Antonio Valenzuela
anlvalen@ucsc.edu
January 29, 2018

Pre-lab Questions:
All answers are at least sourced from the linux API

1. [5] What command will show you which groups you are a member of?
    $ groups

2. [5] What does the environmental variable "$?" hold? (Hint: the command 'echo $?' will should you this on your screen)
    $? is the return code of the last command executed.

3. [5] What key combination will suspend a currently running process and place it as a background process?
    The following will suspend the currently running process where pid is replaced by the actual PID of the process
    $ kill -STOP pid
    source:https://www.ostechnix.com/suspend-process-resume-later-linux/

4. [5] With what command (and arguments) can you find out your kernel version and the "nodename"? [The output should not include any other information]
    "uname" is the kernel version number and "n" is nodename
    $ uname -n

    source:https://www.liquidweb.com/kb/how-to-check-the-kernel-version-in-linux-ubuntu-centos/

5. [5] What is the difference between the paths ".", "..", and "~"? What does the path "/" refer to when not preceded by anything?
    "." - current directory
    ".." - parent directory
    "~" - home directory
    "/" - absolute path

    http://linuxcommand.org/lc3_lts0020.php

6. [5] What is a pid? Which command would you use to find the “pid” for a running process?

The Process Identification Number (PID) is an identification number for processes created in Unix
To obtain PID of running process.
$ pidof [options] pattern

7. [20] Write a single command that will return every user’s default shell. [You may chain commands using piping and redirects] (Hint: See ‘Chapter 19: filters’ of linux-training.be as well as the man page for the /etc/passwd file: https://linux.die.net/man/5/passwd)

8. [10] What is the difference between “sudo” and “su root”?

The root in Unix is the SuperUser or “su”. Certain commands in a terminal require root privileges. It is not advisable that as commands be run being the actual root. Instead, such programs or commands are done using sudo. sudo allows the user to act as root without using root login. Generally, this is seen as more secure.
source: https://help.ubuntu.com/community/RootSudo

9. [10] How would you tell your computer to run a program or script on a schedule or set interval on Linux? E.g. Run this program once every 30 minutes.

Crontab files allow a user to schedule tasks on specific times whether they are reoccuring or not.

source: http://kvz.io/blog/2007/07/29/schedule-tasks-on-linux-using-crontab/

10. [30] Write a shell script that only prints the even numbered lines of each file in the current directory. The output should be filename: line for each even numbered line. You do not need to print line numbers.

sources: https://www.tutorialspoint.com/unix_commands/awk.htm, https://www.computerhope.com/unix/uawk.htm,

The Lab [100 pts]:

2. [30 pts] Save a screenshot of dump and pingall output. Explain what is being shown in the screenshot.

Outputs of the dump and pinout commands are shown in the figures below. The command “pingall” sends a ping from each host to all other hosts in the network to test their related links. Dump displays information regarding the nodes of a network such as assigned addresses.

