Network Working Group	A. Knauf
Internet-Draft	G. Hege
Intended status: Standards Track	T C. Schmidt
Expires: May 03, 2012	HAW Hamburg
	M. Waehlisch
	link-lab & FU Berlin
	October 31, 2011

A RELOAD Usage for Distributed Conference Control (DisCo)
draft-knauf-p2psip-disco-04

Abstract

This document defines a RELOAD Usage for Distributed Conference Control (DisCo) with SIP. DisCo preserves conference addressing through a single SIP URI by splitting its semantic of identifier and locator using a new Kind data structure. Conference members are enabled to select conference controllers based on proximity awareness and to recover from failures of individual resource instances. DisCo proposes call delegation to balance the load at focus peers.

Status of this Memo

Copyright Notice

Table of Contents

*1. Introduction

*2. Terminology

*3. Overview of DisCo

*3.1. Reference Scenario

*3.2. Initiating a Distributed Conference

*3.3. Joining a Conference

*3.4. Conference State Synchronization

*3.5. Call delegation

*3.6. Resilience

*3.7. Topology Awareness

*4. RELOAD Usage for Distributed Conference Control

*4.1. Shared Resource DisCo-Registration

*4.2. Kind Data Structure

*4.3. Variable Conference Identifier

*4.4. Conference Creation

*4.5. Advertising Focus Ability

*4.6. Determining Coordinates

*4.7. Proximity-aware Conference Participation

*4.8. Configuration Document Extension

*5. Conference State Synchronization

*5.1. Event Package Overview

*5.2. <distributed-conference>

*5.3. <version-vector>/<version>

*5.4. <conference-description>

*5.5. <focus>

*5.5.1. <focus-state>

*5.5.2. <users>/<user>

*5.5.3. <relations>/<relation>

*5.6. Distribution of Change Events

*5.7. Translation to Conference-Info Event Package

*5.7.1. <conference-info>

*5.7.2. <conference-description>

*5.7.3. <host-info>

*5.7.4. <conference-state>

*5.7.5. <users>/<user>

*5.7.6. <sidebars-by-ref>/<sidebars-by-value>

*6. Distributed Conference Control with SIP

*6.1. Call delegation

*6.2. Conference Access

*6.3. Media Negotiation and Distribution

*6.3.1. Offer/Answer

*6.4. Restructuring a Conference

*6.4.1. On Graceful Leave

*6.4.2. On Unexpected Leave

*7. DisCo Kind Definition

*8. XML Schema

*9. Relax NG Grammar

*10. Security Considerations

*10.1. Trust Aspects

*11. IANA Considerations

*12. Acknowledgments

*13. References

*13.1. Normative References

*13.2. Informative References

*Appendix A. Change Log

*Authors' Addresses

1. Introduction

This document describes a RELOAD Usage for distributed conference control (DisCo) in a tightly coupled model with SIP [RFC3261]. The Usage provides self-organizing and scalable signaling that allows 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

*RELOAD Usage and definition of conferencing Kind

*Mechanisms for conference synchronization and call delegation

*Mechanism for proximity-aware routing within a conference

*An XML event package for distributed conferences

In this document, the term distributed conferencing 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 [RFC3261] user agents uniformly control and manage a multiparty session. This document defines a new Usage for RELOAD, including an additional Kind code point with a corresponding data structure that complies with the demands for distributed conferences. The 'DisCo' data structure stores the mapping of a single conference URI to multiple conference controllers 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, this document uses the definitions for Shared Resources (ShaRe) [I-D.knauf-p2psip-share].
Delay and jitter are critical issues in multimedia communications. The proposed conferencing scheme supports mechanisms to build an optimized interconnecting graph between conference participants and their responsible conference controllers. Conference members will be enabled to select the closest focus with respect to delay or jitter.

DisCo extends conference control mechanisms to provide a consistent and reliable conferencing environment. Controlling peers maintain a consistent view of the entire conference state. The multiparty system can be re-structured based on call delegation operations.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

We use the terminology and definitions from der RELOAD base draft[I-D.ietf-p2psip-base], the peer-to-peer SIP concepts draft [I-D.ietf-p2psip-concepts], the usage for shared resources draft [I-D.knauf-p2psip-share],and the terminology formed by the framework for conferencing with SIP [RFC4353]. Additionally the following terms are used:

Coordinate Value: An opaque string that describes a host's relative position in the network topology.

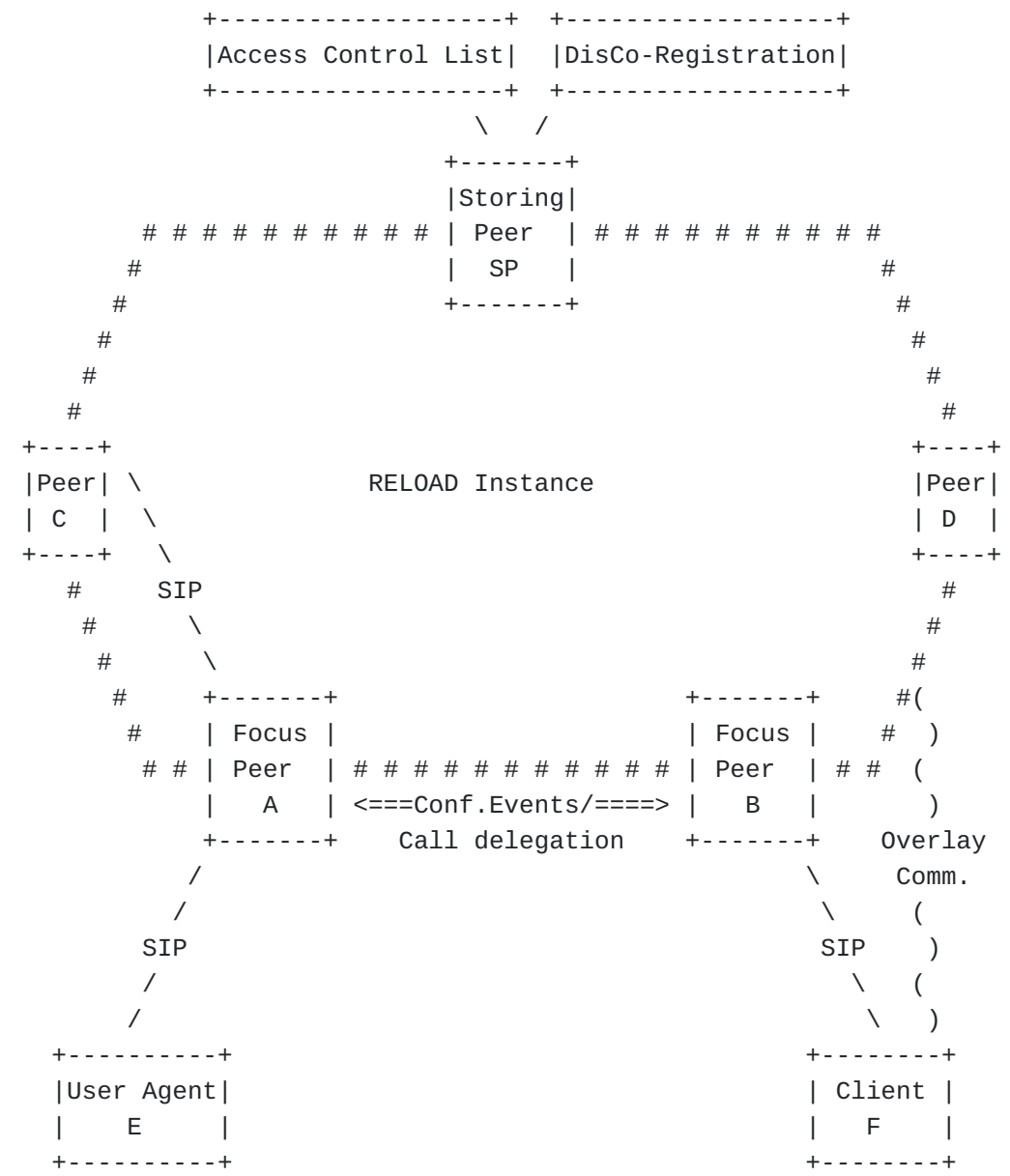
Focus peer: A RELOAD peer that provides SIP conferencing functions and implements the Usage for distributed conferencing. It is called 'active' if already involved in signaling relation to conference participants. Otherwise, if only registered in a distributed conference data structure, it is referred to as a 'potential' focus peer.

