



# GlobalLogic

A Hitachi Group Company

EDUCATION

## Smart Start: Linux/Networking Debugging network applications

Sergii Kudriavtsev

# Agenda

- \* Basic tools: nettop / iftop, dstat
- \* Connectivity tools and scanners: telnet, netcat (nc), nmap
- \* Network data monitoring and debugging applications:
  - tcpdump, iptraf (iptraf-ng), trafshow
- \* Network performance: ttcp -> nuttcp, iperf

# Basic tools: nettop (jnettop) / iftop, dstat

- nettop - Utility to show network traffic (both TCP and UDP v4 and v6) split by process and remote host.
- `$ git clone https://github.com/Emanem/nettop.git`
- `$ apt-get install libncurses5-dev libncursesw5-dev build-essential libpcap-dev`
- `$ make`
- `$ sudo ./nettop`

nettop 0.5 [ 1.25s ( 58/ 43/ 17/ 5/ 5)]					Total	0.83	3.36	KiB/s
кількість результатів: 11 500 (0,25 c)								
PID	CMDLINE				RECV	SENT		
1739	/usr/sbin/sshd -D				0.27	2.99 KiB/s		
	172.22.120.123				0.27	2.99 KiB/s		
-1	(kernel) localhost				574.07	383.51 Byte/s		
	172.22.120.3				289.43	236.01 Byte/s		
					284.64	147.50 Byte/s		
7672	/usr/share/skypeforlinux/skypeforlinux --type=utility --utility-sub-type=network.mojom.NetworkSe				0.00	0.00 Byte/s		
28323	/usr/lib/slack/slack --type=utility --utility-sub-type=network.mojom.NetworkService --lang=en-US				0.00	0.00 Byte/s		
24968	/lib/systemd/systemd-resolved				0.00	0.00 Byte/s		
23711	/usr/sbin/ntpd -p /var/run/ntpd.pid -g -u 133:144				0.00	0.00 Byte/s		
13387	/usr/local/desktopcentralagent/bin/dcondemand &				0.00	0.00 Byte/s		
12055	/opt/google/chrome/chrome --type=utility --utility-sub-type=network.mojom.NetworkService --lang=				0.00	0.00 Byte/s		
12010	/opt/google/chrome/chrome				0.00	0.00 Byte/s		
11528	/usr/lib/chromium-browser/chromium-browser --type=utility --utility-sub-type=network.mojom.Netwo				0.00	0.00 Byte/s		
10835	/sbin/dhclient -d -q -sf /usr/lib/NetworkManager/nm-dhcp-helper -pf /run/dhclient-eth0.pid -lf /				0.00	0.00 Byte/s		
7701	/usr/share/skypeforlinux/skypeforlinux --type=renderer --enable-crashpad --crashpad-handler-pid=				0.00	0.00 Byte/s		
7588	/usr/sbin/in.tftpd --listen --user tftp --address :69 --secure /var/lib/tftpboot				0.00	0.00 Byte/s		
7556	/usr/sbin/openvpn --daemon ovpn-client --status /run/openvpn/client.status 10 --cd /etc/openvpn				0.00	0.00 Byte/s		
5170	/usr/sbin/cups-browsed				0.00	0.00 Byte/s		
5169	/usr/sbin/cupsd -l				0.00	0.00 Byte/s		
1721	/usr/sbin/snmpd -Lsd -lf /dev/null -u Debian-snmp -g Debian-snmp -I -smux mteTrigger mteTriggerC				0.00	0.00 Byte/s		
1644	avahi-daemon: running [kbp1-lhp-a05785.local]				0.00	0.00 Byte/s		
1194	falcon-sensor				0.00	0.00 Byte/s		
1180	/sbin/rpcbind -f -w				0.00	0.00 Byte/s		

- jnettop - View hosts/ports taking up the most network traffic
- `$ sudo jnettop -i eth0`
- `$ sudo jnettop --display text -t 5 --format CSV`
- `$ sudo jnettop --display text -t 5 --format '$srcname$, $srcport$, $dstname$, $dstport$, $totalbps$'`

```
"0.0.0.0", "255.255.255.255", "UDP", "68", "67", "UNKNOWNv4", "", "690", "0", "690", "2", "0", "2", "138", "0",
", "138", "0", "0", "0", "", "?uid"?
"0.0.0.0", "0.0.0.0", "ETHER", "0", "0", "UNKNOWNv4", "UNKNOWNv4", "178", "0", "178", "1", "0", "1", "35", "0",
", "35", "0", "0", "0", "", "?uid"?
```