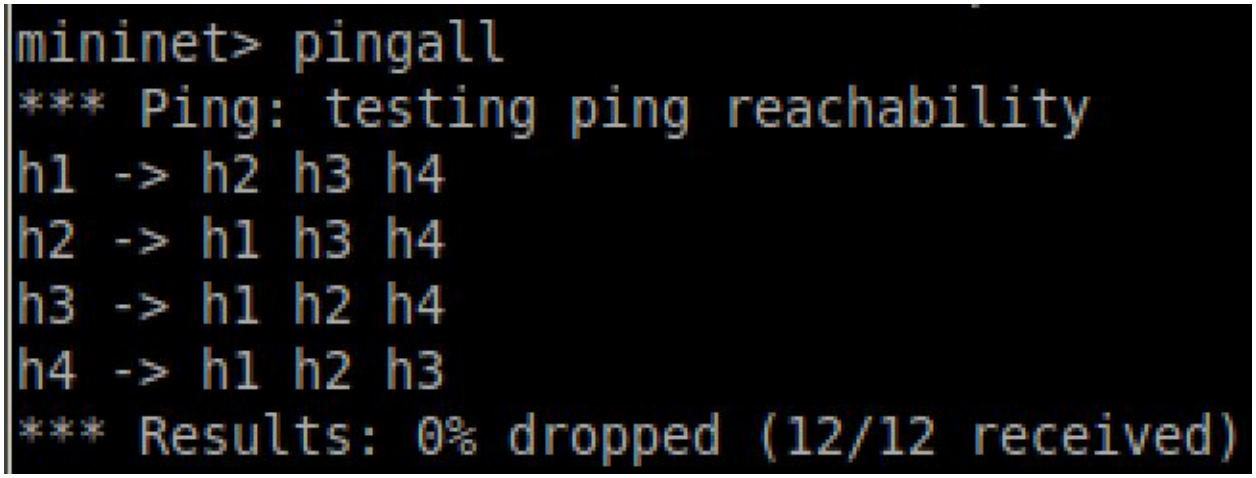


Figure: pingout Output

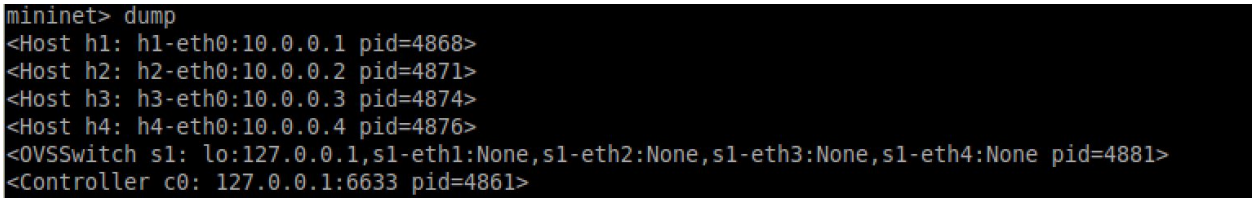


Figure: dump Output

3. [10 pts] Run the iperf command as well, and screenshot the output, how fast is the connect?

The iperf connect shows a speed of 30.7 Gbits per second

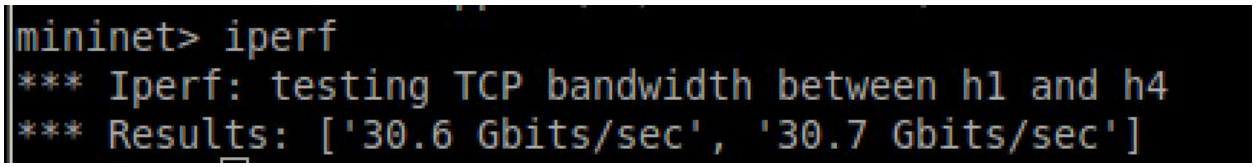
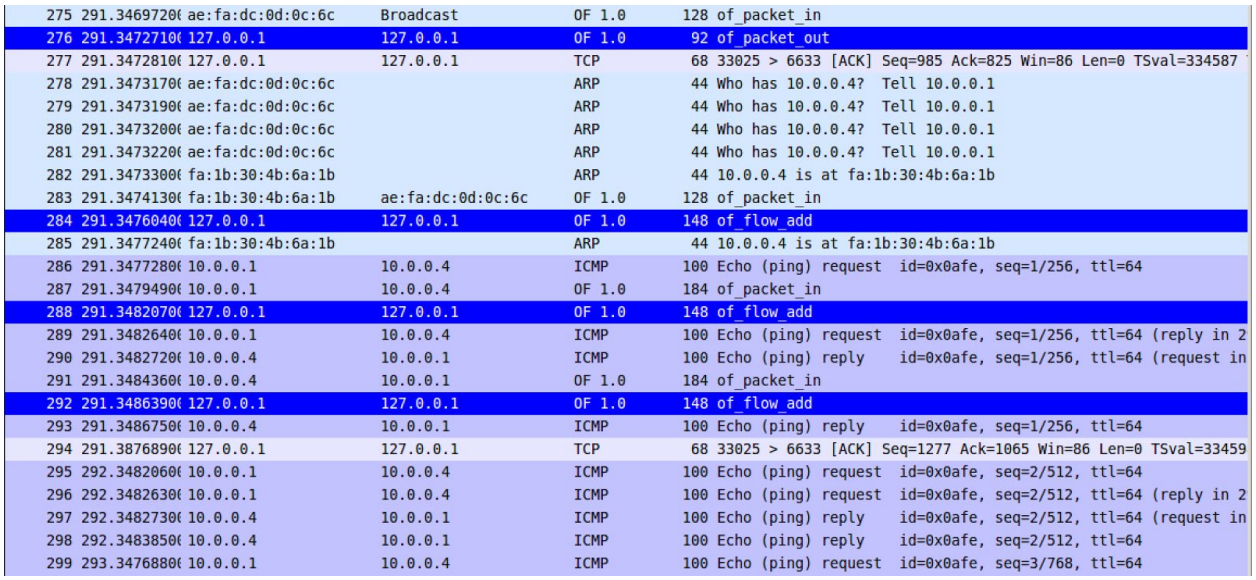