3. Overview of DisCo

3.1. Reference Scenario

The reference scenario for the Distributed Conference Control (DisCo) is shown in Figure 1. Peers are connected via a RELOAD [I-D.ietf-p2psip-base] instance, in which peers A and B are managing a single multiparty conference. The conference is identified by a unique conference URI, but located at peers A and B fulfilling the role of the focus. The mapping of the conference URI to one or more responsible focus peers is stored in a new RELOAD Resource for distributed conferencing within a data structure denoted as DisCo-Registration. The storing peer SP of the distributed conference resource holds this data. The focus peers A and B maintain SIP signaling relations to conference participants, which may have different conference protocol capabilities. In this example, peer A is the focus for the RELOAD peer C and the plain SIP user agent E whereas peer B serves as a focus for RELOAD peer D and the RELOAD client F.
RELOAD peers and clients obtain the contact information for the conference from the storing peer. In contrast to this, the user agent E receives the conference URI not by RELOAD mechanisms, but resolves the ID and joins the conference by plain SIP negotiation.

Focus peers maintain a SIP signaling relation among each other used for notification messages that synchronize the conference focus peers' knowledge about the entire conference state. Additionally, focus peers can transfer calls to each other by a call delegation mechanism.

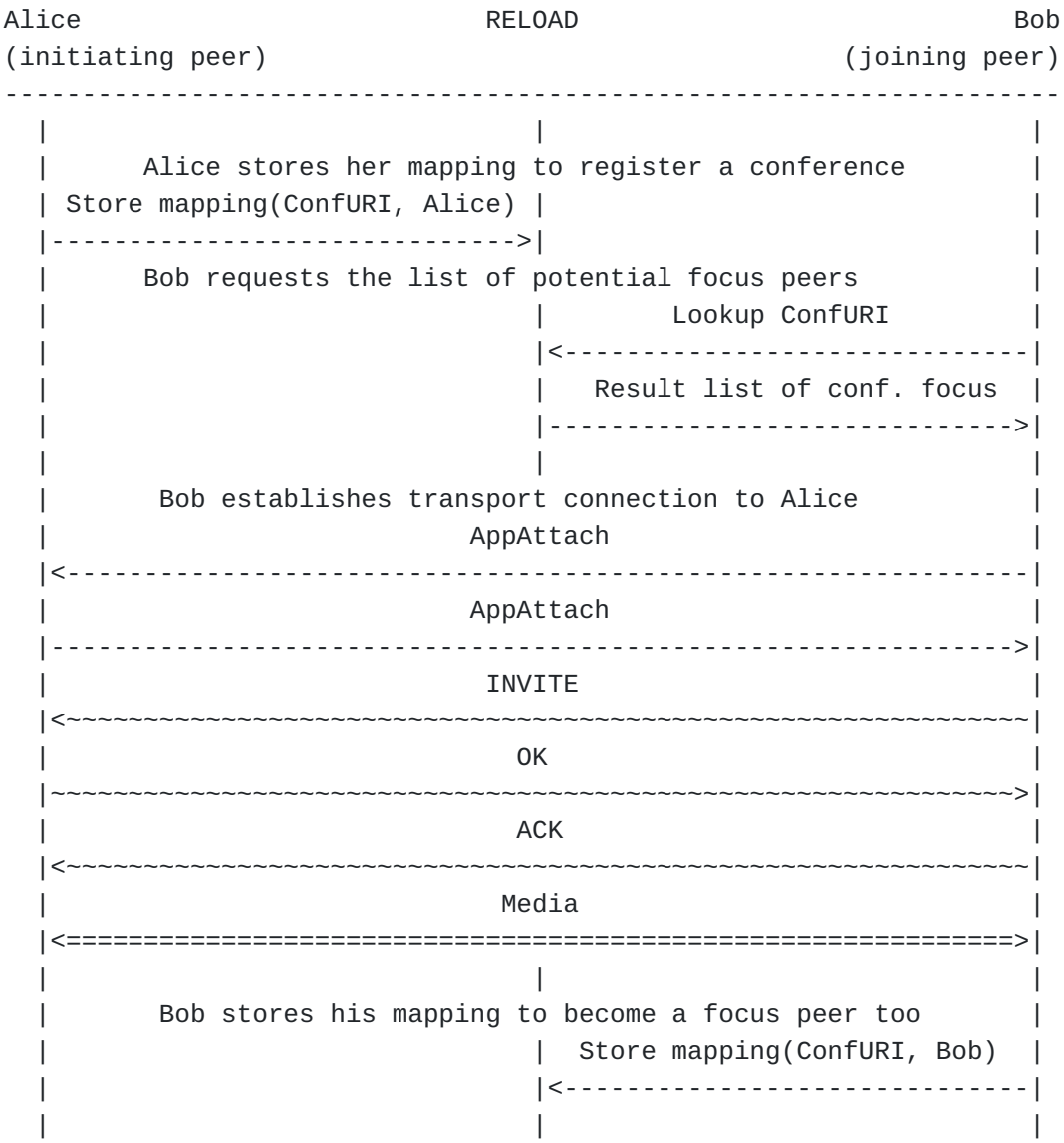


3.2. Initiating a Distributed Conference

To create a conference the initiating user agent announces itself as a focus for the conference. It stores its own contact information (Node-ID) as a DisCo-Registration Kind (cf. Figure 2) in the RELOAD overlay. The hashed conference URI is used as the Resource-ID. This Resource will later contain the contact IDs of all potential focus peers including optional topological descriptors.

3.3. Joining a Conference

A RELOAD-aware node (cf. Bob in Figure 2) intending to join an existing conference requests the list of potential focus peers stored in the DisCo-Registration under the conference's Resource-ID. The node selects any of the focus peers (e.g., Alice) and establishes a connection using AppAttach/ICE [RFC5245]. This transport is then used to send an INVITE to the conference applying the chosen focus as the contact. The selection of the focus peer can optionally be based on proximity information if available.
A conference member proposes itself as a focus for subsequent participants by adding its Node-ID to the DisCo-Registration stored under the conference URI in the RELOAD overlay. The decision whether a peer announces as a focus incorporates bandwidth, power, and other constraints, but details are beyond the scope of this document.



3.4. Conference State Synchronization

Each focus of a conference maintains signaling connections to its related participants independently from other conference controllers. This distributed conference design effects that the entire SIP conference state is jointly held by all focus peers. In DisCo, state synchronization is based on a SIP specific event notifications mechanism [RFC3265].
Each focus peer maintains its view of the entire conference state by subscribing to the other focus peers' XML event package for distributed conferences. This is based on the event package for conference state (cf. [RFC4575]). Details are defined in this document in Section 5.
Receivers of event notifications update their local conference state document to represent a valid view of current total conference state. The event notification package for distributed conferences enables focus peers to synchronize the entire conference state. The event package defines additional XML elements and complex types (see Section 8 for more details), which describe the responsibilities of any focus peer in the conference. By providing a global view each focus peer is enabled to perform additional load balancing operations and enhances the robustness against departures of focus peers.

3.5. Call delegation

Call delegation (see Section 6.1) is a feature used to transfer an incoming participation request to another focus peer. It can be applied to prevent overloading of focus peers. Call delegation is realized through SIP REFER requests, which carry signaling and session description information of the caller to be transferred. This feature is achieved transparently for the transferred user agent by using a source routing mechanism at SIP dialog establishment. Descriptions of overload detection are beyond the scope of this document.

3.6. Resilience

A focus peer can decide to leave the conference or may ungracefully fail. In a traditional conferencing scenario, loss of the conference controller or the media distributor would cause a complete failure of the multiparty conversation. Distributed conferencing uses the redundancy provided by multiple focus peers to reconfigure a current multiparty conversation. Participants that loose their entry point to the conference re-invite themselves via the remaining focus peers or will be re-invited by the latter. This option is based on the conference state and call delegation functions.

3.7. Topology Awareness

DisCo supports the optimization of the conference topology in respect of the underlying network using topological descriptors. An extension

for the RELOAD XML configuration document is defined in Section 4.8 to support landmarking approaches. Each peer intending to create or participate in a distributed conference MAY determine a topological descriptor that describes its relative position in the network. Focus peers store these coordinate values in an additional data field in the DisCo-Registration data structure. This enables peers joining the conference to select the closest focus with respect to its coordinate values.

