PSCR 2022 The Digital Experience

A Platform for Experimental Research in Critical Voice Communications

Jan Janak, Artiom Baloian, Kahlil Dozier,
Dan Rubenstein, Henning Schulzrinne - Columbia University
Lauren Berny, Charles Jennings - John Jay College

Disclaimer

This presentation was produced by guest speaker(s) and presented for publication in the National Institute of Standards and Technology's PSCR 2022 The Digital Experience. The contents of this presentation do not necessarily reflect the views or policies of the National Institute of Standards and Technology or the U.S. Government.

This work was performed under the following financial assistance award 70NANB19H003 from U.S. Department of Commerce, National Institute of Standards and Technology.

Posted with Permission.

Outline

1. Project Objectives
2. Testbed Design
3. Remote Listening Experiments
4. Interactive Game Experiments
5. Summary & Next Steps

Project Objectives

How does the quality of the communication channel affect first responder communications?

- Four phase approach:
  a. **Build a communication testbed**
     i. Emulate real mission critical voice (MCV) systems
     ii. Configurable audio & network impairments
  b. **Experiment with trained first responders**
     i. Communicate using the testbed in a controlled environment
  c. **Measure communication performance**
     i. Analyze data collected during experiments
  d. **Build mathematical models**
     i. Channel conditions -> performance measures

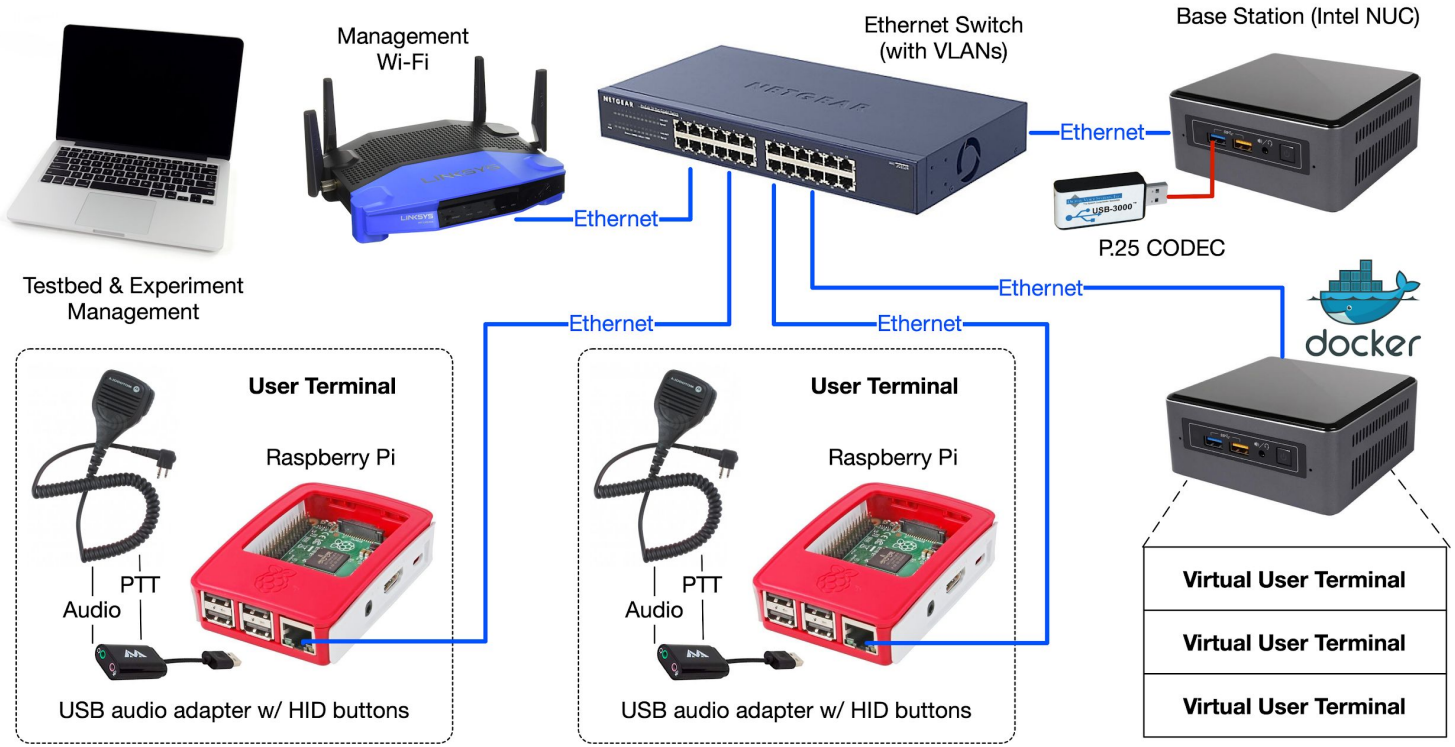
Quality of Experience (QoE) Measures

- Comprehension errors
  - repeat transmitted messages
- Task errors
  - wrong information recorded
- Usage errors
  - pressing push-to-talk (PTT) button too early or too late
- Length and latency of responses
  - pauses between requests and start of transmission
- Subjective ratings of user experience
  - rated frustration with radios