### format variables:

src, srcname, srcport, srcbytes,  
srcpackets, srcbps, srcpps,  
dst, dstname, dstport, dstbytes,  
dstpackets, dstbps, dstpps,  
proto, totalbytes, totalpackets,  
totalbps, totalpps, filterdata

```
run 0:00:04 device eth0      pkt[f]filter: none
[C]ntfilter: on [b]ps-bytes/s [l]ocal aggr: none [r]emote aggr: none
Please wait, shutting down...
```

LOCAL <-> REMOTE	PORT	PROTO	(IP)	PORT	TXBPS TX	RXBPS RX	TOTALBPS TOTAL
kbp1-lhp-a05785 <-> 172.22.120.123							
172.22.120.150	58600	TCP	172.22.120.123		1.15K/s	109b/s	1.26K/s
172.22.120.150	58600	TCP	172.22.120.123		5.77K	546b	6.31K
kbp1-lhp-a05785 <-> 109.197.218.17							
172.22.120.150	22194	TCP	109.197.218.17		240b/s	89b/s	329b/s
172.22.120.150	22194	TCP	109.197.218.17		1.03K	268b	1.29K
kbp1-lhp-a05785 <-> 239.255.255.250							
172.22.120.150	1900	UDP	239.255.255.250		142b/s	0b/s	142b/s
172.22.120.150	1900	UDP	239.255.255.250		642b	0b	642b
kbp1-lhp-a05785 <-> 172.22.120.3							
172.22.120.150	53	UDP	172.22.120.3		46b/s	80b/s	126b/s
172.22.120.150	53	UDP	172.22.120.3		185b	321b	506b
kbp1-lhp-a05785 <-> 172.22.120.3							
172.22.120.150	53	UDP	172.22.120.3		45b/s	79b/s	124b/s
172.22.120.150	53	UDP	172.22.120.3		183b	319b	502b
kbp1-lhp-a05785 <-> 172.22.120.3							
172.22.120.150	53	UDP	172.22.120.3		45b/s	79b/s	124b/s
172.22.120.150	53	UDP	172.22.120.3		183b	319b	502b
kbp1-lhp-a05785 <-> 172.22.120.3							
172.22.120.150	53	UDP	172.22.120.3		45b/s	79b/s	124b/s
172.22.120.150	53	UDP	172.22.120.3		183b	319b	502b
kbp1-lhp-a05785 <-> 172.22.120.3							
172.22.120.150	53	UDP	172.22.120.3		45b/s	79b/s	124b/s
172.22.120.150	53	UDP	172.22.120.3		183b	319b	502b
kbp1-lhp-a05785 <-> 172.22.120.3							
172.22.120.150	40550	UDP	172.22.120.3		45b/s	79b/s	124b/s
172.22.120.150	40550	UDP	172.22.120.3		183b	319b	502b

20.2. Reachability

TOTAL	4.59K/s	1.44K/s	6.03K/s
-------	---------	---------	---------



- iftop - display bandwidth usage on an interface by host
- `$ iftop -i eth1 # Interactive (ncurses) mode:`

	12,5Kb	25,0Kb	37,5Kb	50,0Kb	62,5Kb
kbp1-lhp-a05785	=> 172.22.120.123		4,22Kb	3,25Kb	3,59Kb
	<=		208b	208b	260b
kbp1-lhp-a05785	=> 172.22.120.4		336b	638b	792b
	<=		472b	1,14Kb	1,40Kb
kbp1-lhp-a05785	=> ec2-54-67-119-89.us-west-1.compute.amazonaws.com		0b	338b	662b
	<=		0b	119b	357b
kbp1-lhp-a05785	=> 109.197.218.17		488b	139b	113b
	<=		208b	139b	148b
kbp1-lhp-a05785	=> waw02s16-in-f10.1e100.net	177	175.9k	0b	94b
	<=		0b	128b	80b
kbp1-lhp-a05785	=> ec2-18-197-249-189.eu-central-1.compute.amazonaws.co	177	175.9k	0b	127b
	<=		0b	87b	54b
kbp1-lhp-a05785	=> 52.112.100.8	3.0m	166	66.3k	0b
	<=		0b	110b	69b
kbp1-lhp-a05785	=> 172.22.120.49	1.5m	108	51.2k	0b
	<=		0b	70b	44b
255.255.255.255	=> 172.22.120.49	1.2m	721	9.9k	0b
	<=		0b	152b	95b
kbp1-lhp-a05785	=> pmp-kbp1-1.globallogic.com	85	0.0	0b	42b
	<=		0b	42b	26b
kbp1-lhp-a05785	=> lh-in-f188.1e100.net	92	0.0	0b	42b
	<=		0b	42b	26b
kbp1-lhp-a05785	=> 20.86.226.133	111	4.1k	0b	42b
	<=		0b	0b	0b
172.22.5.41	=> all-systems.mcast.net		128b	26b	16b
	<=		0b	0b	0b
kbp1-lhp-a05785	=> mdns.mcast.net		0b	0b	16b
	<=		0b	0b	0b