4. RELOAD Usage for Distributed Conference Control

4.1. Shared Resource DisCo-Registration

A distributed conference is a cooperative service of several individual devices that use a common identifier. To enable a mapping from one conference identifier to multiple focus peers, this document defines the new Kind data structure DisCo-Registration. The DisCo Kind uses the definitions for Shared Resources [I-D.knauf-p2psip-share] to allow a jointly maintenance by multiple focus peers. Hence, write permission to a DisCo-registration is shared by the conference creator with all authorized focus peers.
DisCo complies with the following requirements for Shared Resources:

Isolated Data Storage: DisCo uses the dictionary data model. Each dictionary key is the Node-ID of the certificate that will be used to sign the stored data

ResourceNameExtension field: A DisCo-Registration can contain the ResourceNameExtension structure an initial field in the Kind data structure. It contains the conference URI of the registered DisCo-record.

4.2. Kind Data Structure

Each DisCo-Registration data structure stores a mapping of a conference identifier to one or multiple focus peers that cooperatively control the conference. Additionally, each DisCo-Registration provides the coordinate value, which indicates the relative network position of the focus peers.
The data structure uses the RELOAD dictionary type. The dictionary key MUST be the Node-ID of the focus peer that is associated with the dictionary entry. This allows a focus peer to update only its own mapping. The DisCo data structure of type DisCoRegistration is constructed as follows:

```
struct {
  /* This field is optional, see documentation */
  ResourceNameExtension res_name_ext;
  opaque coordinate<0..2^16-1>;
  NodeId node_id;
} DisCoRegistration;
```

The DisCoRegistration structure is composed of the following values:

res_name_ext: This field can contain the conference URI. It meets the requirement for the USER-CHAIN-ACL access policy defined in [I-D.knauf-p2psip-share] to enable variable resource names.

coordinate: This field contains a topological descriptor that indicates the relative position of the peer in the network. To support different algorithms the coordinate field is represented as an opaque string.

node_id: This field contains the Node-ID of the peer storing the DisCoRegistration and is used to contact the peer when utilizing its service as a focus.

4.3. Variable Conference Identifier

DisCo-Registrations can be stored based on a variable Resource Name. However, a conference identifier set by a user MUST follow the requirements for Kinds using variable Resource Names as defined in the ShaRe Usage [I-D.knauf-p2psip-share].

4.4. Conference Creation

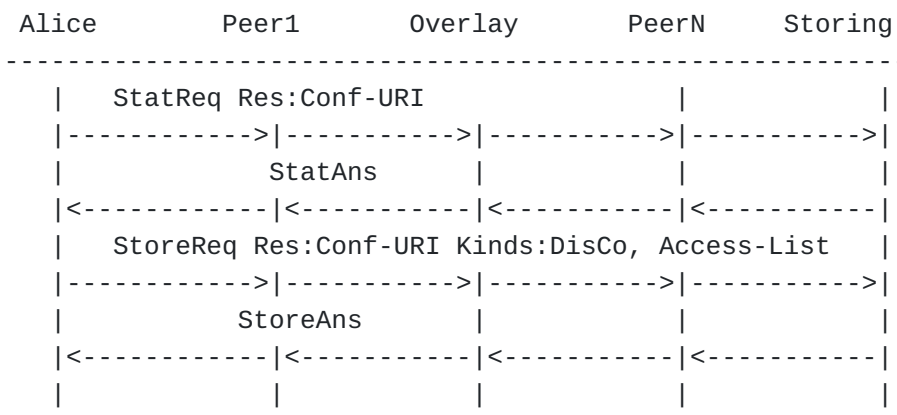
The registration of a distributed conference includes the storage of the following two Kinds (see Figure 4).

DisCo-Registration Kind: contains the conference identifier and the optional coordinate value. If used, the coordinate value MAY be determined previously to the conference registration. The Resource Name and hence the Resource-ID is defined by the hash over the desired conference identifier (using the hash algorithm of the overlay).

Access Control List Kind: contains the conference participants that are allowed to register as focus peers for a conference (see [I-D.knauf-p2psip-share]). Its Resource Name/ID is equal to those of the corresponding DisCo-Registration.

Preliminary to storage of DisCo-Registration and Access Control List (ACL) Kinds the conference creator SHOULD perform a RELOAD StatReq to verify that no former conference is present. If neither a DisCo-Registration nor an associated ACL exist, the conference creator stores

a DisCo-Registration with a valid conference identifier (see Section 4.3) and an ACL referring to the DisCo-Registration Kind-ID.
If DisCo registrations and ACL Kinds from previous conferences are still existing there are two options. First, if conference creator is aware of the indexes from previous ACL Kinds, it refreshes the root item of this ACL and stores its registration as focus peer as DisCo-Registration Kind. Second, If the creator is unaware of indexes, it fetches all Access List Kinds to determine the index of the root item.



Optionally the conference initiator (or any active focus) MAY store an additional RELOAD SIP-Registration in the overlay [I-D.ietf-p2psip-sip] or even at a standard SIP registrar [RFC3261] under a URI for which it has write permission. This allows DisCo-unaware or even legacy SIP user agents to participate in the conference. Those registrations SHOULD always point to a currently active focus, who is prepared to accept legacy user agents. The user agent who initially performed the registrations is responsible for updating them appropriately. How authorization has been obtained to perform those registration is out of scope of this document.
The lifetime of a distributed conference is not necessarily limited by the participation time of its creator. As long as the root item of an Access Control List to a DisCo-Registration is not expired, Authorized Peers are allowed to further provide a conferencing service and even store new focus registrations.

4.5. Advertising Focus Ability

All participants of a distributed conference MAY potentially become a focus peer for their conference. This depends on its capacities such as sufficient processing power (CPU, Memory) for the desired media type, the quality of the network connectivity, and the conference policy. Information about network connectivity with respect to NAT or restricted firewalls can be obtained via ICE [RFC5245] connectivity checks. If a peer is behind a NAT, it SHOULD allow for incoming

connections via AppAttach/ICE. Peers that can only accept connections with the support of TURN should not act as a focus. Nevertheless, under special circumstances it might be reasonable to locate a focus peer behind such a NAT (e.g., within a companies network infrastructure). If a participant is a candidate to become a focus for the conference, it stores its Node-ID and optional coordinate value into the DisCo data structure. An entry in the corresponding ACL [I-D.knauf-p2psip-share] must be present to allow this peer to write the DisCo-Registration resource. By storing the mapping into the data structure a participant becomes a potential focus.

4.6. Determining Coordinates

Each RELOAD peer within the context of a distributed conference MAY be aware of its relative position in the network topology. To construct a topology-aware conference, the DisCo Usage provides the coordinate value within the DisCo-Registration data structure. Focus peers store their relative network position together with the announcement as conference focus. Joining peers that are able to determine their coordinates may then select a focus peer whose relative position is in its vicinity (see Section 4.7).
Some algorithms determine topology information by measuring Round-Trip Times (RTT) towards a set of hosts serving as landmarks (e.g., [landmarks-infocomm02]). To support such algorithms this document describes an extension to the RELOAD XML configuration document that allows to configure the set of landmark hosts peer must use for position estimation (see Section 4.8). Once a focus peer has registered its mapping in the DisCo data structure, it also stores the according coordinates in the same mapping. These <Node-ID,coordinates> vectors are used by peers joining the conference to select the focus peer that is relatively closest to itself.
Because topology-awareness can be obtained by many different approaches a concrete algorithms is out of scope of this document.

