

به نام خدا

گزارش پروژه اول شبکه های کامپیوتری

علی عدالت ۸۱۰۱۹۵۴۲۷

امیر رنجبر ۸۱۰۱۹۵۴۰۲

دکترخونساری

## Ifconfig

```
~/Desktop/Term 6/Network/CA/CA1/CN-CA1 -- mininet@mininet-vm: ~ -- ssh -X -Y -f mininet -p 2222 localhost
Last login: Mon Feb 25 14:07:55 on ttys002
amir ... | CA | CA1 | CN-CA1 ? master ? 1 ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
options=1203<RXCSUM,TXCSUM,TXSTATUS,SW_TIMESTAMP>
inet 127.0.0.1 netmask 0xff000000
inet6 ::1 prefixlen 128
inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
nd6 options=201<PERFORMNUD,DAD>
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=0<> mtu 1280
XHC20: flags=0<> mtu 0
XHC0: flags=0<> mtu 0
en2: flags=8963<UP,BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
options=60<TS04,TS06>
ether 16:00:c4:b9:b5:01
media: autoselect <full-duplex>
status: inactive
en1: flags=8963<UP,BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
options=60<TS04,TS06>
ether 16:00:c4:b9:b5:00
media: autoselect <full-duplex>
status: inactive
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
ether 8c:85:90:5e:ad:46
inet6 fe80::416:3b49:f470:b91a%en0 prefixlen 64 secured scopeid 0x8
inet 172.17.70.10 netmask 0xfffffe00 broadcast 172.17.71.255
nd6 options=201<PERFORMNUD,DAD>
media: autoselect
status: active
p2p0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 2304
ether 0e:85:90:5e:ad:46
media: autoselect
status: inactive
awdl0: flags=8943<UP,BROADCAST,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1404
ether 22:ea:58:e4:ff:9e
inet6 fe80::20ea:58ff:fee4:ff9e%awdl0 prefixlen 64 scopeid 0xa
nd6 options=201<PERFORMNUD,DAD>
media: autoselect
status: active
bridge0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
options=63<RXCSUM,TXCSUM,TS04,TS06>
ether 16:00:c4:b9:b5:00
Configuration:
```

قسمت ۱

جواب (۱) چگونه می توان یک دستور را داخل یک namespace مثلا h1 اجرا کرد؟

ip netns exec <network namespace> <command to run against that namespace>

جواب (۲) چرا هنگام اجرای دستور ip netns exec h1 ping 10.0.0.2 با پیغام Network is unreachable مواجه می شویم؟ برای رفع مشکل چه راهکاری پیشنهاد می کنید.

وقتی برای اولین بار lo loopback device در یک namespace بوجود می آید وضعیت آن down می باشد و به همین دلیل loopback ping با خطای بالا مواجه می شود. برای رفع این مشکل وضعیت این interface را به up تغییر می دهیم و برای این کار از دستور زیر استفاده می کنیم.

ip netns exec h1 ip link set dev lo up

جواب ۳) چگونه می توان به یک interface متعلق به یک host آدرس IP اختصاص داد؟ از دستور زیر برای این کار در host به نام h1 استفاده می شود.

```
ip netns exec h1 ifconfig interface-name ip-address
```

جواب ۴) virtual Ethernet یک interface مجازی است که می توان به یک namespace نسبت داده شود. virtual Ethernet ها با هم پیر می شوند و مانند لوله به هم وصل می شوند و برای اتصال namespace شبکه به جهان بیرون استفاده می شود.

جواب ۵) با چه دستوری می توان یک Virtual Ethernet ساخت؟ با دستور زیر یک جفت Virtual Ethernet ساخته می شود که به هم لینک می شوند.

```
ip link add s1-eth1 type veth peer name h1-eth0
```

جواب ۶) با چه دستوری می توان یک Virtual Ethernet ساخته شده را به یک namespace منتقل کرد.

```
ip link set veth1 netns h1
```