Testbed Design

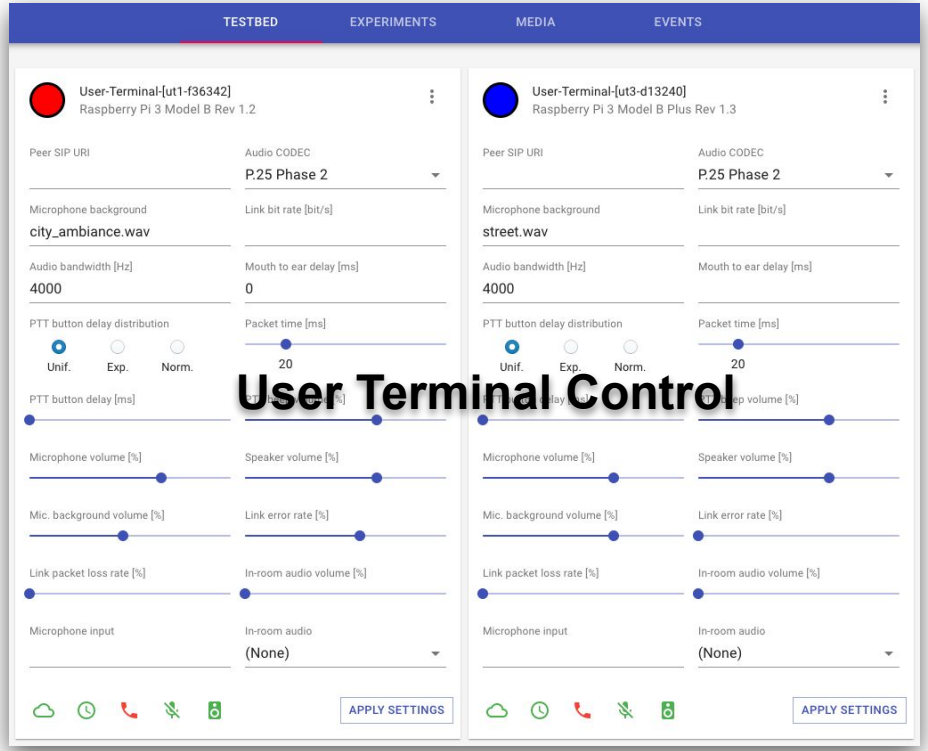
- Open hardware & software
- Affordable off-the-shelf components (Raspberry Pi for user terminals)
- Emulates analog & digital MCV systems (including P.25 Phase 1 & 2)
- Programmable audio & network impairments
- Support for interactive and listening (at-home) experiments
- Testbed and experiment management from the browser
- Python environment for experiment evaluation and analysis