4.7. Proximity-aware Conference Participation

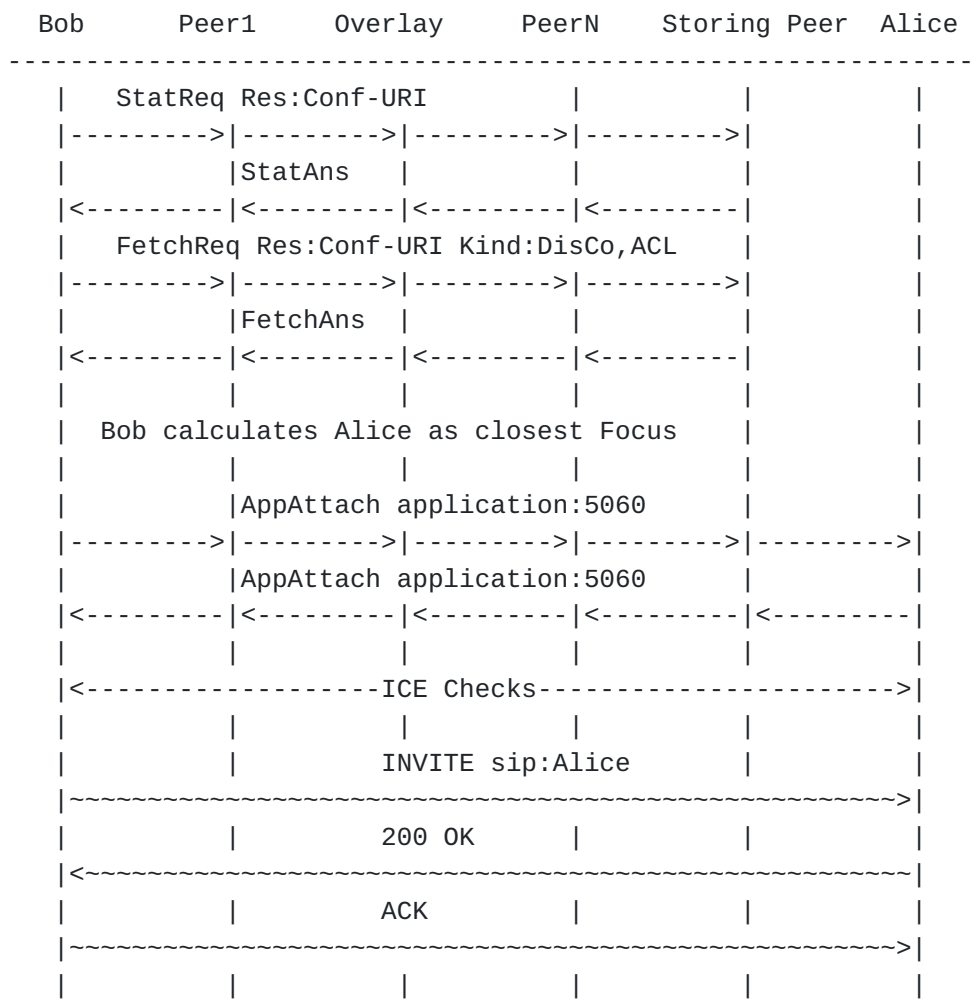
The participation procedure to a distributed conference is composed of three operation.

1. Resolution of the conference identifier.

2. Establishment of of transport connection.

3. SIP signaling to join a conference.

Figure 5 and the following specifications give a more detailed view on the joining procedure.



1. The joining peer MAY determine its own coordinate value (if used).

2. The joining peer sends a StatReq message to obtain all indexes of the Access Control List (ACL) Kinds stored.

3. The joining peer sends a FetchReq message for the DisCo and ACL Kind to the Resource-ID of the conference URI. The FetchReq SHOULD NOT include any specific dictionary keys, but SHOULD fetch for those array ranges previously determined the StatReq. With the ACL items, the joining peer is able to verify whether DisCo-Registrations are stored by authorized focus peers (see [I-D.knauf-p2psip-share]).

4. Using the retrieved coordinates values of the DisCo-Registrations, the joining peer MAY calculate which of than opaque <0..2^16-1> initial field in the Kind data structure focus peers is the relatively closest to itself.

5. The joining peer then establishes a transport using RELOADs AppAttach, respectively, ICE procedures to create a transport to the chosen focus peer.

6. Finally, the established transport is used to create a SIP dialog from the joining peer to the focus peers.

The SIP INVITE request MAY contain the joining peer's topological descriptor as a URI-parameter called 'coord' in the contact-header in base64 encoded form [RFC4648]. An example contact URI is “sip:alice@example.com;coord=PEknbSBhIHRvcG9sb2dpY2FsIGRlc2NyaXB0b3I+”. When the called focus is full and needs to delegate the call it uses the contents of the 'coord'-parameter. It determines the next available focus closest to the calling peer (Section 4.6) using the received descriptor and the other focuses' descriptors from the conference state synchronization document and delegates the call accordingly (see Section 6.1).
A conference focus MAY allow the joining peer to also become a focus (depending on the conference policy see Section 6.2). Therefore, it stores a new ACL Kind that delegates write permission for the DisCo-Registration to the new participant. It sets the 'user_name' field in the ACL Kind to its own user name and sets the 'to_user' field to the user name of the joining peer. If no other conference policy is defined, the focus peer MAY set the allow_delegation flag to true. For further details about the trust delegation using the ACL Kind see [I-D.knauf-p2psip-share].

4.8. Configuration Document Extension

This section defines an additional parameter for the <configuration> element that extends the RELOAD XML configuration document. The proposed <landmarks> element allows RELOAD provider to publish a set of accessible and reliable hosts that SHOULD be used if RELOAD peers use landmarking algorithms to determine relative position in the network topology.
The <landmarks> element serves as container for the <landmark-host> sub-elements, each representing a single host that serves as a landmark. The <landmark-host> uses the following attributes:

address: The IP address (IPv4 or IPv6) of the landmark host.

port: The port on which the landmark host responds for distance estimation.

More than one landmark hosts SHOULD be present in the configuration document.

5. Conference State Synchronization

The global knowledge about signaling and media relations among the conference participants and focus peers defines the global state of a distributed conference. It is composed of the union of every local state at the focus peers. To enable focus peers to provide conference control operations that modify and/or require the global state of a conference, this document defines a mechanism for inter-focus state synchronization. It is based on mutual subscriptions for an Event Package for Distributed Conferences and allows to preserve a coherent knowledge of the global conference state. This XML based event package named 'distributed-conference' MUST be supported by each RELOAD peer that is registered with a DisCo-Registration. Participants of a distributed conference MAY support it. To provide backward compatibility to conference members that do not support the distributed-conference event package, this document defines a translation to the Event Package for Conference State [RFC4575].

5.1. Event Package Overview

The 'distributed-conference' event package is designed to convey information about roles and relations of the conference participants. Conference controllers obtain the global state of the conference and use this information for load balancing or conference restructuring mechanisms in case of a focus failure. Figure Figure 6 gives a general overview of the document hierarchy.

```
distributed-conference
  |
  |-- version-vector
  |    |-- version
  |    |-- version
  |
  |-- conference-description
  |
  |-- focus
  |    |-- focus-state
  |    |    |-- user-count
  |    |    |-- coordinate
  |    |    |-- maximum-user-count
  |    |    |-- active
  |    |    |-- locked
  |    |    |-- conf-uris
  |    |    |-- available-media
  |    |
  |    |-- users
  |    |    |-- user
  |    |    |    |-- endpoint
  |    |    |    |    |-- media
  |    |    |    |    |-- call-info
  |    |
  |    |-- relations
  |    |    |-- relation
  |-- focus
  |    |-- ...
```

The document structure is designed to allow concurrent change events at several focus peers. To prevent race conditions each focus peer has exclusive writing permission to the 'focus' sub element that describes itself. It is achieved by a unique mapping from a focus peer to its XML element using the 'Element Keys' mechanisms for partial notification [RFC4575](sections 4.4-5.). A focus peer is only allowed to update or change that <focus> sub element, whose 'entity' Element Key contains its RELOAD user name. This restriction also applies to the child elements of the 'version-vector' element. Each 'version' number belongs to a specific focus peer maintaining the version number.

The local state of a focus peer is described within a 'focus' element. It provides general information about a focus peer and its signaling and media relations to participants and focus peers. The Conference participants are aggregated within 'users' elements, respectively, 'user' sub elements.

General information about the conference as a whole, is provided within a 'conference-description' element. In contrast to the 'focus' and 'version-vector' elements, conference description is not meant for

concurrent updating. Instead, it provides static conference descriptions that rarely change during a multiparty session.
More detailed descriptions about the event package and its elements are provided in the following sections. The complete XML schema definition is given in Section 8.

5.2. <distributed-conference>

The <distributed-conference> element is the root of a distributed conference XML document. It uses the following attributes:

entity: This attribute contains the conference URI that identifies the distributed conference. A SIP SUBSCRIBE request addressed to this URI initiates an subscribe/notify relation between participants and their related focus peer.

state: This attribute indicates whether the content of a distributed conference document is a 'full', 'partial' or 'deleted' information. It is in accordance with [RFC4575] to enable the partial notification mechanism.