جواب ۷) اینترفیس ها و آدرس های آن هایی که به این namespace اضافه شده اند را نشان می دهد.

```

mininet@mininet-vm: ~
ali@ali-1015PX:~/workspace-UT/CNS$ ssh -X -Y -l mininet -p 2222 localhost
mininet@localhost's password:
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic i686)

 * Documentation:  https://help.ubuntu.com/
Last login: Mon Feb 18 04:10:18 2019
mininet@mininet-vm:~$ sudo mn --custom ~/mininet/custom/p3.py --topo mytopo*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1 s2
*** Adding links:
(20ms delay) (20ms delay) (h1, s1) (20ms delay) (20ms delay) (h2, s1) (15ms delay) (15ms delay) (h3, s2) (1s delay) (1s delay)
(h4, s2) (50ms delay) (50ms delay) (s2, s1)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 2 switches
s1 s2 ...(20ms delay) (20ms delay) (50ms delay) (15ms delay) (1s delay) (50ms delay)
*** Starting CLI:
mininet> nodes
available nodes are:
c0 h1 h2 h3 h4 s1 s2
mininet>

```

```

mininet@mininet-vm: ~
ali@ali-1015PX:~/workspace-UT/CNS$ ssh -X -Y -l mininet -p 2222 localhost
mininet@localhost's password:
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic i686)

 * Documentation:  https://help.ubuntu.com/
Last login: Mon Feb 18 04:10:18 2019
mininet@mininet-vm:~$ sudo mn --custom ~/mininet/custom/p3.py --topo mytopo*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1 s2
*** Adding links:
(20ms delay) (20ms delay) (h1, s1) (20ms delay) (20ms delay) (h2, s1) (15ms delay) (15ms delay) (h3, s2) (1s delay) (1s delay)
(h4, s2) (50ms delay) (50ms delay) (s2, s1)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 2 switches
s1 s2 ...(20ms delay) (20ms delay) (50ms delay) (15ms delay) (1s delay) (50ms delay)
*** Starting CLI:
mininet> nodes
available nodes are:
c0 h1 h2 h3 h4 s1 s2
mininet> net
h1 h1-eth0:s1-eth1
h2 h2-eth0:s1-eth2
h3 h3-eth0:s2-eth1
h4 h4-eth0:s2-eth2
s1 lo: s1-eth1:h1-eth0 s1-eth2:h2-eth0 s1-eth3:s2-eth3
s2 lo: s2-eth1:h3-eth0 s2-eth2:h4-eth0 s2-eth3:s1-eth3
c0
mininet>

```

```
Terminal
mininet@mininet-vm: ~
mininet> exit
*** Stopping 1 controllers
c0
*** Stopping 5 links
.....
*** Stopping 2 switches
s1 s2
*** Stopping 4 hosts
h1 h2 h3 h4
*** Done
completed in 148.716 seconds
mininet@mininet-vm:~$ sudo mn --custom ~/mininet/custom/p3.py --topo mytopo
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1 s2
*** Adding links:
(20ms delay) (20ms delay) (h1, s1) (20ms delay) (20ms delay) (h2, s1) (15ms delay) (15ms delay) (h3, s2) (1s delay) (1s delay)
(h4, s2) (50ms delay) (50ms delay) (s2, s1)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 2 switches
s1 s2 ...(20ms delay) (20ms delay) (50ms delay) (15ms delay) (1s delay) (50ms delay)
*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4
h2 -> h1 h3 h4
h3 -> h1 h2 h4
h4 -> h1 h2 h3
*** Results: 0% dropped (12/12 received)
mininet>
```