Testbed Hardware Architecture

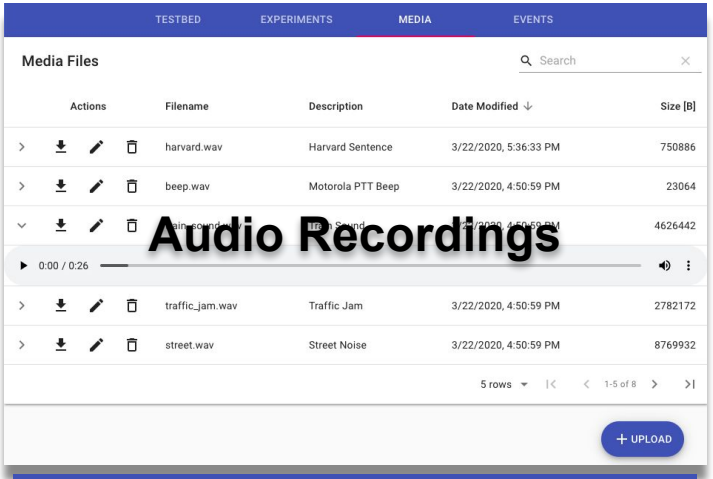


Testbed Prototype at Columbia University

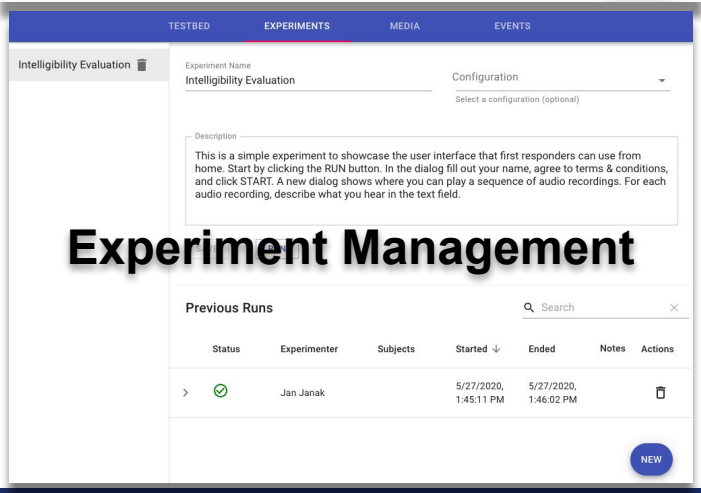
Testbed User Interface



User Terminal Control



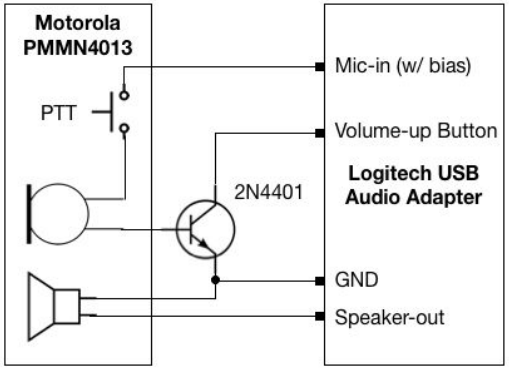
Audio Recordings

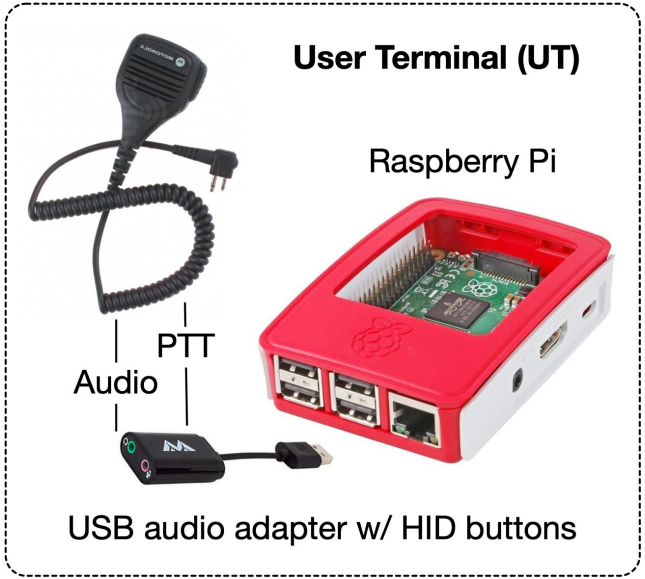


Experiment Management

User Terminal Hardware Architecture

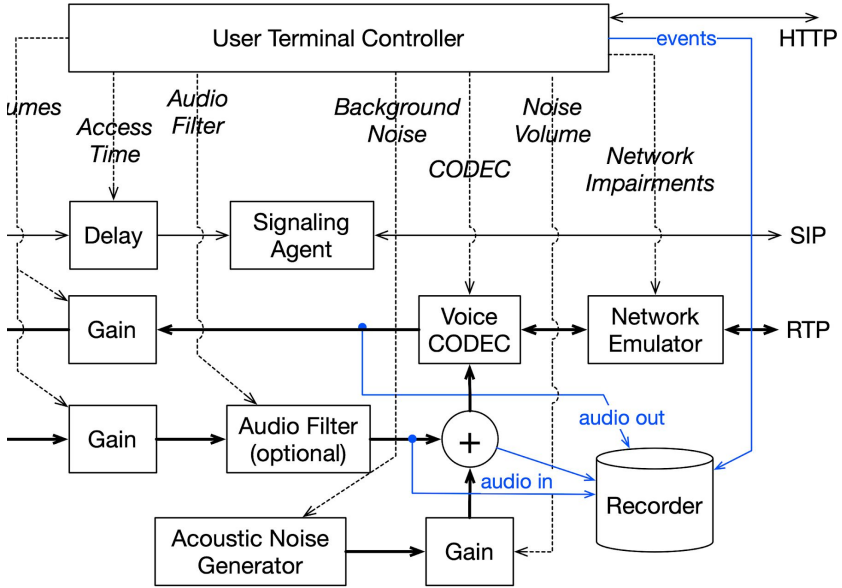
- Connect Motorola speaker to Raspberry Pi
- Emulates PTT communication device
- The device applies audio impairments





User Terminal Software Architecture

- Custom controller application
- PulseAudio for audio processing
- Baresip SIP user agent (P.25 codec)
- Linux kernel network emulator
- Custom event & audio recorder



