

Lab 3文档

Mininet文档

Mininet简介

Mininet (<http://mininet.org/>) 是一个Linux平台的网络仿真工具，使用Linux的cgroup和namespace的功能，结合虚拟网桥（如OpenVSwitch等）通过简单的Python命令建立简单的复杂的网络结构。

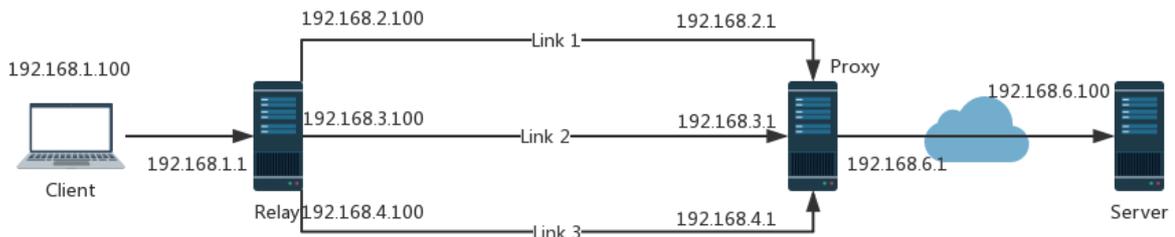
Mininet使用

本次Lab中，Mininet主要作为虚拟网络架构的仿真工具使用，在提供的m3系统镜像中，mininet目录中包含了 `m3_topo.py`，`m3_example.py` 和 `m3_with_tc.py` 三个文件。

Mininet网络链路模拟

参考 `m3_topo.py` 文件，关于mininet的API介绍，可以参考API官方文档 (<http://mininet.org/api/hierarchy.html>)，如果想要学习使用Mininet并修改网络结构可以参考此文档。

`m3_topo.py` 会模拟如下的网络结构（c1是client，h1是relay，h2是proxy，h2是server）：**（需要注意的是，Relay和Proxy的路由表只配置了与Link1，2，3以及自己子网相关的路由，也就是说，Client此时是无法正确的路由至server）**



Mininet交互

Mininet的交互方式有两种，一是在Python脚本中使用 `host.cmd()` 进行，具体的使用方法，可以参考 `m3_example.py` 文件。这个文件在搭建上述网络结构以外还会自动运行m3（**需要提前编译**），并在client和server间运行一次 `iperf` 测试，如果m3运行成功，则 `iperf` 可以正确运行。

第二种交互方式是在Python脚本中调用 `CLI(net)` 进入命令行交互式的方式，使用方式请参考 `help` 指令：

```
Documented commands (type help <topic>):
```

```
=====
```

EOF	gterm	iperfudp	nodes	pingpair	py	switch
dpctl	help	link	noecho	pingpairfull	quit	time
dump	intfs	links	pingall	ports	sh	x
exit	iperf	net	pingallfull	px	source	xterm

```
You may also send a command to a node using:
```

```
<node> command {args}
For example:
mininet> h1 ifconfig
```

The interpreter automatically substitutes IP addresses for node names when a node is the first arg, so commands like

```
mininet> h2 ping h3
should work.
```

Some character-oriented interactive commands require noecho:

```
mininet> noecho h2 vi foo.py
However, starting up an xterm/gterm is generally better:
mininet> xterm h2
```

Mininet脚本的使用

- `m3_topo.py` 脚本可以建立模拟网络结构并进入CLI模式，用户可以在网络节点上运行一些操作，如：

```
$ sudo python m3_topo.py
*** Creating network
...
*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
c1 -> X h1 X
c2 -> X X h2
h1 -> c1 X h2
h2 -> X c2 h1
*** Results: 50% dropped (6/12 received)
mininet> h1 route
Kernel IP routing table
Destination      Gateway          Genmask         Flags Metric Ref    Use Iface
192.168.1.0      0.0.0.0         255.255.255.0  U        0      0      0 h1-eth4
192.168.2.0      0.0.0.0         255.255.255.0  U        0      0      0 h1-eth1
192.168.3.0      0.0.0.0         255.255.255.0  U        0      0      0 h1-eth2
192.168.4.0      0.0.0.0         255.255.255.0  U        0      0      0 h1-eth3
```

- `m3_example.py` 脚本可以建立模拟网络结构并在Relay(h1)和Proxy(h2)上启动m3，并使用iperf进行网络速率测试，最后进入CLI模式，用户可以在m3运行情况下进行一些操作和测试，如：

```
$ sudo python m3_example.py
*** Creating network
...
-----
Client connecting to 192.168.6.100, TCP port 5001
TCP window size: 85.3 KByte (default)
-----
[ 3] local 192.168.1.100 port 57060 connected with 192.168.6.100 port 5001
[ ID] Interval      Transfer      Bandwidth
[ 3]  0.0-10.2 sec   271 MBytes    224 Mbits/sec

*** Routing Table on Router:
*** Starting CLI:
```

- `m3_with_tc.py` 脚本在 `m3_example.py` 基础上运行 `tc` 对链路情况进行模拟，使用 `tc/tc.csv` 文件中的 `thp` 行数据进行模拟，如有需要可以按需进行修改。

TC文档

TC简介

`tc`是Linux上的流量控制工具，拥有多种流量控制方法，也可以用于网络状况模拟。具体的使用方法可以参考Linux man page (<http://man7.org/linux/man-pages/man8/tc.8.html>)。

TC的使用

本Lab中，`tc`用于模拟高铁网络的链路环境，与`tc`相关的代码处于 `~/tc` 目录下，我们使用 `htb` 和 `netem` 分别控制链路的带宽上限和`rtt`、丢包率等参数。目前提供给大家的 `TcController` 程序会从输入的`csv`文件中读取 `thp1,thp2,thp3`三列，并分别用作为3条链路的带宽控制，`rtt`和`loss`有一固定值，此程序主要用于大家设计 Bandwidth-aware schedule时测试使用。

使用示例

```
$ sudo ./TcController tc.csv h1-eth1 h1-eth2 h1-eth3
# 第一个参数是输入的csv文件，后面三个参数分别对应需要控制的三条链路，有顺序要求
```

Surge-m3文档

Surge-m3 介绍

`Surge`是一款用来模拟用户HTTP请求的软件，可以根据给定的数据生成一定量的用户HTTP请求，用于测试 Scheduler的调度能力。

Surge-m3 使用

参考 `README.md`