(2)

```
Firefox Web Browser
root@mininet-vm:~# ifconfig
h1-eth0  Link encap:Ethernet  HWaddr aa:ff:c2:29:73:35
          inet addr:10.0.0.1  Bcast:10.255.255.255  Mask:255.0.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:23 errors:0 dropped:0 overruns:0 frame:0
          TX packets:16 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1302 (1.3 KB)  TX bytes:1008 (1.0 KB)

lo       Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:626 errors:0 dropped:0 overruns:0 frame:0
          TX packets:626 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:104188 (104.1 KB)  TX bytes:104188 (104.1 KB)

root@mininet-vm:~#
mininet> sudo mn --custom ~/mininet/custom/p3.py --topo mytopo
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1 s2
*** Adding links:
(h2, s1) (15ms delay) (15ms delay) (h3, s2) (1s delay) (1s delay)
(h4, s2) (50ms delay) (50ms delay) (s2, s1)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 2 switches
s1 s2 ...(20ms delay) (20ms delay) (50ms delay) (15ms delay) (1s delay) (50ms delay)
*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4
h2 -> h1 h3 h4
h3 -> h1 h2 h4
h4 -> h1 h2 h3
*** Results: 0% dropped (12/12 received)
mininet> xterm h1
mininet>
```

```

Firefox
root@mininet-vm:~# ifconfig
h3-eth0  Link encap:Ethernet  HWaddr 52:d0:8a:95:d3:35
          inet addr:10.0.0.3  Bcast:10.255.255.255  Mask:255.0.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:23 errors:0 dropped:0 overruns:0 frame:0
          TX packets:16 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1302 (1.3 KB)  TX bytes:1008 (1.0 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:666 errors:0 dropped:0 overruns:0 frame:0
          TX packets:666 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:107420 (107.4 KB)  TX bytes:107420 (107.4 KB)

root@mininet-vm:~#

(h2, s1) (15ms delay) (15ms delay) (h3, s2) (1s delay) (1s delay)

*** Starting controller
c0
*** Starting 2 switches
s1 s2 ... (20ms delay) (20ms delay) (50ms delay) (15ms delay) (1s delay) (50ms delay)
*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4
h2 -> h1 h3 h4
h3 -> h1 h2 h4
h4 -> h1 h2 h3
*** Results: 0% dropped (12/12 received)
mininet> xterm h1
mininet> xterm h3
mininet>

```

## قسمت ۴

### Delay

با افزایش دیلی لینک ها از ۲۰ میلی ثانیه به ۹۰ میلی ثانیه مدت زمان لازم برای رفتن پیام ICMP و برگشت آن افزایش می یابد که این موضوع در دو عکس زیر قابل مشاهده است. در اثر افزایش این مدت زمان قبل از رسیدن اولین پکت reply دو پکت درخواست فرستاده می شود.

```

Capturing from h1-eth0 [Wireshark 1.10.6 (v1.10.6 from master-1.10)]
[1] 3011
root@mininet-vm:~#
(process:3011): Gtk-WARNING **: Locale not supported by C library.
Using the fallback 'C' locale.

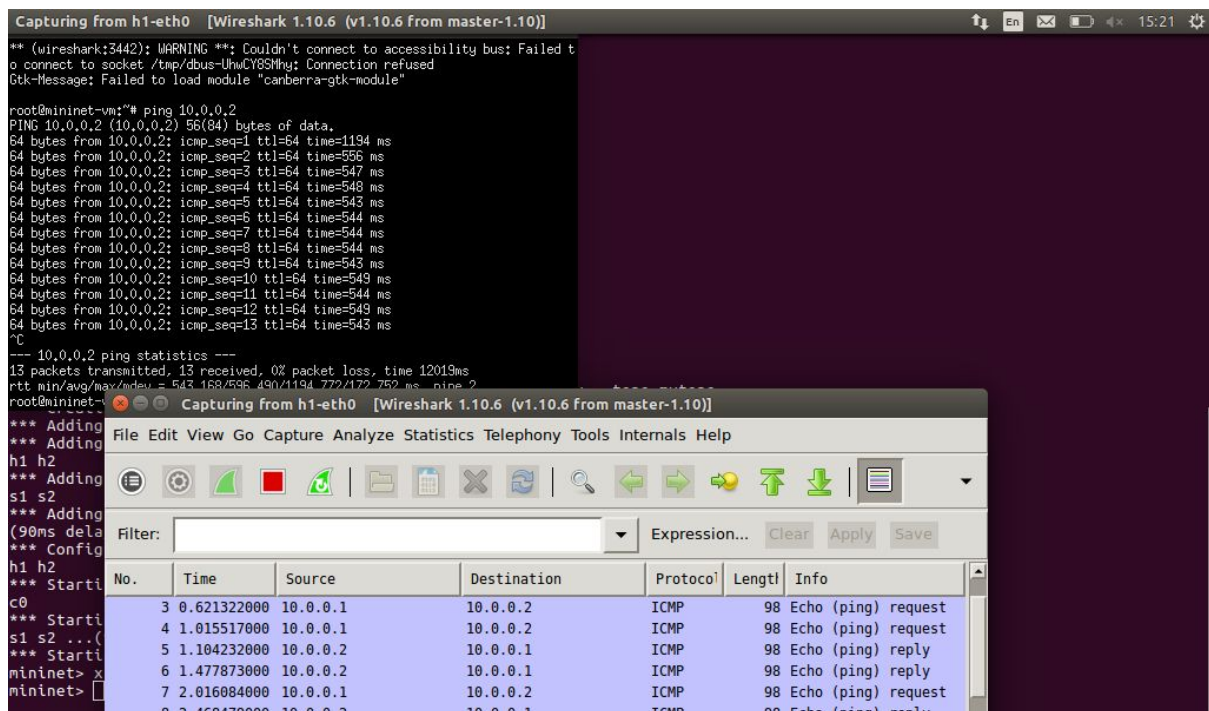
** (wireshark:3011): WARNING **: Couldn't connect to accessibility bus: Failed to
connect to socket /tmp/dbus-UhmCY8Mhy: Connection refused
Gtk-Message: Failed to load module "canberra-gtk-module"

root@mininet-vm:~# ping 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=286 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=123 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=122 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=122 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=122 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=122 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=124 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=123 ms
^C
--- 10.0.0.2 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7013ms
rtt min/avg/max/mdev = 122.703/144.459/286.732/53.817 ms
root@mininet-vm:~#

(h2, s2) (20ms delay) (20ms delay) (s2, s1)

