

# Mechanisms for Robust Multimedia Conferencing Using SIP

Author: Heidi-Maria Rissanen

Supervisor: Prof. Jörg Ott

Instructor: Gonzalo Camarillo, M.Sc.(Tech.)

Oy LM Ericsson Ab, Finland

# Contents

- Background
- Objectives
- Conferencing frameworks
- Prototype implementation
- Robustness in SIP conferences
- Conclusions

# Background

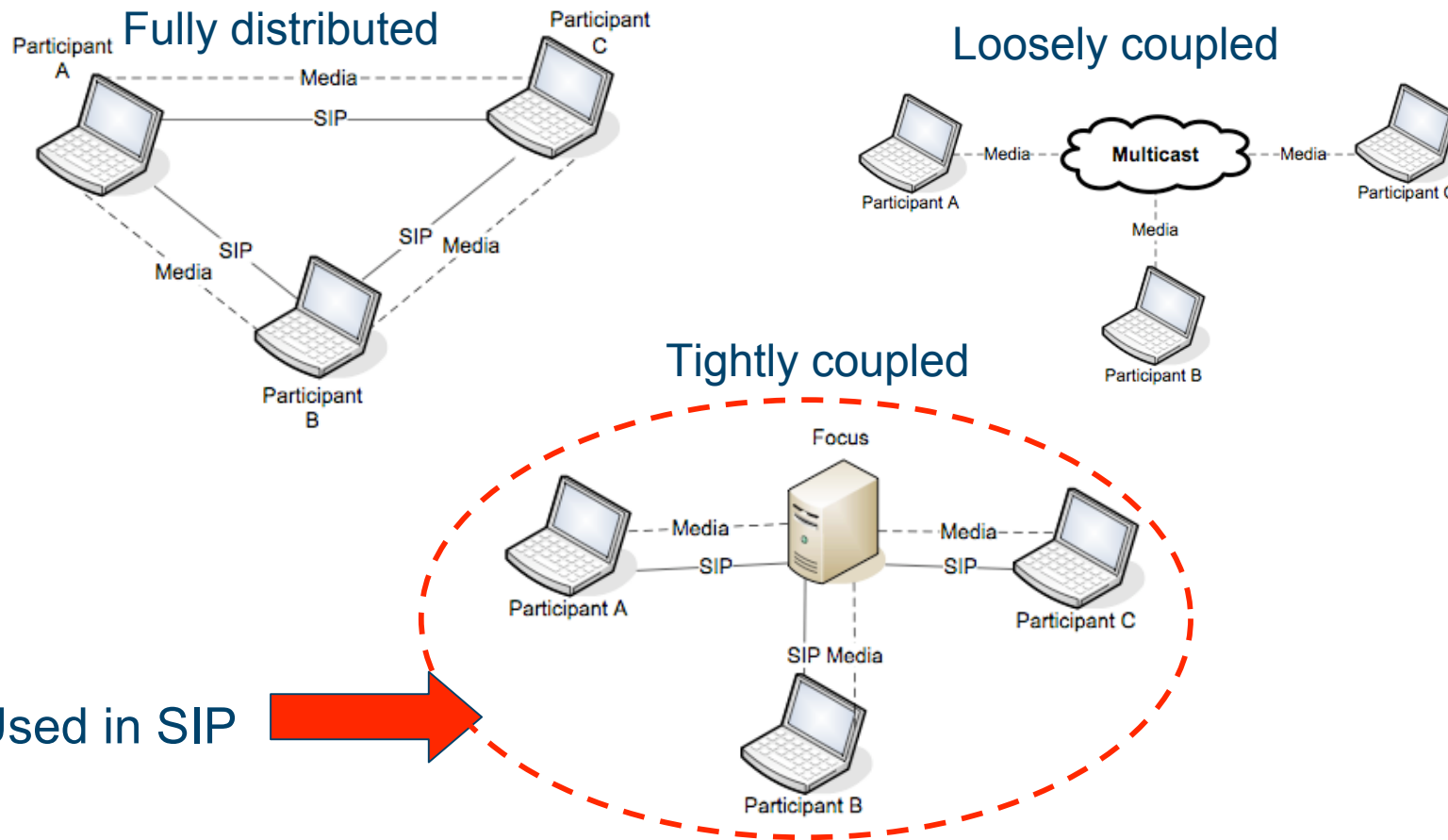
- The Session Initiation Protocol (SIP) is the most widely used Internet telephony standard
- Multimedia conferencing using SIP is steadily increasing
  - Still, there are no built-in mechanisms in the SIP protocol to ensure the robustness of conferencing services