The <distributed-conference> child elements are <vector-version>, <conference-description> and the <focus> elements. An event notification of state 'full' MUST include all these elements. An event notification of state 'partial' MUST contain at least <version-vector> and its sub elements.

5.3. <version-vector>/<version>

The Event Package for Distributed Conferences uses the <version-vector> and its <version> sub elements to enable a coherent versioning scheme. It is based on vector clocks as defined in [timestamps-acsc88]. Each <version> element contains a unsigned integer that describes the state of a specific focus peer and contains the following attributes:

entity: This attribute contains the user name of the focus peer whose local version number is described by this element.

node-id: This attribute contains the Node-ID of the focus peer.

Whenever the local status of a focus peer changes (e.g. a new participant arrived) the version number of the corresponding <version> element MUST be incremented by one. Each change in the local state also triggers a new event notification containing the entire <version-vector> and the changes contained in a <focus> element.
The recipient of a change event needs to update it local XML document. If a received <version> number is higher that the local, it updates the <version> element and its associated <focus> element with the retrieved elements. All other elements remain constant.

If the length or contents of the retrieved <version-vector> is different to the local copy it indicates a incoherent knowledge about the entire conference state. If the retrieved <version-vector> contains any unknown focus peers and any local version numbers for the known focus peers is lower, the receiver SHOULD request a 'full' XML notification.
If any local <version> number is retarded more than one, the receiver SHOULD request a 'full' event notification from the sender. The full state notification updates all <focus> elements whose corresponding <version> element is out of date.

5.4. <conference-description>

The <conference-description> element provides general information and links to auxiliary services for the conference. The following sub elements provide human-readable text descriptions about the conference:

<display-text>: Contains a short textual description about the conference

<subject>: Contains the subject of the conference

<free-text>: Contains a longer textual description about the conference

<keywords>: Contains a list of keywords that match the conference topic. The XML schema definition and semantic is imported from the 'conference-info' event package [RFC4575].

The <service-uris> sub element enables focus peers to advertise auxiliary services for the conference. The XML schema definition and semantic is imported from the 'conference-info' event package [RFC4575].
The <conference-description> element uses the 'state' Element Key to enable the partial notification mechanism.

5.5. <focus>

Each <focus> element describes a focus peer actively controlling the conference. It provides general information about a focus peer (e.g., display-text, languages, etc.), contains conference specific information about the state of a focus peer (user-count, available media types, etc.) and announces signaling and media information about the maintained participants. Additionally, it describes signaling or media relations to further focus peers.
The <focus> element uses the following attributes:

entity: This attribute contains the user name of the RELOAD peer acting as focus peer. It uniquely identifies the focus peer that is allowed to update or change all sub elements of <focus>. All other

focus peers SHOULD NOT update or change sub elements of this <focus> element. A SUBSCRIBE request addressed to the user name initiates a conference state synchronization with the focus peer.

Node-ID This attributed contains the Node-ID of the peer acting as conference focus

state: In accordance to [RFC4575], this attribute indicates whether the content of the <focus> element is a 'full', 'partial' or 'deleted' information. A 'partial' notification contains at maximum a single <focus> element.

The following sub elements of <focus> provide general information about a focus peer: [RFC4575]

<display-text>: Contains a short text description of the user acting as focus peer.

<associated-aors>: This element contains additional URIs that are associated with this user.

<roles:> This element MAY contain human-readable text descriptions about the roles of the user in the conference.

<languages>: This element contains a list of tokens, each describing a language understood by the user.

The XML schema definition and semantic for <associated-aors>, <roles> and <languages> are imported from the 'conference-info' event package Following, a detailed description of the remaining sub elements.

5.5.1. <focus-state>

The <focus-state> element aggregates a set of conference specific information about the RELOAD user acting as focus peer. The following attribute is defined for the <focus-state> element:

status: This attribute indicates whether the content of the <focus-state> element is a 'full', 'partial' or 'deleted' information.

The <focus-state> element has the following sub elements:

<user-count>:
This element contains the number of participants that are connected to the conference via this focus peer at a certain moment.

<coordinate> This element contains the coordinate value Section 4.2 of the focus peer

<maximum-user-count>: This number indicates a threshold of participants a focus peer is able to serve. This value might change during a conference, depending on the focus peers current load.

<conf-uris>: This element MAY contain other conference URIs in order to access the conference via different signaling means. The XML schema definition and semantic is imported from [RFC4575].

<available-media>: This element is imports the <conference-media-type> type XML scheme definitions from [RFC4575]. It allows a focus peer to list its available media streams.

<active>: This boolean element indicates whether a focus peer is currently active. Conference participation requests or a call delegation request SHOULD succeed.

<locked>: In contrast to <active>, this element indicates that a focus peer is not willing to accept anymore participation or call delegation request.

5.5.2. <users>/<user>

The <users>, respectively, each <user> element describes a single participant that is maintained by the focus peer described by the parent <focus> element. The <users> element XML schema definition and its semantic is imported from the 'conference-info' event package [RFC4575].

5.5.3. <relations>/<relation>

The <relations> element serves as container for <relation> elements, each describing a specific connection to another focus peer. The parent element <relations> uses the 'state' attribute to enable the partial notification mechanism. For the <relation> element the following attributes are defined:

entity: This attribute contains the user name of the remote focus.

node-id This attribute contains the Node-ID of the remote focus peer.

The content of each <relation> is a comma separated string that describes the tuple <CONNECTION-TYPE:IDENTIFIER>. The CONNECTION-TYPE

is a string token describing the type of connection (e.g. media, signaling, etc.) and the IDENTIFIER contains a variable connection identifier. It is a generic method to announce any kind of connection to a remote focus. This specification defines following tuples:

media:<label>: This tuple identifies a single media stream. The 'label' variable contains the SDP "label" attribute. (see [RFC4574]).

sync:<call-id>: This tuple indicates a subscription for the Event Package for Distributed Conferences. The 'call-id' variable contains the call-id of the SIP subscription dialog.

5.6. Distribution of Change Events

Each focus peer in a distributed conference must be able to retrieve all change events from all other focus peers. A simple approach would be to inter-connect each focus with all other focus in a full meshed topology. To avoid a full mesh, this document describes a 'mutual' subscription scheme that constructs a spanning tree topology among the focus peers.

A conference participant that becomes a focus peer MUST send a SIP SUBSCRIBE to request the Event Package for Distributed Conferences to its own focus peer. The subscription request is addressed to user name of the active focus peer. The latter interprets this subscription as a request for conference state synchronization. The corresponding NOTIFY message contains a 'full' distributed-conference state XML document (see section Section Section 5.1).

The subscribed focus peer in turn subscribes the upcoming focus peer for the distributed conference event package. The corresponding NOTIFY message carries a 'partial' conference state XML document. It contains the received <version-vector> including a new <version> element for itself and a new <focus> element that describes its local state (see Section 5.5).

Resulting by this subscription scheme, each focus peer has at least one subscription to obtain updates for the conference state and is a notifier for change events originated itself. In a incrementally increasing conference, the 1st and 2nd focus peer have a mutual subscription for conference change events. A 3rd focus could have a mutual subscription with the 1st focus, a 4th focus to the 2nd focus and so forth. The result is a spanning tree topology among the focus peers in which each focus peer is a possible root for distribution tree for conference change events.

However, the fact that event notifications need to traverse one or more intermediate focus peers until conference-wide delivery, demands a forwarding mechanism for change events. On receiving a change event, a notified focus validates based on the <version-vector> whether the incoming state event is not a duplicate to previews notifications. If its not a duplicate, the received change event triggers a new event

notification at the receiver of the change event. It notifies all its subscribers with excepting the sender of the event notification. In this way, the change event will be 'flooded' among the focus peer of a conference.

5.7. Translation to Conference-Info Event Package

The Event Package for Distributed Conferences imports several XML element definitions of the Event Package for Conference State [RFC4575]. This is caused by two reasons. First, the semantic of these elements are fitting the demands to describe the global state of a distributed conference and, second, it facilitates a re-translation to [RFC4575] to enable a backward compatibility to DisCo-unaware clients. Therefore, each focus peer MAY provide a separate [RFC4575] conform event notification service to its connected participants.
The following sections describe the translation to the Event Package for Conference State [RFC4575] by defining translation rules for the root element and its direct sub elements. For a better understanding, the following sections use a notation ci.<ELEMENT> to refer to a sub element of the conference-info element, and disco.<ELEMENT> to refer to an element of the distributed-conference event package.

