**Details of the Organization (with postal address)**

Juniper Networks is a multinational corporation headquartered in Sunnyvale, California that develops and markets networking products. Its products include routers, switches, network management software, network security products and software-defined networking technology.

Juniper was founded in 1996 by Pradeep Sindhu. The company received several rounds of funding from venture capitalists and telecommunications companies before going public in 1999. Juniper grew to $673 million in annual revenues by 2000. By 2001 it had a 37 percent share of the core routers market, challenging Cisco's once-dominant market-share. It grew to $4 billion in revenues by 2004 and $4.63 billion in 2014. Juniper appointed Kevin Johnson as CEO in 2008, Shaygan Kheradpir in 2013 and Rami Rahim in 2014.

Juniper Networks originally focused on core routers, which are used by internet service providers (ISPs) to perform IP address lookups and direct internet traffic. Through the acquisition of Unisphere in 2002, Juniper entered the market for edge routers, which are used by ISPs to route internet traffic to individual consumers. Juniper entered the IT security market with its own JProtect security toolkit in 2003, before acquiring security company NetScreen Technologies the following year. It entered the enterprise segment in the early 2000s, which accounted for one-third of revenues by 2005. As of 2014, Juniper has been focused on developing new software-defined networking products.

**Juniper Networks India Pvt. Ltd.**

Exora Business Park,

Marathalli-Sarjapur Outer Ring Road,

Bangalore, Karnataka 560103

**Name of the External guide with contact details and e-mail address:**

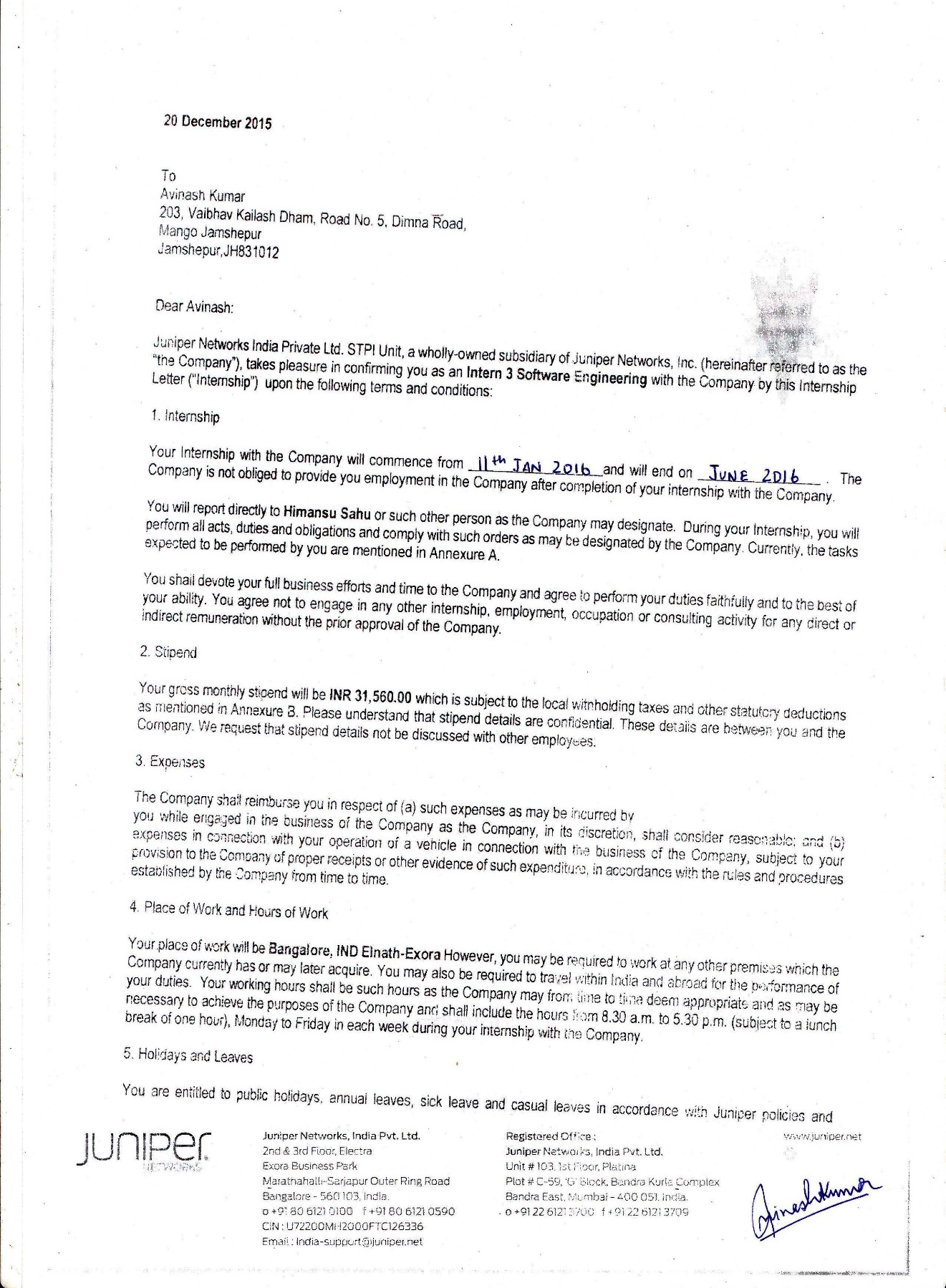
Pavan C V.