Figure: iperf Output

4. Run wireshark, and using the display filter, filter for “of”. Note: When you run wireshark you should do so as “sudo wireshark”. When you choose an interface to capture on, you should select “any”.

a. [20 pts] Run ping from a host to any other host using hX ping -c 5 hY. How many of_packet_in messages show up? Take a screenshot of your results.

6 instances of ‘of_packet_in’ appear. These entries are numbered as [No:] 275, 283, 287, 291, 315, 320.



275	291.3469720(	ae:fa:dc:0d:0c:6c	Broadcast	OF 1.0	128 of_packet_in
276	291.3472710(	127.0.0.1	127.0.0.1	OF 1.0	92 of_packet_out
277	291.3472810(	127.0.0.1	127.0.0.1	TCP	68 33025 > 6633 [ACK] Seq=985 Ack=825 Win=86 Len=0 TSval=334587
278	291.3473170(	ae:fa:dc:0d:0c:6c		ARP	44 Who has 10.0.0.4? Tell 10.0.0.1
279	291.3473190(	ae:fa:dc:0d:0c:6c		ARP	44 Who has 10.0.0.4? Tell 10.0.0.1
280	291.3473200(	ae:fa:dc:0d:0c:6c		ARP	44 Who has 10.0.0.4? Tell 10.0.0.1
281	291.3473220(	ae:fa:dc:0d:0c:6c		ARP	44 Who has 10.0.0.4? Tell 10.0.0.1
282	291.3473300(	fa:1b:30:4b:6a:1b		ARP	44 10.0.0.4 is at fa:1b:30:4b:6a:1b
283	291.3474130(	fa:1b:30:4b:6a:1b	ae:fa:dc:0d:0c:6c	OF 1.0	128 of_packet_in
284	291.3476040(	127.0.0.1	127.0.0.1	OF 1.0	148 of_flow_add
285	291.3477240(	fa:1b:30:4b:6a:1b		ARP	44 10.0.0.4 is at fa:1b:30:4b:6a:1b
286	291.3477280(	10.0.0.1	10.0.0.4	ICMP	100 Echo (ping) request id=0x0afe, seq=1/256, ttl=64
287	291.3479490(	10.0.0.1	10.0.0.4	OF 1.0	184 of_packet_in
288	291.3482070(	127.0.0.1	127.0.0.1	OF 1.0	148 of_flow_add
289	291.3482640(	10.0.0.1	10.0.0.4	ICMP	100 Echo (ping) request id=0x0afe, seq=1/256, ttl=64 (reply in 2
290	291.3482720(	10.0.0.4	10.0.0.1	ICMP	100 Echo (ping) reply id=0x0afe, seq=1/256, ttl=64 (request in
291	291.3484360(	10.0.0.4	10.0.0.1	OF 1.0	184 of_packet_in
292	291.3486390(	127.0.0.1	127.0.0.1	OF 1.0	148 of_flow_add
293	291.3486750(	10.0.0.4	10.0.0.1	ICMP	100 Echo (ping) reply id=0x0afe, seq=1/256, ttl=64
294	291.3876890(	127.0.0.1	127.0.0.1	TCP	68 33025 > 6633 [ACK] Seq=1277 Ack=1065 Win=86 Len=0 TSval=33459
295	292.3482060(	10.0.0.1	10.0.0.4	ICMP	100 Echo (ping) request id=0x0afe, seq=2/512, ttl=64
296	292.3482630(	10.0.0.1	10.0.0.4	ICMP	100 Echo (ping) request id=0x0afe, seq=2/512, ttl=64 (reply in 2
297	292.3482730(	10.0.0.4	10.0.0.1	ICMP	100 Echo (ping) reply id=0x0afe, seq=2/512, ttl=64 (request in
298	292.3483850(	10.0.0.4	10.0.0.1	ICMP	100 Echo (ping) reply id=0x0afe, seq=2/512, ttl=64
299	293.3476880(	10.0.0.1	10.0.0.4	ICMP	100 Echo (ping) request id=0x0afe, seq=3/768, ttl=64