- iftop - display bandwidth usage on an interface by host
- `$ iftop -i eth1 -t # Use text interface without ncurses and print the output to STDOUT.`

```
=====
# Host name (port/service if enabled)          last 2s    last 10s    last 40s cumulative
-----
1 255.255.255.255                               =>         0b         0b         0b         0B
  0.0.0.0                                       <=       1,29Kb       993b       993b       993B
-----
Total send rate:                               0b         0b         0b
Total receive rate:                           1,29Kb       993b       993b
Total send and receive rate:                   1,29Kb       993b       993b
-----
Peak rate (sent/received/total):               0b        1,29Kb      1,29Kb
Cumulative (sent/received/total):              0B         993B       993B
=====
```

# Basic tools: dstat



- dstat - Versatile tool for generating system resource statistics

- `$ sudo dstat -n --net-packets -N enp0s25,total --socket --tcp`

```
net/enp0s25--net/total- pkt/enp0s25--pkt/total- -----sockets----- ----tcp-sockets----
  recv  send: recv  send|#recv #send:#recv #send|tot tcp udp raw frg|lis act syn tim clo
    0    0 :    0    0 |    0    0 :    0    0 |739 31 16  0  0| 13 21  0  3  3
2666B 239B:3035B 239B|18.0  3.00 :20.0  3.00 |739 31 16  0  0| 13 20  0  3  4
5579B 2929B:5884B 2929B|45.0 23.0 :46.0 23.0 |737 29 16  0  0| 13 21  0  3  1
5073B 1692B:5612B 1692B|30.0 11.0 :33.0 11.0 |737 29 16  0  0| 13 21  0  3  1
3312B 134B:4172B 134B|20.0  2.00 :24.0  2.00 |737 29 16  0  0| 13 21  0  3  1
2822B  70B:3191B  70B|17.0  1.00 :19.0  1.00 |737 29 16  0  0| 13 21  0  3  1
3110B  70B:3585B  70B|18.0  1.00 :20.0  1.00 |737 29 16  0  0| 13 21  0  2  1
```

- `$ sudo dstat`

```
You did not select any stats, using -cdngy by default.
--total-cpu-usage-- -dsk/total- -net/total- ---paging-- ---system--
usr sys idl wai stl| read  writ| rcv  send| in  out| int  csw
 4  2  94  0  0| 84k 383k|  0   0| 92B 209B|1393 4063
 5  2  93  0  0|  0   0| 709B 1239B|  0   0|1041 2078
 4  2  94  0  0|  0  48k| 116B 306B|  0   0|1060 2241
 5  2  93  0  0|  0 128k|1001B 643B|  0   0|1169 2397
 6  3  91  0  0|  0  48k|4220B 3263B|  0   0|1309 2738
 5  2  93  0  0|  0   0|3393B 4905B|  0   0|1101 2224
```

# Connectivity tools and scanners: telnet

- telnet - user interface to the TELNET protocol

### TELNET client

- `$ telnet 8.8.8.8 53`

Trying 8.8.8.8...

Connected to 8.8.8.8.

Escape character is '^]'.

Connection closed by foreign host.

`$ telnet a.b.c.d 22`

Trying a.b.c.d...

Connected to a.b.c.d.

Escape character is '^]'.

SSH-2.0-OpenSSH\_7.6p1 Ubuntu-4ubuntu0.7

# Connectivity tools and scanners: netcat (nc)

- netcat (nc)

### **The network swiss army knife**

- server mode

- `$ nc -l -s 10.105.28.82 -p 3333`

- client mode

- direct connection

- `$ nc 10.105.28.82 3333`

- scanning

- `$ nc -z -v localhost 1-1000`

- `localhost [127.0.0.1] 995 (pop3s) open`

- `localhost [127.0.0.1] 993 (imaps) open`

- `localhost [127.0.0.1] 443 (https) open`

- `localhost [127.0.0.1] 143 (imap) open`

- `localhost [127.0.0.1] 110 (pop3) open`

- `localhost [127.0.0.1] 80 (http) open`

- `localhost [127.0.0.1] 25 (smtp) open`

- `localhost [127.0.0.1] 22 (ssh) open`

## EDUCATION

- netcat (nc)
- server + client
  - plain data exchange

server	client
\$ nc -l -s 10.105.28.82 -p 3333	
	\$ nc 10.105.28.82 3333
	data from client
data from client	
data from server	
	data from server
	<CTRL-C>