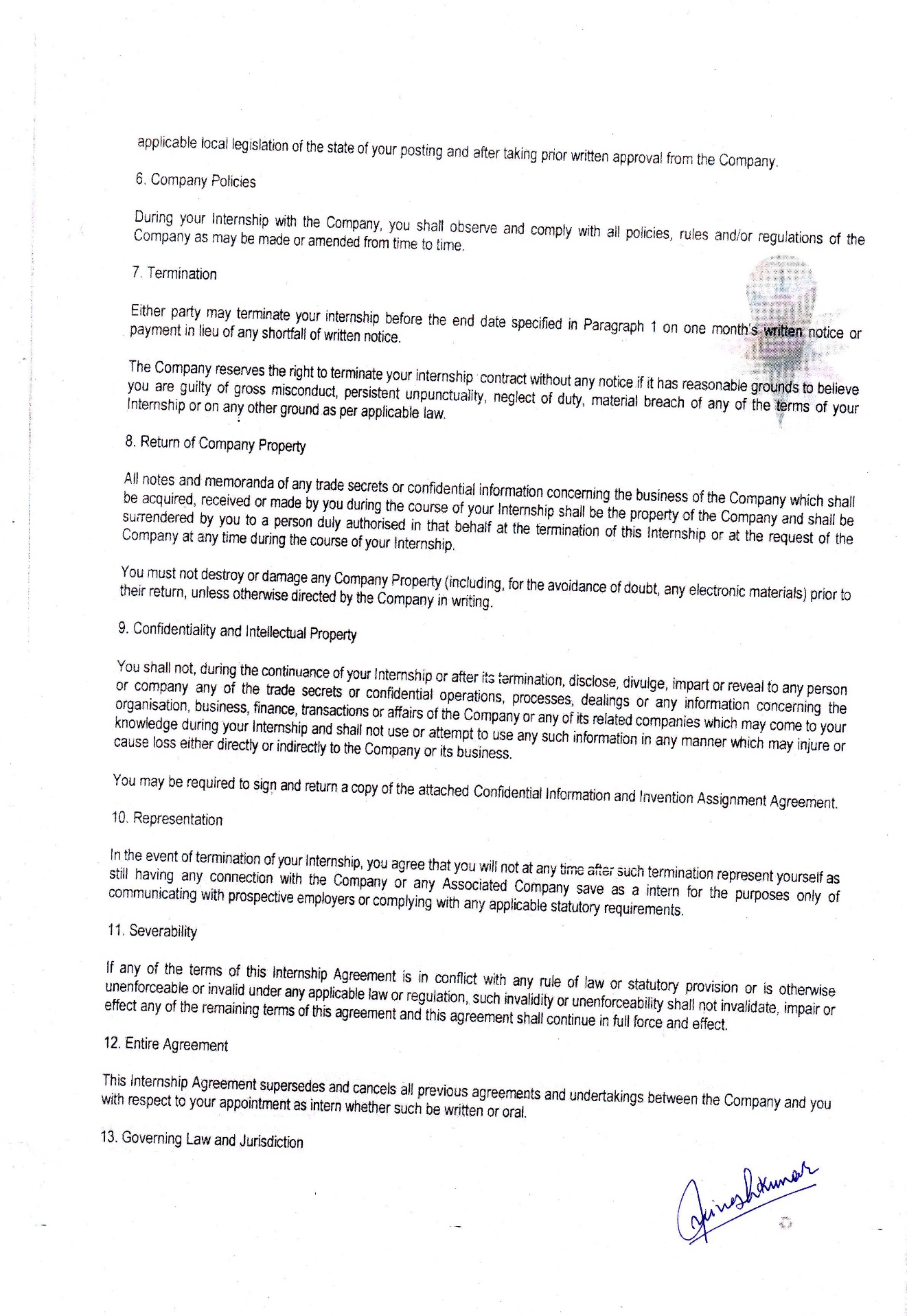
pavancv@juniper.net

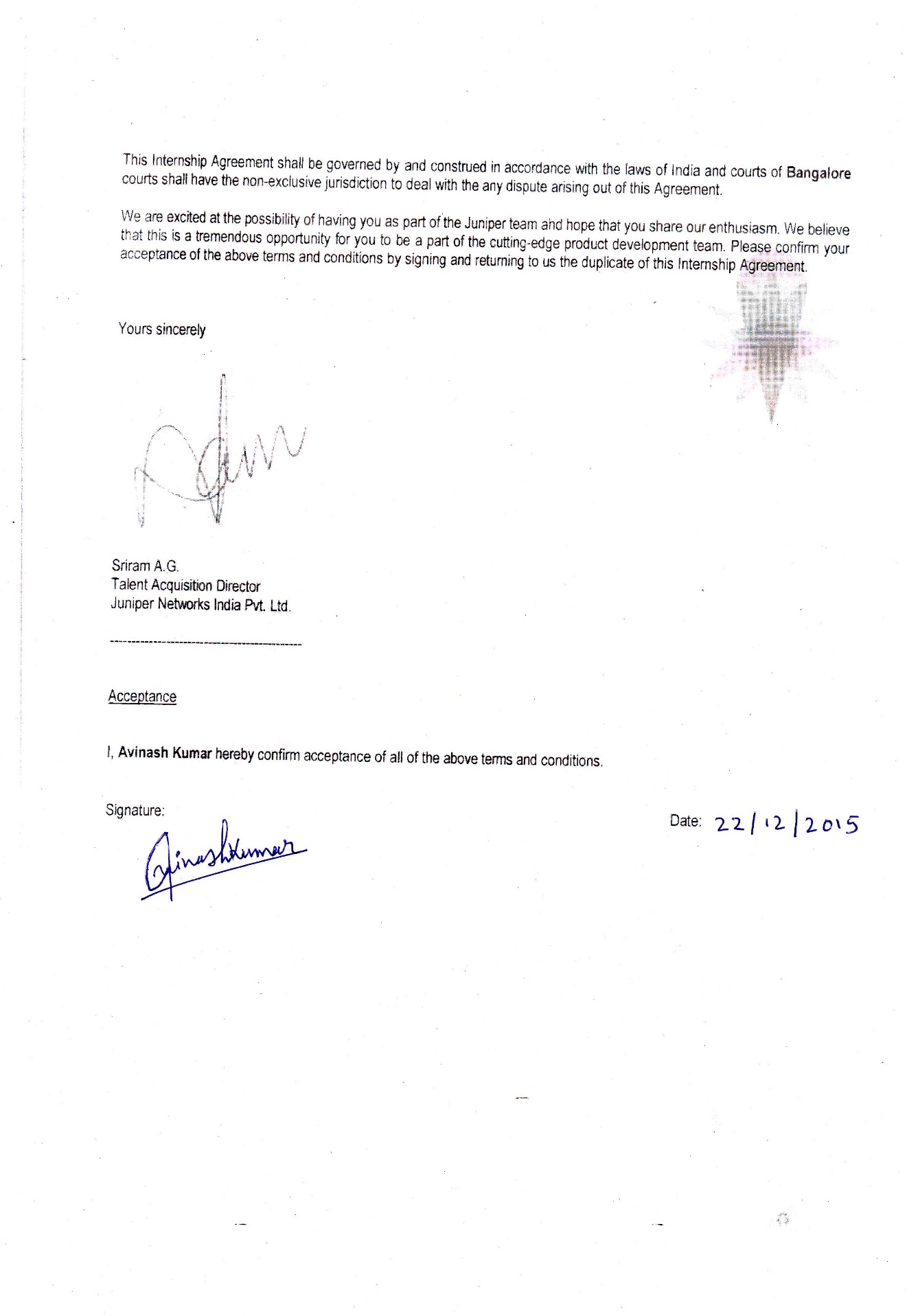
Software Engineer Staff

CDBU Switch

Juniper Networks Pvt. Ltd.

****

****

****

**CONTENTS**

1. Details of Organization and Mentor……………………………….1
2. Confirmation Letter………………………………………………..2
3. Abstract…………………………………………………...……….6
4. Introduction ……………………...………………………………..7
5. Problem Definition……...…………………………………………9
6. Objectives...…………...……………………………………….....10
7. Background…………..…………………………………………..11
8. Methodology………..……………….…………………………...13
9. Implementation Details……………..…………………………....15
10. Progress Till Date…………….……………….………………....20
11. Remaining Work……………….……………….……………….20
12. References……………………….……………….……………...22

# ABSTRACT

The phone-home daemon is a plug and play utility which is developed as part of the Universal Customer Premises Entity Platform. The uCPE platform is an integrated router and switch with extensible Virtual Network Functionality capabilities, the phone home module serves as initial boot up, upgrade and configuration functionality.

The Phone home daemon is responsible for the initial boot up and configuration of the uCPE device when the device is turned on at the customer premises. The utility is expected to automate the connecting to the Phone home server over internet and download the initial configuration and data/image. It is responsible for managing the image version currently present on the uCPE.

The Phone home daemon will have a web interface, to view the current status of the uCPE at all states from initialization to bootstrap completion phase. This will also include detailed log of the state machine, with error codes, info code, states, device id etc. This portal provides for the front end to the daemon which is accessible by connecting a device with a browser to the uCPE.

The end product is a zero touch provisioning system which automates the initial provisioning/ on boarding of the network device which would also include booting device with pre-installed proprietary and configuring the management interfaces.