Supported Impairments

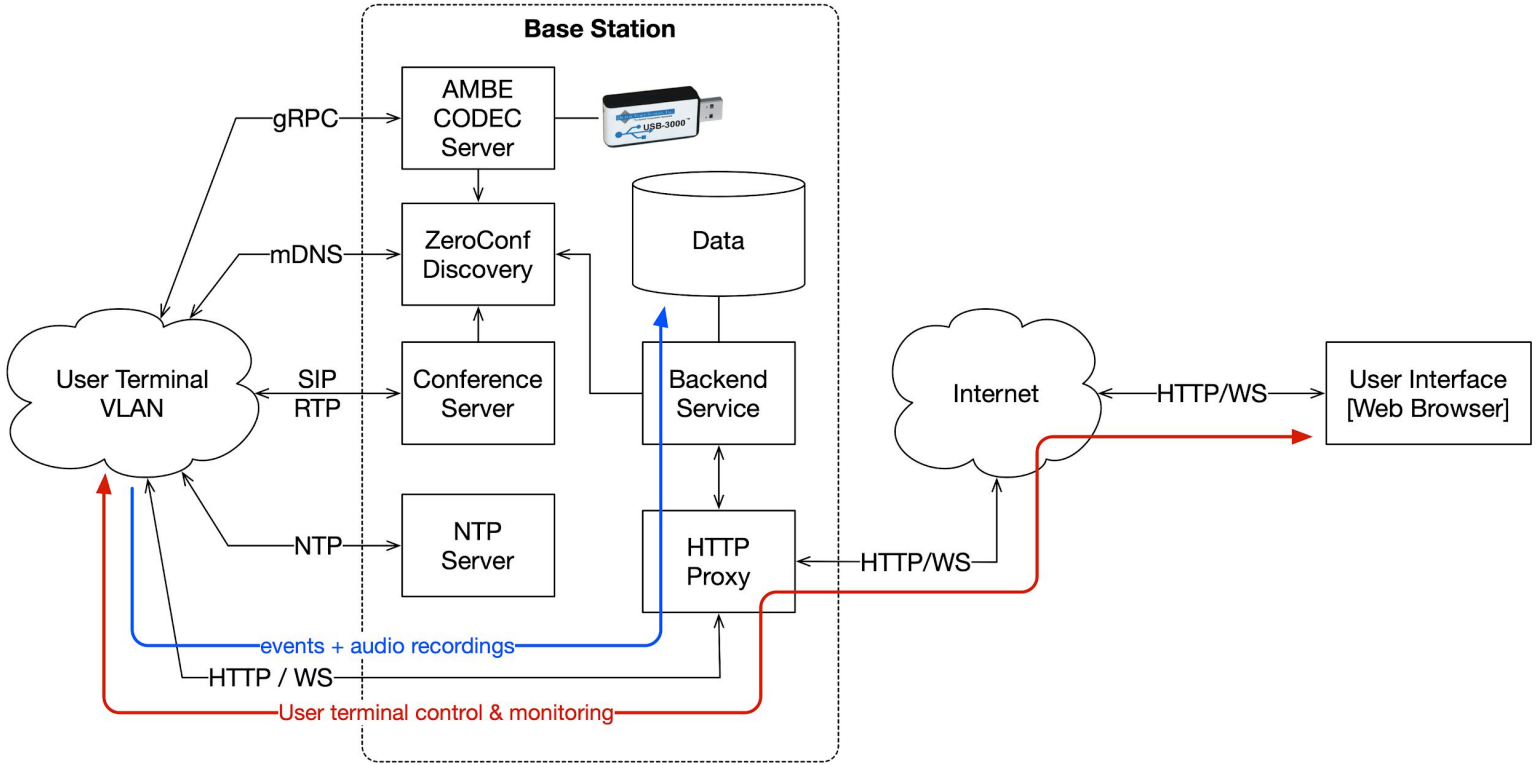
Audio

- Background noise injection
- Fine-grained volume control
- Audio filtering (bandwidth, low-pass)
- Variable PTT delay (mic audio cut-off)
- Voice compression (codec selection)

Network

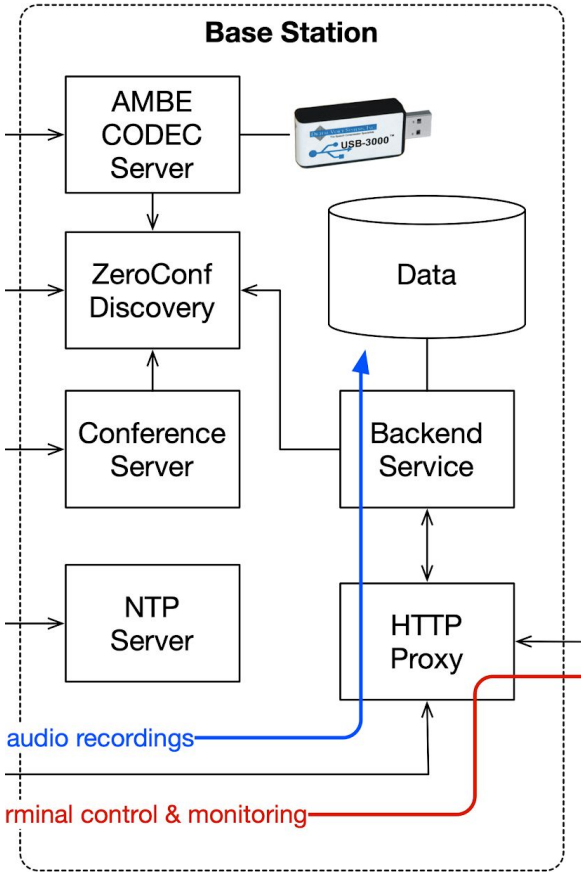
- Configurable mouth-to-ear delay
- Maximum bandwidth limitation
- Modulation rate (baud rate)
- Packet loss
- Bit error injection

