **Calculating the closest entry point:** On successful received the conference information, a peer compares each retrieved coordinates value representing the focus landmark vector with its own. Our approach subtracts the each focus landmark vector with that of the joining peer and builds the scalar product over the result of a substation. The joining peer then chooses that focus with the smallest scalar product result as entry point to the conference.

4. **Connecting to a Focus:** Using contact information deposited dictionary entry, JP establishes a transport connection to the selected focus peer (FP) using the RELOAD's AppAttach operation. It is routed throughout the overlay to FP and indicates a desired SIP signaling connection by setting the *application* field to *5060*. After FP finalized the AppAttach progress JP and FP perform ICE checks [26] to detect whether any of them if located behind a NAT and additional TURN server are needed for application session establishment.

5. **SIP Session establishment:** The established transport connection is then used to enter the ordinary SIP signaling progress thus JP can successfully join the multiparty conversation. Additionally, JP can pass JP writing permission to the DisCo registration by transmitting the shared certificate within a SIP INFO message.

6. **Advertising focus abilities:** JP can optionally advertise itself as available focus peer for the distributed conference, by mapping its contact to the existing DisCo data structure at the storing peer.

The joining peer hereby adds its own landmark vector coordinates as an URI parameter *coord* base64 encoded to its URI in the SIP *contact* header. The *coord*-parameter is used by the requested focus in case of overloading. It then performs the *call delegation* mechanism and selects a focus candidate according to the new participants network positioning. If the selected focus is capable to serve new clients it accepts the SIP call. Further, it published the new membership by achieving the SDCON synchronization mechanism explained in section 3, to keep the conference state consistent.

Because every new member chooses its closest focus, the conference will be constructed unified distributed among all controlling peers, like shown in figure 4. Following this regular construction, participants and focus peers will arrange themselves to an unbalanced distribution tree. To reduce the diameter of this tree, hence minimizing the delay times between the nodes, it is possible to establish cross connections. This kind of mesh optimization are highly dependent on the types of used media streams, and is therefore out of scope of this paper.

4.3 Resilience to Focus Failures

A problem in traditional tightly coupled conferences, originates from the focus that acts as single point of failure. If it breaks down, all signaling and media sessions are disconnected. In our scenario, the distributed structure of the conference prevents the breakdown of the entire multimedia session as one focus peer fails. As a focus fails, it can be substituted by potential or active focus peers re-collecting lost conference participants.

We use this redundancy to build a recovery mechanism. As a DisCo-enabled participant notices that its related focus does not any more deliver signaling or media packets, it will connect to one of the remaining managers of the conference. It therefore achieves the same DisCo protocol steps explained previews, however, without redetermining its landmark vector.

Conference participants not supporting the DisCo Usage will get a different treatment in case of focus error. A conference focus selects one or more active focus peers, that will serve as *backup* focus. The selection is done according to the relative network *coordinates* by choosing the closest peers. The backup selection will be announced to all other conference controller within the conference state XML document. In the case of node appearance, the detecting focus firstly notifies the conference managing peers about failure to share knowledge. It then immediately refers all disconnected participants to the *backup* focus peers. In this way, participants related to the malfunctioning conference controller just notice a temporally connection loss and recover via a re-invitation mechanism. Whenever the malicious focus returns, it re-joins the conference normally. Otherwise, the dictionary entry of this peer will be deleted by the resource owner, after the *lifetime* value expires in RELOAD.

New participating peers who try to connect to a disappeared focus will receive a 404 Not Found response message, according to the RELOAD protocol. These peers then try to connect to the focus, whose landmark coordinates are the second closest to their own. The stored DisCo data is protected against failure of the resource owner, by the provided replication algorithms in the used DHT running the RELOAD P2P SIP instance.

4.4 Security & Trust Aspects

The DisCo Usage defines a set of security and trust aspects in a P2P environment. A common problem in distributed P2P systems arises from the fact, that connections will be established, even though the corresponding partners do not necessarily trust each other. In our conference scenario, we assume that participating peers can authenticate each other in person based on the received voice and video transmissions. Built on this, we introduce a graduated trust delegation system for distributed conferences.