## EDUCATION

- netcat (nc)
  - archived exchange

server	client
\$ nc -l -s 10.105.28.82 -p 3333   gunzip -c	
	\$ cat   gzip   nc localhost 3333
	line1
	line2
	line3
	bye-bye
	<CTRL-D>
	<CTRL-C>
line1	
line2	
line3	

- netcat (nc)
  - encrypted exchange

server	client
<code>\$ read -s PASSWD &amp;&amp; export PASSWD</code>	<code>\$ read -s PASSWD &amp;&amp; export PASSWD</code>
<code>mypassword</code>	<code>mypassword</code>
<code>\$ nc -l -p 3333   openssl des -pass env:PASSWD -d</code>	
	<code>\$ echo "secret message"   openssl des -pass env:PASSWD -e   nc localhost 3333</code>
	<code>&lt;CTRL-C&gt;</code>
<code>secret message</code>	
<code>\$ unset PASSWD</code>	<code>\$ unset PASSWD</code>

# Connectivity tools and scanners: nmap

- nmap

**Network exploration tool and security / port scanner**

- Scanning
  - Get list of available hosts in network

```
$ nmap -sn 192.168.0.1-254
```

```
Starting Nmap 6.25 ( http://nmap.org ) at 2017-02-07 20:32 EET
```

```
Nmap scan report for 192.168.0.1
```

```
Host is up (0.0012s latency).
```

```
Nmap scan report for faust (192.168.0.55)
```

```
Host is up (0.000052s latency).
```

```
Nmap scan report for scully (192.168.0.59)
```

```
Host is up (0.00043s latency).
```

```
Nmap done: 254 IP addresses (3 hosts up) scanned in 8.18 seconds
```

- nmap

**Network exploration tool and security / port scanner**

- Scanning
  - Get list of open ports

```
$ nmap scanme.nmap.org
```

```
Starting Nmap 6.25 ( http://nmap.org ) at 2017-01-31 20:08 EET
```

```
Nmap scan report for scanme.nmap.org (45.33.32.156)
```

```
Host is up (0.19s latency).
```

```
Not shown: 996 closed ports
```

```
PORT      STATE SERVICE
```

```
22/tcp    open  ssh
```

```
80/tcp    open  http
```

```
9929/tcp  open  nping-echo
```

```
31337/tcp open  Elite
```

```
Nmap done: 1 IP address (1 host up) scanned in 13.95 seconds
```

- nmap

### Network exploration tool and security / port scanner

- Scanning
  - Port scanning techniques
    - -sS (TCP SYN scan). Sends TCP SYN packet.  
`$ sudo nmap -sS scanme.nmap.org`
      - requires superuser permission to send a raw packet.
      - fast.
      - allows clear, reliable differentiation between the **open**, **closed**, and **filtered** states
      - Responses:
        - SYN/ACK indicates the port is listening (**open**).
        - RST (reset) is indicative of **closed** port.
        - No response is received after several retransmissions: the port is marked as **filtered**.
        - ICMP unreachable error (type 3, code 0, 1, 2, 3, 9, 10, or 13) is received: the port is marked as **filtered**.



- nmap

### Network exploration tool and security / port scanner

- Scanning
  - Port scanning techniques
    - -sT (TCP connect scan). Nmap calls `connect()` call to establish connection.  
`$ nmap -sT scanme.nmap.org`
      - no need in superuser permission.
      - slow.
      - remote host will accept this connection and may log.
      - less efficient than TCP SYN scan.

- nmap

### Network exploration tool and security / port scanner

- Scanning
  - Port scanning techniques
    - -sU (UDP scans).

```
$ sudo nmap -sU scanme.nmap.org
```

      - requires superuser permission to send a raw packet.
      - slow.
      - hard to detect open ports.
      - Responses:
        - UDP packet (rarely): port is **open**.
        - ICMP port unreachable error (type 3, code 3) is received: the port is **closed**.
        - Other ICMP unreachable errors (type 3, codes 0, 1, 2, 9, 10, or 13) mark the port as **filtered**.
        - No response is received after retransmissions: the port is classified as **open|filtered**.
  - NOTE. Nmap's results are based on packets returned by the target machines (or firewalls in front of them). Such hosts may be untrustworthy and send responses intended to confuse or mislead Nmap.