```

No.	Time	Source	Destination	Protocol	Length	Info
3	0.145091000	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request
4	0.266112000	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply
5	1.006736000	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request
6	1.110911000	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply
7	2.003766000	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request

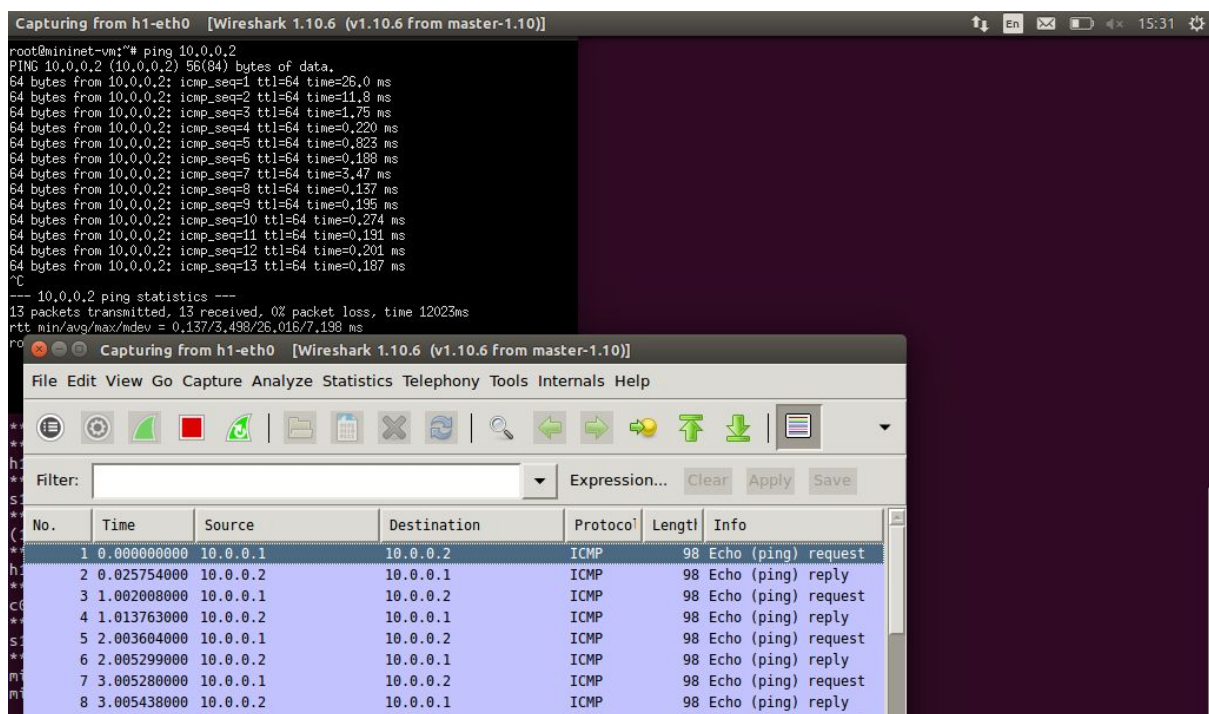
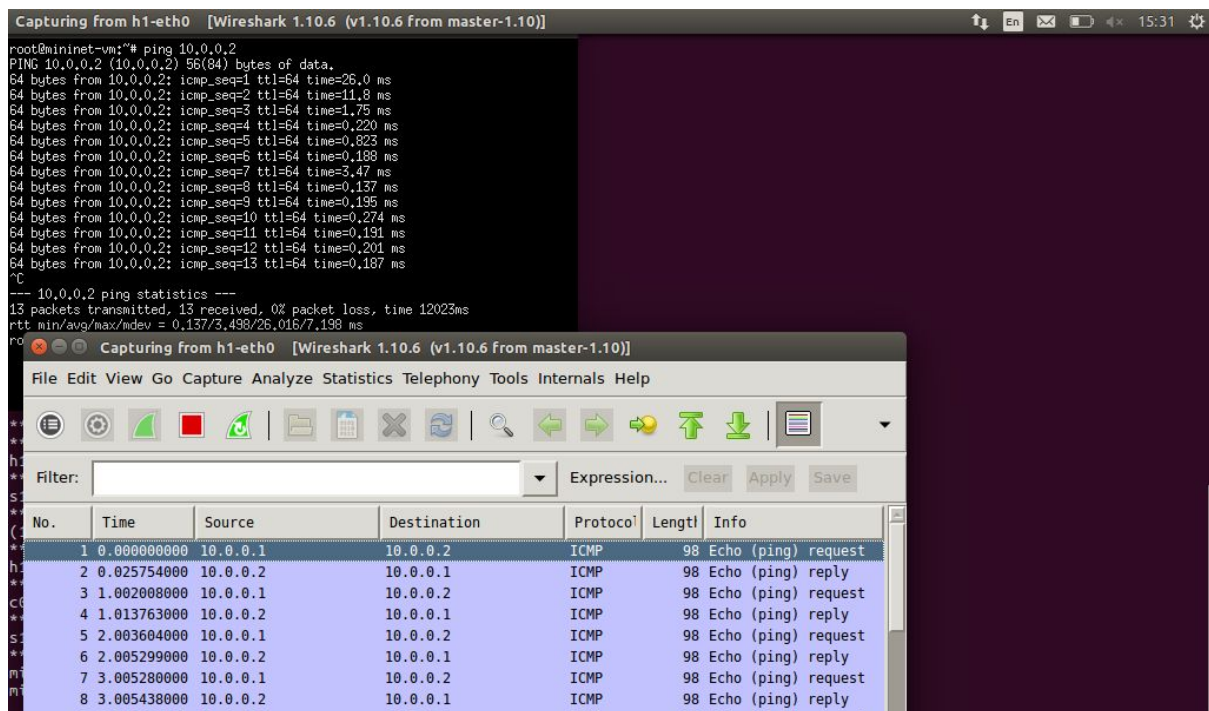


عکس اول مربوط به دیلی ۲۰ میلی ثانیه و عکس دوم مربوط به دیلی ۹۰ میلی ثانیه است.

## Bandwidth

با افزایش bandwidth و پهنای باند می توان تعداد بیت بیشتری را در ثانیه انتقال داد اما در ping چون حجم پکت ارسالی بسیار کم است و تعداد محدودی پکت ارسال می شود پس ping به پهنای باید زیادی احتیاج ندارد و با افزایش پهنای باند تغییری در مدت زمان میانگین ارسال و برگشت پکت های ping ایجاد نمی شود. در دو عکس زیر این موضوع قابل مشاهده است. تفاوت اندک زمان ها به دلیل تعداد متفاوت پکت های capture شده است. تعداد درخواست قبل از اولین پاسخ با افزایش پهنای باند تغییر نمی کند.





