Design Updates for a RELOAD Usage for Distributed Conference Control (DisCo)

draft-knauf-p2psip-disco-00

Alexander Knauf
Gabriel Hege
Thomas Schmidt
Matthias Wählisch

alexander.knauf@haw-hamburg.de, hege@fhtw-berlin.de,
{t.schmidt,waehlisch}@ieee.org

Outline

1. Motivation
2. The DisCo Conferencing Scheme
3. Problem Statement
4. Shared Resources & Trust Delegation
5. Conference State Synchronization
6. Media Distribution

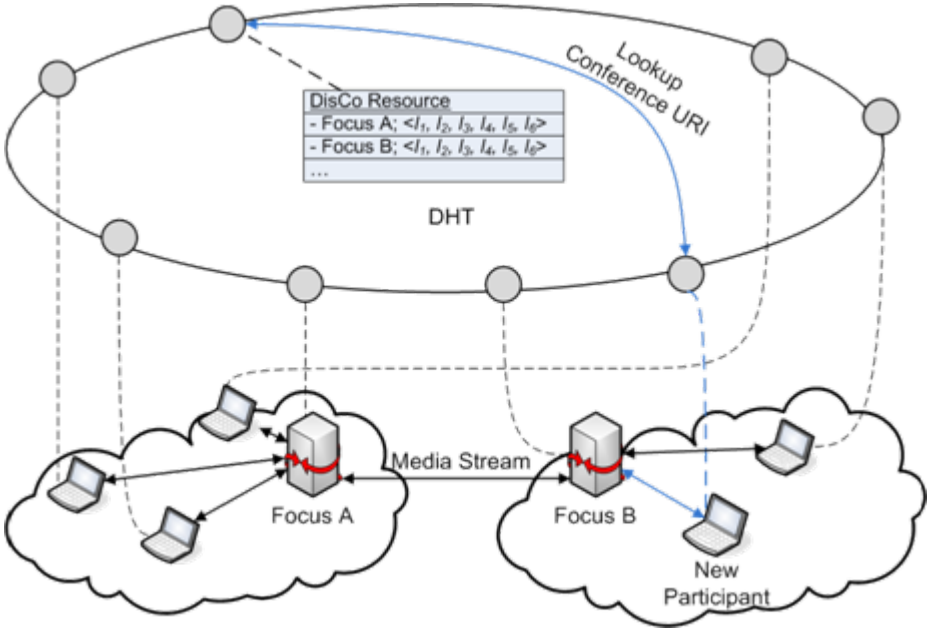
Motivation

- SIP and RELOAD designed for session establishment in a P2P-fashion
- No mechanism for group communication/overlay multicast in RELOAD
- Entry points to groups need to be located by joining peers
  - SIP Usage for RELOAD only locates single entities
- Management of group membership and state synchronization within a group are not treated in P2PSIP

The DisCo Conferencing Scheme

Goal: Create a tightly coupled conference, but distribute focus transparently among multiple focus peers

- Register conference in RELOAD
- Focus peers register themselves as entry points to the conference
- Conference State is synchronized among focus peers
- Media distribution by focus peers
- Multiple focus peers appear as a single entity to legacy clients using source routing



Problem Statement

Publishing conference registration and entry points in RELOAD:

- Problem: shared (write) access to conference registration
  - RELOADs security model based on certificates
  - Allowing write access only to owners of a matching certificate
- New RELOAD Usage for shared resources

State synchronization:

- Initial idea: Event Package for Conference State (RFC4575)
  - Not suited for distributed maintenance of conference state
- New event package for distributed updating

Media distribution:

- No central mixer, and no multicast functionality
- Use focuses for media distribution, do mixing at end devices
  - Results in re-negotiations on join/leave events

Problem: securing a shared resource without contacting enrollment server

Basic problem:

- Trust Anchor needed
  - Self-signed or creator-based certificates don't work alone
  - Malicious peer could take over the shared resource by creating own certificate for the resource (name)
- Approach: Tie resource name to creator
  - Restrict allowed resource URIs by pattern matching
  - Example: URI pattern: *-conf-$USER@$DOMAIN
    User Name: alice@example.com
    Allowed: pretty-conf-alice@example.com
    NOT allowed: alice-conference@example.com

Problem: Trust Delegation Chain

1st Approach: USER-CHAIN-MATCH

- Resource creator produces shared certificate, signed with his private key
- Storing peer verifies certificate chain
- Distribute private key for shared resource to all managing peers

Remaining Problems:

- Enhanced chance of shared certificate being compromised, because multiple peers have the private key
- Still no solution for certificate revocation (short lifetime, revoc. List?)
- Certificate can only be used as long as creator is present (+ lifetime)

Problem: Trust Delegation Chain (2)