295	292.3482060(	10.0.0.1	10.0.0.4	ICMP	100 Echo (ping) request id=0x0afe, seq=2/512, ttl=64
296	292.3482630(	10.0.0.1	10.0.0.4	ICMP	100 Echo (ping) request id=0x0afe, seq=2/512, ttl=64 (reply in 2
297	292.3482730(	10.0.0.4	10.0.0.1	ICMP	100 Echo (ping) reply id=0x0afe, seq=2/512, ttl=64 (request in
298	292.3483850(	10.0.0.4	10.0.0.1	ICMP	100 Echo (ping) reply id=0x0afe, seq=2/512, ttl=64
299	293.3476880(	10.0.0.1	10.0.0.4	ICMP	100 Echo (ping) request id=0x0afe, seq=3/768, ttl=64
300	293.3476970(	10.0.0.1	10.0.0.4	ICMP	100 Echo (ping) request id=0x0afe, seq=3/768, ttl=64 (reply in 3
301	293.3477060(	10.0.0.4	10.0.0.1	ICMP	100 Echo (ping) reply id=0x0afe, seq=3/768, ttl=64 (request in
302	293.3477080(	10.0.0.4	10.0.0.1	ICMP	100 Echo (ping) reply id=0x0afe, seq=3/768, ttl=64
303	294.3476770(	10.0.0.1	10.0.0.4	ICMP	100 Echo (ping) request id=0x0afe, seq=4/1024, ttl=64
304	294.3476850(	10.0.0.1	10.0.0.4	ICMP	100 Echo (ping) request id=0x0afe, seq=4/1024, ttl=64 (reply in
305	294.3476940(	10.0.0.4	10.0.0.1	ICMP	100 Echo (ping) reply id=0x0afe, seq=4/1024, ttl=64 (request i
306	294.3476960(	10.0.0.4	10.0.0.1	ICMP	100 Echo (ping) reply id=0x0afe, seq=4/1024, ttl=64
307	295.3476850(	10.0.0.1	10.0.0.4	ICMP	100 Echo (ping) request id=0x0afe, seq=5/1280, ttl=64
308	295.3476930(	10.0.0.1	10.0.0.4	ICMP	100 Echo (ping) request id=0x0afe, seq=5/1280, ttl=64 (reply in
309	295.3477020(	10.0.0.4	10.0.0.1	ICMP	100 Echo (ping) reply id=0x0afe, seq=5/1280, ttl=64 (request i
310	295.3477030(	10.0.0.4	10.0.0.1	ICMP	100 Echo (ping) reply id=0x0afe, seq=5/1280, ttl=64
311	296.0006770(	127.0.0.1	127.0.0.1	OF 1.0	76 of_echo_request
312	296.0009650(	127.0.0.1	127.0.0.1	OF 1.0	76 of_echo_reply
313	296.0009760(	127.0.0.1	127.0.0.1	TCP	68 33025 > 6633 [ACK] Seq=1285 Ack=1073 Win=86 Len=0 TSval=33575
314	296.3556660(	fa:1b:30:4b:6a:1b		ARP	44 Who has 10.0.0.1? Tell 10.0.0.4
315	296.3578450(	fa:1b:30:4b:6a:1b	ae:fa:dc:0d:0c:6c	OF 1.0	128 of_packet_in
316	296.3586860(	127.0.0.1	127.0.0.1	OF 1.0	148 of_flow_add
317	296.3587400(	127.0.0.1	127.0.0.1	TCP	68 33025 > 6633 [ACK] Seq=1345 Ack=1153 Win=86 Len=0 TSval=33584
318	296.3591730(	fa:1b:30:4b:6a:1b		ARP	44 Who has 10.0.0.1? Tell 10.0.0.4
319	296.3591900(	ae:fa:dc:0d:0c:6c		ARP	44 10.0.0.1 is at ae:fa:dc:0d:0c:6c
320	296.3599710(	ae:fa:dc:0d:0c:6c	fa:1b:30:4b:6a:1b	OF 1.0	128 of_packet_in