# Network data monitoring and debugging applications: tcpdump

- tcpdump

**A Tool for network monitoring and data acquisition**

- Dumping network data

- `$ tcpdump -vv -i enp0s25 -w tcpdump.log`

- Viewing tcpdump's logs

- `$ tcpdump -n -r tcpdump.log | less`

- Flags of TCP packets:

- S - SYN (tcp-syn)
    - F - FIN (tcp-fin)
    - P - PUSH (tcp-push)
    - R - RST (tcp-rst)
    - U - URG (tcp-urg)
    - W - ECN CWR
    - E - ECN-Echo
    - . - ACK (tcp-ack)
    - none - no flags set

## EDUCATION

- tcpdump
- Test scanning with nc

tcpdump	nc server	nc client
\$ tcpdump -vv -i lo -w tcpdump.log 'tcp port 3333'		
	\$ nc -l -s 127.0.0.1 -p 3333	
		\$ nc 127.0.0.1 3333
		msg from client
	msg from client	
	msg from server	
		msg from server
		<CTRL-C>
<CTRL-C>		
\$ tcpdump -n -A --number -tttt -r tcpdump.log		

- tcpdump DUMPs:

Client to server: green background, server to client: yellow background

```
1 2017-02-08 13:45:42.564088 IP 127.0.0.1.51158 > 127.0.0.1.3333: Flags [ S ], seq 2049537805, win 43690, options
[mss 65495,sackOK,TS val 8948558 ecr 0,nop,wscale 7], length 0
E..<C@.8.8v.....z)w.....0.....
...N.....

2 2017-02-08 13:45:42.564105 IP 127.0.0.1.3333 > 127.0.0.1.51158: Flags [ S. ], seq 2355732552, ack 2049537806, win
43690, options [mss 65495,sackOK,TS val 8948558 ecr 8948558,nop,wscale 7], length 0
E..<..@.@.<.....i.Hz)w.....0.....
...N...N...

3 2017-02-08 13:45:42.564117 IP 127.0.0.1.51158 > 127.0.0.1.3333: Flags [ . ], ack 1, win 342, options [nop,nop,TS
val 8948558 ecr 8948558], length 0
E..4.D@.8.0).....z)w..i.I...V.(.....
...N...N

4 2017-02-08 13:45:47.943991 IP 127.0.0.1.51158 > 127.0.0.1.3333: Flags [ P. ], seq 1:17, ack 1, win 342, options
[nop,nop,TS val 8953938 ecr 8948558], length 16
E..D.E@.8.0l.....z)w..i.I...V.8.....
...R...Nmsg from client

5 2017-02-08 13:45:47.944002 IP 127.0.0.1.3333 > 127.0.0.1.51158: Flags [ . ], ack 17, win 342, options [nop,nop,TS
val 8953938 ecr 8953938], length 0
E..4+.@.@.....i.Iz)w....V.(.....
...R...R

6 2017-02-08 13:45:52.712025 IP 127.0.0.1.3333 > 127.0.0.1.51158: Flags [ P. ], seq 1:17, ack 17, win 342, options
[nop,nop,TS val 8958706 ecr 8953938], length 16
E..D+.@.@.....i.Iz)w....V.8.....
.....Rmsg from server
```



- tcpdump DUMPS:

Client to server: green background, server to client: yellow background

```
7 2017-02-08 13:45:52.712037 IP 127.0.0.1.51158 > 127.0.0.1.3333: Flags [ .], ack 17, win 342, options [nop,nop,TS
val 8958706 ecr 8958706], length 0
E..4.F@.@.(.....z)w..i.Y...V.(.....
.....

8 2017-02-08 13:45:55.271991 IP 127.0.0.1.51158 > 127.0.0.1.3333: Flags [ F.], seq 17, ack 17, win 342, options
[nop,nop,TS val 8961266 ecr 8958706], length 0
E..4.G@.@.z.....z)w..i.Y...V.(.....
.....

9 2017-02-08 13:45:55.272135 IP 127.0.0.1.3333 > 127.0.0.1.51158: Flags [ F.], seq 17, ack 18, win 342, options
[nop,nop,TS val 8961266 ecr 8961266], length 0
E..4+ @.@.....i.Yz)w....V.(.....
.....

10 2017-02-08 13:45:55.272157 IP 127.0.0.1.51158 > 127.0.0.1.3333: Flags [ .], ack 18, win 342, options [nop,nop,TS
val 8961267 ecr 8961266], length 0
E..4.H@.@.y.....z)w..i.Z...V.(.....
.....
```

- tcpdump DUMPs:

Client to server: green background, server to client: yellow background

- Filtering by TCP flags

- Show packets with SYN flag:

```
$ tcpdump -n -A -tttt -r tcpdump.log 'tcp[tcpflags] & tcp-syn != 0'
```

```
13:07:31.986393 IP 10.105.28.82.42154 > 176.37.18.212.3038: Flags [ S], seq 1765949417,  
win 29200, options [mss 1460,sackOK,TS val 6658005 ecr 0,nop,wscale 7], length 0  
13:07:31.989522 IP 176.37.18.212.3038 > 10.105.28.82.42154: Flags [ S.], seq 551062087,  
ack 1765949418, win 28960, options [mss 1387,sackOK,TS val 548544849 ecr  
6658005,nop,wscale 7], length 0  
13:13:57.483879 IP 172.17.48.42.49331 > 10.105.28.170.8081: Flags [ SEW], seq 4261555761,  
win 8192, options [mss 1460,nop,wscale 8,nop,nop,sackOK], length 0
```

