|  |  |  |  |
| --- | --- | --- | --- |
| 自动化框架 | 文档编号 | 版本 |  |
| 1 | V1.0 |  |
|  | | 共 7页 |

自动化框架

历史更改

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NO | 更新日期 | 版本 | 修改描述 | 作者 | 审核 |
| 0 | 2021年07月27日 | V1.0 | 第一版 | 孙建涛 | 孙建涛 |
| 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |

目录

[1 简介 2](#_Toc78285675)

[1.1文档目的及范围 2](#_Toc78285676)

[1.2应用场景描述 2](#_Toc78285677)

[1.3框架设计的原则 2](#_Toc78285678)

[2 自动化设计思路 3](#_Toc78285679)

[3 设计约束 3](#_Toc78285680)

[4 硬件配置 3](#_Toc78285681)

[5 物理组网 4](#_Toc78285682)

[4 特性设计 5](#_Toc78285683)

[4.1 整体架构设计与描述 5](#_Toc78285684)

[4.2 自动化框架的选择 5](#_Toc78285685)

[4.3 发包工具的实现与选择 6](#_Toc78285686)

[4.4 杂包的处理 6](#_Toc78285687)

[4.5 保证发送报文不乱序 7](#_Toc78285688)

[4.6 流量内容和乱序校验 7](#_Toc78285689)