# CHAPTER 1

# INTRODUCTION

* 1. **GENERAL INTRODUCTION TO THE TOPIC**

Phone home Daemon Client is a module that is used for day zero configuration of the integrated switch and router devices provided by Juniper Networks. This daemon is responsible to perform a zero touch provisioning system.

The daemon automatically connects to a server (Phone home server) and tries updating the configuration and downloads the appropriate images and installs them on the switch. This process can be seen on the captive portal and the logs that are displayed there. The logs each state and important messages on the system log too. The environment it run in provides the phone home client with the capability to configure DHCP server, other virtual machines, reboot/restart entire switch and install downloaded image from PHS.

The PHC is three distinct interfaces to the outside world.

1. Process configuration from user
2. Perform plug and play (Download configuration image and upgrade)
3. Interface with user via LAN ports.
   1. **ORGANIZATION**

Juniper Networks is a multinational corporation headquartered in Sunnyvale, California that develops and markets networking products. Its products include routers, switches, network management software, network security products and software-defined networking technology.

* 1. **AREA OF COMPUTER SCIENCE**

Computer Networks and Unix Systems will be the area of computer science that this project can be classified into. The client runs on a network device and is responsible for its provisioning. This client uses Unix environment to run a Daemon that automatically spawns with the system on initial configuration.

This project also applies some of the fundamental concepts of Software Engineering in development cycle, but can still be rightly classifies into Computer Network as Software solution

* 1. **HARDWARE AND SOFTWARE REQUIREMENTS**
     1. **Hardware Requirements**
* This daemon runs on uCPE which are integrated switch and router.
  + 1. **Software Requirements**
* The design assumes POSIX complaint thread library support from the underlying platform/OS
* The daemon depends on the availability of DHCP server, client.
* HTTP Server with CGI script support
* Python support
* SHA1/MD5 Utilities to verify image
* Basic Linux utilities like grep and awk.

**CHAPTER 2**

**PROBLEM DEFINITION**

The main aim is to implement a Zero Touch Provisioning system that will take care of the Day zero configuration and will be responsible for Bootstrapping OS to applying custom Configuration and take it to Provisioning Complete stage.

The new approach is used as existing approach are either Ad hoc which uses custom scripts or are Proprietary (Plug and Play).

This Daemon should be able to provide for the ZTP. The web UI will be used to convey the ZTP progress as well as to authenticate upgrade, if required. The utility is expected to automate the connecting to the Phone home server over internet and download the initial configuration and data/image. It is responsible for managing the image version currently present on the uCPE.

**CHAPTER 3**

**OBJECTIVES**

Automate the initialization and bootstrap process of the uCPE at the customer premise thus improving the overall user experience with device. To achieve this following goals needs to be achieved.

* The Daemon should be able to get the device id and contact the Phone home server automatically after the device is started at customer premise.
* Maintain different threads for configuration, CGI socket and state of the daemon.
* Maintain a trace of the events in the syslog of all the states for debugging and crash report.
* Verifying the image integrity before installation or upgrade.
* To have a captive portal which displays the ZTP status while the process is running.

The end product will be a daemon running on the uCPE which will also have a portal as the front end. The portal should be accessible in any external device that supports a browser and which can be connected to the uCPE through a LAN cable to its management port.

.

**CHAPTER 4**

**BACKGROUND**

* 1. DAEMON

In multitasking computer operating systems, a daemon is a computer program that runs as a background process, rather than being under the direct control of an interactive user. Traditionally, the process names of a daemon end with the letter d, for clarification that the process is, in fact, a daemon, and for differentiation between a daemon and a normal computer program. For example, syslogd is the daemon that implements the system logging facility, and sshd is a daemon that services incoming SSH connections.

In a Unix environment, the parent process of a daemon is often, but not always, the init process. A daemon is usually either created by a process forking a child process and then immediately exiting, thus causing init to adopt the child process, or by the init process directly launching the daemon. In addition, a daemon launched by forking and exiting typically must perform other operations, such as dissociating the process from any controlling terminal (tty). Such procedures are often implemented in various convenience routines such as daemon in Unix.

Systems often start daemons at boot time and serve the function of responding to network requests, hardware activity, or other programs by performing some task. Daemons can also configure hardware (like udevd on some Linux systems), run scheduled tasks (like cron), and perform a variety of other tasks.

* 1. CGI SCRIPT

A ***CGI script*** is any program that runs on a web server.

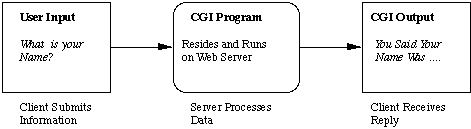


Figure 4.2.1 The Common Gateway Interface

CGI stands for **C**ommon **G**ateway **Interface** CGI defines a standard way in which information may be passed to and from the browser and server. Any program or script that can process information according to the CGI specification can, in theory, be used to code a CGI script.