- tcpdump DUMPs:

Client to server: green background, server to client: yellow background

- Filtering by TCP flags

- Show packets whether with SYN or ACK flags:

```
$ tcpdump -n -A -tttt -r tcpdump.log 'tcp[tcpflags] & (tcp-syn|tcp-ack) != 0'
13:07:31.813309 IP 10.105.28.82.58476 > 77.120.114.98.2222: Flags [ .], ack 5476, win
1424, options [nop,nop,TS val 6657832 ecr 2083753677], length 0
13:07:31.986447 IP 10.105.28.82.47742 > 80.242.105.76.11700: Flags [ S], seq 325561646,
win 29200, options [mss 1460,sackOK,TS val 6658005 ecr 0,nop,wscale 7], length 0
13:07:31.989522 IP 176.37.18.212.3038 > 10.105.28.82.42154: Flags [ S.], seq 551062087,
ack 1765949418, win 28960, options [mss 1387,sackOK,TS val 548544849 ecr
6658005,nop,wscale 7], length 0
2017-02-08 13:07:32.413282 IP 10.105.28.82.47742 > 80.242.105.76.11700: Flags [ F.], seq
120, ack 156, win 229, options [nop,nop,TS val 6658432 ecr 14822612], length 0
2017-02-08 13:07:33.501546 IP 77.120.114.98.2222 > 10.105.28.82.58476: Flags [ P.], seq
9692:9768, ack 1, win 358, options [nop,nop,TS val 2083755365 ecr 6659511], length 76
2017-02-08 13:07:40.267316 IP 192.4.116.173.60222 > 10.105.28.54.17921: Flags [ R.], seq
1023632408, ack 2768700647, win 510, length 0
2017-02-08 13:13:57.483879 IP 172.17.48.42.49331 > 10.105.28.170.8081: Flags [ SEW], seq
4261555761, win 8192, options [mss 1460,nop,wscale 8,nop,nop,sackOK], length 0
```

- tcpdump DUMPs:

Client to server: **green background**, server to client: **yellow background**

- Filtering by TCP flags

- Show packets **with both SYN and ACK** flags:

```
$ tcpdump -n -A -tttt -r tcpdump.log '(tcp[tcpflags] & (tcp-syn) != 0) &&  
(tcp[tcpflags] & (tcp-ack) != 0)'
```

```
13:07:31.989522 IP 176.37.18.212.3038 > 10.105.28.82.42154: Flags [ S.], seq 551062087,  
ack 1765949418, win 28960, options [mss 1387,sackOK,TS val 548544849 ecr  
6658005,nop,wscale 7], length 0
```

- wireshark (free and open-source packet analyzer)

[illegible]

# Network data monitoring and debugging applications: iptraf (iptraf-ng)



- iptraf (iptraf-ng)

### Interactive Colorful IP LAN Monitor

- `$ iptraf-ng`
- General statistics and logging
  - `Configure... -> Logging: On -> Exit configuration`
  - `IP traffic monitor -> All interfaces -> Input log file name (/var/log/iptraf-ng/*.log)`
- Filtering
  - `Filters... -> IP... -> Define new filter... -> src TCP port 3333 -> Press <I>`
    - `Source port: 3333 to 3333`
    - `Protocols to match: TCP 'Y'`
    - `Press <ENTER>`
  - `Press <CTRL-X>`
  - `Apply filter...`
    - `Select filter with name "src TCP port 3333"`
  - `Return to the main screen`
  - `IP traffic monitor -> lo -> Input log file name (if necessary)`

- iptraf (iptraf-ng)

### Interactive Colorful IP LAN Monitor

- `$ iptraf-ng`
- Filtering
  - Run server listening on port #3333, connect client to it and exchange data
    - Server  
`$ nc -l -s 127.0.0.1 -p 3333`
    - Client  
`$ nc 127.0.0.1 3333`
  - Observe statistics for connection on local interface on TCP port 3333 in `iptraf-ng`

TCP Connections (Source Host:Port)	Packets	Bytes	Flag	Iface
127.0.0.1:51158	= 8	464	--A- 1o	
127.0.0.1:3333	= 6	360	-PA- 1o	

