asyncio并发管理网络设备

In [ ]:

```
#coding:utf8
import paramiko
import os
import time
import asyncio

class ConnectError(Exception):
    pass


class sshCisco:

    def __init__(self, host, username, password, pripwd, port):
        self.conn = sshCisco.__sshConn(host, username, password, pripwd, port)

    @staticmethod
    def __sshConn(host, username, password, pripwd, port):
        ssh = paramiko.SSHClient()
        ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())
        ssh.load_system_host_keys()
        try:
            ssh.connect(host, port, username, password, allow_agent=False, look_for_keys=False)
        except Exception as e:
            raise ConnectError("paramiko ssh connect %s error! %s" % (host, e))
        else:
            ssh_shell = ssh.invoke_shell()  # 用户模式
            ssh_shell.send(bytes('enable\n', encoding='utf8'))
            time.sleep(1)
            ssh_shell.send(bytes(pripwd + '\n', encoding='utf8'))  # 进入特权模式
            time.sleep(1)
            line = str(ssh_shell.recv(1024), encoding='utf8')
            if line.endswith("#"):
                return ssh_shell
            else:
                raise ConnectError("paramiko login cisco privilege level error! %s" % host)

    async def async_executeCmd(self, cmd):
        result = ''
        if cmd == "disconnect":
            self.close()
        self.conn.send(bytes(cmd + '\n', encoding='utf8'))
        await asyncio.sleep(2)
        line = str(self.conn.recv(65535), encoding='utf8')
        result += line
        if result.endswith("#"):
            return result
        else:
            while 1:
                self.conn.send(bytes(' ', encoding='utf8'))
                await asyncio.sleep(2)
                line = str(self.conn.recv(65535), encoding='utf8')
                result += line
                if line.endswith("#"):
                    break
            return result

    def executeCmd(self, cmd):
        result = ''
        if cmd == "disconnect":
            self.close()
        self.conn.send(bytes(cmd + '\n', encoding='utf8'))
        time.sleep(2)
        line = str(self.conn.recv(65535), encoding='utf8')
        result += line
        if result.endswith("#"):
            return result
        else:
            while 1:
                self.conn.send(bytes(' ', encoding='utf8'))
                time.sleep(2)
                line = str(self.conn.recv(65535), encoding='utf8')
                result += line
                if line.endswith("#"):
                    break
            return result

    def close(self):
        print('%s Connection interrupt!!' %self.host)
        self.conn.close()
        exit()


if __name__ == '__main__':
    x = sshCisco("172.16.70.150", "ccna", "ccna", "ccna","22")
    print(x.executeCmd("show run"))
```