# Objectives

- Theoretical part of the thesis
  - Propose mechanisms for improving the robustness of SIP conferencing
- Practical part of the thesis
  - Implement the SIP conferencing functionality on a Session Border Controller (SBC)
    - SBC is a device that can be used in Voice over IP (VoIP) networks to control both signaling and media streams
- Research methods
  - Literature survey
  - Prototype implementation

# Conferencing

- Conferencing can be done in several ways:

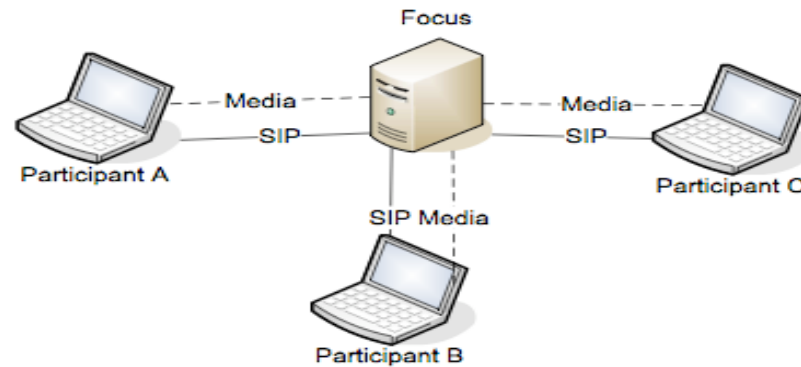


# Conferencing frameworks

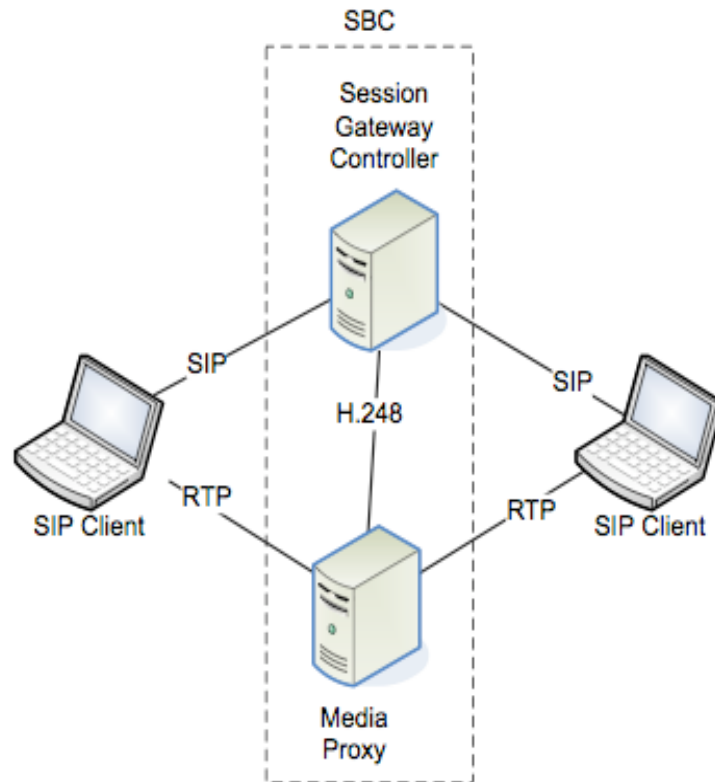
- There are two conferencing frameworks defined by the IETF (Internet Engineering Task Force)
  - The SIPPING conferencing framework
  - The XCON conferencing framework
- Frameworks define the logical entities and terminology to be used for conferencing
- SIPPING uses SIP as the signaling protocol, XCON does not depend on any particular signaling protocol
  - The XCON framework is SIPPING compatible

# Conferencing frameworks (cont.)

- The central component of the conferencing model is a conference server called focus
  - Has a signaling relationship with every conference participant
  - Is responsible e.g. for the media streams of the session, conference policy, notifications about the state changes of the conference.
  - Participants contact the focus by using a unique conference URI
    - e.g. sip:discussion\_on\_travel@conference.com