# Network data monitoring and debugging applications: trafshow

- trafshow

### Full screen visualization of the network traffic

- `$ trafshow`
- Filtering
  - See `tcpdump(1)` man page for optional filtering expression.
  - `$ trafshow -i lo 'tcp port 3333'`
  - Run server listening on port #3333, connect client to it and exchange data
    - Server  
`$ nc -l -s 127.0.0.1 -p 3333`
    - Client  
`$ nc 127.0.0.1 3333`
  - Observe statistics for connection on local interface on TCP port 3333 in `trafshow`

Source	Destination	Protocol	Size	CPS
localhost,51158	localhost,3333	tcp	172	15
localhost,3333	localhost,51158	tcp	120	35

# Network performance: nuttcp

- nuttcp

**Network performance measurement tool** - Its most basic usage is to determine the raw TCP (or UDP) network layer throughput by transferring memory buffers from a source system across an interconnecting network to a destination system, either transferring data for a specified time interval, or alternatively transferring a specified number of bytes.

- Server mode

- `$ nuttcp -S`

- Client mode

- `$ nuttcp -il server_hostname`

```
99.3750 MB / 1.00 sec = 833.6163 Mbps 0 retrans
93.3750 MB / 1.00 sec = 783.2683 Mbps 0 retrans
96.3125 MB / 1.00 sec = 807.9044 Mbps 0 retrans
100.1875 MB / 1.00 sec = 840.4799 Mbps 0 retrans
98.3750 MB / 1.00 sec = 825.2277 Mbps 0 retrans
98.8125 MB / 1.00 sec = 828.8886 Mbps 0 retrans
99.1875 MB / 1.00 sec = 832.0342 Mbps 0 retrans
96.2500 MB / 1.00 sec = 807.4237 Mbps 0 retrans
100.3125 MB / 1.00 sec = 841.4755 Mbps 0 retrans
93.5000 MB / 1.00 sec = 784.3419 Mbps 0 retrans
```

```
977.2500 MB / 10.02 sec = 818.3851 Mbps 7 %TX 14 %RX 0 retrans 0.87 msRTT
```

- Client mode reverse direction

- `$ nuttcp -il -r server_hostname`

- nuttcp
- Send 300 Mbps of UDP in bursts of 20 packets for 5 seconds
  - `$ nuttcp -u -Ri300m/50 -i 1 -T5 server_hostname`  
35.7607 MB / 1.00 sec = 299.9241 Mbps 0 / 36619 ~drop/pkt 0.00 ~%loss  
35.7617 MB / 1.00 sec = 300.0018 Mbps 0 / 36620 ~drop/pkt 0.00 ~%loss  
35.5645 MB / 1.00 sec = 298.3730 Mbps 2 / 36420 ~drop/pkt 0.00549 ~%loss  
33.3262 MB / 1.00 sec = 279.5649 Mbps 2253 / 36379 ~drop/pkt 6.19 ~%loss  
34.6924 MB / 1.00 sec = 291.0208 Mbps 1095 / 36620 ~drop/pkt 2.99 ~%loss  
  
175.1836 MB / 5.00 sec = 293.7936 Mbps 99 %TX 11 %RX 3350 / 182738 drop/pkt 1.83 %loss

# This amount of loss is tolerable.

- Send a 300 Mbps in bursts of 50 packets:
  - `$ nuttcp -u -Ri300m/50 -i 1 -T5 server_hostname`  
23.8232 MB / 1.00 sec = 199.8394 Mbps 12238 / 36633 ~drop/pkt 33.41 ~%loss  
25.2227 MB / 1.00 sec = 211.5836 Mbps 10783 / 36611 ~drop/pkt 29.45 ~%loss  
25.1846 MB / 1.00 sec = 211.2405 Mbps 10816 / 36605 ~drop/pkt 29.55 ~%loss  
24.2969 MB / 1.00 sec = 203.8392 Mbps 11754 / 36634 ~drop/pkt 32.08 ~%loss  
25.1484 MB / 1.00 sec = 210.8927 Mbps 10864 / 36616 ~drop/pkt 29.67 ~%loss  
  
123.7090 MB / 5.00 sec = 207.5136 Mbps 99 %TX 12 %RX 56470 / 183148 drop/pkt 30.83 %loss

The level of loss become excessive, showing that some device in the path needs bigger buffers. Note that bursts of > 50 packets is common with TCP over paths with a RTT > 20 ms.

- nuttcp 10G+ UDP testing
- nuttcp is definitely the best tool for high-speed UDP testing. To get a full 10Gbps using UDP requires the right MTU size (9K), the right packet size (8972), and an larger buffer size.