* 1. UNIX DOMAIN SOCKET

Sockets provide point-to-point, two-way communication between two processes. Sockets are very versatile and are a basic component of interprocess and intersystem communication. A socket is an endpoint of communication to which a name can be bound. It has a type and one or more associated processes.

he UNIX domain provides a socket address space on a single system. UNIX domain sockets are named with UNIX paths. Sockets can also be used to communicate between processes on different systems. The socket address space between connected systems is called the Internet domain.

* 1. ZERO TOUCH PROVISIONING

Zero Touch Provisioning allows you to provision new Juniper Networks switches in your network automatically, without manual intervention. When you physically connect a switch to the network and boot it with a default factory configuration, it attempts to upgrade the Junos OS software automatically and auto install a configuration file from the network.

**CHAPTER 5**

**METHODOLOGY**

The project uses the agile methodology of Software Development Life Cycle. Agile approaches are typically used in software development to help businesses respond to unpredictability.

The Agile framework used for the project is Scrum. Scrum is an iterative and incremental [agile software development](http://en.wikipedia.org/wiki/Agile_software_development) framework for managing product development. A key principle of Scrum is its recognition that during a project the customers can change their minds about what they want and need.

The project can be divided into three main phases which will represent three main modules of the daemon responsible for making the whole system work.

**5.1 Phone home daemon state machine**

This is the crux of the whole system which maintains the states of the daemon. This is what will be responsible for spawning new threads and having an overall monitor on the system. The daemon goes through various states throughout its life cycle. These states include the Initialization, Connecting, Activation required, Activated, Downloading, Download Complete, Updating, Bootstrap complete or failed etc. This thread is also responsible for restarting the configuration process or re initializing the daemon.

The state machine has keeps all the information in the memory and no database as such is required. This is also responsible to downloading the images from the server and then performing the upgrade of the device followed by a reboot.

**5.1.1 Dependencies and Libraries required**

The environment in which the daemon runs is assumed to be POSIX complaint and thus should support standard POSIX multithreading environment. The multithreaded nature of the daemon is kept to support loose coupling of the system and make the working asynchronous. The events are generated using the system signals like SIGHUP, SIGHTERM etc.

**5.2 The Phone home server**

The Phone home server is the server that the daemon must contact after the initialization process is complete. This server maintains latest images for the devices and will also be used for authentication by using some activation code that will be provided by the end user. This image downloaded from the server should be verified for its integrity and then only be used for update. The servers can be configured in the daemon and if one server fails it should go on to try connecting to the next server or any redirect servers that is already configured on the device.

This server is contacted over the wan port of the device after the first connection is made, for a good download speed of the images for the device and seamless upgrade of the system. The server maintains a record and configuration for all the registered devices and when contacted will accordingly respond with the correct image URL to corresponding devices.

**5.3 The front-end for management of PHC**

This is the module which is the face to the ZTP protocol to shows the status of the whole bootstrap process to the end user onto a device that is connected to the uCPE in the same network through the lan port. This is a captive portal with a sever running and listening and responding on a port. The response from the daemon is transferred to the server which in turn is rendered to the browser. This also shows a detailed log of the events.

This portal will also ask the user to enter activation codes if required and then validate accordingly using the Phone Home server. The portal is set to be accessible using a static IP that is preconfigured on the switch.

**CHAPTER 6**

**IMPLEMENTATION DETAILS**

**6.1 PHC Architecture**

The phone daemon runs on top of Juniper’s Switch which requires Unix environment to execute. The daemon is part of a Linux container that is managed by the host hypervisor that also monitors other Virtual Machines.

**6.1.1 Linux Containers**

LXC (Linux Containers) is an operating-system-level virtualization method for running multiple isolated Linux systems (containers) on a control host using a single Linux kernel.

The Linux kernel provides the cgroups functionality that allows limitation and prioritization of resources (CPU, memory, block I/O, network, etc.) without the need for starting any virtual machines, and namespace isolation functionality that allows complete isolation of an applications' view of the operating environment, including process trees, networking, user IDs and mounted file systems.

LXC combines kernel's cgroups and support for isolated namespaces to provide an isolated environment for applications. Docker can also use LXC as one of its execution drivers, enabling image management and providing deployment services.

**6.1.2 Virtual Machine**

In computing, a virtual machine (VM) is an emulation of a particular computer system. Virtual machines operate based on the computer architecture and functions of a real or hypothetical computer, and their implementations may involve specialized hardware, software, or a combination of both.

A virtual machine (VM) is a software implementation of a machine (for example, a computer) that executes programs like a physical machine. Virtual machines are separated into two major classes, based on their use and degree of correspondence to any real machine, a *system virtual machine and* A *process virtual machine.*

**6.1.3 Hypervisor**

A hypervisor or virtual machine monitor (VMM) is a piece of computer software, firmware or hardware that creates and runs virtual machines.

A computer on which a hypervisor is running one or more virtual machines is defined as a host machine. Each virtual machine is called a guest machine. The hypervisor presents the guest operating systems with a virtual operating platform and manages the execution of the guest operating systems. Multiple instances of a variety of operating systems may share the virtualized hardware resources.

