

---

## **RFIdentify: Speaker Presence Sensing**

*Blake Arnold, Willi Ballenthin*

`{baa2121, wrb2102}@columbia.edu`

*Internet Real-Time Lab, Columbia University  
New York, NY 10027*

---

Internet Real-Time Lab  
Fall 2010

Computer Science  
Columbia University  
New York, New York 10027  
URL: <http://www.cs.columbia.edu/irt/>

# RFIdentify: Speaker Presence Sensing

Blake Arnold, Willi Ballenthin

{baa2121, wrb2102}@columbia.edu

Internet Real-Time Lab, Columbia University  
New York, NY 10027

*Columbia University*

Fall 2010

## Abstract

Designing a location aware system to allow large conferences to work seamlessly has been a problem in the past. There have been past try's at trying to create a location aware network and many successful implementations. But none of these successes has yet to allow an instantaneous notification of presence.

## 1 Introduction

Location aware hardware is slowly gaining ground in the past few years. The hardware has continued to grow from Sonar to GPS to cellphone triangulation. The main motivation for this technology is ease of use from the user side. New services have been created to help third parties track users of such devices as means of recommendations or alerts.

This technology can be useful during a large conference with numerous speakers. Keeping track of which speakers are currently presenting is a daunting task. It requires constantly looking at a time and the speaker order while also keeping track of any changes to the speaker order.

In this paper, we show how location aware hardware can be used to allow better speaker recognition during a conference. Are main goal is to alert the attendees when new speakers start to present. This allows the attendee to determine when a speaker of interest begins his/her talk.

Past designs to create a speaker aware system have been different in the past.

## 2 Related Work

1. **F8 Conference** - Facebook Presence for all attendees as well as speakers.
2. **Marc Petit-Huguenin** - Original designer for microphone RFID system

## 3 Outline

In Section 4, we introduce the use cases of the new software and hardware combination. In Section 5 describes are strategy and design of the new system with further details in Section 6. Finally, we describe future work in Section 7."

## 4 Problem

Following IETF presentations remotely requires a lot of work by the end user. They must download the pdf of the slides, determine who is speaking by a XMPP message and then listen to the audio feed through a video stream program. The system has a heavy reliance on humans inputting data, such as the current speaking, uploading slides

and speaking into the microphone. All these things elements can lead to human error and a bad experience by the end user.

## 5 Approach

By using an RFID reader near the presenter location, the speaker can effectively be identified. This eliminates the need for human interaction to send out the name of the current speaker.

The use of a RFID tag can raise security concerns from the users. RFID readers have been shown to be able to read tags from few meters away. This can be alleviated with use of a shielding device that blocks RFID frequencies from being intercepted by unwanted readers. Another strategy is to have low frequency card that only have an unique ID associated with the card. A database would then store the information of the card holder. If anyone reads the card, they will not be able to gather any useful information about the person.

### 5.1 Server

The server Stores a database of RFID tag ID numbers associated with each presenter. The database allows quick access to the speakers information, as well as other information such as their email address or the location of their presentation.

1. **Dependencies** The Server requires a network to receive incoming presence responses from RFID readers.

### 5.2 Client

1. **Dependencies** -
2. **Configure Handling** -
3. **Avahi Loop** -
4. **Reader Loop** -

## 6 Technologies

The Client and server were implementing using open source software and other well known libraries to create a well tested code base.

### 6.1 Server

1. **Apache** - Server core implementation for handling client requests. The httpcore.jar used in the implementation handles low level functionality of client handling from opening ports to passing the requests onto higher level HTTP request handlers.
2. **SIMPLE** - SIP message passing implementation. Used to alert a presence server that a current speaker is presenting.

### 6.2 Client

1. **avahi/mDNS** - DESCRIPTION
2. **pthreads** -
3. **cURL** - DESCRIPTION

## **6.3 Hardware**

1. **DLP** -
2. **FTDI/Big Reader** -

## **7 Summary and Future Work**

Create code for SenseEverything  
often repeats the main result

## **8 Acknowledgements**

## **9 Bibliography**

<http://www.facebook.com/presence/>