

我们使用 mininet 中的 iperf 工具在网络中生成 UDP 流量，iperf 客户端传送数据流到 iperf 的服务端，由服务端接收并记录相关信息。

mininet 自带控制台可供使用的命令虽然已经比较丰富，但却并未给出较为明确的 API 接口来支持用户自定义命令。

为了依照 Payless 论文进行指定的主机间 CBR 业务流，我们需要将自定义命令添加到 mininet 中，在 mininet 中完成新命令的拓展。

在 mininet 中进行功能拓展主要分为 3 步：

1. 修改 mininet/net.py: net 模块实现 Mininet 类，是仿真平台 mininet 的主体类，该类完成节点添加配置、网络基本功能和一些选项功能的实现。我们需要将我们自定义的函数定义在该类中。

调用 iperf 命令的 iperf_extend 函数

```
765     def iperf_extend( self, hosts=None, udpBw='10M', seconds=5, port=5001):
766         """Run iperf between two hosts using UDP.
767             hosts: list of hosts; if None, uses opposite hosts
768         """
769         if not hosts:
770             return
771         else:
772             assert len( hosts ) == 2
773             client, server = hosts
774             filename = client.name[1:] + '.out'
775             output( '*** Iperf: testing bandwidth between ' )
776             output( "%s and %s\n" % ( client.name, server.name ) )
777             iperfArgs = 'iperf -u '
778             bwArgs = '-b ' + udpBw + ' '
779             print "***start server***"
780             server.cmd( iperfArgs + '-s -i 5 ' + '&' )
781             print "***start client***"
782             client.cmd(
783                 iperfArgs + '-t ' + str(seconds) + ' -c ' + server.IP() + ' ' + bwArgs
784                 + '&')
```

Iperf_test 指定 Payless 论文的情形

```
787     def iperfTest(self):
788         base_port = 5001
789         host_list = []
790         host_list = [h for h in self.hosts]
791
792         client = host_list[0]
793         server = host_list[7]
794         period = 4
795         self.iperf_extend(hosts = [client, server], udpBw='10M', seconds=period, port=base_port)
796
797         sleep(10)
798         period = 5
799         self.iperf_extend(hosts = [client, server], udpBw='10M', seconds=period, port=base_port)
800
801         sleep(2)
802         client = host_list[1]
803         server = host_list[6]
804         period = 5
805         self.iperf_extend(hosts = [client, server], udpBw='20M', seconds=period, port=base_port)
806
807         sleep(2)
808         client = host_list[2]
809         server = host_list[5]
810         period = 11
811         self.iperf_extend(hosts = [client, server], udpBw='20M', seconds=period, port=base_port)
812
813         sleep(9)
814         client = host_list[1]
815         server = host_list[6]
816         period = 5
817         self.iperf_extend(hosts = [client, server], udpBw='20M', seconds=period, port=base_port)
818
819         sleep(7)
820         client = host_list[1]
821         server = host_list[6]
822         period = 3
823         self.iperf_extend(hosts = [client, server], udpBw='50M', seconds=period, port=base_port)
824
825         sleep(5)
826         self.iperf_extend(hosts = [client, server], udpBw='50M', seconds=period, port=base_port)
827
828         sleep(13)
829         period = 5
830         self.iperf_extend(hosts = [client, server], udpBw='50M', seconds=period, port=base_port)
831         print "test has done"
```

2. 修改 mininet/cli.py: cli 模块定义了 CLI 类，为 mininet 提供命令行接口，用于解析用户输入的命令，之前定义的自定义命令需要在 CLI 类中通过注册函数注册这条自定义命令。

```
228     def do_iperftest( self,line):
229         """Multi iperf UDP test between nodes"""
230         args = line.split()
231         if not args:
232             self.mn.iperfTest()
233
```

3. 修改 bin/mn: 在完成命令注册与绑定后，需要在 mininet 执行解释器中注册命令与对应执行函数的映射关系。

```
88 # optional tests to run
89 TESTS = [ 'cli', 'build', 'pingall', 'pingpair', 'iperf', 'all', 'iperfudp','iperftest',
90           'none' ]
91
92 ALTSPELLING = { 'pingall': 'pingAll',
93                 'pingpair': 'pingPair',
94                 'iperfudp': 'iperfUdp',
95                 'iperfUDP': 'iperfUdp',
96                 'iperftest': 'iperfTest' }
```

最后，进入 mininet/util 目录，重新编译安装 mininet

```
$~/mininet/util/install.sh -n
```

重启 mininet，sudo mn --switch ovs --controller ref --topo tree,depth=3,fanout=2 创建 Payless 中拓扑结构

输入 iperft，可用 tab 补全 iperftest，从而可使用 iperftest 进行 CBR 业务流的测试。

Wireshark 进行监测