RELOAD provides a set of access control policies, defining whether a peer is allowed to perform a certain store request or not. For our distributed conferencing resource, we use the defined *user-match* access policy. Each stored data can be written if the request is signed with a key associated with a certificate whose hashed user name equals the resource's overlay ID. Since our DisCo resource needs to be updated by multiple peers, using the *user-node-match* policy used by the SIP Usage for RELOAD is not an option. We use our trust delegation mechanism, allowing peers to obtain write access to the shared resource. To receive write permissions for the distributed conference overlay resource, the private key for the certificate of the stored resource will be transmitted within a SIP `INFO` message to allowed focus peers. Using this key, a user is able to authenticate itself against the owner of the conference data structure, and can register as potential focus.

On conference creation, the initiating peer can setup the distributed conference policy for a layered authentication. In an *open access* model, every peer interested to join the conference can do so by just inviting one of the multi-party focus peers. No authentication is required for participating. An open access model may be suitable for a group conversation of public interest, for example a political debate. Because in this open model, an attacker could easily become a focus peer and send malicious packages, we define an *open access focus authenticate* model. The conference initiator can specify that peers wanting to become a focus need to authenticate themselves using any of the standard authentication mechanisms allowed in SIP. The corresponding credentials need to be transmitted to those peers by a non-SIP, non-overlay mechanism. As a new participant invites the conference, it uses ordinary SIP authorization. After validation of the presented credentials the called focus is then allowed to pass the conference's certificate key to the recently joined conference member. To create a closed multimedia conference, it is also possible to set an authentication scheme required for participation in a *closed access* model. Thus, only users who present valid authorization credentials are allowed to join. By combining the *closed access* and the *focus authenticate* model, our layered access model defines different permissions for clients joining the conference only and peers that are allowed to become focus, dependent on their credentials. Focus peers obtain the information needed to validate participants' credentials within the conference XML document (e.g. a conference password or certificate), to be able to authorize new members. The used access model will be stored in the conference state XML extension, thus every controlling peer is aware of the used access model.

Providing these access layers, a user initiating a conference is able to setup its desired privacy policies for the multi-party conversation. It can be suggested, that in closed conferences an unknown conference member will be detected by the participating users, for example by not recognizing its voice or outwards appearance in video. Those unsuspected users can be excluded by a conference focus peer by disconnecting signaling and media sessions.

4.5 Supporting Conference-unaware Parties

Participation a virtual and distributed conference is not be exclusive for those peers that implemented our RELOAD Usage definition. Standard compliant participation is transparently provided to peers unaware of the distributed conference construction. This section describes the backward compatibility to user applications implementing the SIP Usage [18] for RELOAD, and describes how connectivity to SIP-only user agents is achieved.

The SIP Usage for RELOAD defines a *kind* data structure for storing an AoR for a SIP user agent. It likewise uses the *destination list* feature in RELOAD and provides the storage of GRUUs as contact addresses for SIP session establishment. To provide backward compatibility to RELOAD peers only implementing the SIP Usage, a conference initiator can decide to register the conference URI as SIP-Registration *kind* parallel to the DisCo kind. The SIP Usage registration is then performed using the *destination list* feature, registering the amount of active and potential focus peers as entries in the destination list. Peers attending to join requesting resolving conference URI using SIP-Registration kind-ID, retrieve the destination list containing the conference entry points. The connection to a conference focus then will be achieved in accordance with the SIP Usage. Those peers will not be aware of the distributed structure of the multi-party conversation.

Because the SIP usage for RELOAD access model is *user-node-match*, other focus peers will not be able to update the stored data. The conference initiator must update the SIP-Registration kind continuously, on appearance or disappearance of focus peers. Hence, it depends to the conference initiator to keep the destination list up to date and valid. To achieve maximal accessibility in the case that the latter permanently leaves the multi-party conversation, it has to set the *lifetime* value on a high level. By just using the SIP-registration kind, conference joins can not be performed under proximity selection. However, a conference created with DisCo can provide access to the multi-party, although a client does not implement our usage.