figure: output for ‘h1 ping -c h4’ (used for both a and b)

b. [20 pts] What is the source and destination IP addresses for these entries? Find another packet that matches the “of” filter with the OpenFlow typefield set to

OFPT_PACKET_OUT. What is the source and destination IP address for this entry? Take screenshots showing your results.

Source and destination for the numbered entries are as follows:

No. 275: ae:fa:dc:0d:oc:6c -> broadcast

No. 283: fa:1b:20:4b:6a:1b -> ae:fa:dc:0d:oc:6c

No: 287: 10.0.0.1 -> 10.0.0.4

No. 291: 10.0.0.4 -> 10.0.0.1

No. 315: fa:1b:30:4b:6a:1b -> ae:fa:dc:0d:0c:6c

No. 320: ae:fa:dc:0d:0c:6c -> fa:1b:30:4b:6a:1b

Packet that matches the “of” filter with the typefield set to OFT_PACKET_OUT is entry No. 276. Its source and destination IP are 127.0.0.1 -> 120.0.0.1

figure: event of packet with typefield set to OFT_PACKET_OUT

c. [20 pts] Replace the display filter for “of” to “icmp && not of”. Run pingall again, how many entries are generated in wireshark? What types of icmp entries show up? Take a screenshot of your results

51 entries are displayed when pingall is run on the display filter “icmp && not of”. These entries are of type “request” or “reply”

```
    22 14.20991000( 10.0.0.1          10.0.0.2          ICMP      100 Echo (ping) request  id=0x0c81, seq=1/256, ttl=64
    25 14.21029200( 10.0.0.1          10.0.0.2          ICMP      100 Echo (ping) request  id=0x0c81, seq=1/256, ttl=64 (reply in 26)
    26 14.21030000( 10.0.0.2          10.0.0.1          ICMP      100 Echo (ping) reply    id=0x0c81, seq=1/256, ttl=64 (request in 25)
    29 14.21064600( 10.0.0.2          10.0.0.1          ICMP      100 Echo (ping) reply    id=0x0c81, seq=1/256, ttl=64
    41 14.21281500( 10.0.0.1          10.0.0.3          ICMP      100 Echo (ping) request  id=0x0c83, seq=1/256, ttl=64
    44 14.21311900( 10.0.0.1          10.0.0.3          ICMP      100 Echo (ping) request  id=0x0c83, seq=1/256, ttl=64 (reply in 45)
    45 14.21312500( 10.0.0.3          10.0.0.1          ICMP      100 Echo (ping) reply    id=0x0c83, seq=1/256, ttl=64 (request in 44)
    48 14.21340800( 10.0.0.3          10.0.0.1          ICMP      100 Echo (ping) reply    id=0x0c83, seq=1/256, ttl=64
    49 14.21433500( 10.0.0.1          10.0.0.4          ICMP      100 Echo (ping) request  id=0x0c84, seq=1/256, ttl=64
    52 14.21458900( 10.0.0.1          10.0.0.4          ICMP      100 Echo (ping) request  id=0x0c84, seq=1/256, ttl=64
    53 14.21459000( 10.0.0.1          10.0.0.4          ICMP      100 Echo (ping) request  id=0x0c84, seq=1/256, ttl=64
    54 14.21459200( 10.0.0.1          10.0.0.4          ICMP      100 Echo (ping) request  id=0x0c84, seq=1/256, ttl=64