- `$ nuttcp -l8972 -T30 -u -w4m -Ru -il server_hostname`



# Network performance: iperf

- **iperf - Perform network throughput tests**

**iPerf is an open-source tool written in the C programming language. Moreover, it works in a client-server model and supports UDP and TCP. Therefore, we need to have two systems that both have the tool installed. First, we need to initiate the server. After that, we need to connect to the server from the client machine.**

- Initiating the Server
- `$ iperf -s`

```
-----  
Server listening on TCP port 5001  
TCP window size: 128 KByte (default)  
-----
```

The server is now listening on TCP port 5001. By default, iPerf uses TCP and will listen on port 5001.

Optional flags that we can include:

- u will make the server use UDP rather than TCP
- p will change the default port

- iperf
  - Configure the server use **UDP** and listen on port 5003:

```
# iperf -s -u -p 5003
```

```
-----
```

```
Server listening on UDP port 5003
```

```
Receiving 1470 byte datagrams
```

```
UDP buffer size: 208 KByte (default)
```

```
-----
```

```
The server is now listening on UDP port 5003.
```

- iperf

- Connecting to the Server From the Client using TCP

After initiating the server, we should connect to it from the client machine:

- `$ iperf -c $SERVER_IP`

```
-----  
Client connecting to 192.168.0.2, TCP port 5001
```

```
TCP window size: 85.0 KByte (default)  
-----
```

```
[ 3] local 192.168.0.1 port 38616 connected with 192.168.0.2 port 5001
```

```
[ ID] Interval          Transfer      Bandwidth
```

```
 [ 3]  0.0-10.1 sec   6.12 MBytes  5.10 Mbits/sec
```

`-i` specifies the interval time in seconds. 10 is the default

`-t` specifies the time to run the test in seconds

`-p` specifies the port. 5001 is the default

`-w` specifies the TCP window size. 85 KB is the default

- iperf
  - Connecting to the Server From the Client using TCP

```
■ iperf -c 5.182.18.49 -i 5 -t 15 -w 416K -p 5003
```

```
iperf -c 5.182.18.49 -i 5 -t 15 -w 416K -p 5003
```

```
-----  
Client connecting to 5.182.18.49, TCP port 5003
```

```
TCP window size: 416 KByte  
-----
```

```
[ 3] local 192.168.1.24 port 47300 connected with 5.182.18.49 port 5003
```

[ ID]	Interval	Transfer	Bandwidth
[ 3]	0.0- 5.0 sec	2.88 MBytes	4.82 Mbits/sec
[ 3]	5.0-10.0 sec	2.75 MBytes	4.61 Mbits/sec
[ 3]	10.0-15.0 sec	2.75 MBytes	4.61 Mbits/sec
[ 3]	0.0-15.2 sec	8.38 MBytes	4.62 Mbits/sec

Interval time is set to 5 seconds. Test duration is set to 15 seconds and the TCP window size to 416 KB. Port is changed to 5003. Most importantly, the server needs to be listening on port 5003 instead of 5001 for the connection to be established.

- iperf
  - Connecting to the Server From the Client using UDP

UDP can be used instead of TCP.

Certainly, the server needs to be using UDP as well.

```
■ $ iperf -c 5.182.18.49 -u
```

```
-----  
Client connecting to 5.182.18.49, UDP port 5001
```

```
Sending 1470 byte datagrams, IPG target: 11215.21 us (kalman adjust)
```

```
UDP buffer size: 208 KByte (default)  
-----
```

```
[ 3] local 192.168.1.24 port 45640 connected with 5.182.18.49 port 5001
```

```
[ 3] WARNING: did not receive ack of last datagram after 10 tries.
```

```
[ ID] Interval      Transfer    Bandwidth
```

```
[ 3] 0.0-10.0 sec 1.25 MBytes 1.05 Mbits/sec
```

```
[ 3] Sent 892 datagrams
```

- iperf
  - Connecting to the Server From the Client using UDP

Bandwidth is much lower than TCP.

The reason is that iPerf limits the bandwidth for UDP to 1Mbits/sec by default. However, limit can be overridden by adding the -b flag:

```
■ $ iperf -c 5.182.18.49 -u -b 1000M
```

```
-----  
Client connecting to 5.182.18.49, UDP port 5001
```

```
Sending 1470 byte datagrams, IPG target: 11.22 us (kalman adjust)
```

```
UDP buffer size: 208 KByte (default)  
-----
```

```
[ 3] local 192.168.1.24 port 56981 connected with 5.182.18.49 port 5001
```

```
[ 3] WARNING: did not receive ack of last datagram after 10 tries.
```

```
[ ID] Interval          Transfer          Bandwidth
```

```
[ 3] 0.0-10.0 sec    180 MBytes    151 Mbits/sec
```

```
[ 3] Sent 128140 datagrams
```



# Thank You