The participation for ordinary SIP user agents is performed by another mechanism. Since a virtualized conference URI is stored in a RELOAD overlay, a standard SIP user agent can not resolve it with traditional mechanisms and a direct participation is not possible. Instead, participation will be achieved through third-party initiated from within the conference. An established multi-party member requests its related focus to invite the new attendees sending SIP REFER requests. By using the protocol mechanisms for transparent focus distribution explained in section 3, the requested conference manager invites the new attendee sending a SIP INVITE request.

4.6 Evaluation

To verify our concept of the proximity-aware focus selection, we conducted experimental measurements based on the PlanetLab platform [28]. PlanetLab nodes are globally distributed and thus allows for geographically placement of conference peers. Although this real-world experimental facility is biased in the sense that significant nodes are located at well-connected university networks, it gives a good approximation of delay characteristics for this part of the Internet.

CAIDA Monitor	Location
mnl-ph.ark.caida.org	
nrt-jp.ark.caida.org	ASIA
she-cn.ark.caida.org	
dub-ie.ark.caida.org	
lej-de.ark.caida.org	Europe
her-gr.ark.caida.org	
pna-es.ark.caida.org	
sea-us.ark.caida.org	
mty-mx.ark.caida.org	
amw-us.ark.caida.org	North America
yto-ca.ark.caida.org	
wbu-us.ark.caida.org	
hlz-nz.ark.caida.org	Oceania
gig-br.ark.caida.org	South America
scl-cl.ark.caida.org	

Table 1: Selected landmark nodes chosen from CAIDA measurement monitors

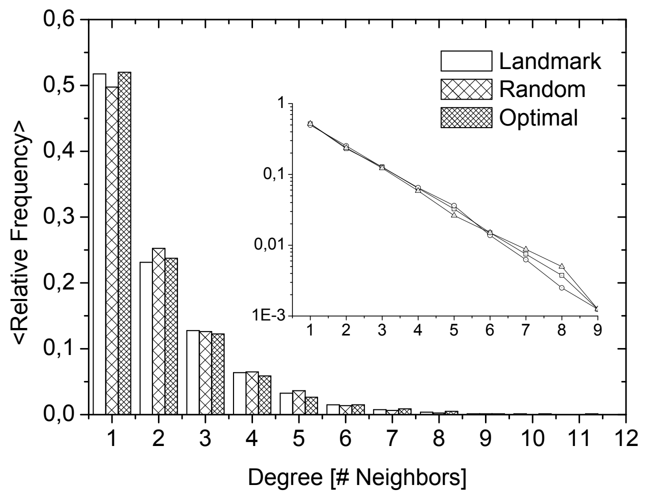


Figure 8: Degree distribution

Experimental Setup

The experiment considers small and medium size conferences. The principle setup is the following: We deployed DisCo on a varying number of peers chosen from a predefined subset of all PlanetLab nodes. These peers create focus instances and corresponding relationships based on the landmarking approach described in section 4.2. Relative network positions are determined using 15 landmark nodes outside of the PlanetLab system. The experiment is conducted until measurements are converged.

100 nodes are selected from the overall PlanetLab nodes to create the list of potential DisCo peers. To mitigate local system disturbances, we included only hosts that exhibit an appropriate system load. The selected nodes are located in Asia, Europe, South and North America to emulate a globally distributed conference and observe long range delay effects.

In general, the quality of landmark approaches depends on an appropriate number of landmark nodes and their placement. However, there is no common sense on the number of dimensions to create a coordinate system [29]. They may range typically of 7 to 9 [30], but also depend on the dataset. In order to evaluate proximity-awareness for an almost generic scenario with respect to the selected DisCo peer, i.e., without any dedicated landmark optimization, 15 landmarks are chosen from the set of CAIDA [31] monitor points (cf. Table 1). This has two advantages: First, CAIDA monitors are globally reachable and not located behind NATs or firewalls, which is important and a realistic deployment assumption for landmark nodes. Second, they are globally distributed covering different geographic locations. The landmark selection process omits suspicious nodes that reply unusually on ICMP echos.