5.7.1. <conference-info>

The root element of Event Package for Conference State uses the attributes 'entity', 'state' and 'version' and is the counterpart of the <distributed-conference> root element in the DisCo Event Package. The former two attributes 'entity' and 'state' are equal in both root elements and can be seamlessly translated.
According to [RFC4575], the 'version' attribute SHOULD be incremented by one at any time a new notification being sent to a subscriber. Hence, in DisCo the 'version' attributed increments with each change event that originated by focus peer and each reception of a change events of remote focus peer.

5.7.2. <conference-description>

The <conference-description> element exists in both event packages, conference-info and distributed-conference. Thus, the following elements are seamlessly translatable: <keywords>, <display-text>, <subject>, <free-text> and <service-uris>.
The sub elements <conf-uris>, <maximum-user-count> and <available-media> in conference-info have there counterparts below the \focus\focus-state element of the distributed-conference event package. Each describes a local state of a focus peer in the conference. Hence, the intersection of every disco.<conf-uris>, disco.<available-media> and the sum over each disco.<maximum-user-count> element of each disco.<focus> element in distributed-conference, specifies the content of the corresponding conference-info elements.

5.7.3. <host-info>

According to [RFC4575] the ci.<host-info> element contains information about the entity hosting the conference. For participants in a distributed conference, the hosting entity is their focus peer. Thus, the ci.<host-info> element contains information about a focus peer.

5.7.4. <conference-state>

The ci.<conference-state> element allows subscribers obtain information about overall state of a conference. Its sub elements ci.<user-count>, ci.<active> and ci.<locked> are reused as sub elements of \focus\focus-state to describe the local state of a focus peer in a distributed conference. The translation rules from the distributed-conference to the conference-info event package are the following:

<user-count>: The sum over each value of the disco.<user-count> element defines the corresponding ci.<user-count>.

<active>: The boolean ci.<active> element is the logical concatenation over all disco.<active> elements by an OR-operator.

<locked> The boolean ci.<locked> element is the logical concatenation over all disco.<locked> elements by an AND-operator.

5.7.5. <users>/<user>

The distributed-conference event package imports the definitions of the ci.<users> and ci.<user> elements under a parent disco.<focus> element for each focus peer in a conference. Thus, the aggregation over all disco.<users> elements specifies the content of the corresponding ci.<users> element.

5.7.6. <sidebars-by-ref>/<sidebars-by-value>

In accordance to [RFC4575], if a participant is connected to a sidebar, its responsible focus peer creates a new <user> by referencing to the corresponding sidebar conference.

6. Distributed Conference Control with SIP

Distributed conference control with SIP defined in this document refers to multiparty conversation in a tightly coupled model that is controlled by several independent entities. It enables a resilient conferencing service for P2P scenarios and provides mechanisms for load-balancing among the focus peers. This section describes additional control operations for distributed conferences with SIP.

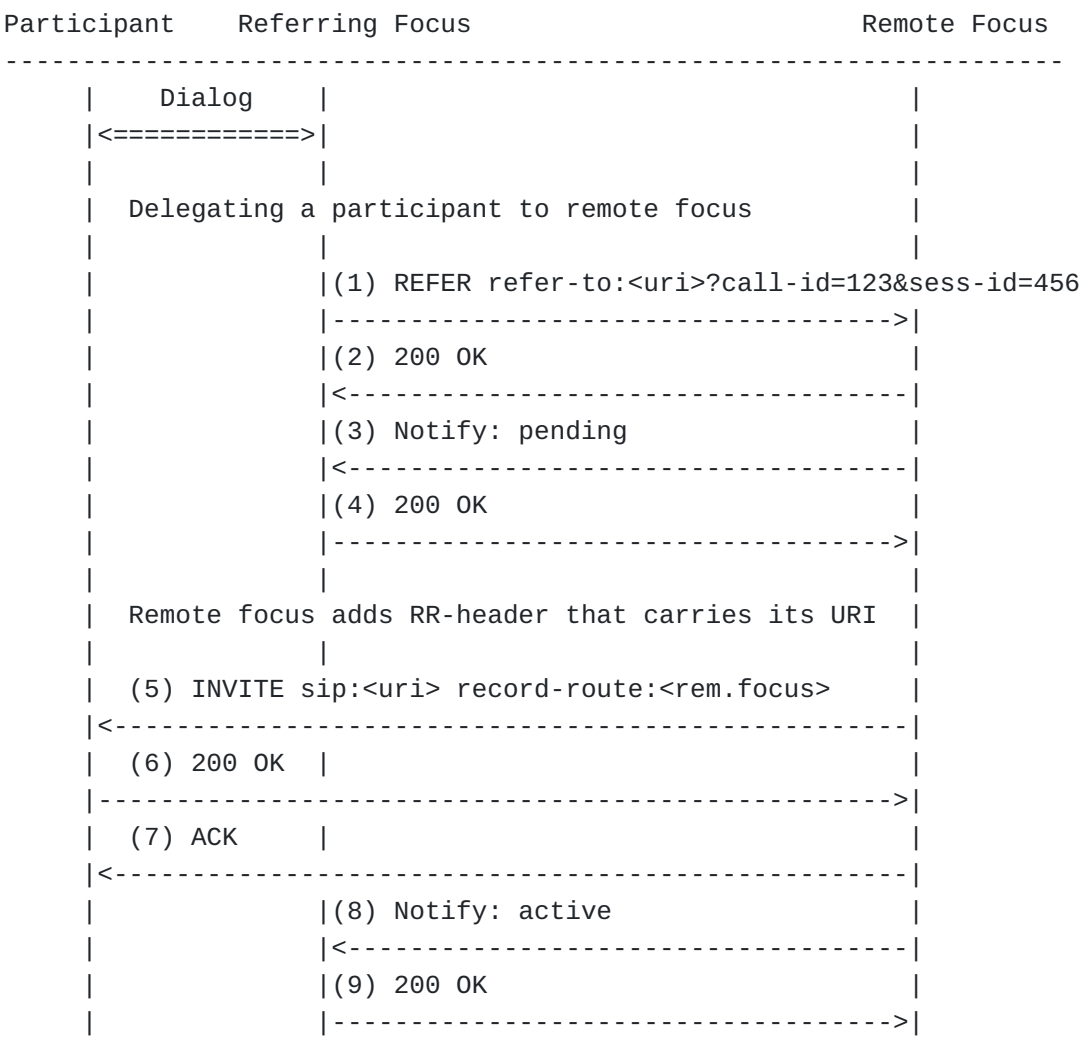
6.1. Call delegation

Distributed conference control enables load-balancing by a mechanism for call delegation. Call delegations are performed by focus peers that are running out of capacities to serve more participants. Incoming participation requests are then transferred to other established focus peer or conference participants that are registered as potential focus peers in the overlay. Call delegations use SIP REFER requests [RFC3515] that contain additional session information and are achieved transparently to the transferred party.
A focus peer initiates a call delegation by sending SIP REFER request containing the URI of the participant in the Refer-To header field. Additionally, the focus peer appends the following parameter to the URI of the participant:

call-id: Contains the call-ID of the initial SIP dialog between the referred participant and the referring focus peer.

sess-id: Contains the 'session identifier' value of the original SDP 'o=' field of the original offer/answer process between referred participant and referring focus peer.

If the recipient accepts the REFER request, it generates a re-INVITE towards the referred party and sets the SIP call-id header and the SDP 'session-identifier' field in the SDP offer, according to the URI parameter values of the initial REFER request. The From header field and contact header are set to the conference URI with setting the 'isfocus' tag to contact header. This identifies the peer as a focus to the conference and identifies this re-INVITE as a request of the SIP dialog between the party and the conference. To ensure that further signaling messages will be routed correctly, the new focus adds a Record-Route header field that contains its contact information (URI, IP-address,..).
An example call flow for call delegation is shown in Figure 7.



Note, subscriptions for the event packages 'distributed-conference' and 'conference-info' are in scope of a specific focus peer and its connected participants. Hence, after a successful call delegation, the referring focus peer SHOULD terminate any subscription to the referred participant. The notifier SHOULD include a reason parameter "deactivated" to indicate a migration of the subscription as defined in [RFC3265]. The new SUBSCRIBE request by the party MUST be sent via the SIP dialog to the conference.