    55 14.21459300( 10.0.0.1          10.0.0.4          ICMP      100 Echo (ping) request  id=0x0c84, seq=1/256, ttl=64 (reply in 56)
    56 14.21459900( 10.0.0.4          10.0.0.1          ICMP      100 Echo (ping) reply    id=0x0c84, seq=1/256, ttl=64 (request in 55)
    59 14.21495500( 10.0.0.4          10.0.0.1          ICMP      100 Echo (ping) reply    id=0x0c84, seq=1/256, ttl=64
    60 14.21663100( 10.0.0.2          10.0.0.1          ICMP      100 Echo (ping) request  id=0x0c85, seq=1/256, ttl=64
    63 14.21693700( 10.0.0.2          10.0.0.1          ICMP      100 Echo (ping) request  id=0x0c85, seq=1/256, ttl=64 (reply in 64)
    64 14.21694400( 10.0.0.1          10.0.0.2          ICMP      100 Echo (ping) reply    id=0x0c85, seq=1/256, ttl=64 (request in 63)
    67 14.21723800( 10.0.0.1          10.0.0.2          ICMP      100 Echo (ping) reply    id=0x0c85, seq=1/256, ttl=64
    79 14.22064900( 10.0.0.2          10.0.0.3          ICMP      100 Echo (ping) request  id=0x0c86, seq=1/256, ttl=64
    82 14.22095600( 10.0.0.2          10.0.0.3          ICMP      100 Echo (ping) request  id=0x0c86, seq=1/256, ttl=64 (reply in 83)
    83 14.22096300( 10.0.0.3          10.0.0.2          ICMP      100 Echo (ping) reply    id=0x0c86, seq=1/256, ttl=64 (request in 82)
    86 14.22125500( 10.0.0.3          10.0.0.2          ICMP      100 Echo (ping) reply    id=0x0c86, seq=1/256, ttl=64
    87 14.22219000( 10.0.0.2          10.0.0.4          ICMP      100 Echo (ping) request  id=0x0c87, seq=1/256, ttl=64
```

```
    87 14.22219000( 10.0.0.2          10.0.0.4          ICMP      100 Echo (ping) request  id=0x0c87, seq=1/256, ttl=64
    90 14.22251300( 10.0.0.2          10.0.0.4          ICMP      100 Echo (ping) request  id=0x0c87, seq=1/256, ttl=64 (reply in 91)
    91 14.22252100( 10.0.0.4          10.0.0.2          ICMP      100 Echo (ping) reply    id=0x0c87, seq=1/256, ttl=64 (request in 90)
    94 14.22281500( 10.0.0.4          10.0.0.2          ICMP      100 Echo (ping) reply    id=0x0c87, seq=1/256, ttl=64
    95 14.22418500( 10.0.0.3          10.0.0.1          ICMP      100 Echo (ping) request  id=0x0c88, seq=1/256, ttl=64
    98 14.22462600( 10.0.0.3          10.0.0.1          ICMP      100 Echo (ping) request  id=0x0c88, seq=1/256, ttl=64 (reply in 99)
    99 14.22463300( 10.0.0.1          10.0.0.3          ICMP      100 Echo (ping) reply    id=0x0c88, seq=1/256, ttl=64 (request in 98)
   102 14.22497700( 10.0.0.1          10.0.0.3          ICMP      100 Echo (ping) reply    id=0x0c88, seq=1/256, ttl=64
   103 14.22669000( 10.0.0.3          10.0.0.2          ICMP      100 Echo (ping) request  id=0x0c89, seq=1/256, ttl=64
