THE UNIVERSITY OF ALABAMA | College of Engineering

AVoIP: Ad-Hoc Voice over Internet Protocol for Small Single-board Computers

Checkpoint Report of Term Project

Students:
Trupeshkumar R. Patel
trpatel2@crimson.ua.edu
David Coleman
dmcoleman1@crimson.ua.edu
Xiaoming Guo
xguo29@crimson.ua.edu

Supervisor:
Dr. Xiaoyan Hong
hxy@cs.ua.edu

Department of Computer Science

April 11, 2022

AVoIP: Ad-Hoc Voice over Internet Protocol for Small Single-board Computers

(Checkpoint Report)

Trupesh R. Patel*, David Coleman*, Xiaoming Guo*
*Department of Computer Science
College of Engineering
University of Alabama
Tuscaloosa, AL 35487
{$trpatel2$, $dmcoleman1$, $xguo29$}@crimson.ua.edu

***Abstract*— Asterisk PBX is a software implementation of Private Branch Exchange (PBX), a system used to handle telecommunication services, allowing VoIP (Voice over Internet Protocol) services on computers, including less powerful devices such as Raspberry Pi. However, using smaller single-board devices come with the limitation of computing power and storage capacity. As a result of these limitations are dropping packets, cutting off the callers while handling a more significant number of calls simultaneously. In this project, the authors provides the proof-of-concept by analyining the packets delays, network jitters, end-to-end network latency, and bandwidth between an Asterisk PBX system (which is install on Raspberry Pi 4) and IP clients that are using VoIP as service. This analyis includes internet through wired, wireless, cellular wireless connection.**

I. Introduction & Background

Voice always has multiple functionalities among the information flowing at the edge of the IoT network. It carries valuable content, reflects the conditions of the environment, and can be used to command other entities through acoustic actuators and phone calls.

Implementing voice systems at the edge of an IoT network typically faces challenges, namely that edge devices are always constrained by computing power, bandwidth contention, and energy consumption. Therefore, it is not feasible to implement a complete TCP/IP stack on each node and give all nodes the ability to connect to remote entities outside the local network.

IP-PBX (IP-based Private Branch Exchange) provides a comprehensive solution to address the aforementioned issues. As the prototype of IP PBX, the design of the traditional PBX system is to serve a private organization, in which both the geographic area and the communication connection are limited to a specific scope. Interconnections between internal phones are without cost, while only central office lines provide connections to the public switched telephone network (PSTN). This scheme meets our expectations for edge IoT communication – internal communication does not occupy egress bandwidth, and some switcher servers still reserve the communication egress to the outside. Leveraging VoIP technology, IP-PBX has ported the PBX scheme to the Internet, replacing telephone lines with packet-switching networks. The IP-based paradigm offers better scalability and lowers the cost same as the Internet brings to other domains.

Several mature implementations of the IP-PBX paradigm are available, including 3CX and Asterisk PBX. Asterisk is an open-source software package that can run all the PBX functions, usually on a Linux operating system platform. It contains the functions of PBX and some other additional features. Along with the essential telephony services, voicemail services, conference calling, interactive voice response, and call queuing are also provided by Asterisk. It also provides multi-party calling, display caller ID (display calling number). To interact with digital telephone equipment and analog telephone equipment, Asterisk needs the support of PCI hardware, the most famous of which is provided by the Digium platform.

From the architecture perspective, Asterisk serves as a middleware function, connecting the underlying telephony technology and the upper-level telephony applications. Both PBX and IVR (Interactive Voice Response) functionalities are integrated within Asterisk. Using compatible PCI hardware, Asterisk supports traditional telephone lines, including TDM (Time Division Multiplexing), TI/El PRI/PRA&RBS (Robbed Bit Signal) mode, analog telephone line/analog telephone (POTS), ISDN (Integrated Services Digital Network) and BRI (Basic Rate) and PRI (Primary Rate). Also, due to the PCI hardware support feature, Raspberry Pi can be used to implement the Asterisk instance.

By using small single-board computers such as Raspberry Pi, any types of businesses can deploy the cost affordable IP based PBX system. Raspberry Pi is small and light weight device, and runs on low power consumsption. So, having it for single tasks like VoIP with multiple functionallity provided by Asterisk PBX is very easy and economical. However, every since this device are smaller and lighter, it can only handle limited number of tasks. So, this paper will analyze the boudrise of Raspberry Pi by packets delays, network jitters, end-to-end network latency, and bandwidth through wired, wireless, cellular wireless connection.

This paper is organized as following convention. The next Section II, experiment setup information on this project. Section III shows detailed decription on each experments.

II. Software & Hardware Setup

abc

III. EXPERIMENT

A. Basic SIP Analysis

The Session Initiation Protocol (SIP) is an application layer control protocol for establishing, changing and terminating multimedia sessions, where the sessions can be IP telephony, multimedia sessions or multimedia conferences. SIP is the core protocol of many IP-based PBX applications, including Asterisk.

To demonstrate the main components of the SIP communication, a simplified scenario is implemented, as shown in Figure 1. Two softphone applications are running on the same computer, and an IP-based PBX application (Asterisk), running on a Raspberry Pi 4, is serving as the PBX server.

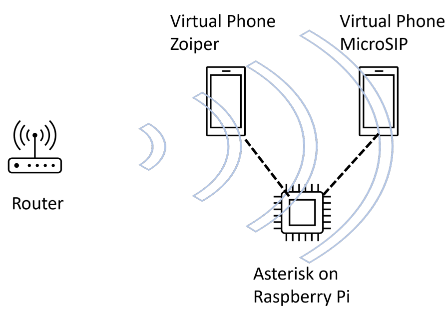

Fig. 1: Simplified Scenario

The SIP process starts with registration. All SIP terminals as User Agents should register with the registration server to inform their location, session capability, and other information.

Usually, when the SIP terminal (User Agent) is powered on or configured to perform a registration operation, a registration request message (REGISTER) is sent to the registration server, carrying all the information that needs to be registered. After receiving the registration request message, the registration server sends a response message to the terminal to inform it that the request message has been received. If the registration is successful, it will send a "200 OK" message to the terminal. However, the most common response is a "401 Unauthorized" message because authorization is required in most of the implantations for security purposes. As shown in Figure 2.

The SIP protocol adopts the Client-Server pattern, in which the calls are established between User Agents through the proxy server.

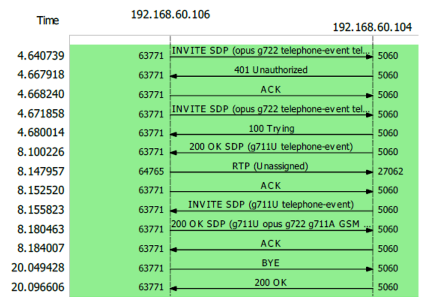

Fig. 2: Zoiper to Asterisk

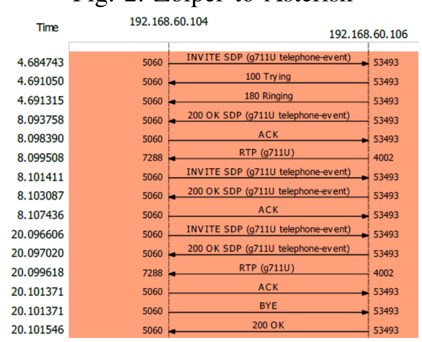

Fig. 3: Zoiper to MicroSIP