6.2. Conference Access

A conference policy defines who is allowed to participate in a multimedia conference. In many cases, a group conversation can be an open discussion free to participate, while in other occasions a closed privacy of a multiparty session is demanded. In distributed conferences, it is also an issue which of the conference participants is allowed to become a controller of the multiparty session.
Thus it must be decided whether:

*A peer is allowed to participate in a conference

*A peer is allowed to become a focus of the conference

Standard SIP authentication mechanisms can be used to authenticate and accordingly authorize joining participants.

6.3. Media Negotiation and Distribution

This section describes a basic scheme for media negotiation and distribution, which is done in an ad-hoc fashion. Each focus peer forwards all media streams it receives from the conference to all connected peers it is responsible for and similarly all streams from its peers to its responsible focus. This results in the media stream naturally following the SIP signalling paths. Implementations MAY choose to use a more sophisticated scheme, e.g. employing cross connections between different sub-trees of the conference, but MUST take measures to prevent loops in media routing.
When a new peer has been attached to a focus, new media steams may be available to the focus, which need to be forwarded to the conference. To accomplish this, the new media streams need to be signalled to the other participants. This is usually done by sending a SIP re-INVITE and modifying the media sessions, adding the new streams to the SDP. This renegotiation can be costly since it needs to be propagated through the whole conference. Also, distributing all media streams separately to all participants can be quite bandwidth intensive. Both problems can partially be mitigated by focus peers performing mixing of media streams, thus trading bandwidth and signalling overheads for computational load on focus peers.

6.3.1. Offer/Answer

A peer joining a conference negotiates media types and media parameters with its designated focus using the standard SDP offer/ answer protocol [RFC3264]. The focus SHOULD offer all existing media streams that it receives from the conference.
A new participant does not necessarily know about all media streams present in a conference beforehand, and thus some of the media streams might not be included in the initial SDP offer sent by the joining peer. An SDP answer sent by the corresponding focus MUST NOT contain any media streams not matching an offer (cf. [RFC3264] Section 6). A joining peer which is aware of the distributed nature of the conference, SHOULD NOT send an SDP offer in the initial INVITE message, but instead send an empty INVITE to which the focus replies with an OK, containing the offer. This prevents the need for a second offer/answer-dialog to modify the session. But for compatibility the normal behavior with the INVITE containing the offer MUST be supported.
For new media streams (e.g. those sent by the new participant) the focus SHOULD only offer media types and codecs which are already used in the conference and which will probably be accepted when forwarded to

neighboring peers, unless the focus is prepared to do transcoding and/or mixing of the received streams.
A focus has two options when distributing media streams from a new participant to the conference. The focus can either mix the new streams into his own, thus averting the need to modify the already established media sessions with neighboring peers or in case the focus is not willing or able to do mixing of the media streams, he SHOULD modify the media sessions with all attached peers by sending a re-INVITE, adding the new media streams coming from the newly joined participant to the SDP. This SHOULD subsequently be done by all other focus peers upon receiving the new stream, resulting in the stream being distributed across the conference.

6.4. Restructuring a Conference

Distributed conference control provides the possibility to delegate calls to remote focus peers. This feature is used to restructure a conference in case of departure of a focus peer. Following, this section presents restructuring schemes for graceful and unexpected leaves of conference focus peers.

6.4.1. On Graceful Leave

In a graceful case, the leaving focus peer (LP) accomplishes the following procedure:

* LP deletes its mapping in the DisCo-Registration by storing the "non-existing" value as described in the RELOAD base document [I-D.ietf-p2psip-base]. Afterwards, it fetches the lasted version of the DisCo-Registration to obtain all potential focus peers.

* LP calculates for all its participants the closest focus among all active and potential focus peer using the algorithm described in Section 4.6. LP then delegates all participants to those focus peers.

* LP then announces its leave by sending a NOTIFY to all its subscribers for the extended conference event package, setting its <focus> state to 'deleted'. Thereafter, it ends its own SIP conference dialog by sending by to its related focus peer.

Since the state synchronization topology in a distributed conference is commonly arranged in a spanning tree, a leave of a focus peer effects a gab in the tree structure. Those focus peers which had the leaving focus peer as their parent, are supposed to reconnect to the synchronization graph by subscribing the parent focus of the leaving peer.

6.4.2. On Unexpected Leave

If an unexpected leave is detected by a participant (e.g. missing signaling and/or media packets) it MUST repeat the joining procedure as described in Section 4.7.

7. DisCo Kind Definition

This section formally defines the DisCo kind.

Name
: DISCO-REGISTRATION

Kind IDs
: The Resource name DISCO-REGISTRATION Kind-ID is the AoR of the conference. The data stored is the DisCoRegistrationData, that contains the Node-ID of a peer acting as a focus for the conference and optionally a coordinates value describing a peer's relative network position.

Data Model
: The data model for the DISCO-REGISTRATION Kind-ID is dictionary. The dictionary key is the Node-ID of the peer action as focus.

Access Control
: USER-CHAIN-ACL

The data stored for the Kind-ID DISCO-REGISTRATION is of type DisCoRegistration. It contains a "coordinates" value, that describes the peers relative network position and the Node-ID of the registered focus peer.

8. XML Schema

The XML schema for the event package for distributed conferences is:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns:ci="urn:ietf:params:xml:ns:conference-info"
     xmlns="urn:ietf:params:xml:ns:distributed-conference"
     targetNamespace="urn:ietf:params:xml:ns:distributed-conference"
     elementFormDefault="qualified"
     attributeFormDefault="unqualified">
<!--
    This imports the definitions in conference-info
-->
<xs:import namespace="urn:ietf:params:xml:ns:conference-info"
        schemaLocation="http://www.iana.org/assignments/xml-registry/
                       schema/conference-info.xsd"/>
<xs:import namespace="http://www.w3.org/XML/1998/namespace"
     schemaLocation="http://www.w3.org/2001/03/xml.xsd"/>
  <!--
   A DISTRIBUTED CONFERENCE ELEMENT
-->
<xs:element name="distributed-conference"
            type="distributed-conference-type"/>
  <!--
    DISTRIBUTED CONFERENCE TYPE
-->
  <xs:complexType name="distributed-conference-type">
    <xs:sequence>
      <xs:element name="version-vector"
            type="version-vector-type" minOccurs="1"/>
      <xs:element name="conference-description"
      type="conference-description-type"
      minOccurs="0" maxOccurs="1"/>
      <xs:element name="focus"
      type="focus-type"
      minOccurs="0"
      maxOccurs="unbounded"/>
       <xs:any namespace="##other" processContents="lax"/>
     </xs:sequence>
     <xs:attribute name="state" type="ci:state-type"/>
     <xs:attribute name="entity" type="xs:anyURI"/>
     <xs:anyAttribute namespace="##other" processContents="lax"/>
  </xs:complexType>
<!--
    VERSION VECTOR TYPE
-->
  <xs:complexType name="version-vector-type">
    <xs:sequence>
      <xs:element name="version"
                  type="version-type"
                  minOccurs="1"
                  maxOccurs="unbounded"/>
```

```
      <xs:any namespace="##other" processContents="lax"/>
    </xs:sequence>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
  </xs:complexType>
<!--
    CONFERENCE DESCRIPTION TYPE
-->
  <xs:complexType name="conference-description-type">
    <xs:sequence>
      <xs:element name="display-text"
                  type="xs:string" minOccurs="0"/>
      <xs:element name="subject" type="xs:string" minOccurs="0"/>
      <xs:element name="free" type="xs:string" minOccurs="0"/>
      <xs:element name="keywords"
                  type="ci:keywords-type" minOccurs="0"/>
      <xs:element name="service-uris"
                  type="ci:uris-type" minOccurs="0"/>
      <xs:any namespace="##other" processContents="lax"/>
    </xs:sequence>
    <xs:attribute name="state" type="ci:state-type"/>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
  </xs:complexType>
<!--
    FOCUS TYPE
-->
  <xs:complexType name="focus-type">
    <xs:sequence>
      <xs:element name="display-text"
                  type="xs:string" minOccurs="0"/>
      <xs:element name="associated-aors"
                  type="ci:uris-type" minOccurs="0"/>
      <xs:element name="roles"
                  type="ci:user-roles-type" minOccurs="0"/>
      <xs:element name="languages"
                  type="ci:user-languages-type" minOccurs="0"/>
      <xs:element name="focus-state"
                  type="focus-state-type" minOccurs="0"/>
      <xs:element name="users"
                  type="ci:users-type" minOccurs="0"/>
      <xs:element name="relations"
                  type="relations-type" minOccurs="0"/>
      <xs:any namespace="#other" processContents="lax"/>
    </xs:sequence>
    <xs:attribute name="entity" type="xs:anyURI"/>
    <xs:attribute name="node-id" type="xs:string"/>
    <xs:attribute name="state" type="ci:state-type"/>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
  </xs:complexType>
<!--
```