Performance Metrics

We analyze the quality of our proximity-aware self organization of (focus) peers based on the following metrics:

Degree corresponds to the number of neighbors. Nodes that have a degree of 1 only participate in the conference without replicate data. Nodes with a larger degree operate as focus. This metric, thus, reflects implicitly the load on a peer.

Delay Stretch measures the ratio of the average delay caused by the overlay and the average delay using native distribution. It follows the idea of the relative average delay (RAD) defined by Castro et al. [32]. This metric represents the relative delay penalty.

Foci Ratio describes the ratio of overlay peers that attain the role of a focus. This metric quantifies the distribution of conference management load among peers.

The results are compared with a complete random selection of focus nodes, and an optimal solution of the focus and peer topology.

Results

The degree distribution of inter-peer relations was measured and displayed in Figure 8. For all schemes, the majority of nodes are single-attached and thus pure leafe nodes. Focus nodes that exhibit a degree ≥ 2 dominantly admit low degrees and thus suffer little load of packet replication and forwarding. More significantly and clearly visible from the insert, the probability of higher degrees exponentially decreases leaving negligible weight to the occurrence of overloaded peers or unsuitable conferencing demands.

A more sensitive measure on detailed conference performance is given by the delays imposed mutually related peer neighborhoods on the overlay. Figure 9 compares the average delay stretch of our landmarking scheme with a random neighbor selection and the optimal set-up. While a routing via arbitrary conference neighbors in our global conference evaluation may lead to alienating delay enhancements of 15 to 30 times, our landmarking scheme remains within favorable bounds around 2 to 3, which is very close to the optimal solution. Most importantly, the delay stretch remains constant with respect to the numbers of conferencing parties and thus promotes arbitrary scalability of our proposed adaptive self-organization scheme. It should be noted that randomized neighbor selection leads to a linearly increasing stretch.

Finally, we examine the relative portion of peers that attain the role of a conference controller assuming the absence

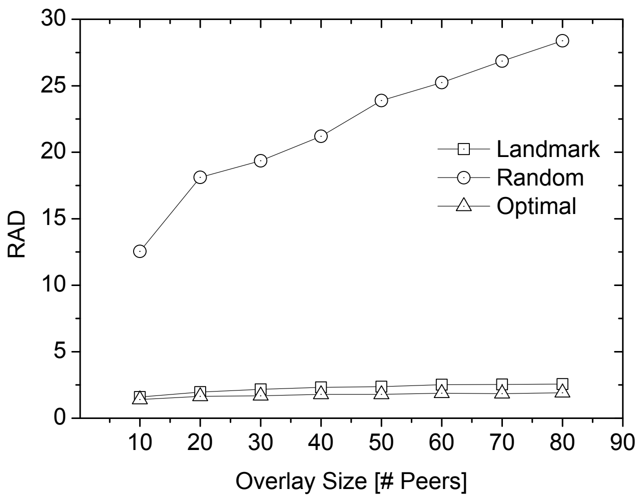


Figure 9: Delay stretch based on the Average Delay Ratio (RAD)

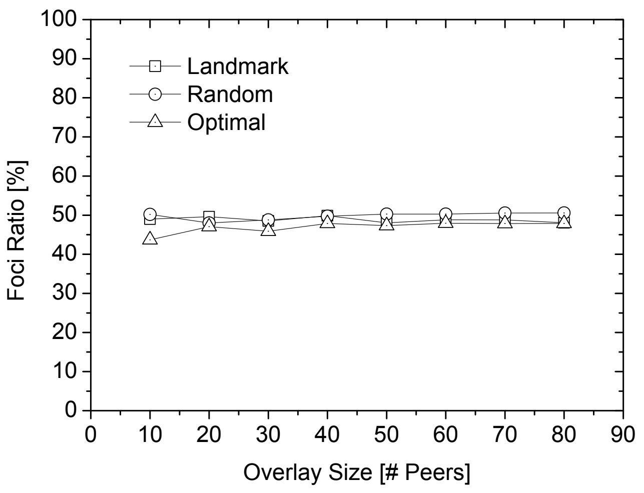


Figure 10: Ratio of overlay peers attaining the role of a focus

of NATs and firewalls. As displayed in Figure 10, the relative portions of focus peers is bound to about 50 %, independent of the conference size, as well as the adaptation scheme in use. Peers thus encounter a probability of 0.5 to be utilized as conference supporters at a scale the remains fully independent of conversational parties.