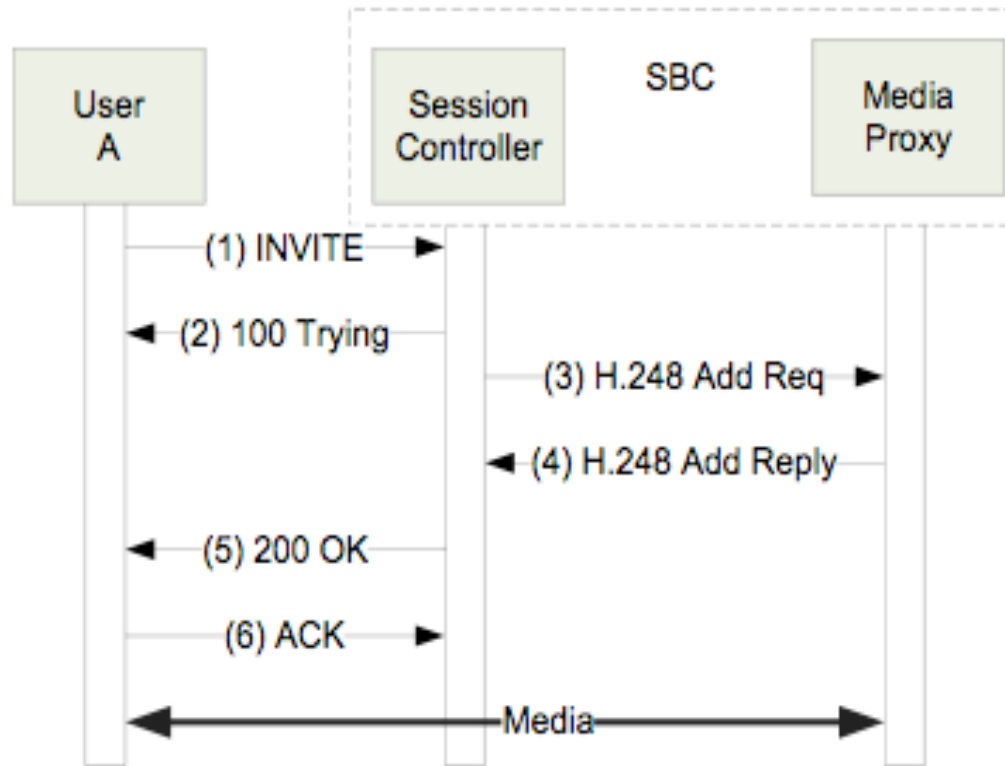
# Prototype implementation



- Session Border Controllers are used for managing the signaling and media streams of VoIP calls
- The objective of the prototype was to implement conference focus functionality on an existing Session Border Controller (SBC)
  - Support small-scale SIP conference sessions