```
    VERSION TYPE
-->
  <xs:complexType name="version-type">
    <xs:simpleContent>
      <xs:extension base="xs:unsignedInt">
        <xs:attribute name="entity" type="xs:anyURI"/>
        <xs:attribute name="node-id" type="xs:string"/>
        <xs:anyAttribute namespace="##other" processContents="lax"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
  <!--
    FOCUS STATE TYPE
-->
  <xs:complexType name="focus-state-type">
    <xs:sequence>
      <xs:element name="user-count"
                  type="xs:unsignedInt" minOccurs="0"/>
      <xs:element name="coordinate"
                  type="xs:string" minOccurs="0"/>
      <xs:element name="maximal-user-count"
                  type="xs:unsignedInt" minOccurs="0"/>
      <xs:element name="conf-uris"
                  type="ci:uris-type" minOccurs="0"/>
      <xs:element name="available-media"
                  type="ci:conference-media-type" minOccurs="0"/>
      <xs:element name="active" type="xs:boolean" minOccurs="0"/>
      <xs:element name="locked" type="xs:boolean" minOccurs="0"/>
      <xs:any namespace="##other" processContents="lax"/>
    </xs:sequence>
    <xs:attribute name="state" type="ci:state-type"/>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
  </xs:complexType>
  <!--
    RELATIONS TYPE
-->
  <xs:complexType name="relations-type">
    <xs:sequence>
      <xs:element name="relation"
                  type="relation-type"
                  minOccurs="0" maxOccurs="unbounded"/>
      <xs:any namespace="##other" processContents="lax"/>
    </xs:sequence>
    <xs:attribute name="state" type="ci:state-type"/>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
  </xs:complexType>
  <!--
    RELATION TYPE
-->
```

```
  <xs:complexType name="relation-type">
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="entity" type="xs:anyURI"/>
        <xs:anyAttribute namespace="##other" processContents="lax"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:schema>
```

9. Relax NG Grammar

The grammar for the Landmark configuration document extension is:

```
<!--
    LANDMARKS ELEMENT
-->
parameter &= element landmarks {
    attribute version { xsd:int }
    <!--
        LANDMARK-HOST ELEMENT
    -->
    element landmark-host {
        attribute address { xsd:string },
        attribute port { xsd:int }
    }*
}?
```

10. Security Considerations

10.1. Trust Aspects

TODO

11. IANA Considerations

TODO: register Kind-ID code point at the IANA

12. Acknowledgments

This work was stimulated by fruitful discussions in the P2PSIP working group and SAM research group. We would like to thank all active members for constructive thoughts and feedback. In particular, the authors would like to thank (in alphabetical order) David Bryan, Toerless Eckert, Lothar Grimm, Cullen Jennings, Peter Musgrave, Joerg Ott, Peter Pogrzeba, Brian Rosen, and Jan Seedorf.

13. References

13.1. Normative References

[RFC2119]	Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
[RFC3261]	Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M. and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.
[RFC3264]	Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, June 2002.
[RFC3265]	Roach, A.B., "Session Initiation Protocol (SIP)-Specific Event Notification", RFC 3265, June 2002.
[RFC3515]	Sparks, R., "The Session Initiation Protocol (SIP) Refer Method", RFC 3515, April 2003.
[RFC4574]	Levin, O. and G. Camarillo, "The Session Description Protocol (SDP) Label Attribute", RFC 4574, August 2006.
[RFC4648]	Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, October 2006.
[RFC4575]	Rosenberg, J., Schulzrinne, H. and O. Levin, "A Session Initiation Protocol (SIP) Event Package for Conference State", RFC 4575, August 2006.
[RFC5245]	Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", RFC 5245, April 2010.
[I-D.ietf-p2psip-base]	Jennings, C, Lowekamp, B, Rescorla, E, Baset, S and H Schulzrinne, "REsource LOcation And Discovery (RELOAD) Base Protocol", Internet-Draft draft-ietf-p2psip-base-19, October 2011.
[I-D.knauf-p2psip-share]	Knauf, A, Hege, G, Schmidt, T and M Waehlisch, "A Usage for Shared Resources in RELOAD (ShaRe)", Internet-Draft draft-knauf-p2psip-share-02, October 2011.

13.2. Informative References

[I-D.ietf-p2psip-concepts]	Bryan, D, Matthews, P, Shim, E, Willis, D and S Dawkins, "Concepts and Terminology for Peer to Peer SIP", Internet-Draft draft-ietf-p2psip-concepts-04, October 2011.
[RFC4353]	Rosenberg, J., "A Framework for Conferencing with the Session Initiation Protocol (SIP)", RFC 4353, February 2006.

[I-D.ietf-p2psip-sip]	Jennings, C, Lowekamp, B, Rescorla, E, Baset, S and H Schulzrinne, "A SIP Usage for RELOAD", Internet-Draft draft-ietf-p2psip-sip-06, July 2011.
[timestamps-acsc88]	Fidge, C., "Timestamps in Message-Passing Systems that Preserve the Partial Ordering", Proceedings of 11th Australian Computer Science Conference, pp. 56-66, February 1988.
[landmarks-infocomm02]	Ratnasamy, , Handley, , Karp, and Shenker, "Topologically-Aware Overlay Construction and Server Selection", Proc. of 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '02) pp. 1190-1199, 2002.

Appendix A. Change Log

The following changes have been made from version draft-knauf-p2psip-disco-02.

1. Adapted mechanisms for storing DisCo-Registrations to new requirements of Shared Resources draft [I-D.knauf-p2psip-share]

The following changes have been made from version draft-knauf-p2psip-disco-02.

1. DisCo-Registration uses now only the USER-CHAIN-ACL access control policy.

2. Adapted mechanisms for storing DisCo-Registrations to new requirements of Shared Resources draft [I-D.knauf-p2psip-share]

The following changes have been made from version draft-knauf-p2psip-disco-01.

1. The conference registration is now based on the Shared Resources draft [I-D.knauf-p2psip-share]:

   *DisCo-Registration Kind now meets the requirements for ShaRe.

   *Conference creation procedure now uses the ShaRe Access List.

   *Replaced USER-CHAIN-MATCH access policy for DisCo-Registration. Now uses USER-CHAIN-ACL or USER-PATTERN-MATCH.

2. Allow focus peers behind NAT

3. Added a 'node-id' attribute to the event package XML <version> element.

4. Added a 'node-id' attribute to the event package XML <focus> element.

5. Added a 'coordinate' child element to the event package XML <focus> element.

6. Corrected typos/wording

The following changes have been made from version draft-knauf-p2psip-disco-00.

1. Updated references.

2. Corrected typos.

3. New Section: Conference State Synchronization

4. XML Event Package for Distributed Conferences

5. New mechanism for generating chained conference certificates

6. Allow shared writing of resources using Access Control Policy USER-CHAIN-MATCH

7. Media Negotiation mechanism in a distributed conference

8. New Section: Distributed Conference Control with SIP

Authors' Addresses

Alexander Knauf Knauf HAW Hamburg Berliner Tor 7 Hamburg, D-20099 Germany Phone: +4940428758067 EMail: alexander.knauf@haw-hamburg.de URI: http://inet.cpt.haw-hamburg.de/members/knauf

Gabriel Hege Hege HAW Hamburg Berliner Tor 7 Hamburg, D-20099 Germany Phone: +4940428758067 EMail: hege@fhtw-berlin.de URI: http://inet.cpt.haw-hamburg.de/members/hege

Thomas C. Schmidt Schmidt HAW Hamburg Berliner Tor 7 Hamburg, D-20099 Germany EMail: schmidt@informatik.haw-hamburg.de URI: http://inet.cpt.haw-hamburg.de/members/schmidt

Matthias Waehlisch Waehlisch link-lab & FU Berlin Hoenower Str. 35 Berlin, D-10318 Germany EMail: mw@link-lab.net URI: http://www.inf.fu-berlin.de/~waehl