2nd Approach: USER-CHAIN-ACL

- Delegation list in RELOAD
  1. Creator X initiates resource and signs with its private key: X → X, kind-ID (signed by X)
  2. Creator delegates to Y: X → Y, kind-ID (signed by X)
  3. Y can delegate further: Y → Z, kind-ID (signed by Y)
  4. Storing peer authorizes writing permission based on the ACL
  5. Accessing peer must verify List Chain (not certificate chain)

- All list entries contain Kind-ID of the corresponding shared resource
- Allows for subtree-revocation by nulling delegation line
- Works in absence of creator (or intermediate node) if lifetime appropriately extended

Conference State Synchronization in DisCo

- Each controller in a distributed conference *maintains its own* signaling relations to the participants
  - **Problem:** Controllers don't have a global knowledge about the current state of a conference e.g.:
    - Who is in the conference?
    - Who is a focus to the conference?
    - Where to delegate a call in case of overloading?

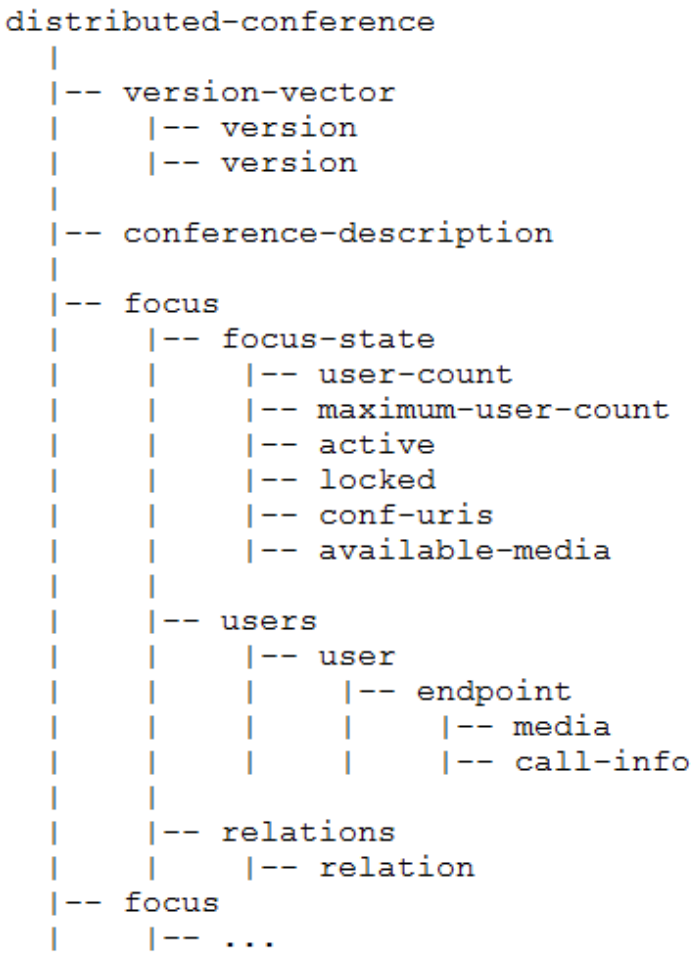
- **Initial approach in version -00:** Focus peers to a conference subscribe to each other for an extended Event Package for Conference State [RFC4575]
  - Problem (identified at ietf78 in samrg session): RFC4575 NOT convenient to synchronize the state in a distributed conference

Event Package for Distributed Conferences

- **Proposal for -01:** Definition of a new Event Package for synchronizing the conference state in a distributed conference
  - Designed to convey information about roles and relations of the conference participants
  - Enables a coherent global knowledge to a conference
  - Handles concurrency and racing conditions
    - E.g., Uses a version scheme based an vector times
  - XML Imports of several element definitions of RFC4575
    - As they are still suitable for distributed conferences

Event Package Overview

- <distributed-conference>:
  - Root element
- <version-vector>:
  - Enables a coherent version scheme
- <conference-description>:
  - General information about a DisCo
- <focus>:
  - Describes a participant in role of a focus and its responsibilities to other participants
- <relations>:
  - Describes a focus peers relations to adjacent focus peers



Offer/Answer within Distributed Conferences

- Focuses are responsible for distributing media to connected peers
- Ad-hoc scheme:
  - A Focus distributes all media streams to all connected peers
  - Focus may choose to do mixing/recoding
  - When a new peer joins:
    - Focus offers all media streams it receives to the joining peer
    - Joining peer offers its media streams to the focus
  - *Either*: Focus modifies media sessions to all connected peers, offering the new stream
  - *OR:* mix the new stream with existing streams to prevent the need for SIP re-INVITE
  - Media streams naturally follow signaling connections

Upcoming Work

Separate contribution into

1. Abstract part that supports shared distributed resources that are
   - Addressable by a uniform URI
   - Synchronized within RELOAD
   - ▶ Research options for SAM community?

2. Application part that designs media conferencing based on the abstract mechanisms & scheme

Thanks!

Opinions / Questions?