Base Station Software Architecture



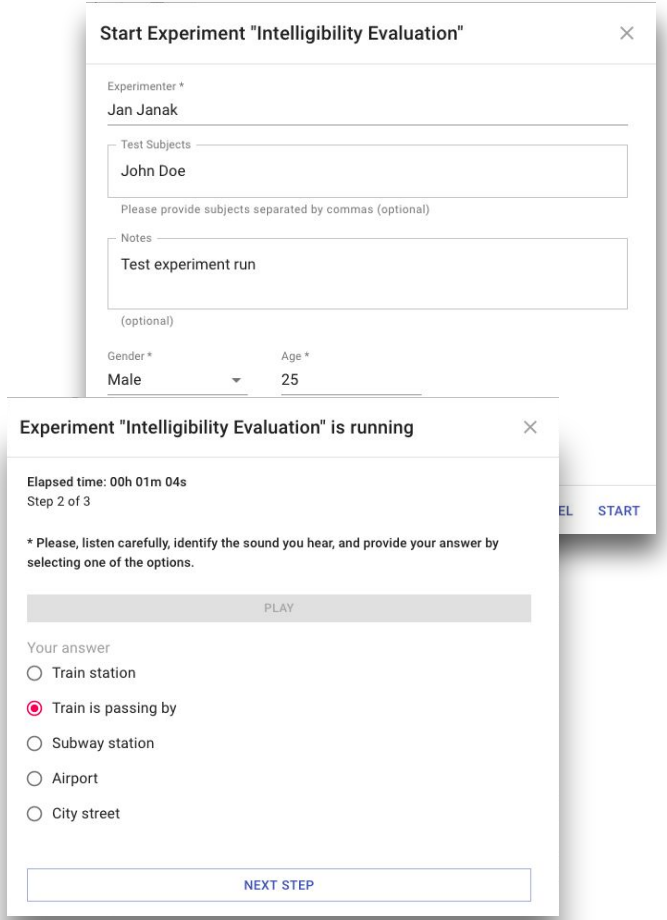
Base Station Services

- Custom Advanced Multi-Band Excitation (AMBE) codec server
- ZeroConf for service discovery
- SIP conference server emulates RF channels
- NTP for user terminal time synchronization
- MongoDB for data storage



Remote Listening Experiments

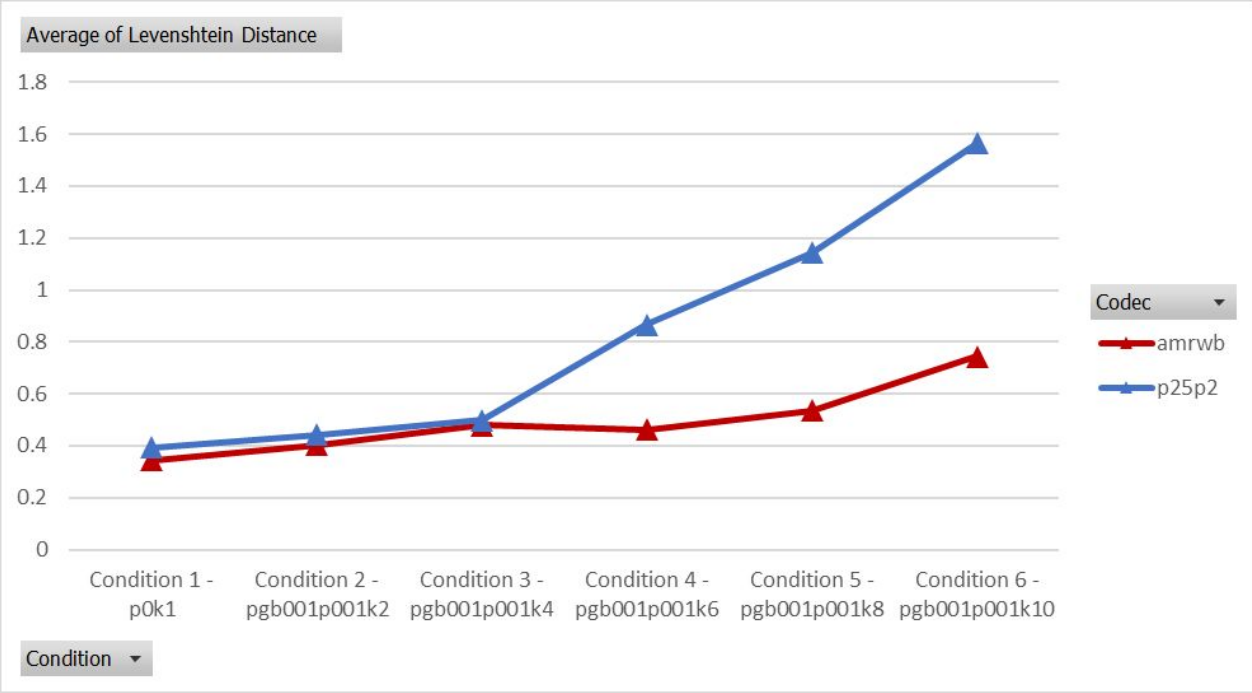
- A series of impaired audio recordings
- Browser UI for playback & data collection
- Anti-cheat design (play once only, no pause)
- Accessible from first responders' homes