**6.1.4 Architectural overview of PHC**

Below is the architectural overview of the Phone home daemon with Hypervisor as the main host and a virtual machine

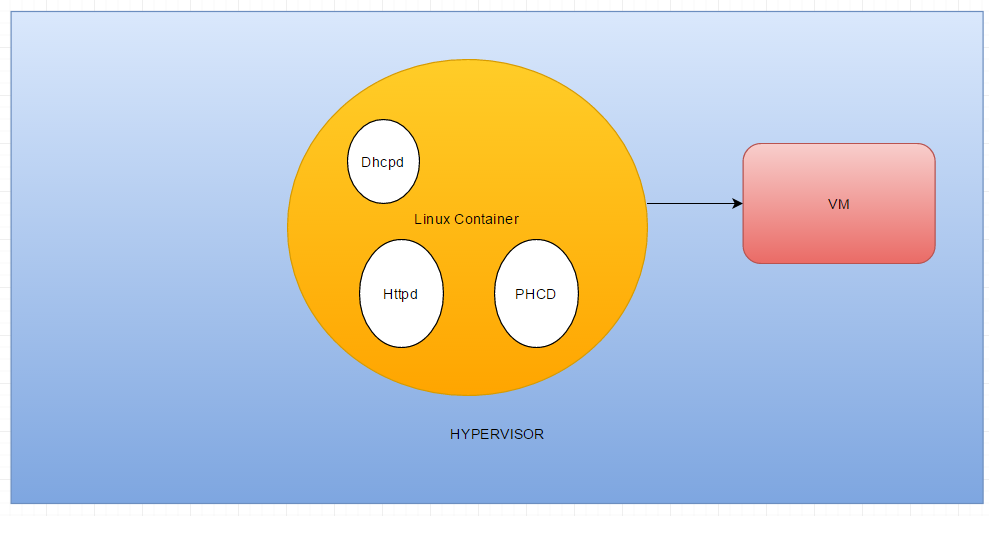


Figure 6.1.1 PHCD Architecture overview

**6.2 Daemon Core implementation**

This was done by separating the configuration, management and cgi socket on dedicated independent threads. These threads were implemented using pthreads api and proper locking mechanism was used to lock the main singleton structure while other threads also try using it. Below is the Sequence diagram for the daemon.