5. CONCLUSION AND OUTLOOK

In this paper, we presented a virtual and distributed conference control solution, self-organizing and adapting to the demands of a scalable, infrastructure-resilient multi-party conversation. Presenting a protocol scheme that transparently splits a SIP conference focus onto multiple peers, an address virtualization of the conference URI separates the logical ID from any physical instance. We demonstrate how this concept is implemented in a RELOAD DHT of P2PSIP, providing independence from any server infrastructure. To meet the requirements of a transient P2P environment, the presented protocol schemes maintain operations for call delegation, load balancing and state synchronization. To reduce signaling delays, we proposed a method for routing with respect to relative network position of peers and to enable a proximity-aware focus selection.

The conducted experimental measurements revealed close to optimal results for our presented concepts. We showed that the signaling delay remains constant during an increasing conference. Furthermore, our measurements on the PlanetLab platform displayed, that our proximity-aware focus selection achieves a low delay stretch. Reducing the edge degree per node and diameter of the arising tree-like mesh topology, we expect to apply further optimizing algorithms for future work. We propose to bring the concept of a virtualized and distributed conference Usage into the IETF standardization process.

6. REFERENCES

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

[2] 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.

[3] C. Jennings, B. Lowekamp, E. Rescorla, S. Baset, and H. Schulzrinne, "REsource LOcation And Discovery (RELOAD) Base Protocol," IETF, Internet-Draft – work in progress 12, November 2010.

[4] J. Rosenberg, "A Framework for Conferencing with the Session Initiation Protocol (SIP)," IETF, RFC 4353, February 2006.

[5] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," IETF, RFC 3550, July 2003.

[6] O. Levin and R. Even, "High-Level Requirements for Tightly Coupled SIP Conferencing," IETF, RFC 4245, November 2005.

[7] J. Rosenberg, H. Schulzrinne, and O. Levin, "A Session Initiation Protocol (SIP) Event Package for Conference State," IETF, RFC 4575, August 2006.

[8] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in *SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*. New York, NY, USA: ACM Press, 2001, pp. 149–160.

[9] S. Ratnasamy, M. Handley, R. M. Karp, and S. Shenker, "Application-Level Multicast Using Content-Addressable Networks," in *Networked Group Communication, Third International COST264 Workshop, NGC 2001, London, UK, November 7-9, 2001, Proceedings*, ser. LNCS, J. Crowcroft and M. Hofmann, Eds., vol. 2233. London, UK: Springer–Verlag, 2001, pp. 14–29.

[10] A. Rowstron and P. Druschel, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems," in *IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, ser. LNCS, vol. 2218. Berlin Heidelberg: Springer–Verlag, Nov. 2001, pp. 329–350.

[11] P. Maymounkov and D. Mazières, "Kademlia: A peer-to-peer information system based on the xor metric," in *Proc. of the 1st Int. Workshop on Peer-to*

Peer Systems (IPTPS '02), Cambridge, MA, USA, 2002, pp. 53–65.

[12] "The BitTorrent Homepage," http://www.bittorrent.com/, 2010.

[13] S. Ratnasamy, M. Handley, R. M. Karp, and S. Shenker, "Topologically-aware overlay construction and server selection," in *Proc. of 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '02)*, Washington, DC, USA, 2002, pp. 1190–1199.

[14] Z. Xu, C. Tang, and Z. Zhang, "Building topology-aware overlays using global soft-state," in *Proc. of the 23rd Int. Conf. on Distributed Computing Systems (ICDCS '03)*. Washington, DC, USA: IEEE Computer Society, 2003, p. 500.

[15] K. Singh and H. Schulzrinne, "Peer-to-peer internet telephony using sip," in *Proc. of the int. workshop on Network and operating systems support for digital audio and video (NOSSDAV '05)*. New York, NY, USA: ACM, 2005, pp. 63–68.