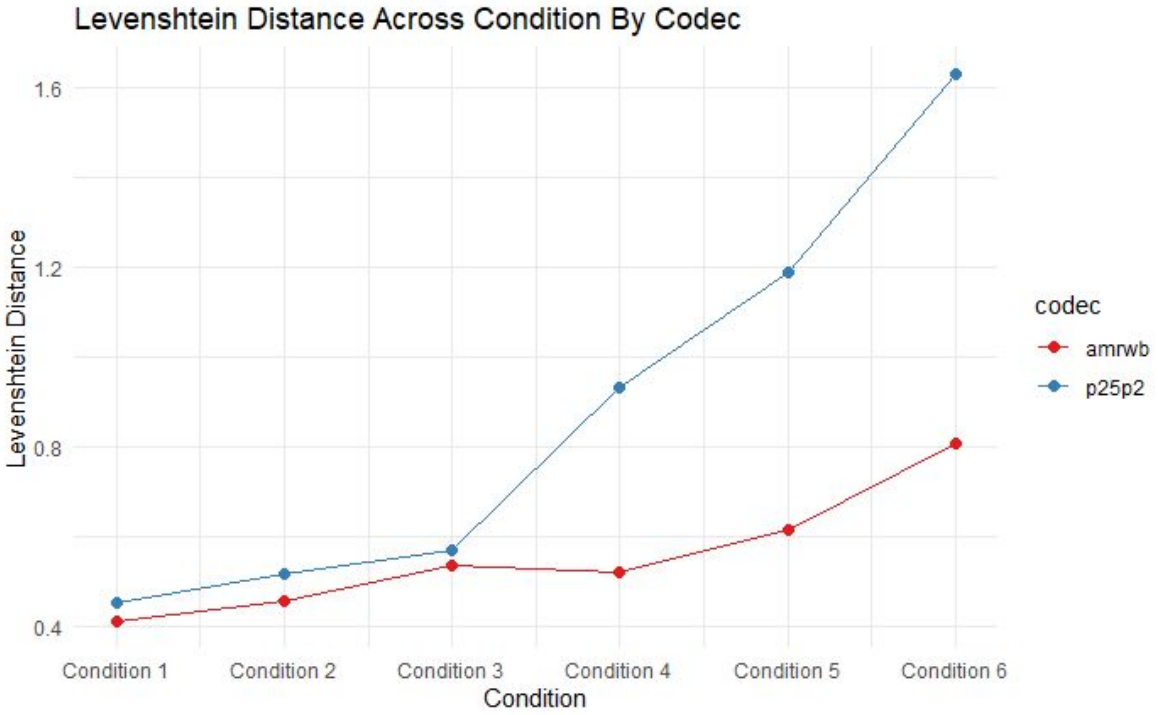
License Plate Listening Experiment

- New Jersey license plates using NATO alphabet
- 48 participants x 72 trials = 3,456 total trials
- Correlated (Gilbert-Elliot model) frame loss and bit errors
- Compare codec performance: P.25 Phase 2 versus AMR

Listening Experiment Results



Listening Experiment Results (continued)



Differences Between Codecs:
Condition 1, p0k1 (reference group)
Condition 2, pgb001p001k2 (p = .847)
Condition 3, pgb001p001k4 (p = .933)
Condition 4, pgb001p001k6 (p <.001)
Condition 5, pgb001p001k8 (p <.001)
Condition 6, pgb001p001k10 (p <.001)

Note: Graph is predicted values

Interactive Experiment in EDGE* Environment

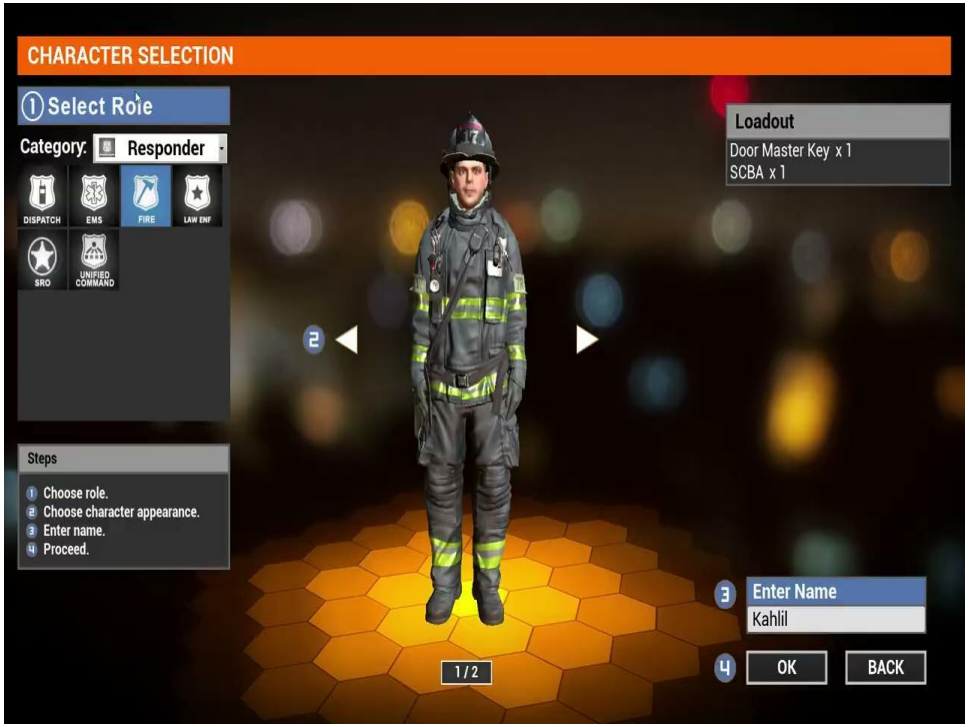
- Virtual training platform for first responders
- Coordinated response to critical incidents
- Developed by the U.S. Department of Homeland Security (DHS)
- Built on the Unreal game engine
- Two training environments: hotel and school

* Enhanced Dynamic Geo-Social Environment (EDGE)