# 1 简介

## 1.1文档目的及范围

本文档主要是来设计分流器功能自动化设计。目的是：

1. 解决测试对测试仪绝对依赖的问题。
2. 释放人力资源。
3. 提升测试效率。
4. 增加测试覆盖率。

范围为测试人员及运维人员

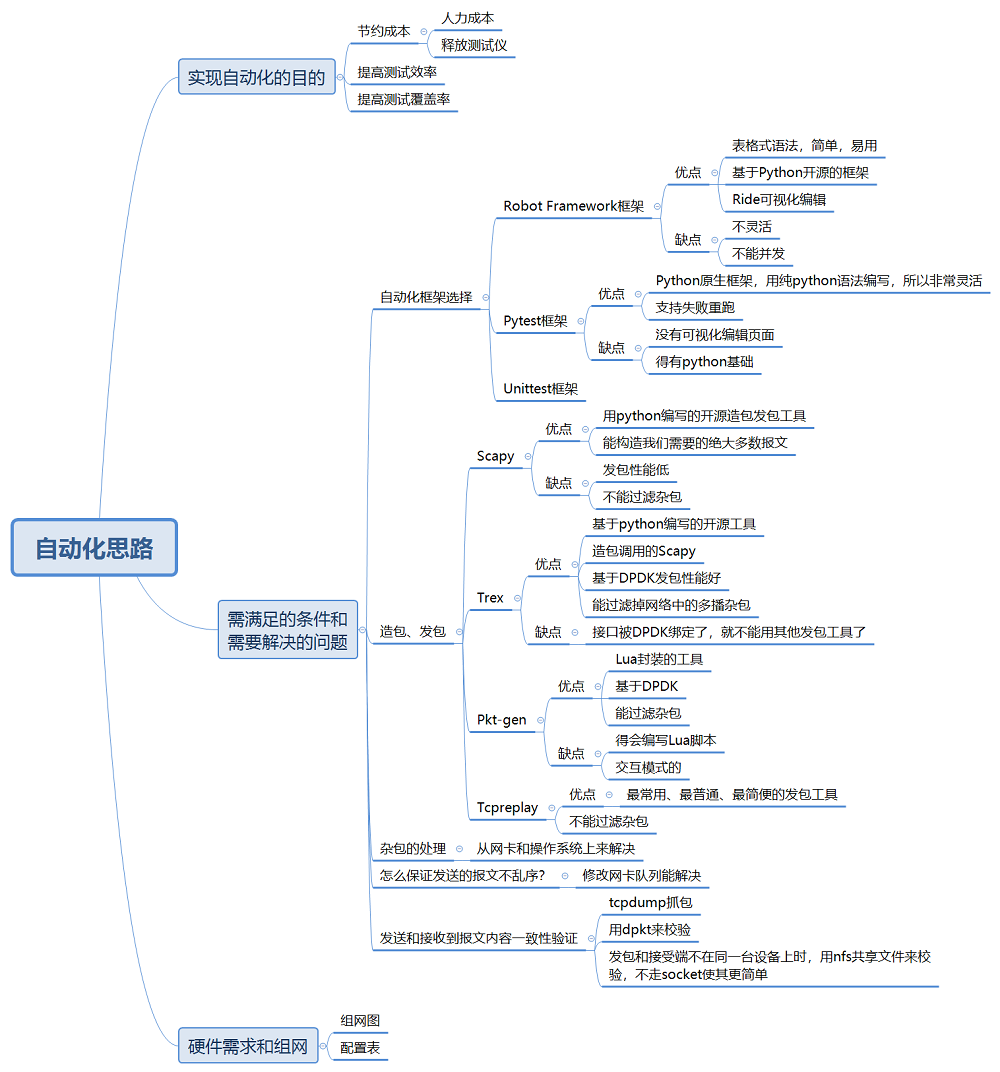
## 1.2应用场景描述

适用于功能测试的能实现自动化的各个场景。

## 1.3框架设计的原则

本着简单、易用的原则。稍微会点python语法的，了解基本的Robot Framework的都能够编写自动化测试用例。

# 2 自动化设计思路

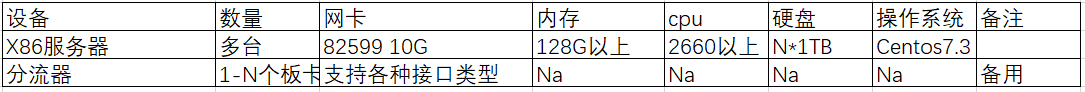


# 设计约束

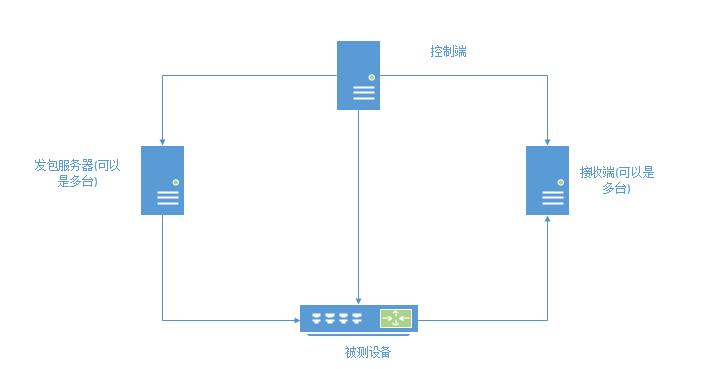
1、此设计只适合功能自动化测试。

2、此自动化设计不依赖于测试仪器。

# 4 硬件配置



# 5 物理组网



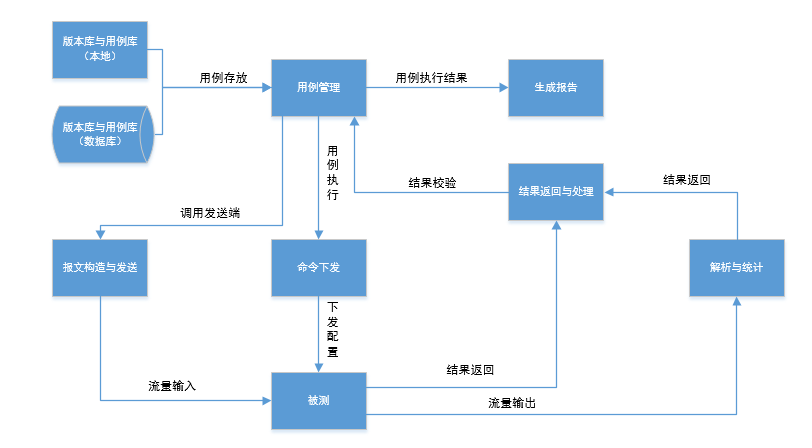
组网

描述：

1. 控制端、发包端、接收端各在一台设备上也可以在一台设备上。
2. 设备管理方面所有设备都互相连通。
3. 业务上控制端->发送端->被测设备->接收端->控制端；控制端->被测设备。
4. 发送端和接收端可以是多台设备。

# 4 特性设计

## 4.1 整体架构设计与描述



主要包括用例管理部分、报文构造与发送部分、命令下发部分、报文接收与结果校验部分。

1. 用例管理模块下发命令配置，被测侧返回命令下发结果。
2. 用例管理模块调用发送端模块进行报文构造与发送，被测侧输出流量到接受模块。
3. 接收模块进行解析与统计，并返回结果。
4. 用例管理获取到结果后进行处理，最终生成测试报告。

## 4.2 自动化框架的选择

1. Robot Framework框架

主要用于测试驱动类型的开发与验收中，基于python的开源框架，也可以基于.Net的IronPython和基于Java的Jpython上运行，兼容多种平台。优点是表格是语法、简单。缺点不灵活、不能够并行测试。