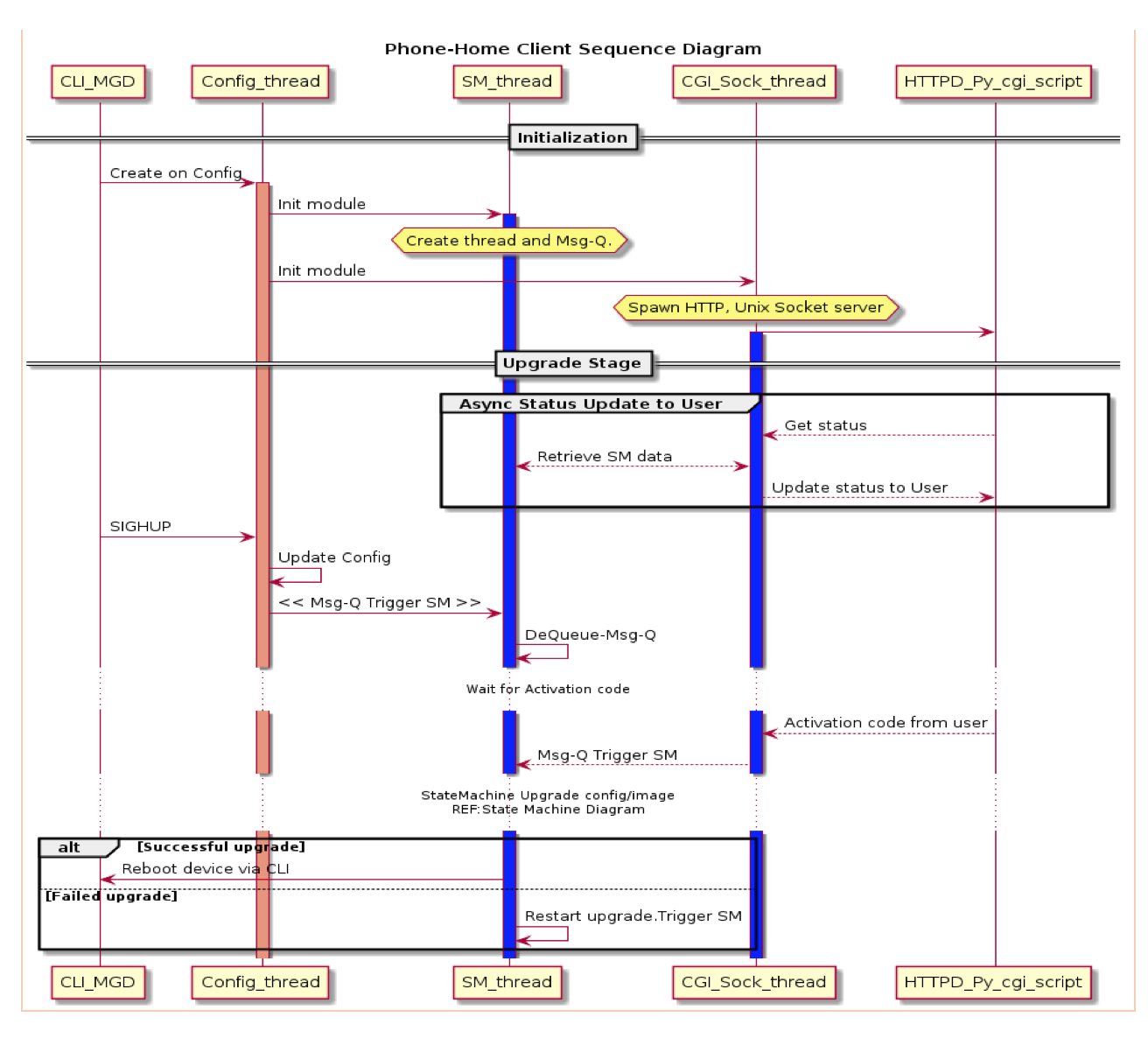


Figure 6.2.1

The daemon core is the state machine which runs on a separate thread. The other threads as mentioned in the Figure 6.2.1 represent their respective jobs.

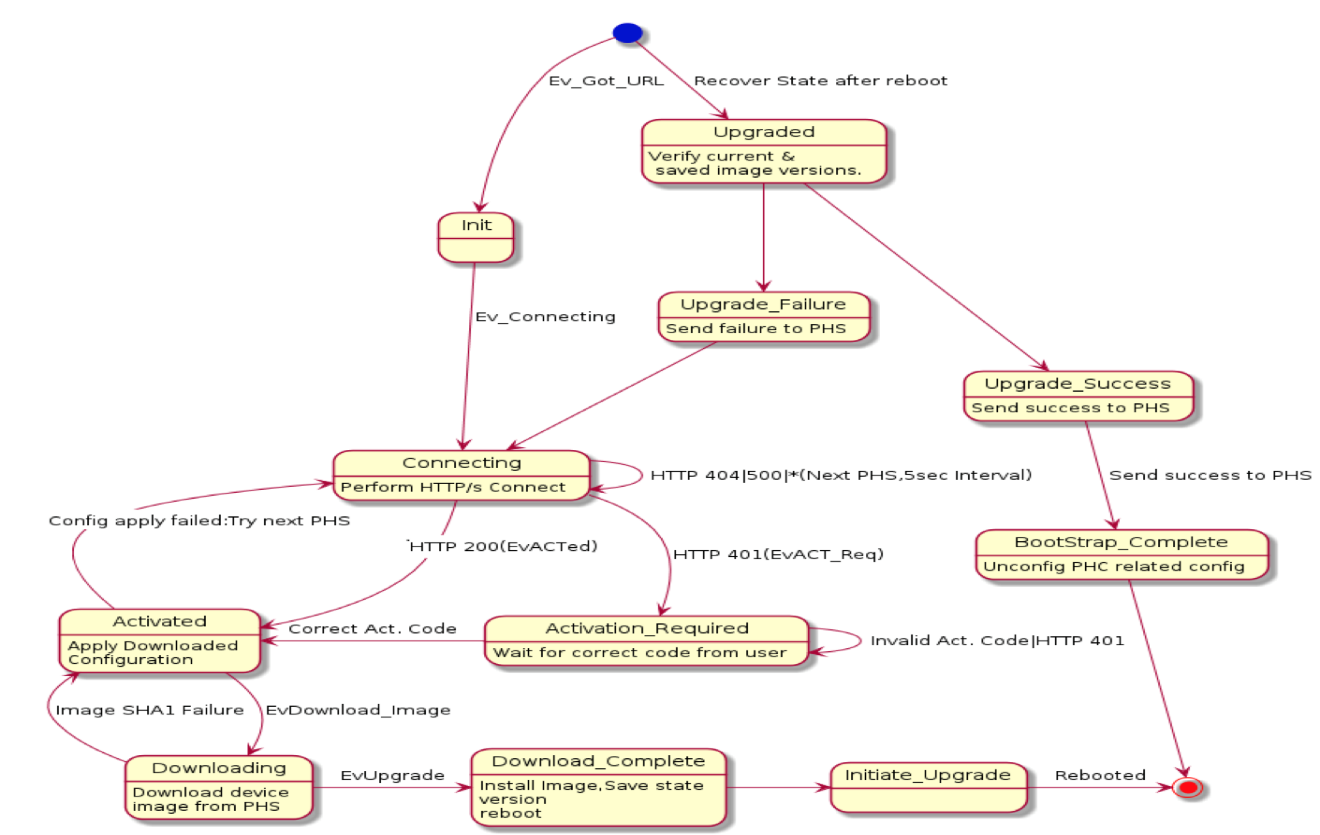


Figure 6.2.2

Figure 6.6.2 the representation of the state diagram of the daemon.