   106 14.22711700( 10.0.0.3          10.0.0.2          ICMP      100 Echo (ping) request  id=0x0c89, seq=1/256, ttl=64 (reply in 107)
   107 14.22712400( 10.0.0.2          10.0.0.3          ICMP      100 Echo (ping) reply    id=0x0c89, seq=1/256, ttl=64 (request in 106)
   110 14.22777500( 10.0.0.2          10.0.0.3          ICMP      100 Echo (ping) reply    id=0x0c89, seq=1/256, ttl=64
   122 14.22927200( 10.0.0.3          10.0.0.4          ICMP      100 Echo (ping) request  id=0x0c8a, seq=1/256, ttl=64
   125 14.22958000( 10.0.0.3          10.0.0.4          ICMP      100 Echo (ping) request  id=0x0c8a, seq=1/256, ttl=64 (reply in 126)
   126 14.22958600( 10.0.0.4          10.0.0.3          ICMP      100 Echo (ping) reply    id=0x0c8a, seq=1/256, ttl=64 (request in 125)
   129 14.22987300( 10.0.0.4          10.0.0.3          ICMP      100 Echo (ping) reply    id=0x0c8a, seq=1/256, ttl=64
   130 14.23090000( 10.0.0.4          10.0.0.1          ICMP      100 Echo (ping) request  id=0x0c8b, seq=1/256, ttl=64
   133 14.23115700( 10.0.0.4          10.0.0.1          ICMP      100 Echo (ping) request  id=0x0c8b, seq=1/256, ttl=64 (reply in 134)
   134 14.23116400( 10.0.0.1          10.0.0.4          ICMP      100 Echo (ping) reply    id=0x0c8b, seq=1/256, ttl=64 (request in 133)
   137 14.23152400( 10.0.0.1          10.0.0.4          ICMP      100 Echo (ping) reply    id=0x0c8b, seq=1/256, ttl=64
   138 14.23264400( 10.0.0.4          10.0.0.2          ICMP      100 Echo (ping) request  id=0x0c8c, seq=1/256, ttl=64
   141 14.23309300( 10.0.0.4          10.0.0.2          ICMP      100 Echo (ping) request  id=0x0c8c, seq=1/256, ttl=64 (reply in 142)
   142 14.23310000( 10.0.0.2          10.0.0.4          ICMP      100 Echo (ping) reply    id=0x0c8c, seq=1/256, ttl=64 (request in 141)
   145 14.23346800( 10.0.0.2          10.0.0.4          ICMP      100 Echo (ping) reply    id=0x0c8c, seq=1/256, ttl=64
```

```
   145 14.23346800( 10.0.0.2          10.0.0.4          ICMP      100 Echo (ping) reply    id=0x0c8c, seq=1/256, ttl=64
   146 14.23618200( 10.0.0.4          10.0.0.3          ICMP      100 Echo (ping) request  id=0x0c8d, seq=1/256, ttl=64
   149 14.23664700( 10.0.0.4          10.0.0.3          ICMP      100 Echo (ping) request  id=0x0c8d, seq=1/256, ttl=64 (reply in 150)
   150 14.23665500( 10.0.0.3          10.0.0.4          ICMP      100 Echo (ping) reply    id=0x0c8d, seq=1/256, ttl=64 (request in 149)
   153 14.23696300( 10.0.0.3          10.0.0.4          ICMP      100 Echo (ping) reply    id=0x0c8d, seq=1/256, ttl=64
```

figure: output of pingall when run on display filter “icmp && not of”