[16] D. A. Bryan, B. B. Lowekamp, and C. Jennings, "Sosimple: A serverless, standards-based, p2p sip communication system," in *Proc. of the 1st Int. Workshop on Advanced Architectures and Algorithms for Internet Delivery and Applications(AAA-IDEA '05)*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 42–49.

[17] A. Fessi, H. Niedermayer, H. Kinkelin, and G. Carle, "A cooperative sip infrastructure for highly reliable telecommunication services," in *Proc. of the 1st int. conf. on Principles, systems and applications of IP telecommunications (IPTComm '07)*. New York, NY, USA: ACM, 2007, pp. 29–38.

[18] C. Jennings, B. Lowekamp, E. Rescorla, S. Baset, and H. Schulzrinne, "A SIP Usage for RELOAD," IETF, Internet-Draft – work in progress 05, July 2010.

[19] S. Romano, A. Amirante, T. Castaldi, L. Miniero, and A. Buono, "A Framework for Distributed Conferencing," IETF, Internet-Draft – work in progress 08, December 2010.

[20] K. Tirasoontorn, S. Kamolphiwong, and S. Sae-Wong, "Distributed p2p-sip conference construction," in *Int. Conf. on Mobile Technology, Applications, and Systems (Mobility '08)*. New York, NY, USA: ACM, 2008, pp. 1–5.

[21] Y.-H. Cho, M.-S. Jeong, J.-W. Nah, W.-H. Lee, and J.-T. Park, "Policy-Based Distributed Management Architecture for Large-Scale Enterprise Conferencing Service Using SIP," *Selected Areas in Communications, IEEE Journal on*, vol. 23, no. 10, pp. 1934–1949, Oct. 2005.

[22] T. C. Schmidt and M. Wählisch, "Group Conference Management with SIP," in *SIP Handbook: Services, Technologies, and Security*, S. Ahson and M. Ilyas, Eds. Boca Raton, FL, USA: CRC Press, December 2008, pp. 123–158, on invitation. [Online]. Available: http://www.crcpress.com/product/isbn/9781420066036

[23] "The Zattoo Homepage," http://www.zattoo.com/, 2010.

[24] A. Knauf, T. C. Schmidt, and M. Wählisch, "Scalable Distributed Conference Control in Heterogeneous Peer-to-Peer Scenarios with SIP," in *Mobimedia '09: Proc. of the 5th International ICST Mobile Multimedia Communications Conference*, ser. ACM Digital Library. Brussels, Belgium: ICST, Sep. 2009, pp. 1–5. [Online]. Available: http://dx.doi.org/10.4108/ICST.MOBIMEDIA2009.7436

[25] "The NIST JAIN-SIP homepage," http://jain-sip.dev.java.net/, 2009.

[26] J. Rosenberg, "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols," IETF, RFC 5245, April 2010.

[27] A. Knauf, G. Hege, T. C. Schmidt, and M. Wählisch, "A RELOAD Usage for Distributed Conference Control (DisCo)," individual, IETF Internet Draft – work in progress 01, December 2010. [Online]. Available: http://tools.ietf.org/html/draft-knauf-p2psip-disco

[28] "The PlanetLab homepage," http://planet-lab.org/, 2010.

[29] B. Abrahao and R. Kleinberg, "On the Internet Delay Space Dimensionality," in *Proc. of the 8th ACM SIGCOMM Conf. on Internet Measurement (IMC'08)*. New York, NY, USA: ACM, 2008, pp. 157–168.

[30] L. Tang and M. Crovella, "Virtual Landmarks for the Internet," in *Proc. of the 3rd ACM SIGCOMM Conf. on Internet Measurement (IMC'03)*. New York, NY, USA: ACM, 2003, pp. 143–152.

[31] "The Cooperative Association for Internet Data Analysis homepage," http://www.caida.org/home/, 2010.

[32] M. Castro, M. B. Jones, A.-M. Kermarrec, A. Rowstron, M. Theimer, H. Wang, and A. Wolman, "An Evaluation of Scalable Application-level Multicast Built Using Peer-to-peer Overlays," in *Proceedings of the Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies (Infocom 2003)*, vol. 2. Washington, DC, USA: IEEE Computer Society, 2003, pp. 1510–1520.