The whole daemon is made in c and compiled using bmake which is BSD version of gnu make. Vim code editor was used to make this.

**6.3 PHC Portal**

This Portal is backed up by a HTTP server that is capable of running CGI script. The front end of the portal is made using Web technologies like HTML, CSS, Javascript. The Portal sends ajax request to the server which responds with a Http reply which further gets rendered on the front end using Javascript. The server gets all the dynamic data by executing a CGI script that reads from Unix sockets onto which the daemons is writing to.

The scripts are written in python with no external modules included. This script is responsible to send the http request with proper data to the portal. The dynamic data is being sent as plain text.

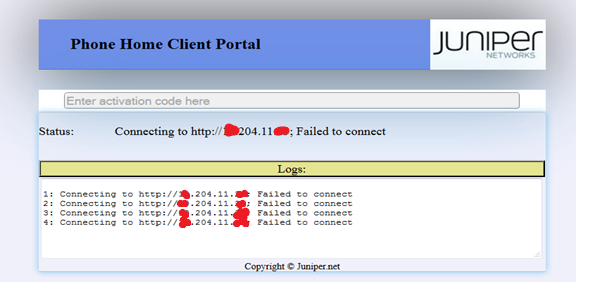


Figure 6.3.1 The Portal for the PHC daemon

**CHAPTER 7**

**PROGRESS TILL DATE AND THE REMAINING WORK**

**7.1 PROGRESS TILL DATE**

* The Phone home daemon state machine is working.
* The configuration can be done using cli.
* The portal is up and running.
* The daemon was returning plain text info previously it is changed to give a JSON response now so that it can be easily consumed on the front end side and will almost work as an Api call to the CGI script, moreover the messages displayed have been refactored and made more generic.
* Few bugs were reported that has been fixed such as configuration error and deadlock in states and some regarding race condition too.

**7.2 REMAINING WORK**

The phone home client is up and running but as and when a bug or a new requirement comes up from the customer it is being modified. There are some know issues and improvements that is to be worked upon in the next stage.

**7.2.1 Revamping the Portal**

The portal needs a complete revamp to give it a more professional look and feel for a better UI/UX. This will need to be done with minimum dependencies, strictly speaking zero dependencies on external resources or libraries. The Portal will get a new design and along with it refactoring of javascript code for easy maintenance has to be done, which will be done by making the code more modular and object oriented. The portal might include SVG images as animation that will need to be made.

Some of the new features being expected are

* Fully Responsive, to support device of all screen size
* Should be cross browser compatible
* Minimum dependencies but with good UI/UX
* Fast page load speed.

**7.2.1 Ethernet over USB support**

Currently LAN cable is used to connect to the switch and configure. The LAN cable connection has some big drawback, such as

* LAN Ports are not available on many devices such as Macbooks and Tablets. The idea is that any device including a phone should be able to connect using USB interface and configure the switch.
* With LAN ports the problem is that VMs take time to come up and the user has to wait for long time until they boot. This limitation is a setback on the user experience that needs to be resolved.

Ethernet over USB will allow any device with a browser and appropriate drivers to be able to connect to the switch, manage it and have access to the portal. It will overcome the above mentioned bottlenecks and improve the overall user experience.

**CHAPTER 8**

**REFERENCES**

[1] The Design of the Operating System by Maurice Bach

[2] Python official documentation (https://www.python.org/doc)

[3] Linux Man pages

[4] https://linuxcontainers.org

Project Details

|  |  |  |  |
| --- | --- | --- | --- |
| *Student Details* | | | |
| **Student Name** | AVINASH KUMAR | | |
| Register Number | 120905532 | Branch / Roll No | CSE |
| Email Address | avi.loyola@gmail.com | Phone No (M) | +918867600811 |
|  | | | |
| *Project Details* | | | |
| **Project Title** | **Phone Home Client Daemon, For Zero Touch Provisioning** | | |
| Project Duration | 6 months | Date of reporting | 11th January |
|  |  | | |
| *Organization Details* | | | |
| **Organization Name** | **Juniper Networks Pvt. Ltd.** | | |
| Full postal address with pin code, telephone number, Fax and email | Juniper Networks, India Pvt. Ltd, Elnath-Exora Marathalli, Sarjapur Outer Ring Road BANGALORE, KARNATAKA, 560103 | | |
| Website address | http://www.**juniper**.net | | |
|  |  | | |
| *External Guide Details* | | | |
| **Supervisor Name** | **Pavan C V** | | |
| Designation | Software Engineer Staff | | |
| Email address | pavancv@juniper.net | Phone No (M) |  |
|  |  | | |
| *Internal Guide Details* | | | |
| **Faculty Name** | **Mrs. Rajashree Krishna** | | |
| Full contact address with pin code | Department of Computer Science  Manipal Institute of Technology  Manipal- 576104 | | |
| Email address | raji.krish@manipal.edu | | |