Source: https://www.dhs.gov/science-and-technology/EDGE

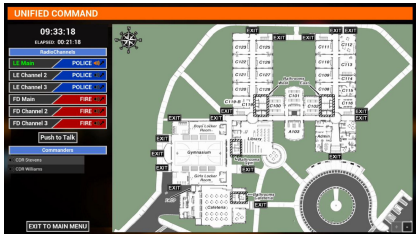
EDGE Gameplay

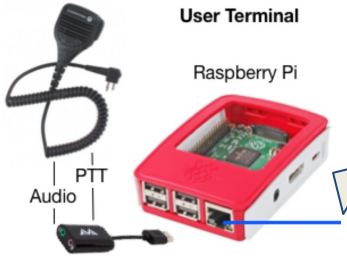
https://www.youtube.com/watch?v=nFoLQ4M2CRc



EDGE+MCV Testbed @ Columbia

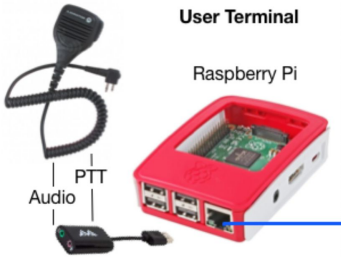
Dispatcher





USB audio adapter w/ HID buttons

First responder (test subject)