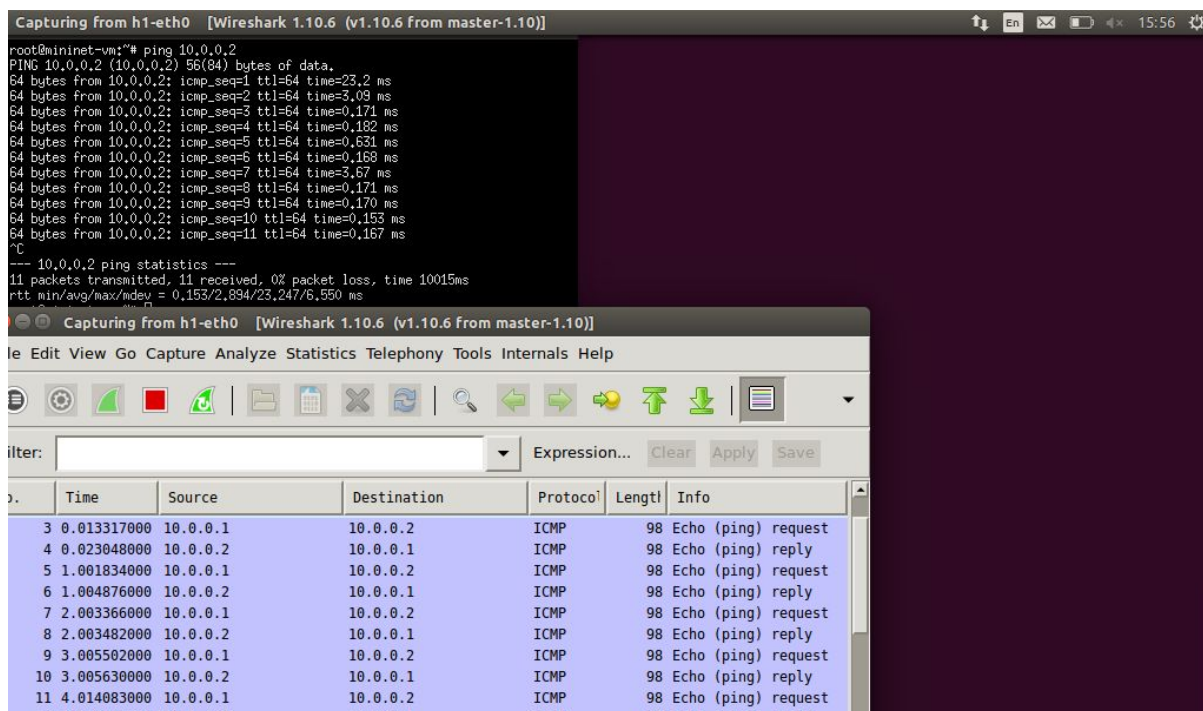
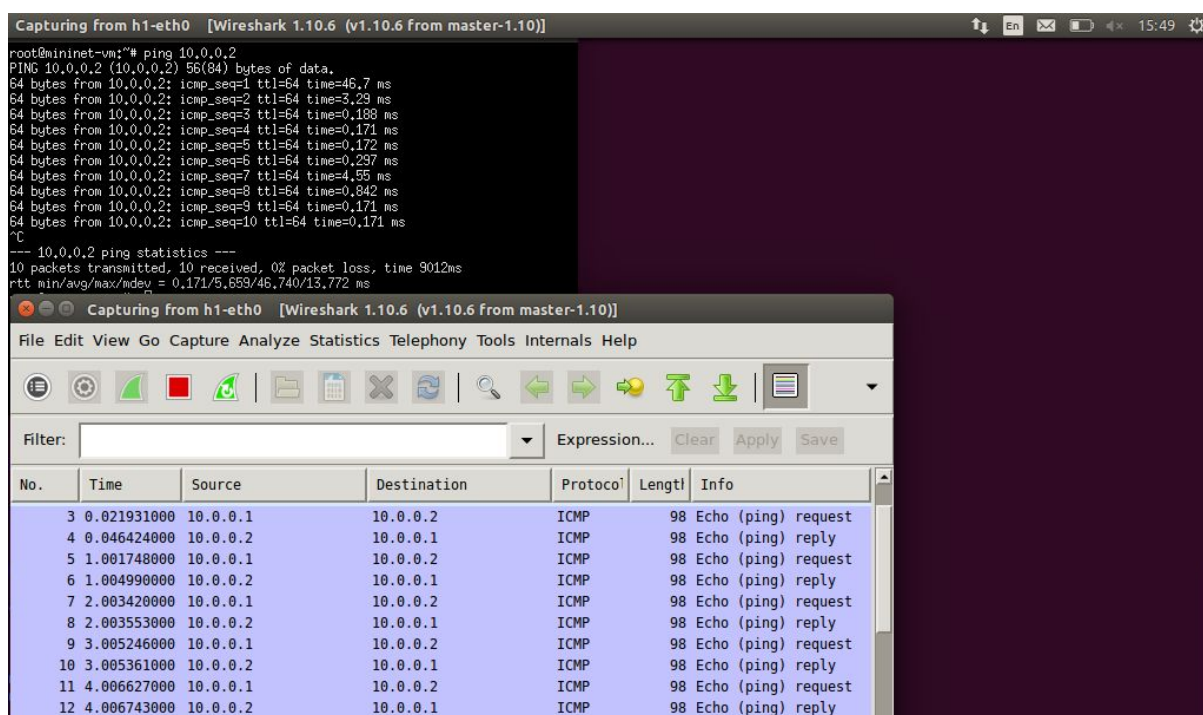
عکس اول مربوط به پهنای باند ۱ مگابیت بر ثانیه و عکس دوم مربوط به پهنای باند ۱۵ مگابیت بر ثانیه است.