# Session setup



# Example message

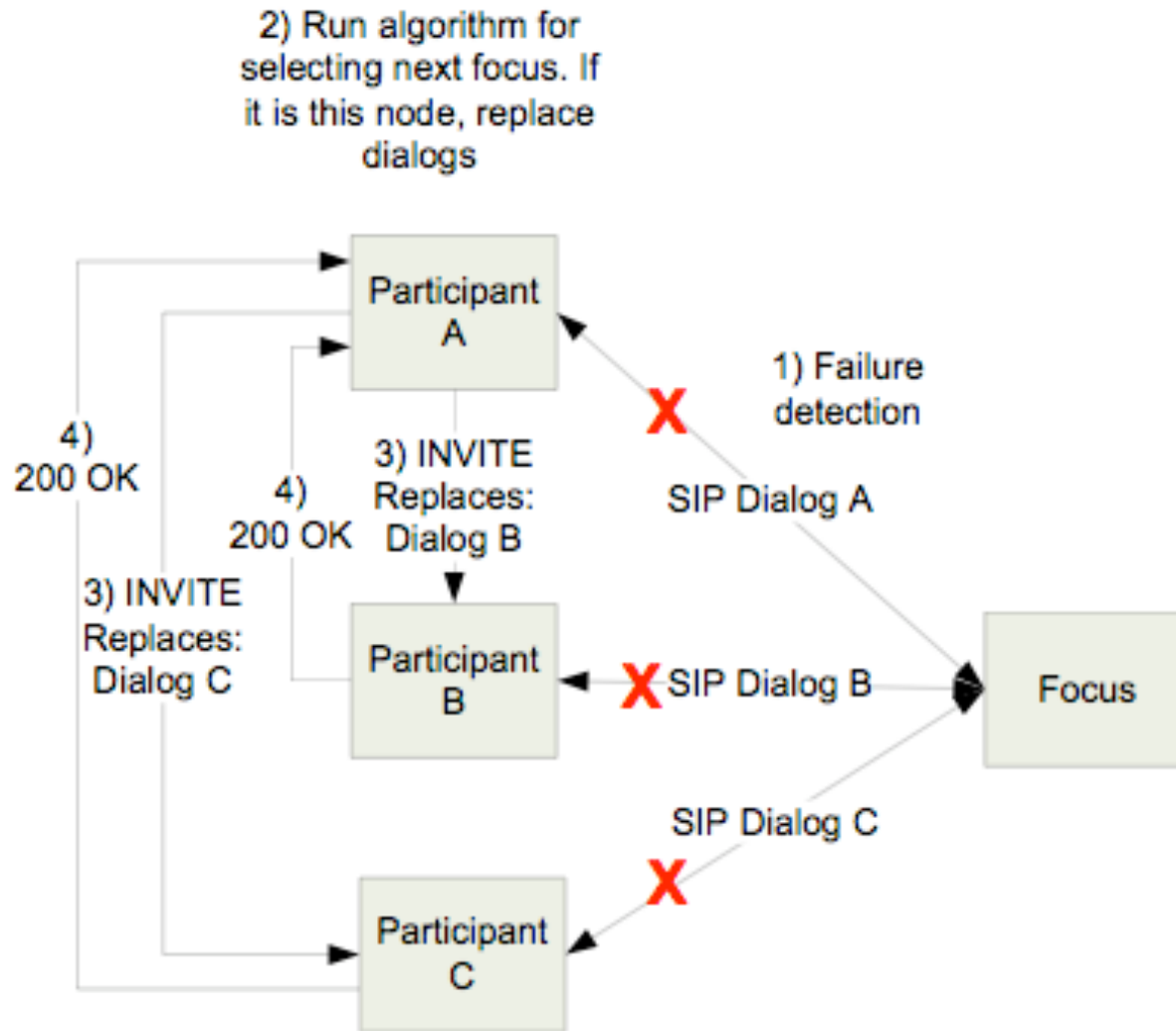
```
INVITE sip:conference@131.160.36.15:5060 SIP/2.0
Via: SIP/2.0/UDP 127.0.0.2:5062
From: sippl <sip:sippl@127.0.0.2:5062>;tag=15
To: <sip:conference@131.160.36.15:5060>
Call-ID: 1-18098@127.0.0.2
CSeq: 1 INVITE
Contact: sip:sippl@127.0.0.2:5062
Max-Forwards: 70
Subject: Conference Call
Content-Type: application/sdp
Content-Length: 134
```

```
v=0
o=user1 53655765 2353687637 IN IP4 127.0.0.1
s=-
c=IN IP4 131.160.36.15
t=0 0
m=audio 30106 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

# Robustness in SIP conferences (1)

- A robust system will continue operating normally even if there is a failure or incorrect input is passed into the system
- To improve the robustness of SIP conferencing we need a mechanism for
  - replicating the state of the conference session into the system
  - detecting the failure of the focus
  - identifying nodes capable of acting as a focus
  - electing the next focus
  - transferring the session to the new focus

# Robustness in SIP conferences (2)



# Robustness in SIP conferences (3)

- For state replication and for identifying possible backup focus candidates we use two existing SIP extensions with slight modifications
  - As a participant joins, it indicates its conferencing capabilities in the INVITE message
    - Participants have different roles in the conference
    - The focus informs every node that is “possible focus” about the state changes in the conference
    - The backup focus candidates then have the needed information to re-establish signaling and media streams

# Robustness in SIP conferences (4)

- For detecting the failure we use periodic session refreshes as already defined in SIP
- For electing the focus we use a simple deterministic algorithm
  - Every backup focus candidates runs independently

# Conclusions

- Theoretical part
  - Most of the needed functionality for robust conferencing using SIP is already defined
    - Only slight modifications were needed in two specifications
- Practical part
  - The SIP conferencing functionality was successfully implemented, the concept was proved to true
- Suggestions for future work
  - How the mechanisms could be used for load balancing of conferencing foci
  - Large-scale conferences
  - More advance algorithms for backup focus election

**ERICSSON** 

**TAKING YOU FORWARD**