USB audio adapter w/ HID buttons

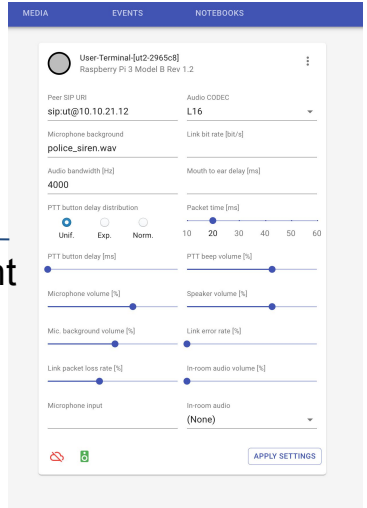
voice

voice

Testbed base station

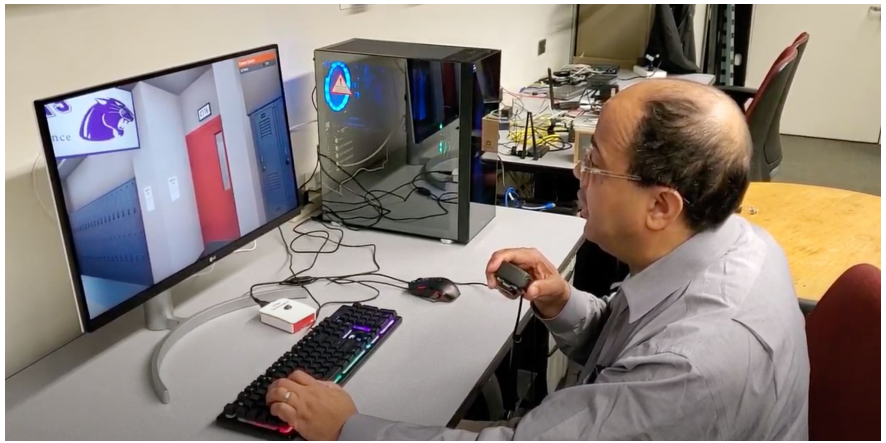
Impairment control

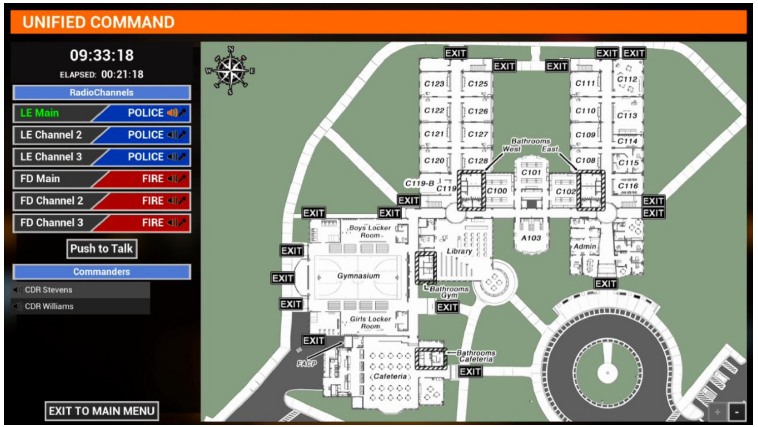
Experimenter

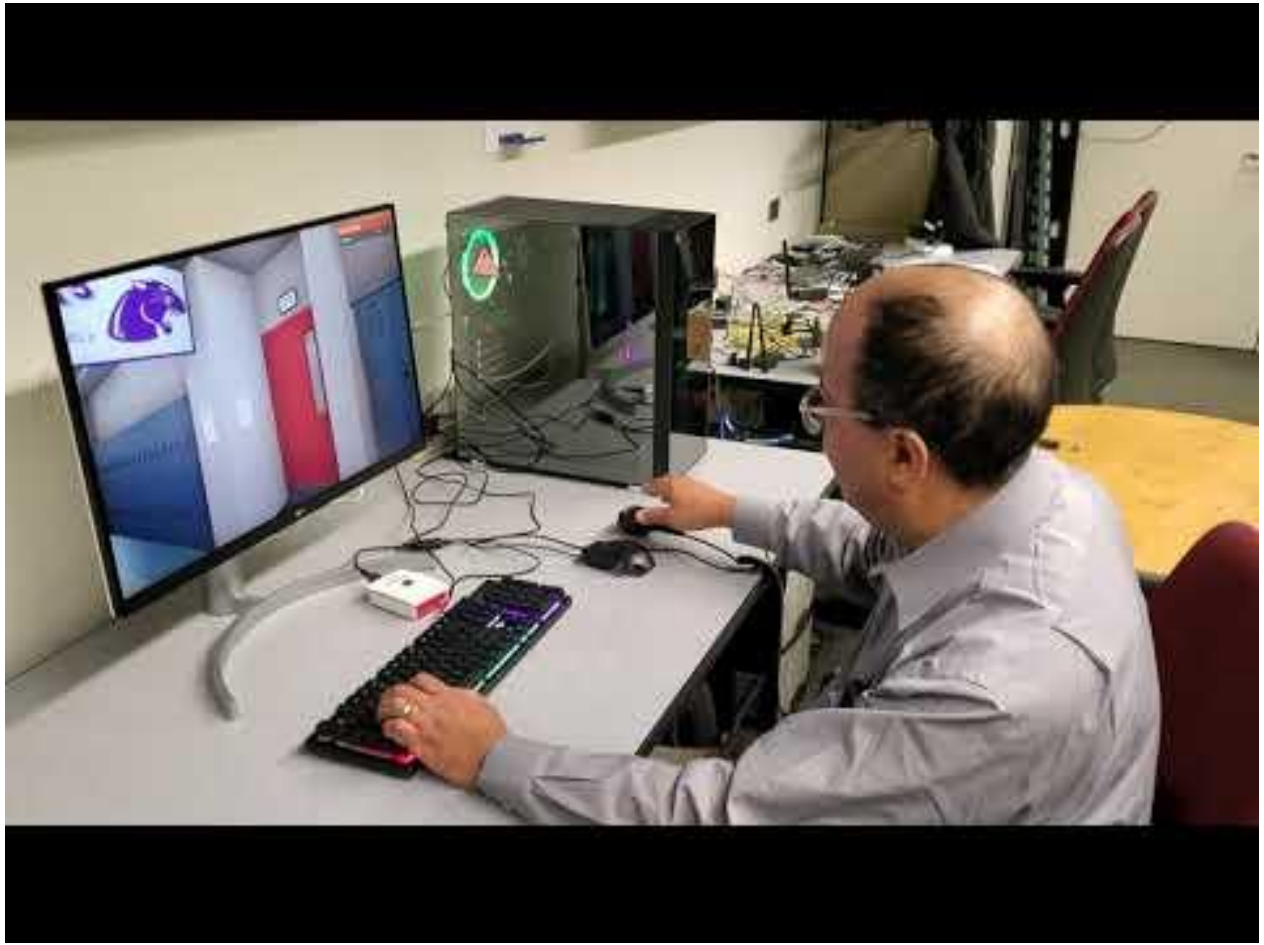


EDGE First Responder Experiment

- Human test subject in EDGE
- Two-way communication with dispatcher via MCV testbed
  - Variables: MtE delay, PTT delay, access time



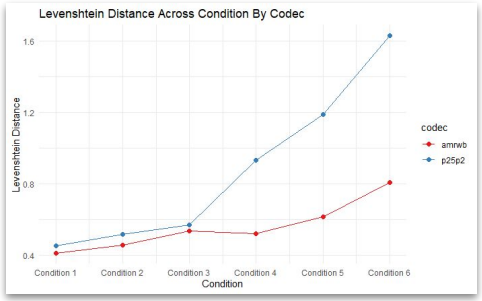


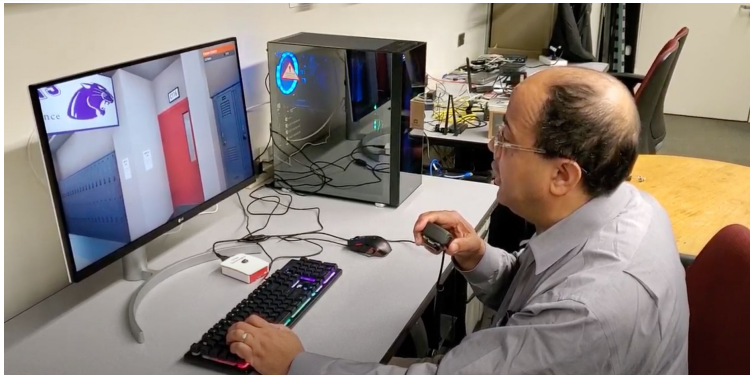


Platform for Experimental Research in Critical Voice Communications

Summary

- Open testbed for experiments with MCV
  - Remote listening experiments
  - Interactive delay-based experiments
- Ongoing human subject experiments
  - Listening experiment performed & evaluated
  - Interactive (EDGE) experiment underway





https://irtlab.gitlab.io/mcv-testbed

Next Steps

1. Run experiments with first responders on Columbia University campus
2. Measure true Mouth-to-Ear (MtE) delay of the testbed
3. Develop mathematical models to estimate performance measures from channel condition (impairment levels)

Additional Resources

- Project website is online
- Testbed source code available under the MIT license
- Free software developed for the project:
  - PulseAudio client for NodeJS: https://github.com/janakj/pulseaudio.js
  - Client library for DVSI AMBE vocoder chips: https://github.com/janakj/ambe

Project Website

https://irtlab.gitlab.io/mcv-testbed