秉着简单易用、容易上手的原则选择了RF框架。

2. Unittest框架

单元测试框架不仅可以适用于单元测试，还可以适用WEB自动化测试用例的开发与执行，该测试框架可组织执行测试用例，并且提供了丰富的断言方法，判断测试用例是否通过，最终生成测试结果。

unittest为python内置库,不需要单独安装。

3.Pytest框架

pytest是一个非常成熟的全功能的Python测试框架，纯python编写，得有一定的python基础，支持参数化，能够支持简单的单元测试和复杂的功能测试。

Pytest是基于Unittest的扩展的测试框架，提供了更多的功能和可扩展性。支持失败重跑而Unittest不具备。

## 4.3 发包工具的实现与选择

1、Tcpreplay

此工具是Linux的发包工具，发包口不需要配置IP地址的情况下就能够发送pcap报文，简单易用，不需要再开发。缺点是网卡协议栈的杂包过滤不掉。

2、Pkt-gen

此工具能够构造大部分常见报文并且能够发包，是基于DPDK的，lua封装的工具。能够屏蔽掉网卡协议栈杂包，是交互模式的。需要调用lua脚本，如果是发送pcap文件的，直接shell执行脚本就可以。需要安装DPDK+python3+lua环境。

3、scapy

用python编写的交互式数据包处理程序，它能让用户发送、抓取、伪造网络报文。也能发送pcap文件。性能上差一些。

4、Trex

Trex发包工具是基于DPDK的，封装了scapy。在性能和功能都很不错。

对于发包工具的选择：根据测试场景来定，某个场景需要那个就用哪个。

## 4.4 杂包的处理

1. 如果操作系统为Centos7.3，网卡为82599，那么就不会主动发送组播报文。

2. 如果操作系统为Centos7.8，网卡为82599，那么得把NetworkManager 关掉命令为：

# systemctl stop NetworkManager、# systemctl stop NetworkManager

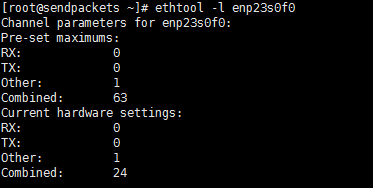
# ip link set *enp101s0f3* multicast off

3. 如果操作系统为Centos7.8并且网卡为X710，上述第2种方法无效，应该是X710网卡底层主动发送的，这个需要改其驱动。

## 4.5 保证发送报文不乱序

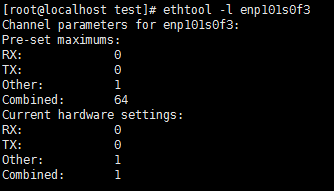
网卡配置修改：

1、查看通道ethtool -l enp23s0f0



2、发包端修改万兆网卡口接收包队列。

ethtool -L enp101s0f3 combined 1



3、接收端也做同样设置。

4、不要配置为混杂模式，以免有杂包。

5、如果用tcpreplay发包，不要使用-t来发送，这样会造成报文乱序。

## 4.6 流量内容和乱序校验

* 校验思路：

1. 发包端的pcap是事先已经准备好的，把它放到nfs共享目录下。
2. 接收端开启tcpdump抓包。
3. 用tcpreplay向被测设备发送报文。
4. 被测设备经过处理，转发给接收端。
5. 接收端抓到报文后把文件放到nfs共享目录下。
6. 这时开始校验发送和接收的pcap文件内容和报文顺序。

* 遇到的问题：

1. 到底用什么工具校验内容最好，还是自己写代码去解析pcap文件然后对比？
2. 由于接收端抓包报文时用的时间戳是当前的时间戳，所以和发送端的pcap中时间戳不一致？
3. 比对的性能如何？时间能否接受？
4. 乱序怎么验证？

* 问题怎么解决：

1. 针对与第一个问题，首先思路是先找一找有没有类似的能满足这种需求的工具，如果找不到只能自己写代码来解析。经过大海捞针的寻找，终于找到了，它就是dpkt，这个工具可以把pcap文件头、pcap报文头和时间戳给剥离掉，所以呢直接比对二进制就行了，完美解决。如果自己的代码能力很强自己写也可以。
2. 针对第二个问题上边已经解决了。
3. 针对于性能问题，经过在linux上测试，校验10个G报文花费了4分多钟，测功能应该足够了，这个问题解决。
4. 乱序问题其实也解决了，dpkt逐条检查，如果两个文件中的顺序不对应，会比对失败。