

某局点希望我司设备实现如下条件判定：

执行dis ospf peer X.X.X.X，如果邻居正常就对与OSPF无关的A接口执行undo shutdown，如果邻居不正常就对A接口执行shutdown，每3秒执行一次判定。这里如果用传统协议去做，很难实现。

无。

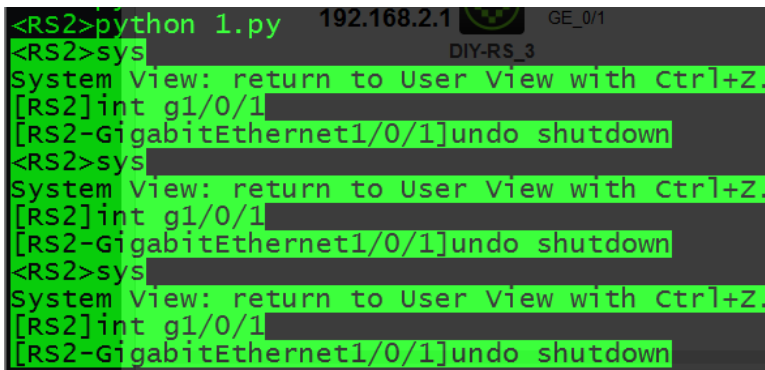
传统协议都是按照实际网络需求编写的，自由度上受到了限制，如果我们亲手去设计一个小协议，那么很多刁钻的问题都可以轻松解决。我司V7平台引入了嵌入式语言的做法大大提高了解决网络问题的自由度，这里我们用Python举例来实现上面的需求。

首先对应需求编写一份Python代码：

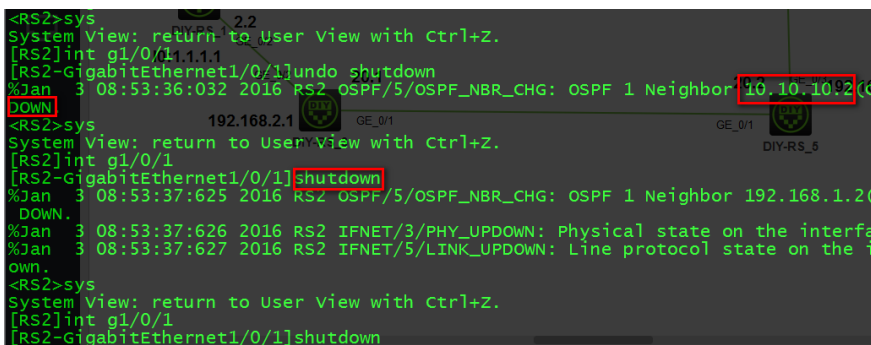
```
import time
import re
import comware

while True:
    results = comware.CLI(& # 39;dis ospf peer 192.168.4.2& # 39;, False).get_output()//设备自己执行dis ospf peer 192.168.4.2这条命令
    access = False
    p = re.compile(& # 39;Full& # 39;)//用关键字Full去匹配执行命令后显示出来的信息
    for result in results:
        string = str(result)
        if p.findall(string):
            access = True
            break
    if access:
        comware.CLI(& # 39;sys ;int g1/0/1 ;undo shutdown& # 39;,True)//如果显示出来的信息中有Full,那么就执行sys intg1/0/1 undo shutdown
    else:
        comware.CLI(& # 39;sys ;int g1/0/1 ;shutdown& # 39;,True)//如果显示出来的信息中没有Full,那么就执行sys int g1/0/1 shutdown
    time.sleep(3)//3表示程序每3S执行一次
```

将文件的名字改为1.py，然后将此文件传到设备的FLASH中去，让设备运行此文件：



从上图我们可以看到当执行Python文件后，代码已经生效，由于此时的邻居是full的，因此设备会不停的对G1/0/1口执行undo shutdown，如果此时我们将邻居DOWN掉：



我们可以看到当邻居DOWN掉后，根据代码的原意，设备对G1/0/1口执行了shutdown操作。

此时如果我们将OSPF邻居变为full状态：

```
<RS2>sys 10:1.1.1.1
System View: return to User View with Ctrl+Z.
[RS2]int g1/0/1
[RS2-GigabitEthernet1/0/1]shutdown
%Jan 3 08:56:36:2016 RS2 OSPF/5/OSPF_NBR_CHG: OSPF 1 Neighbor 10.10.10.2
to FULL
<RS2>sys
System View: return to User View with Ctrl+Z.
[RS2]int g1/0/1
[RS2-GigabitEthernet1/0/1]undo shutdown
%Jan 3 08:56:38:715 2016 RS2 IFNET/3/PHY_UPDOWN: Physical state on the interfa
%Jan 3 08:56:38:716 2016 RS2 IFNET/5/LINK_UPDOWN: Line protocol state on the
p.
<RS2>sys
System View: return to User View with Ctrl+Z.
[RS2]int g1/0/1
[RS2-GigabitEthernet1/0/1]undo shutdown
```

我们可以看到根据代码的原意，设备开启了G1/0/1接口，实现了我们的需求。