## Switch Queue

با افزایش سایز بافر SwitchQueue داریم که پکت های بیشتری می توانند بافر شوند و کنترل شلوغی و بار بر روی یک سوییچ بهتر می تواند انجام شود که این موضوع در دیلی میانگین ارسال



و برگشت ping می تواند تاثیر گزار باشد. هر چه SwitchQueue کمتر باشد شلوغی سوییچ بیشتر می شود و احتمال از دست رفتن پکت نیز افزایش می یابد و ممکن است تعداد پکت ارسالی قبل از اولین پاسخ افزایش یابد. با افزایش SwitchQueue شلوغی کمتر می شود و دیلی میانگین نیز می تواند کاهش یابد. دو عکس زیر به ترتیب برای SwitchQueue برابر ۱ و ۱۵ هستند.



تعداد سوییچ ها

با افزایش تعداد سوییچ های سری متصل به هم بین host ها مسیری که پکت ها باید طی کنند طولانی تر می شود و اگر برای هر لینک یک دیلی در نظر بگیریم پس میانگین زمان ارسال و پاسخ در ping با افزایش تعداد سوییچ ها زیاد می شود. با توجه به این که از پروتکل انتقالی مثل TCP که reliable است استفاده نمی شود با افزایش مسیر ارسال و برگشت احتمال این که پکت دچار مشکل شود بالاتر می رود. که بدلیل افزایش این احتمال ممکن است چند پکت در خواست ارسال شود سپس اولین پکت پاسخ برگردد. وضعیت در حالتی که ۲ و ۷ سوییچ سری داشته باشیم به ترتیب در دو عکس زیر آمده است.

The image shows a terminal window and a Wireshark packet capture window. The terminal window displays the output of a ping command from 10.0.0.1 to 10.0.0.2. The output shows 8 packets transmitted, 8 received, 0% packet loss, and a time of 7018ms. The Wireshark window shows the capture of these packets, with the packet list table below.

No.	Time	Source	Destination	Protocol	Length	Info
3	0.024598000	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request
4	0.039848000	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply
5	1.001694000	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request
6	1.007559000	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply
7	2.003331000	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request

Capturing from h1-eth0 [Wireshark 1.10.6 (v1.10.6 from master-1.10)]

```
root@mininet-vm:~# ping 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data:
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=134 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=6.67 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.448 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.359 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.647 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=0.510 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=4.98 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=0.216 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=0.339 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=0.217 ms
64 bytes from 10.0.0.2: icmp_seq=11 ttl=64 time=0.717 ms
64 bytes from 10.0.0.2: icmp_seq=12 ttl=64 time=0.193 ms
64 bytes from 10.0.0.2: icmp_seq=13 ttl=64 time=0.224 ms
^C
--- 10.0.0.2 ping statistics ---
13 packets transmitted, 13 received, 0% packet loss, time 12012ms
rtt min/avg/max/mdev = 0.193/11.503/134.009/35.420 ms
```

Capturing from h1-eth0 [Wireshark 1.10.6 (v1.10.6 from master-1.10)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter:  Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
3	0.078612000	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request
4	0.133740000	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply
5	1.001182000	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request
6	1.007803000	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply
7	2.002590000	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request
8	2.002987000	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply
9	3.004061000	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request
10	3.004377000	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply