GS Operations - C10M

Concurrent 10 million connection and Load Test - DPDK/mTCP/Pktgen

Project Name	C10M
Purpose	Develop an opensource based packet generator for BIGIP load evaluation/trouble-shooting using technology below DPDK http://www.dpdk.org/ is an user space drivers and libraries for fast packet processing, it can generates 10M/pps, 10M/cps mTCP A Highly Scalable User-level TCP Stack for Multicore Systems. MoonGen https://github.com/emmericp/MoonGen to generate raw packet like SYN/RST/ACK/UDP/ICMP flooding Pktgen-DPDK http://dpdk.org/browse/apps/pktgen-dpdk/ Another raw packet generator from Intel, runs with latest DPDK repo
Project Contacts	Vincent Li

Background:

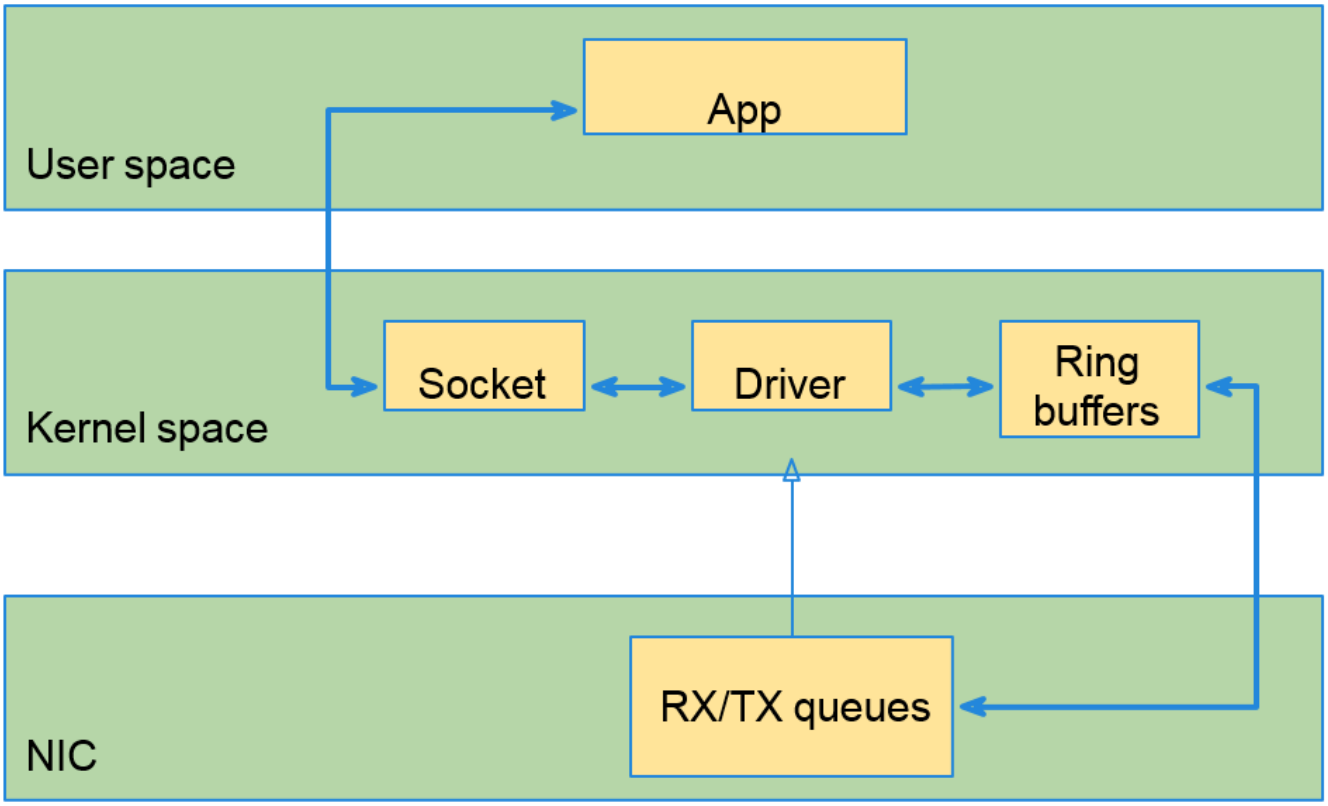
The Secret to 10 Million Concurrent Connections -The Kernel is the Problem, Not the Solution

http://highscalability.com/blog/2013/5/13/the-secret-to-10-million-concurrent-connections-the-kernel-i.html

Problem:

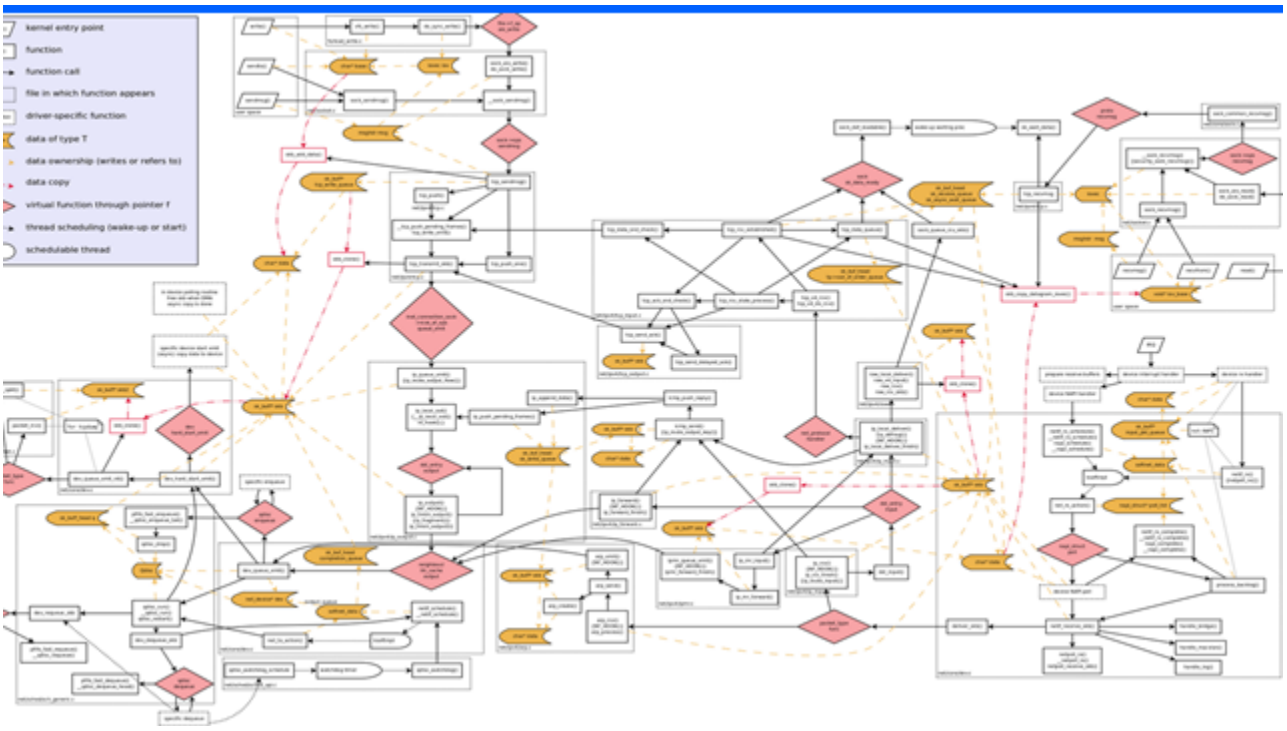
http://www.slideshare.net/garyachy/dpdk-44585840?qid=254b419f-1d44-44f1-99c4-87f13b7d5fe4&v=default&b=&from_search=8

Simplified packet processing in Linux:



Real packet processing in Linux:

Linux network data flow



- System calls
- Context switching on blocking I/O
- Data Copying from kernel to user space
- Interrupt handing in kernel

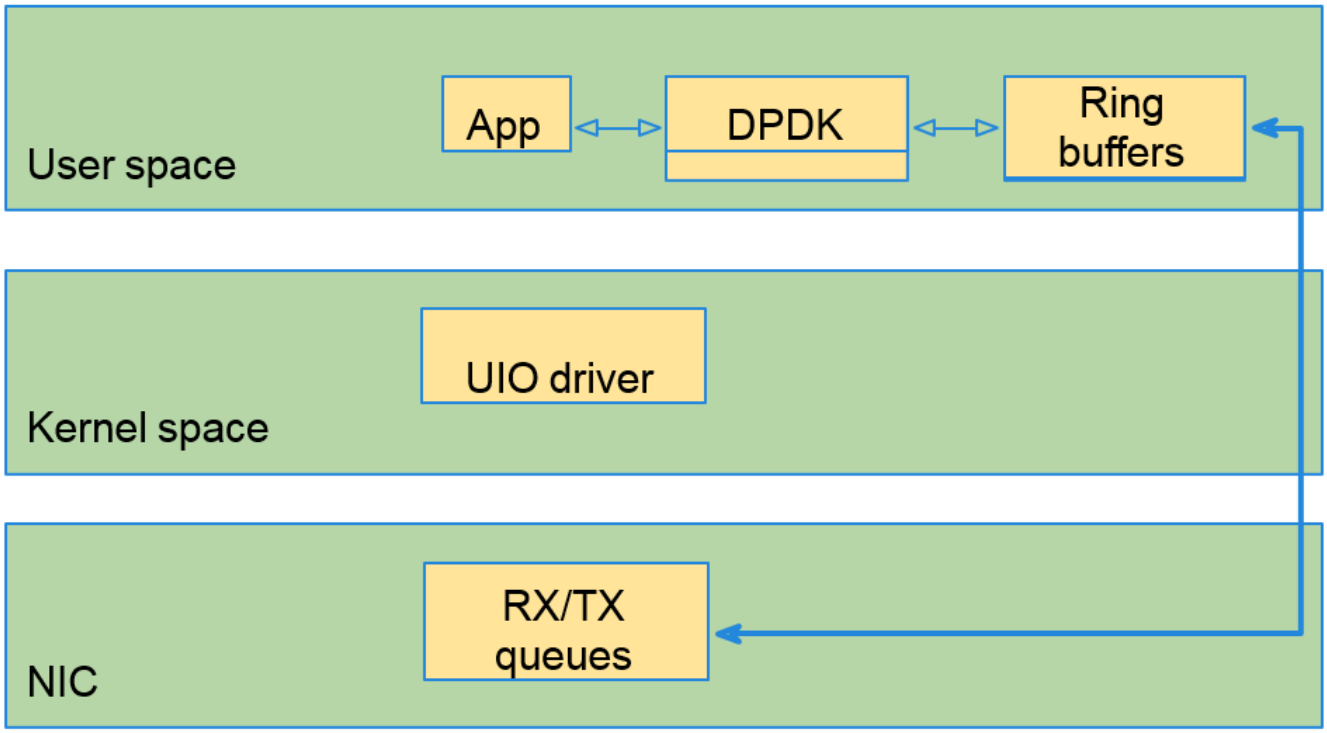
Expense of sendto :

- sendto - system call: 96ns
- sosend_dgram - lock sock_buff, alloc mbuf, copy in: 137ns
- udp_output - UDP header setup: 57ns
- ip_output - route lookup, ip header setup: 198ns
- ether_output - MAC lookup, MAC header setup: 162ns
- ixgbe_xmit - device programing: 220ns

Total: 950ns

Solution:

Packet processing with DPDK



- Processor affinity (separate cores)
- Huge pages(no swap, TLB)
- UIO (no copying from kernel)
- Polling (no interrupts overhead)
- Lockless synchronization(avoid waiting)
- Batch packets handling
- SSE, NUMA awareness

UIO for example:

Kernel space (UIO framework) <------>/dev/uioX<------>userspace epoll/mmap<-------->App

Problem:

http://www.ndsl.kaist.edu/~kyoungsoo/papers/mtcp.pdf

Limitaions of the Kernel's TCP stack

- Lack of connection locality
- Shared file descriptor space
- Inefficient per-packet processing

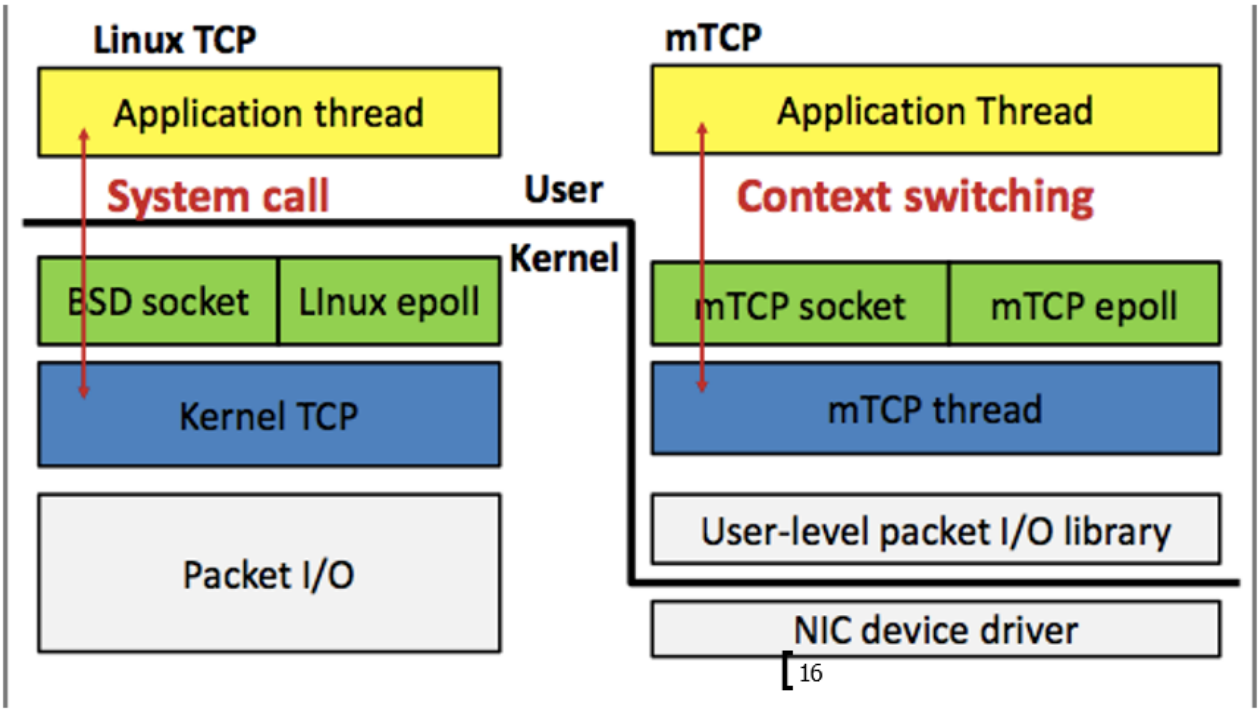
- System call overhead

Solution:

- Batching in packet I/O, TCP processing, user applications (reduce system call overhead)
- Connection locality on multicore systems - handling same connection on same core, avoid cache pollution (solve connection locality)
- No descriptor sharing between mTCP thread

<PROJECT DATA SPACE>

mTCP: A Highly Scalable User-level TCP Stack for Multicore Systems https://github.com/eunyoung14/mtcp



clone of mTCP in ES codeshare http://git.es.f5net.com/index.cgi/codeshare/tree/vli/mtcp

clone addition:

- Change apachebench configure script to compile with dpdk support
- Ported SSL BIO onto mTCP to enable apachebench to perform SSL test
- Add SSL clienthello stress test based on epwget and ssl-dos

- Add command line option in epwget and apachebenach to enable source address pool to congest servers
- Increase mTCP SYN BACKLOG to increase concurrent connection
- Changed DPDK .config to compile DPDK as combined shared library
- Tuned send/receive buffer size in epwget.conf to achieve ~7 million concurrent connection on Dell Poweredge R710 II 72G MEM, 16 core, Intel NIC 82599ES

mTCP installation

https://github.com/eunyoung14/mtcp has detail installation

```
- DPDK VERSION -
----------------
1. Set up Intel's DPDK driver. Please use our version of DPDK.
   We have only changed the lib/igb_uio/ submodule. The best
   method to compile DPDK package is to use DPDK's tools/setup.sh
   script. Please compile your package based on your own hardware
   configuration. We tested the mTCP stack on Intel Xeon E5-2690
   (x86_64) machine with Intel 82599 Ethernet adapters (10G). We
   used the following steps in the setup.sh script for our setup:

    - Press [10] to compile the package
          - Press [13] to install the driver
          - Press [17] to setup 1024 2MB hugepages
          - Press [19] to register the Ethernet ports
          - Press [31] to quit the tool

  - check that DPDK package creates a new directory of compiled
  libraries. For x86_64 machines, the new subdirectory should be
  *dpdk-2.1.0/x86_64-native-linuxapp-gcc*

  - only those devices will work with DPDK drivers that are listed
  on this page: http://dpdk.org/doc/nics. Please make sure that your
  NIC is compatible before moving on to the next step.

2. Next bring the dpdk-registered interfaces up. Please use the
   setup_iface_single_process.sh script file present in dpdk-2.1.0/tools/
   directory for this purpose. Please change lines 49-51 to change the IP
   address. Under default settings, run the script as:
       # ./setup_iface_single_process.sh 3

   This sets the IP address of your interfaces as 10.0.x.3.

3. Create soft links for include/ and lib/ directories inside
   empty dpdk/ directory:
       # cd dpdk/
    # ln -s <path_to_dpdk_2_1_0_directory>/x86_64-native-linuxapp-gcc/lib lib
    # ln -s <path_to_dpdk_2_1_0_directory>/x86_64-native-linuxapp-gcc/include include

4. Setup mtcp library:
    # ./configure --with-dpdk-lib=$<path_to_mtcp_release_v3>/dpdk
      ## And not dpdk-2.1.0!
      ## e.g. ./configure --with-dpdk-lib=`echo $PWD`/dpdk
    # cd mtcp/src
       # make
  - check libmtcp.a in mtcp/lib
  - check header files in mtcp/include

5. make in util/:
       # make

6. make in apps/example:
       # make
  - check example binary files

7. Check the configurations
  - epserver.conf for server-side configuration
  - epwget.conf for client-side configuration
  - you may write your own configuration file for your application
  - please see README.config for more details
    -- for the latest version, dyanmic ARP learning is *DISABLED*

8. Run the applications!
```

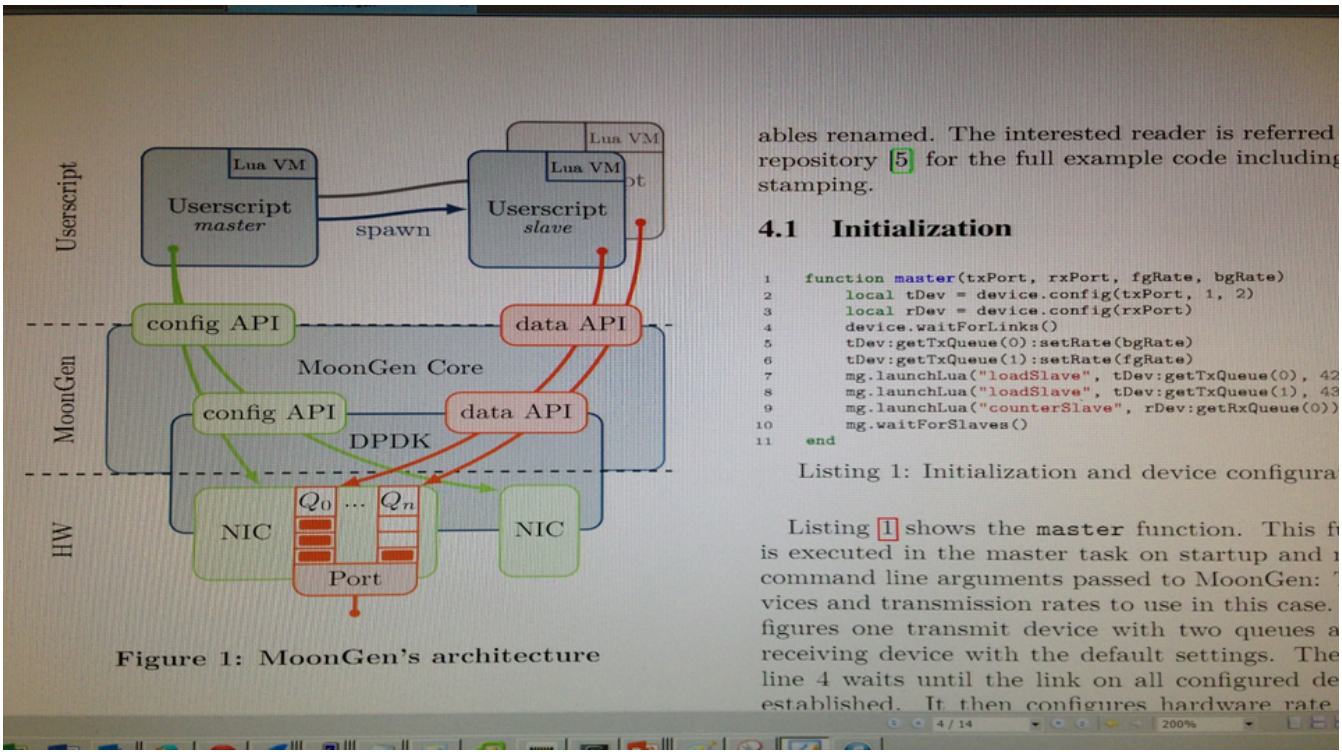
mTCP App configuration

################ mtcp configuration file ################

The underlying I/O module you want to use. Please

```
# enable only one out of the two.
io = dpdk
num_cores = 8
num_ip = 64
# Number of memory channels per processor socket (dpdk-only)
num_mem_ch = 4
#------ DPDK ports -------#
#port = dpdk0 dpdk1
port = dpdk0
#port = dpdk0:0
#port = dpdk0:1

# Enable multi-process support (under development)
#multiprocess = 0 master
#multiprocess = 1

# Receive buffer size of sockets
rcvbuf = 512
# Send buffer size of sockets
sndbuf = 512
# Maximum concurrency per core
max_concurrency = 1000000
# Maximum number of socket buffers per core
# Set this to small value if there are many idle connections
max_num_buffers = 1000000
# TCO timeout seconds
# (tcp_timeout = -1 can disable the timeout check)
tcp_timeout = 30
```

MoonGen: fully scriptable high-speed packet generator built on DPDK and LuaJIT. https://github.com/emmericp/MoonGen



Userscript Master

```
 1 local dpdk              = require "dpdk"
 2 local memory     = require "memory"
 3 local device     = require "device"
 4 local stats              = require "stats"
 5
 6
 7 function master(txPorts, minIp, numIps, rate)
 8         if not txPorts then
 9                 printf("usage: txPort1[,txPort2[,...]] [minIP numIPs rate]")
10                 return
11         end
12         txPorts = tostring(txPorts)
13         minIp = minIp or "10.0.0.1"
14         numIps = numIps or 100
15         rate = rate or 0
16         for currentTxPort in txPorts:gmatch("(%d+),?") do
17                 currentTxPort = tonumber(currentTxPort)
18                 local txDev = device.config({ port = currentTxPort })
19                 txDev:wait()
20                 txDev:getTxQueue(0):setRate(rate)
21                 dpdk.launchLua("loadSlave", currentTxPort, 0, minIp, numIps)
22         end
23         dpdk.waitForSlaves()
24 end
25
```

Userscript slave

```
26 function loadSlave(port, queue, minA, numIPs)
27         --- parse and check ip addresses
28
29         local minIP, ipv4 = parseIPAddress(minA)
30         if minIP then
31                 printf("INFO: Detected an %s address.", minIP and "IPv4" or "IPv6")
32         else
33                 errorf("ERROR: Invalid minIP: %s", minA)
34         end
35
36         -- min TCP packet size for IPv6 is 74 bytes (+ CRC)
37         local packetLen = ipv4 and 60 or 74
38
39         --continue normally
40         local queue = device.get(port):getTxQueue(queue)
41         --local srcport = math.random(0, 2^16 - 1)
42         local mem = memory.createMemPool(function(buf)
43                 buf:getTcpPacket(ipv4):fill{
44                         ethSrc="a0:36:9f:a1:4d:6d", ethDst="52:54:00:2E:62:A2",
45                         --- ip4Dst="10.9.1.2",
46                         ip4Dst="10.9.3.6",
47                         ip6Dst="fd06::1",
48                         tcpDst="80",
49                         tcpSyn=1,
50                         tcpSeqNumber=1,
51                         tcpWindow=10,
52                         pktLength=packetLen }
53         end)
54
55         local lastPrint = dpdk.getTime()
56         local totalSent = 0
57         local lastTotal = 0
58         local lastSent = 0
59         local bufs = mem:bufArray(128)
60         local counter = 0
61         local c = 0
62
63         local txStats = stats:newDevTxCounter(queue, "plain")
```

```
64           while dpdk.running() do
65                   -- faill packets and set their size
66                   bufs:alloc(packetLen)
67                   for i, buf in ipairs(bufs) do
68                           local pkt = buf:getTcpPacket(ipv4)
69
70                           --increment IP
71                           if ipv4 then
72                                   pkt.ip4.src:set(minIP)
73                                   pkt.ip4.src:add(counter)
74                                   pkt.tcp.src = math.random(0, 2^16 - 1)
75                           else
76                                   pkt.ip6.src:set(minIP)
77                                   pkt.ip6.src:add(counter)
78                           end
79                           counter = incAndWrap(counter, numIPs)
80
81                           -- dump first 3 packets
82                           if c < 3 then
83                                   buf:dump()
84                                   c = c + 1
85                           end
86                   end
87                   --offload checksums to NIC
88                   bufs:offloadTcpChecksums(ipv4)
89
90                   totalSent = totalSent + queue:send(bufs)
91                   txStats:update()
92           end
93           txStats:finalize()
94   end
```

clone of MoonGen in ES codeshare **http://git.es.f5net.com/index.cgi/codeshare/tree/vli/MoonGen**

- improved tcp syn flooding with random src ip and src port
- added DNS flooding script to test Victoria2 DNS DDOS Hardware protection
- added icmp echo flooding

MoonGen Installation:

1. Install the dependencies (see below)
2. git submodule update --init
3. ./build.sh
4. ./setup-hugetlbfs.sh
5. Run MoonGen from the build directory

How to Run MoonGen script:

command syntax: build/MoonGen examples/<scriptname> <dpdk port> <min src ip> <# of src ip> <rate>

#build/MoonGen examples/dns-flood-victoria.lua 0 10.0.0.1 16000000 10000

Hardware SPEC:

Dell Poweredge R710 72G MEM, 16 core, Intel NIC 82599 2.40GHz 20 cores 64G MEM

DUT: Victoria B2250 Intel(R) Xeon(R) CPU E5-2658 v2 @

Dell PowerEdge R210 II (used $300) 8 core, 32G MEM Intel 1G NIC I350 (cpu64-rhel6) 4 cores 16G MEM

DUT: BIGIP KVM VE CPU: QEMU Virtual CPU version

Load Test Example

1, DNS flooding without HW acceleration

#build/MoonGen examples/dns-flood-victoria.lua 0 10.0.0.1 16000000 10000

Device: id=0] Sent 13710082176 packets, current rate 4.51 Mpps, 3246.01 MBit/s, 3967.35 MBit/s wire rate.

[Device: id=0] Sent 13714591360 packets, current rate 4.51 Mpps, 3246.53 MBit/s, 3967.98 MBit/s wire rate.

[Device: id=0] Sent 13719099520 packets, current rate 4.51 Mpps, 3245.79 MBit/s, 3967.08 MBit/s wire rate.

top - 12:07:02 up 1 day, 20:38, 1 user, load average: 5.22, 7.46, 9.27

Tasks: 777 total, 19 running, 758 sleeping, 0 stopped, 0 zombie

Cpu(s): 50.6%us, 40.2%sy, 0.0%ni, 9.2%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st

Mem: 66080376k total, 65722732k used, 357644k free, 108700k buffers

Swap: 5242872k total, 0k used, 5242872k free, 4048612k cached

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
17859	root	1	-19	57.9g	145m	123m	R	100.8	0.2	56:44.33	tmm.0 -T 10 --tmid **<==========HIGH CPU**
17741	root	1	-19	57.9g	145m	123m	R	100.5	0.2	58:08.51	tmm.0 -T 10 --tmid
17853	root	1	-19	57.9g	145m	123m	R	100.5	0.2	56:46.73	tmm.0 -T 10 --tmid
17854	root	1	-19	57.9g	145m	123m	R	100.5	0.2	56:46.97	tmm.0 -T 10 --tmid
17855	root	1	-19	57.9g	145m	123m	R	100.5	0.2	56:46.06	tmm.0 -T 10 --tmid
17856	root	1	-19	57.9g	145m	123m	R	100.5	0.2	56:37.67	tmm.0 -T 10 --tmid

```
 17857 root      1 -19 57.9g 145m 123m R 100.5  0.2  56:45.54 tmm.0 -T 10 --tmid
 17858 root      1 -19 57.9g 145m 123m R 100.5  0.2  56:45.70 tmm.0 -T 10 --tmid
 17860 root      1 -19 57.9g 145m 123m R 100.5  0.2  56:45.65 tmm.0 -T 10 --tmid
 17852 root      1 -19 57.9g 145m 123m R 100.2  0.2  56:50.91 tmm.0 -T 10 --tmid
 20110 root     RT   0     0    0    0 S 80.6  0.0   0:27.55 [enforcer/11]
 20111 root     RT   0     0    0    0 R 80.6  0.0   0:27.56 [enforcer/15]
 20116 root     RT   0     0    0    0 R 80.6  0.0   0:27.50 [enforcer/13]
 20108 root     RT   0     0    0    0 R 80.2  0.0   0:27.55 [enforcer/19]
 20109 root     RT   0     0    0    0 R 80.2  0.0   0:27.57 [enforcer/17]
 20112 root     RT   0     0    0    0 S 80.2  0.0   0:27.55 [enforcer/5]
 20113 root     RT   0     0    0    0 R 80.2  0.0   0:27.52 [enforcer/1]


-------------------------------------------------------------------
Ltm::Virtual Server: vs_dns_10g
-------------------------------------------------------------------
Status
  Availability     : unknown
  State            : enabled
  Reason           : The children pool member(s) either don't have service check
  CMP              : enabled
  CMP Mode         : all-cpus
  Destination      : 10.3.3.249:53
  PVA Acceleration : none
Traffic                     ClientSide  Ephemeral  General
  Bits In                     11.5G         0        -
  Bits Out                    16.7G         0        -
  Packets In                  20.0M         0        -
  Packets Out                 20.0M         0        -
  Current Connections         27.1M         0        -
  Maximum Connections         27.1M         0        -
  Total Connections           28.8M         0        -
```

2, DNS flooding with HW acceleration

#build/MoonGen examples/dns-flood-victoria.lua 0 10.0.0.1 16000000 10000

[Device: id=0] Sent 13710082176 packets, current rate 4.51 Mpps, 3246.01 MBit/s, 3967.35 MBit/s wire rate.

[Device: id=0] Sent 13714591360 packets, current rate 4.51 Mpps, 3246.53 MBit/s, 3967.98 MBit/s wire rate.

[Device: id=0] Sent 13719099520 packets, current rate 4.51 Mpps, 3245.79 MBit/s, 3967.08 MBit/s wire rate.

https://docs.f5net.com/display/PDDESIGN/DNS+DDoS+HW+Acceleration+-+Validation

```
sys fpga firmware-config {
    type l7-intelligent-fpga
```

```
}
ltm profile dns /Common/dns_fpga {
    app-service none
    enable-hardware-query-validation yes
    enable-hardware-response-cache yes
}
ltm virtual /Common/vs_dns_10g {
    destination /Common/10.3.3.249:53
    ip-protocol udp
    mask 255.255.255.255
    profiles {
        /Common/dns_fpga { }
        /Common/udp_immediate { }
    }
    rules {
        /Common/dns_responder
    }
    source 0.0.0.0/0
    translate-address enabled
    translate-port enabled
}


top - 14:51:05 up  3:30,  1 user,  load average: 0.12, 0.05, 0.01
Tasks: 771 total,   1 running, 770 sleeping,   0 stopped,   0 zombie
Cpu(s):  4.2%us,  0.5%sy,  0.0%ni, 95.2%id,  0.0%wa,  0.1%hi,  0.0%si,  0.0%st
Mem:  66080272k total, 63094488k used,  2985784k free,    61152k buffers
Swap:  5242876k total,        0k used,  5242876k free,  1352852k cached
  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 6428 root       1 -19 58.4g 151m 122m S 12.6  0.2   3:19.62 tmm.0 <================TMM CPU usage drops significantly
 6435 root       1 -19 58.4g 151m 122m S 11.3  0.2   2:42.67 tmm.4
 6432 root       1 -19 58.4g 151m 122m S 10.9  0.2   2:44.57 tmm.1
 6433 root       1 -19 58.4g 151m 122m S 10.9  0.2   2:42.78 tmm.2
 6434 root       1 -19 58.4g 151m 122m S 10.9  0.2   2:40.69 tmm.3
 6436 root       1 -19 58.4g 151m 122m S 10.9  0.2   2:41.53 tmm.5
 6437 root       1 -19 58.4g 151m 122m S 10.9  0.2   2:42.68 tmm.6
 6438 root       1 -19 58.4g 151m 122m S 10.9  0.2   2:40.92 tmm.7
 6439 root       1 -19 58.4g 151m 122m S 10.9  0.2   2:41.87 tmm.8
 6440 root       1 -19 58.4g 151m 122m S 10.6  0.2   2:41.49 tmm.9
28351 root     -91   0 97592  81m  31m S  2.0  0.1   7:00.29 bcmINTR
28589 root      20   0 97592  81m  31m S  2.0  0.1   5:43.36 bcmCNTR.0
```

profile_dns_stat

name vs_name queries drops hw_malformed hw_inspected hw_cache_lookups hw_cache_responses

--- --- --- --- --- --- --- ---

/Common/dns_fpga /Common/vs_dns_10g 6981 0 0 29727032 29726590 29720435

3, SYN flooding without hardware acceleration

#build/MoonGen examples/l3-tcp-syn-flood.lua 0 10.0.0.1 16000000 10000

[Device: id=0] Sent 7061632 packets, current rate 7.06 Mpps, 3615.47 MBit/s, 4745.31 MBit/s wire rate.

ltm profile fastl4 /Common/fl4_fpga {

app-service none

defaults-from /Common/fastL4

hardware-syn-cookie disabled

pva-offload-dynamic disabled

software-syn-cookie enabled

}

top - 10:53:51 up 42 min, 1 user, load average: 0.24, 0.23, 0.65

Tasks: 769 total, 10 running, 759 sleeping, 0 stopped, 0 zombie

Cpu(s): 35.4%us, 1.7%sy, 0.1%ni, 62.9%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st

Mem: 66080376k total, 62740700k used, 3339676k free, 45784k buffers

Swap: 5242872k total, 0k used, 5242872k free, 1199508k cached

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND

19290 root 1 -19 57.9g 145m 123m R 71.5 0.2 0:14.38 tmm.0 -T 10 --tmid **<==========HIGH CPU**

19291 root 1 -19 57.9g 145m 123m R 70.2 0.2 0:14.53 tmm.0 -T 10 --tmid

19293 root 1 -19 57.9g 145m 123m S 70.2 0.2 0:14.37 tmm.0 -T 10 --tmid

19267 root 1 -19 57.9g 145m 123m R 69.8 0.2 0:30.32 tmm.0 -T 10 --tmid

19292 root 1 -19 57.9g 145m 123m R 69.8 0.2 0:14.38 tmm.0 -T 10 --tmid

19298 root 1 -19 57.9g 145m 123m R 69.8 0.2 0:14.72 tmm.0 -T 10 --tmid

19295 root 1 -19 57.9g 145m 123m R 69.5 0.2 0:14.73 tmm.0 -T 10 --tmid

19296 root 1 -19 57.9g 145m 123m R 69.5 0.2 0:14.03 tmm.0 -T 10 --tmid

19297 root 1 -19 57.9g 145m 123m R 69.2 0.2 0:14.14 tmm.0 -T 10 --tmid

19294 root 1 -19 57.9g 145m 123m R 65.2 0.2 0:13.31 tmm.0 -T 10 --tmid

4 SYN flooding with HW acceleration

#build/MoonGen examples/l3-tcp-syn-flood.lua 0 10.0.0.1 16000000 10000

[Device: id=0] Sent 7061632 packets, current rate 7.06 Mpps, 3615.47 MBit/s, 4745.31 MBit/s wire rate.

ltm profile fastl4 /Common/fl4_fpga {

app-service none

```
    defaults-from /Common/fastL4
    hardware-syn-cookie enabled
    pva-offload-dynamic enabled
    software-syn-cookie enabled
}
```

```
top - 10:50:08 up 38 min,  1 user,  load average: 0.06, 0.36, 0.81
Tasks: 769 total,   1 running, 768 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.8%us,  0.2%sy,  0.0%ni, 98.5%id,  0.5%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:  66080376k total, 62740552k used,  3339824k free,    45324k buffers
Swap:  5242872k total,        0k used,  5242872k free,  1199492k cached
  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
19267 root      1 -19 57.9g 145m 123m S  3.6  0.2   0:11.87 tmm <===========CPU usage drops significantly with hardware acceleration
19293 root      1 -19 57.9g 145m 123m S  1.3  0.2   0:01.72 tmm
19296 root      1 -19 57.9g 145m 123m S  1.3  0.2   0:01.36 tmm
19297 root      1 -19 57.9g 145m 123m S  1.3  0.2   0:01.37 tmm
19290 root      1 -19 57.9g 145m 123m S  1.0  0.2   0:01.38 tmm
19292 root      1 -19 57.9g 145m 123m S  1.0  0.2   0:01.73 tmm
19294 root      1 -19 57.9g 145m 123m S  1.0  0.2   0:01.35 tmm
19295 root      1 -19 57.9g 145m 123m S  1.0  0.2   0:02.12 tmm
19298 root      1 -19 57.9g 145m 123m S  1.0  0.2   0:02.11 tmm
19291 root      1 -19 57.9g 145m 123m S  0.7  0.2   0:01.73 tmm
```

5, 10M concurrent HTTP connection

#epwget 10.3.3.249/ 160000000 -N 16 –c 10000000

```
[CPU 0] dpdk0 flows: 625000, RX:   96382(pps) (err:     0),  0.10(Gbps), TX:  413888(pps),  0.64(Gbps)
[CPU 1] dpdk0 flows: 625000, RX:  101025(pps) (err:     0),  0.10(Gbps), TX:  398592(pps),  0.61(Gbps)
[CPU 2] dpdk0 flows: 625000, RX:  106882(pps) (err:     0),  0.11(Gbps), TX:  418432(pps),  0.64(Gbps)
[CPU 3] dpdk0 flows: 625000, RX:  101497(pps) (err:     0),  0.10(Gbps), TX:  405952(pps),  0.62(Gbps)
[CPU 4] dpdk0 flows: 625000, RX:  107375(pps) (err:     0),  0.11(Gbps), TX:  427008(pps),  0.66(Gbps)
[CPU 5] dpdk0 flows: 625000, RX:   96012(pps) (err:     0),  0.10(Gbps), TX:  404352(pps),  0.62(Gbps)
[CPU 6] dpdk0 flows: 625000, RX:  100834(pps) (err:     0),  0.10(Gbps), TX:  405504(pps),  0.62(Gbps)
[CPU 7] dpdk0 flows: 625000, RX:  102572(pps) (err:     0),  0.11(Gbps), TX:  401024(pps),  0.62(Gbps)
[CPU 8] dpdk0 flows: 635366, RX:  111319(pps) (err:     0),  0.12(Gbps), TX:  410880(pps),  0.63(Gbps)
[CPU 9] dpdk0 flows: 625000, RX:  102179(pps) (err:     0),  0.11(Gbps), TX:  391104(pps),  0.60(Gbps)
[CPU10] dpdk0 flows: 625000, RX:   98014(pps) (err:     0),  0.10(Gbps), TX:  408320(pps),  0.63(Gbps)
[CPU11] dpdk0 flows: 625000, RX:  102712(pps) (err:     0),  0.11(Gbps), TX:  398976(pps),  0.61(Gbps)
[CPU12] dpdk0 flows: 625000, RX:  105891(pps) (err:     0),  0.11(Gbps), TX:  415616(pps),  0.64(Gbps)
```

[CPU13] dpdk0 flows: 625000, RX: 97728(pps) (err: 0), 0.10(Gbps), TX: 390592(pps), 0.60(Gbps)

[CPU14] dpdk0 flows: 625001, RX: 100570(pps) (err: 0), 0.10(Gbps), TX: 407872(pps), 0.63(Gbps)

[CPU15] dpdk0 flows: 625000, RX: 103412(pps) (err: 0), 0.11(Gbps), TX: 391296(pps), 0.60(Gbps)

[ALL] dpdk0 flows: 10010366, RX: 1634404(pps) (err: 0), 1.69(Gbps), TX: 6489408(pps), 9.96(Gbps) <================

top - 15:25:26 up 23:57, 1 user, load average: 0.16, 0.33, 0.43

Tasks: 778 total, 17 running, 761 sleeping, 0 stopped, 0 zombie

Cpu(s): 45.1%us, 30.6%sy, 0.0%ni, 24.3%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st

Mem: 66080376k total, 62855960k used, 3224416k free, 136316k buffers

Swap: 5242872k total, 0k used, 5242872k free, 1182216k cached

  PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND

17283 root 1 -19 57.9g 145m 123m R 94.1 0.2 1322:36 tmm.0 -T 10 --tmid <================

17286 root 1 -19 57.9g 145m 123m R 94.1 0.2 1322:37 tmm.0 -T 10 --tmid

17281 root 1 -19 57.9g 145m 123m R 93.8 0.2 1322:39 tmm.0 -T 10 --tmid

17284 root 1 -19 57.9g 145m 123m R 93.8 0.2 1322:37 tmm.0 -T 10 --tmid

17282 root 1 -19 57.9g 145m 123m R 93.4 0.2 1322:37 tmm.0 -T 10 --tmid

17287 root 1 -19 57.9g 145m 123m R 93.4 0.2 1322:37 tmm.0 -T 10 --tmid

17288 root 1 -19 57.9g 145m 123m R 93.4 0.2 1322:36 tmm.0 -T 10 --tmid

17289 root 1 -19 57.9g 145m 123m R 93.4 0.2 1322:36 tmm.0 -T 10 --tmid

17043 root 1 -19 57.9g 145m 123m R 92.8 0.2 1325:48 tmm.0 -T 10 --tmid

17285 root 1 -19 57.9g 145m 123m R 92.1 0.2 1322:37 tmm.0 -T 10 --tmid

31507 root RT 0 0 0 0 R 32.0 0.0 0:00.97 [enforcer/19]

31508 root RT 0 0 0 0 R 32.0 0.0 0:00.97 [enforcer/13]

31509 root RT 0 0 0 0 R 32.0 0.0 0:00.97 [enforcer/15]

31510 root RT 0 0 0 0 S 31.7 0.0 0:00.96 [enforcer/9]

31511 root RT 0 0 0 0 S 31.7 0.0 0:00.96 [enforcer/7]

31512 root RT 0 0 0 0 R 31.4 0.0 0:00.95 [enforcer/3]

31515 root RT 0 0 0 0 S 16.8 0.0 0:00.51 [enforcer/1]

[root@localhost:/S1-green-P:Active:Standalone] config # tail -f /var/log/ltm

Nov 4 15:25:29 slot1/bigip1 warning tmm7[17043]: 011e0003:4: Aggressive mode sweeper: /Common/default-eviction-policy (70000000002d6) (global memory) 9864 Connections killed

Nov 4 15:25:29 slot1/bigip1 warning tmm7[17043]: 011e0002:4: sweeper_policy_bind_deactivation_update: Aggressive mode /Common/default-eviction-policy deactivated (70000000002d6) (global memory). (12793204/15051776 pages)

Nov 4 15:25:29 slot1/bigip1 warning tmm6[17043]: 011e0003:4: Aggressive mode sweeper: /Common/default-eviction-policy (60000000002d2) (global memory) 10122 Connections killed

Nov 4 15:25:29 slot1/bigip1 warning tmm6[17043]: 011e0002:4: sweeper_policy_bind_deactivation_update: Aggressive mode /Common/default-eviction-policy deactivated (60000000002d2) (global memory). (12792703/15051776 pages)

Nov 4 15:25:29 slot1/bigip1 warning tmm3[17043]: 011e0003:4: Aggressive mode sweeper: /Common/default-eviction-policy (30000000002de) (global memory) 10877 Connections killed

Nov 4 15:25:29 slot1/bigip1 warning tmm3[17043]: 011e0002:4: sweeper_policy_bind_deactivation_update: Aggressive mode /Common/default-eviction-policy deactivated (30000000002de) (global memory). (12787088/15051776 pages)

Nov 4 15:25:29 slot1/bigip1 warning tmm4[17043]: 011e0003:4: Aggressive mode sweeper: /Common/default-eviction-policy

(40000000002c2) (global memory) 10306 Connections killed

Nov 4 15:25:29 slot1/bigip1 warning tmm4[17043]: 011e0002:4: sweeper_policy_bind_deactivation_update: Aggressive mode /Common/default-eviction-policy deactivated (40000000002c2) (global memory). (12787088/15051776 pages)

Every 1.0s: tmsh show ltm virtual vs_http_10g Wed Nov 4 15:27:15 2015

Availability : unknown

State : enabled

Reason : The children pool member(s) either don't have service check

ing enabled, or service check results are not available yet

CMP : enabled

CMP Mode : all-cpus

Destination : 10.3.3.249:80

PVA Acceleration : none

Traffic	ClientSide	Ephemeral	General
Bits In	329.8G	0	-
Bits Out	90.4G	0	-
Packets In	287.6M	0	-
Packets Out	150.2M	0	-
Current Connections	6.1M	0	- <==========
Maximum Connections	6.7M	0	-
Total Connections	39.8M	0	-

mTCP perf top output ~70% cycles in Userspace

Samples: 1M of event 'cycles', Event count (approx.): 441906428558

8.25%	epwget	[.] SendTCPPacket
7.93%	[kernel]	[k] _raw_spin_lock
7.16%	epwget	[.] GetRSSCPUCore
7.15%	epwget	[.] IPOutput
4.26%	libc-2.19.so	[.] memset
4.10%	epwget	[.] ixgbe_xmit_pkts
3.62%	[kernel]	[k] clear_page_c
3.26%	epwget	[.] WriteTCPControlList
3.24%	[vdso]	[.] 0x0000000000000cf9
2.95%	epwget	[.] AddtoControlList
2.70%	epwget	[.] MTCPRunThread
2.66%	epwget	[.] HandleRTO
2.51%	epwget	[.] CheckRtmTimeout
2.10%	libpthread-2.19.so	[.] pthread_mutex_unlock
1.83%	epwget	[.] dpdk_send_pkts
1.68%	epwget	[.] HTInsert

1.65% epwget [.] CreateTCPStream

1.42% epwget [.] MPAllocateChunk

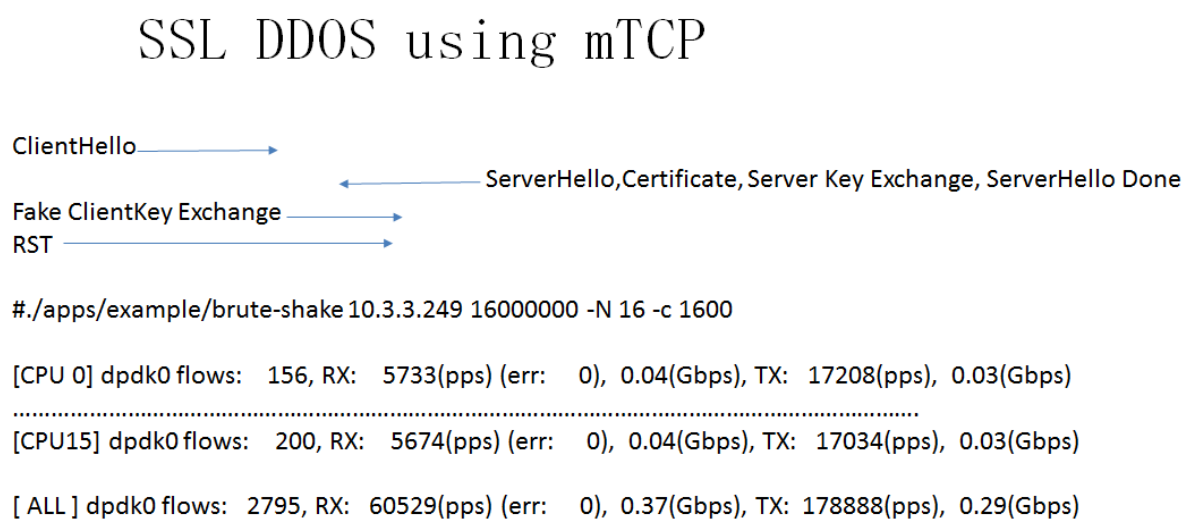
1.29% epwget [.] TCPCalcChecksum

1.24% epwget [.] dpdk_recv_pkts

1.20% epwget [.] mtcp_getsockopt

1.12% epwget [.] rx_recv_pkts

6, SSL DDOS test using mTCP



#./apps/example/brute-shake 10.3.3.249 16000000 -N 16 -c 1600

[CPU 0] dpdk0 flows: 156, RX: 5733(pps) (err: 0), 0.04(Gbps), TX: 17208(pps), 0.03(Gbps)

.....

[CPU15] dpdk0 flows: 200, RX: 5674(pps) (err: 0), 0.04(Gbps), TX: 17034(pps), 0.03(Gbps)

[ALL] dpdk0 flows: 2795, RX: 60529(pps) (err: 0), 0.37(Gbps), TX: 178888(pps), 0.29(Gbps)

top - 09:10:21 up 22:58, 1 user, load average: 10.45, 4.43, 1.67

Tasks: 782 total, 19 running, 763 sleeping, 0 stopped, 0 zombie

Cpu(s): 50.6%us, 40.1%sy, 0.1%ni, 9.1%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st

Mem: 66080376k total, 62923192k used, 3157184k free, 138624k buffers

Swap: 5242872k total, 0k used, 5242872k free, 1259132k cached

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND

21480 root 1 -19 57.9g 145m 123m R 100.0 0.2 81:24.41 tmm <==========

21503 root 1 -19 57.9g 145m 123m R 100.0 0.2 48:05.30 tmm

21504 root 1 -19 57.9g 145m 123m R 100.0 0.2 47:23.12 tmm

21505 root 1 -19 57.9g 145m 123m R 100.0 0.2 47:06.70 tmm

21506 root 1 -19 57.9g 145m 123m R 100.0 0.2 46:55.21 tmm

21507 root 1 -19 57.9g 145m 123m R 100.0 0.2 46:12.27 tmm

21508 root 1 -19 57.9g 145m 123m R 100.0 0.2 46:56.27 tmm

21509 root 1 -19 57.9g 145m 123m R 100.0 0.2 47:01.32 tmm

21510 root 1 -19 57.9g 145m 123m R 100.0 0.2 46:48.54 tmm

21511 root 1 -19 57.9g 145m 123m R 100.0 0.2 47:06.64 tmm

1675 root RT 0 0 0 0 R 80.6 0.0 2:07.06 enforcer/9

1669 root RT 0 0 0 0 R 80.2 0.0 2:07.03 enforcer/1

1670 root RT 0 0 0 0 R 80.2 0.0 2:07.03 enforcer/15

1673 root RT 0 0 0 0 R 80.2 0.0 2:07.03 enforcer/3

1677 root RT 0 0 0 0 R 80.2 0.0 2:07.02 enforcer/13

1671 root RT 0 0 0 0 S 79.9 0.0 2:07.04 enforcer/19

1672 root RT 0 0 0 0 R 79.9 0.0 2:07.02 enforcer/5

profile_clientssl_stat

name vs_name cur_conns max_conns encrypted_bytes_in encrypted_bytes_out sess_cache_cur_entries

---

/Common/clientssl /Common/vs_https 16235 25533 31164613471 66885162672 2621439

ssess_cache_lookups sess_cache_overflows fatal_alerts fully_ hw_accelerated_conns partially_hw_accelerated_conns

---

47843605 42545150 55576 0 0

SR 1-3391074511/#1-1K40BX7 example

```
root@pktgen-template:/home/admin/mtcp# ./apps/example/brute-shake 10.1.72.69 16000000 -N 8 -c 12000
```

7, ApacheBench(ab) mTCP port https test to Victoria

#ab -n 16000 -N 16 -c 8000 -L 64 https://10.3.3.249/

Loading mtcp configuration from : /etc/mtcp/config/mtcp.conf

Loading interface setting

EAL: Detected lcore 0 as core 0 on socket 0

.....

Checking link statusdone

Port 0 Link Up - speed 10000 Mbps - full-duplex

Benchmarking 10.3.3.249 (be patient)

CPU6 connecting to port 443

CPU7 connecting to port 443

CPU8 connecting to port 443

CPU9 connecting to port 443

CPU10 connecting to port 443

CPU5 connecting to port 443

CPU11 connecting to port 443

CPU12 connecting to port 443

CPU13 connecting to port 443

CPU14 connecting to port 443

CPU15 connecting to port 443

CPU4 connecting to port 443

CPU2 connecting to port 443

CPU3 connecting to port 443

CPU1 connecting to port 443

CPU0 connecting to port 443

.....

[ALL] dpdk0 flows: 5016, RX: 9651(pps) (err: 0), 0.04(Gbps), TX: 14784(pps), 0.02(Gbps)

Ltm::Virtual Server: vs_https

CMP : enabled

 CMP Mode : all-cpus

 Destination : 10.3.3.249:443

 PVA Acceleration : none

Traffic ClientSide Ephemeral General

 Bits In 49.2G 0 -

 Bits Out 71.0G 0 -

 Packets In 47.1M 0 -

 Packets Out 30.4M 0 -

 Current Connections 6.3K 0 -

 Maximum Connections 146.0K 0 -

 Total Connections 4.3M 0 -

 PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND

12864 root 1 -19 57.9g 145m 123m S 5.0 0.2 53:09.44 tmm

13087 root 1 -19 57.9g 145m 123m S 3.0 0.2 14:01.00 tmm

13088 root 1 -19 57.9g 145m 123m S 3.0 0.2 13:32.00 tmm

13091 root 1 -19 57.9g 145m 123m S 3.0 0.2 13:25.59 tmm

13093 root 1 -19 57.9g 145m 123m S 3.0 0.2 13:34.57 tmm

13094 root 1 -19 57.9g 145m 123m S 3.0 0.2 13:46.66 tmm

13086 root 1 -19 57.9g 145m 123m S 2.6 0.2 14:09.38 tmm

13089 root 1 -19 57.9g 145m 123m S 2.6 0.2 13:42.05 tmm

13090 root 1 -19 57.9g 145m 123m S 2.6 0.2 13:47.88 tmm

13092 root 1 -19 57.9g 145m 123m S 2.3 0.2 13:40.11 tmm

8, ApacheBench(ab) mTCP port https test to BIGIP VE (KVM)

#ab -n 1000000 -c 8000 -N 8 -L 64 https://10.9.3.6/

Checking link status.....done

Port 0 Link Up - speed 1000 Mbps - full-duplex

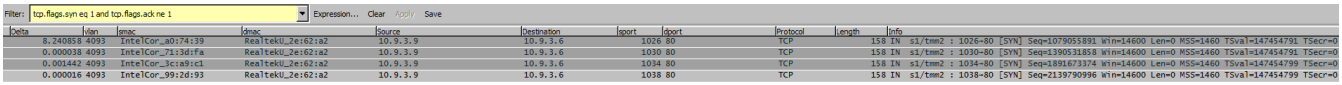
Benchmarking 10.9.3.6 (be patient)

```
CPU6 connecting to port 443
CPU7 connecting to port 443
CPU5 connecting to port 443
CPU4 connecting to port 443
CPU3 connecting to port 443
CPU2 connecting to port 443
CPU1 connecting to port 443
CPU0 connecting to port 443
[ ALL ] dpdk0 flows:   8000, RX:   13443(pps) (err:     0),  0.01(Gbps), TX:   13953(pps),  0.01(Gbps)

top - 13:12:22 up 4 min,  1 user,  load average: 3.34, 2.01, 0.82
Tasks: 395 total,   4 running, 391 sleeping,   0 stopped,   0 zombie
Cpu(s): 13.2%us,  6.5%sy,  0.0%ni, 64.5%id, 15.6%wa,  0.0%hi,  0.1%si,  0.0%st
Mem:  14403128k total, 14060912k used,   342216k free,    22400k buffers
Swap:  1048568k total,        0k used,  1048568k free,   863780k cached
  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  P COMMAND
13954 root      RT   0 12.0g 124m 104m R 92.4  0.9   0:27.17 0 tmm.0 -T 4 --tmid 0 --npus 4 --platform Z100 -m -s 12088
14125 root      RT   0 12.0g 124m 104m R 92.0  0.9   0:13.28 1 tmm.0 -T 4 --tmid 0 --npus 4 --platform Z100 -m -s 12088
14126 root      RT   0 12.0g 124m 104m S 92.0  0.9   0:12.36 2 tmm.0 -T 4 --tmid 0 --npus 4 --platform Z100 -m -s 12088
14127 root      RT   0 12.0g 124m 104m S 92.0  0.9   0:13.15 3 tmm.0 -T 4 --tmid 0 --npus 4 --platform Z100 -m -s 12088
-------------------------------------------------------------------------------
Ltm::Virtual Server: vs_https
-------------------------------------------------------------------------------
Status
Traffic                     ClientSide  Ephemeral  General
  Bits In                     428.8M         0        -
  Bits Out                    786.2M         0        -
  Packets In                  505.9K         0        -
  Packets Out                 423.1K         0        -
  Current Connections           9.4K         0        -
  Maximum Connections          12.9K         0        -
```

9 Generate TCP/HTTP connection with random src MAC

http://sourceforge.net/p/curl-loader/mailman/message/33614941/

```
root@pktgen:/usr/src/mtcp# git diff mtcp/src/eth_out.c
diff --git a/mtcp/src/eth_out.c b/mtcp/src/eth_out.c
index 7fd1097..b064e9c 100644
--- a/mtcp/src/eth_out.c
+++ b/mtcp/src/eth_out.c
@@ -181,6 +181,44 @@ FlushSendChunkBuf(mtcp_manager_t mtcp, int nif)
 }
 #endif /* E_PSIO */
 /*----------------------------------------------------------------------------*/
+
+/**
+ * Get a pseudo-random value.
+ *
+ * This function generates pseudo-random numbers using the linear
+ * congruential algorithm and 48-bit integer arithmetic, called twice
+ * to generate a 64-bit value.
+ *
+ * @return
+ *   A pseudo-random value between 0 and (1<<64)-1.
+ */
+static inline uint64_t
+rte_rand(void)
+{
+        uint64_t val;
+        val = lrand48();
+        val <<= 32;
+        val += lrand48();
+        return val;
+}
+
+
+static void
+generate_random_mac_addr(struct ethhdr *mac_addr)
+{
+        uint64_t random;
+
+        /* Set Organizationally Unique Identifier (OUI) prefix. */
+        mac_addr->h_source[0] = 0xa0;
+        mac_addr->h_source[1] = 0x36;
+        mac_addr->h_source[2] = 0x9f;
+        /* Force indication of locally assigned MAC address. */
+//        mac_addr->h_source[0] |= ETHER_LOCAL_ADMIN_ADDR;
+        /* Generate the last 3 bytes of the MAC address with a random number. */
+        random = rte_rand();
+        memcpy(&mac_addr->h_source[3], &random, 3);
+}
+
 uint8_t *
 EthernetOutput(struct mtcp_manager *mtcp, uint16_t h_proto,
                int nif, unsigned char* dst_haddr, uint16_t iplen)
@@ -213,8 +251,9 @@ EthernetOutput(struct mtcp_manager *mtcp, uint16_t h_proto,
 #endif

        ethh = (struct ethhdr *)buf;
+       generate_random_mac_addr(ethh);
        for (i = 0; i < ETH_ALEN; i++) {
-               ethh->h_source[i] = CONFIG.eths[nif].haddr[i];
+               //ethh->h_source[i] = CONFIG.eths[nif].haddr[i];
                ethh->h_dest[i] = dst_haddr[i];
        }
        ethh->h_proto = htons(h_proto);
```



10 mTCP hack to generate TCP 3WHS final ACK with 1 byte payload

SR 1-2235587071/ IXIA were sending 1byte payload in TCP 3WHS final ACK to cause BIGIP memory sweeper kicking in, unable to reach advertised platform tcp connection spec numbers, mTCP patch below to simulate that

```
oot@pktgen:/home/dpdk/mtcp# git diff mtcp/src/tcp_out.c
iff --git a/mtcp/src/tcp_out.c b/mtcp/src/tcp_out.c
ndex e346958..edebe19 100644
-- a/mtcp/src/tcp_out.c
++ b/mtcp/src/tcp_out.c
@ -432,6 +432,8 @@ SendControlPacket(mtcp_manager_t mtcp, tcp_stream *cur_stream, uint32_t cur_ts)
{
        struct tcp_send_vars *sndvar = cur_stream->sndvar;
        int ret = 0;
         char payload = 'y';
         uint16_t payloadlen = 1;

        if (cur_stream->state == TCP_ST_SYN_SENT) {
                /* Send SYN here */
@ -445,7 +447,8 @@ SendControlPacket(mtcp_manager_t mtcp, tcp_stream *cur_stream, uint32_t cur_ts)

        } else if (cur_stream->state == TCP_ST_ESTABLISHED) {
                /* Send ACK here */
                ret = SendTCPPacket(mtcp, cur_stream, cur_ts, TCP_FLAG_ACK, NULL, 0);
//              ret = SendTCPPacket(mtcp, cur_stream, cur_ts, TCP_FLAG_ACK, NULL, 0);
                ret = SendTCPPacket(mtcp, cur_stream, cur_ts, TCP_FLAG_ACK, (uint8_t *)&payload, payloadlen);
```

Jul 18 14:08:47 slot1/cluster1 warning tmm3[22490]: 011e0002:4: sweeper_policy_bind_deactivation_update: Aggressive mode /Common/default-eviction-policy deactivated (3000000000001) (global memory). (1524040/1793024 pages)

Jul 18 14:08:47 slot1/cluster1 warning tmm[22490]: 011e0003:4: Aggressive mode sweeper: /Common/default-eviction-policy (1) (global memory) 20244 Connections killed

Jul 18 14:08:47 slot1/cluster1 warning tmm[22490]: 011e0002:4: sweeper_policy_bind_deactivation_update: Aggressive mode /Common/default-eviction-policy deactivated (1) (global memory). (1524010/1793024 pages)

filter	3.2G	3.6G	1
umem	1.2G	10.2G	1
xdata	4.0G	4.6G	2048

11 ICMP ping flooding to BIGIP VE

build/MoonGen examples/icmp-flood.lua 0 10.0.0.1 16000000 10000

top - 12:10:54 up 1:55, 1 user, load average: 0.24, 0.06, 0.02

Tasks: 381 total, 2 running, 379 sleeping, 0 stopped, 0 zombie

Cpu0 : 16.2%us, 11.3%sy, 0.0%ni, 13.1%id, 0.0%wa, 0.3%hi, 59.1%si, 0.0%st

Cpu1 : 2.1%us, 2.4%sy, 0.0%ni, 94.6%id, 0.0%wa, 0.0%hi, 0.3%si, 0.6%st

Cpu2 : 3.5%us, 3.2%sy, 0.0%ni, 90.6%id, 0.0%wa, 0.0%hi, 1.8%si, 0.9%st

Cpu3 : 1.2%us, 1.8%sy, 0.3%ni, 96.0%id, 0.0%wa, 0.0%hi, 0.3%si, 0.3%st

Mem: 14403128k total, 14267112k used, 136016k free, 22252k buffers

Swap: 1048568k total, 1224k used, 1047344k free, 571908k cached

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ P COMMAND

```
2889 root     RT   0 12.0g 124m 104m S 154.7  0.9   3:58.44 0 tmm.0 -T 4 --tmid 0 --npus 4 --platform Z100 -m -s 12088
3054 root     RT   0 12.0g 124m 104m S  8.8  0.9   2:48.53 3 tmm.0 -T 4 --tmid 0 --npus 4 --platform Z100 -m -s 12088
3053 root     RT   0 12.0g 124m 104m S  8.5  0.9   2:12.29 2 tmm.0 -T 4 --tmid 0 --npus 4 --platform Z100 -m -s 12088
3052 root     RT   0 12.0g 124m 104m R  7.6  0.9   2:06.63 1 tmm.0 -T 4 --tmid 0 --npus 4 --platform Z100 -m -s 12088
```

Technical tips for load generation

1 mTCP thread pre-allocate memory pools for TCP send and receive buffer from configured maximum number of buffers.

when the load generator has limited memory, it is recommended to reduce the size of TCP send and receive buffer and number of buffers in application configuration file.

also configure the BIGIP DUT to respond small packet size (< 64 bytes) because large response payload size would trigger mTCP payload merge length error

for example: setting TCP receive and send buffer in epwget.conf

Receive buffer size of sockets

rcvbuf = 1024

Send buffer size of sockets

sndbuf = 1024

2 mTCP default receive window is 64 bytes when window scale not negotiated, TMOS 11.4.0 and later disabled window scale unless increase send/receive buffer > 65535

This result in BIGIP send 64 bytes to mTCP in each segment while doing load testing, cause BIGIP buffers data in longer period and connection stays active longer period

see SR 1-1526064281/. Linux client would not notice this difference when window scale not negotiated because Linux client advertise receive window > MSS

3 When BIGIP syncookie activated, there are tons of TCP retransmission between mTCP and BIGIP, filed an issue with mTCP https://github.com/eunyoung14/mtcp/issues/23

also BIGIP bug is filed in Bug 554761

4 Victoria2 DNS DDOS Hardware validation skips # of question check, Bug 558251 DNS query with multiple question RR marked as malformed in SW and not in HW, without HW validation of # of DNS question count, TMM CPU usage tops at 100% under the DNS flooding test

5 BIGIP v12.0.0 SW syncookies and windowscaling will cause 3WHS to fail on L7 VIP, this would also affect tcp/http load test Bug 555020

6 SSL performance test with modified multithread apachebench with SSL support SR 1-1602453049/ (use AES256-SHA to get better performance number, not TLS_DHE_RSA_WITH_*)

7 Bug 486688 SYN cookie learning for network and/or wildcard port virtual servers : test code can be found here http://ickernel.blogspot.com/2016/01/syn-flood-network-virtual-server-in.html

8 ID 621988 SR 1-2402757471 same src ipport udp packet drops on udp virtual with mirror

9 DPDK + mTCP on VMware ESXi VM and Linux perf to resolve SR 1-2653101191

Development:

1. **Ported mTCP/DPDK to VMware ESXi VM (patch https://docs.f5net.com/download/attachments/290265627/ESXi-VM-mTCP.txt?api=v2)**

mTCP/DPDK in VM (vli_ubuntu_1604_DPDK):

(Note: Ubuntu 1604 changes mTCP DPDK netdev name dpdk0 once igb_uio bounded, to disable it, add GRUB_CMDLINE_LINUX="net.ifnames=0" and run update-grub)

Figure 3. Standard para-virtual device connectivity

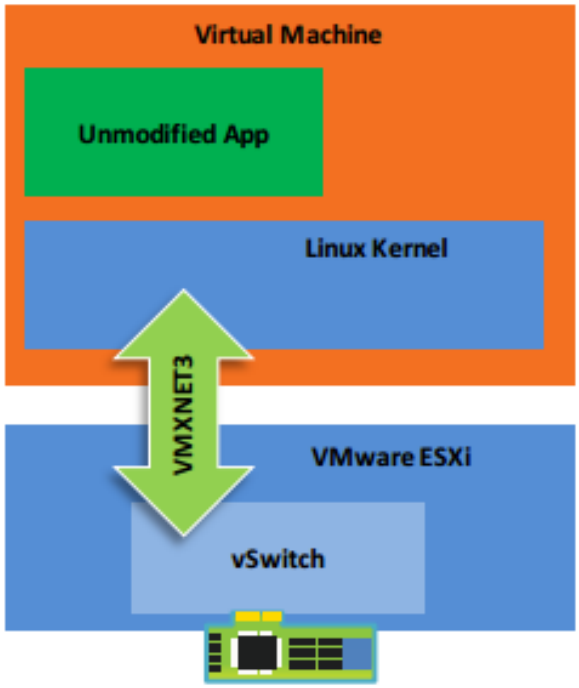
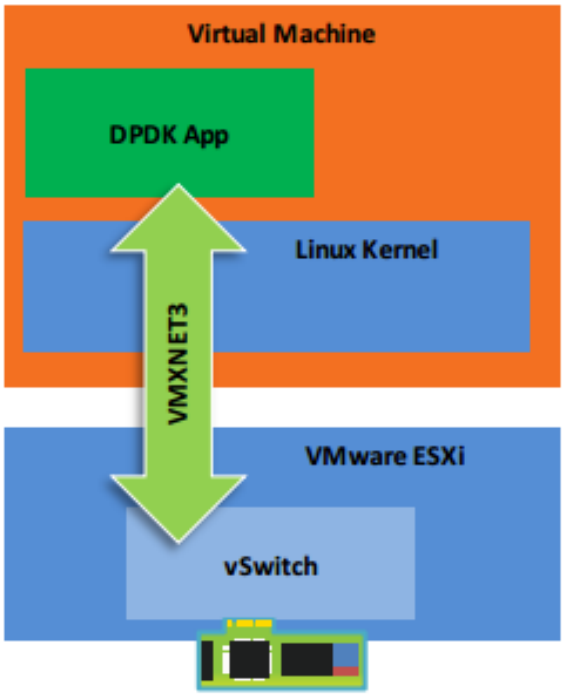


Figure 4. Intel DPDK para-virtual device connectivity



root@ubuntu-dpdk:/home/dpdk/mtcp# ./apps/example/epwget 10.169.72.88 1600000000 -N 8 -c 1600000

Loading mtcp configuration from : /etc/mtcp/config/epwget.conf
Loading interface setting

EAL: PCI device 0000:03:00.0 on NUMA socket -1

EAL: probe driver: 15ad:7b0 rte_vmxnet3_pmd

EAL: Not managed by a supported kernel driver, skipped

EAL: PCI device 0000:0b:00.0 on NUMA socket -1

EAL: probe driver: 15ad:7b0 rte_vmxnet3_pmd

Interface name: dpdk0

Configurations:

Number of CPU cores available: 8

Number of CPU cores to use: 8

Number of source ip to use: 32

Maximum number of concurrency per core: 1000000

Maximum number of preallocated buffers per core: 1000000

Receive buffer size: 2048

Send buffer size: 2048

TCP timeout seconds: 30

TCP timewait seconds: 0

NICs to print statistics: dpdk0

Interfaces:

name: dpdk0, ifindex: 0, hwaddr: 00:50:56:86:43:CF, ipaddr: 10.0.0.1, netmask: 255.255.255.0

Number of NIC queues: 8

[CPU 0] dpdk0 flows: 200194, RX: 30694(pps) (err: 0), 0.03(Gbps), TX: 61504(pps), 0.09(Gbps)

[CPU 1] dpdk0 flows: 200114, RX: 30832(pps) (err: 0), 0.02(Gbps), TX: 61760(pps), 0.09(Gbps)

[CPU 2] dpdk0 flows: 200059, RX: 31250(pps) (err: 0), 0.03(Gbps), TX: 61696(pps), 0.09(Gbps)

[CPU 3] dpdk0 flows: 200030, RX: 31149(pps) (err: 0), 0.02(Gbps), TX: 61632(pps), 0.09(Gbps)

[CPU 4] dpdk0 flows: 200010, RX: 31682(pps) (err: 0), 0.03(Gbps), TX: 61632(pps), 0.09(Gbps)

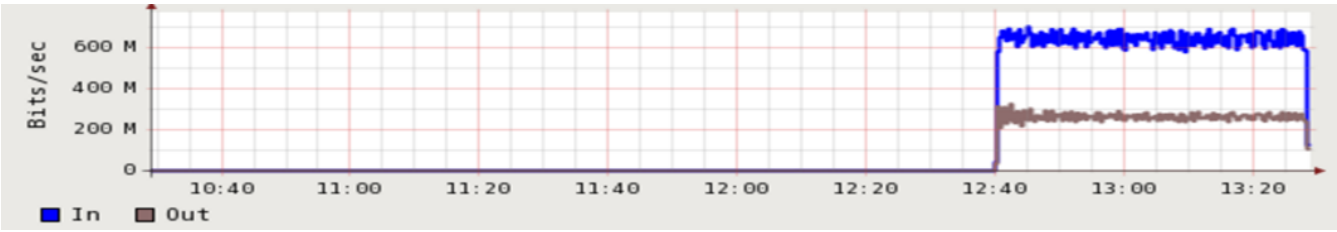
[CPU 5] dpdk0 flows: 200005, RX: 31424(pps) (err: 0), 0.03(Gbps), TX: 61696(pps), 0.09(Gbps)

[CPU 6] dpdk0 flows: 200000, RX: 32209(pps) (err: 0), 0.03(Gbps), TX: 61824(pps), 0.09(Gbps)

[CPU 7] dpdk0 flows: 200001, RX: 45255(pps) (err: 0), 0.04(Gbps), TX: 61632(pps), 0.09(Gbps)

[ALL] dpdk0 flows: 1600413, RX: 264495(pps) (err: 0), 0.22(Gbps), TX: 493376(pps), 0.74(Gbps)

BIGIP 10200 -



2 Port MoonGen/DPDK to VMware ESXi VM (MoonGen DPDK is not synced with Intel DPDK, running MoonGen on VM can be challenging, Developed syn, dns flooding in DPDK Pktgen to replace MoonGen)

3 Intel Pktgen-DPDK in VMware ESXi VM (patches can found in attachment syn-dns-flood.txt)

a: syn flooding

some comments in SR 1-2068544601/#1-ZO4P5N

root@pktgen-template:/home/admin/pktgen-dpdk/dpdk/examples/pktgen-dpdk# ./app/app/x86_64-native-linuxapp-gcc/pktgen -c ff --
-P -m [0:1-7].0

Pktgen> load synflood.txt

Pktgen>page range

Port # Port-0
dst.ip : 10.2.72.97
inc : 0.0.0.1
min : 10.2.72.97
max : 10.2.72.200
:

```
src.ip         :       10.1.72.154
   inc         :         0.0.0.1
   min         :       10.1.72.154
   max         :       10.1.72.254
               :
ip_proto       :              TCP
               :
dst.port / inc :         80/    0
    min / max  :          0/    0
               :
src.port / inc :       1025/    1
    min / max  :       1025/65512
               :
vlan.id / inc  :          1/    0
   min / max   :          1/4095
               :
pkt.size / inc :         64/    0
    min / max  :         64/1518
               :
dst.mac        :   00:23:e9:6c:74:83
   inc         :   00:00:00:00:00:00
   min         :   00:00:00:00:00:00
   max         :   00:00:00:00:00:00
               :
src.mac        :   00:50:56:86:10:76
   inc         :   00:00:00:00:00:00
   min         :   00:00:00:00:00:00
   max         :   00:00:00:00:00:00
               :
gtpu.teid / inc :         0/    0
     min / max :         0/    0
-- Pktgen Ver: 2.9.17 (DPDK 16.04.0-rc2)  Powered by Intel® DPDK ----------


Pktgen> start 0

Pktgen> stop 0


root@pktgen-template:/home/admin/pktgen-dpdk/dpdk/examples/pktgen-dpdk# cat synflood.txt
#
# Pktgen - Ver: 2.9.17 (DPDK 16.04.0-rc2)
# Copyright (c) <2010-2016>, Intel Corporation. All rights reserved., Powered by Intel® DPDK

# Command line arguments: (DPDK args are defaults)
# ./app/app/x86_64-native-linuxapp-gcc/pktgen -c ff -n 3 -m 512 --proc-type primary -- -P -m [0:1-7].0

###############################################################################
# Pktgen Configuration script information:
#   GUI socket is Not Enabled
#   Flags 00040004
#   Number of ports: 1
#   Number ports per page: 4
#   Number descriptors: RX 512 TX: 512
#   Promiscuous mode is Enabled


###############################################################################
# Global configuration:
geometry 132x44
mac_from_arp disable

############################ Port  0 ###################################
#
# Port:  0, Burst: 32, Rate:100%, Flags:c0000010, TX Count:Forever
#          SeqCnt:0, Prime:1 VLAN ID:0001, Link: <UP-10000-FD>
#
# Set up the primary port information:
set 0 count 0
set 0 size 64
set 0 rate 100
set 0 burst 32
```

```
set 0 sport 1234
set 0 dport 5678
set 0 prime 1
type ipv4 0
proto tcp 0
set ip dst 0 10.2.72.97
#set ip dst 0 10.1.72.8
set ip src 0 10.1.72.154/24
#vpr 172.24.15.7
#set mac 0 00:23:E9:12:AA:01
#10200 172.24.40.6
#set mac 0 00:23:E9:E5:F2:C3
#4200 172.24.46.39
#set mac 0 00:23:E9:63:5B:83
#5200 172.24.19.15
set mac 0 00:23:E9:6C:74:83
vlanid 0 1

pattern 0 zero
user.pattern 0 0123456789abcdef

latency 0 disable
mpls 0 disable
mpls_entry 0 0
qinq 0 disable
qinqids 0 0 0
gre 0 disable
gre_eth 0 disable
gre_key 0 0
#
# Port flag values:
icmp.echo 0 disable
pcap 0 disable
range 0 enable
process 0 disable
capture 0 disable
rxtap 0 disable
txtap 0 disable
vlan 0 disable

#
# Range packet information:
src.mac start 0 00:50:56:86:10:76
src.mac min 0 00:00:00:00:00:00
src.mac max 0 00:00:00:00:00:00
src.mac inc 0 00:00:00:00:00:00
dst.mac start 0 00:23:E9:6C:74:83
#dst.mac start 0 00:23:E9:E5:F2:C3
#dst.mac start 0 00:23:E9:12:AA:01
#dst.mac start 0 00:23:E9:63:5B:83
#dst.mac start 0 00:50:56:86:84:90
dst.mac min 0 00:00:00:00:00:00
dst.mac max 0 00:00:00:00:00:00
dst.mac inc 0 00:00:00:00:00:00

src.ip start 0 10.1.72.154
src.ip min 0 10.1.72.154
src.ip max 0 10.1.72.254
src.ip inc 0 0.0.0.1

dst.ip start 0 10.2.72.97
dst.ip min 0 10.2.72.97
dst.ip max 0 10.2.72.200
dst.ip inc 0 0.0.0.1

#dst.ip start 0 10.1.72.8
#dst.ip min 0 10.1.72.8
#dst.ip max 0 10.1.72.8
#dst.ip inc 0 0.0.0.1

src.port start 0 1025
src.port min 0 1025
```

```
src.port max 0 65512
src.port inc 0 1

dst.port start 0 80
dst.port min 0 0
dst.port max 0 0
dst.port inc 0 0

vlan.id start 0 1
vlan.id min 0 1
vlan.id max 0 4095
vlan.id inc 0 0

pkt.size start 0 64
pkt.size min 0 64
pkt.size max 0 1518
pkt.size inc 0 0

#
# Set up the sequence data for the port.
set 0 seqCnt 0

################################# Done #################################
```

b: dns flooding

Pktgen> load dnsflood.txt

Pktgen> start 0

Pktgen> stop 0

root@pktgen-template:/home/admin/pktgen-dpdk/dpdk/examples/pktgen-dpdk# cat dnsflood.txt

```
#
# Pktgen - Ver: 2.9.17 (DPDK 16.04.0-rc2)
# Copyright (c) <2010-2016>, Intel Corporation. All rights reserved., Powered by Intel® DPDK

# Command line arguments: (DPDK args are defaults)
# ./app/app/x86_64-native-linuxapp-gcc/pktgen -c ff -n 3 -m 512 --proc-type primary -- -P -m [0:1-7].0

###########################################################################
# Pktgen Configuration script information:
#   GUI socket is Not Enabled
#   Flags 00040004
#   Number of ports: 1
#   Number ports per page: 4
#   Number descriptors: RX 512 TX: 512
#   Promiscuous mode is Enabled


###########################################################################
# Global configuration:
geometry 132x44
mac_from_arp disable

########################## Port  0 #################################
#
# Port:  0, Burst: 32, Rate:100%, Flags:c0000010, TX Count:Forever
#          SeqCnt:0, Prime:1 VLAN ID:0001, Link: <UP-10000-FD>
#
# Set up the primary port information:
set 0 count 0
set 0 size 64
set 0 rate 100
set 0 burst 32
set 0 sport 1234
set 0 dport 5678
set 0 prime 1
type ipv4 0
proto udp 0
set ip dst 0 10.2.72.97
```

```
#set ip dst 0 10.1.72.8
set ip src 0 10.1.72.154/24
#vpr 172.24.15.7
#set mac 0 00:23:E9:12:AA:01
#10200 172.24.40.6
#set mac 0 00:23:E9:E5:F2:C3
#4200 172.24.46.39
#set mac 0 00:23:E9:63:5B:83
#5200 172.24.19.15
set mac 0 00:23:E9:6C:74:83
vlanid 0 1

#pattern 0 user
#user.pattern 0 000011111111111111

latency 0 disable
mpls 0 disable
mpls_entry 0 0
qinq 0 disable
qinqids 0 0 0
gre 0 disable
gre_eth 0 disable
gre_key 0 0
#
# Port flag values:
icmp.echo 0 disable
pcap 0 disable
range 0 enable
range.proto 0 udp
process 0 disable
capture 0 disable
rxtap 0 disable
txtap 0 disable
vlan 0 disable

#
# Range packet information:
src.mac start 0 00:50:56:86:10:76
src.mac min 0 00:00:00:00:00:00
src.mac max 0 00:00:00:00:00:00
src.mac inc 0 00:00:00:00:00:00
dst.mac start 0 00:23:E9:6C:74:83
#dst.mac start 0 00:23:E9:E5:F2:C3
#dst.mac start 0 00:23:E9:12:AA:01
#dst.mac start 0 00:23:E9:63:5B:83
#dst.mac start 0 00:50:56:86:84:90
dst.mac min 0 00:00:00:00:00:00
dst.mac max 0 00:00:00:00:00:00
dst.mac inc 0 00:00:00:00:00:00

src.ip start 0 10.1.72.154
src.ip min 0 10.1.72.154
src.ip max 0 10.1.72.254
src.ip inc 0 0.0.0.1

dst.ip start 0 10.2.72.97
dst.ip min 0 10.2.72.97
dst.ip max 0 10.2.72.200
dst.ip inc 0 0.0.0.1

#dst.ip start 0 10.1.72.8
#dst.ip min 0 10.1.72.8
#dst.ip max 0 10.1.72.8
#dst.ip inc 0 0.0.0.1

src.port start 0 1025
src.port min 0 1025
src.port max 0 65512
src.port inc 0 1

dst.port start 0 53
dst.port min 0 0
```

```
dst.port max 0 0
dst.port inc 0 0

vlan.id start 0 1
vlan.id min 0 1
vlan.id max 0 4095
vlan.id inc 0 0

pkt.size start 0 74 #note packet size setting has to match the whole udp dns payload, can be improved in code to deal with variable length of domain name query
pkt.size min 0 74
pkt.size max 0 1518
pkt.size inc 0 0

#
# Set up the sequence data for the port.
set 0 seqCnt 0

################################## Done ###################################
```

Bug 486688 test validation

-----CLIENT **VMware ESXi VM vli_ubuntu_1404_ab_template** with DPDK pktgen

root@pktgen-template:/home/admin/pktgen-dpdk/dpdk/examples/pktgen-dpdk# ./app/app/x86_64-native-linuxapp-gcc/pktgen -c ff -- -P -m "[0:0-7].0 "

```
 Copyright (c) <2010-2016>, Intel Corporation. All rights reserved.
   Pktgen created by: Keith Wiles -- >>> Powered by Intel® DPDK <<<

Lua 5.3.2  Copyright (C) 1994-2015 Lua.org, PUC-Rio
>>> Packet Burst 32, RX Desc 512, TX Desc 512, mbufs/port 4096, mbuf cache 512

=== port to lcore mapping table (# lcores 8) ===
   lcore:    0    1    2    3    4    5    6    7
port   0:  D: T  0: 1  0: 1  0: 1  0: 1  0: 1  0: 1  0: 1 =  1: 8
Total   :  1: 1  0: 1  0: 1  0: 1  0: 1  0: 1  0: 1  0: 1
    Display and Timer on lcore 0, rx:tx counts per port/lcore

Configuring 1 ports, MBUF Size 1920, MBUF Cache Size 512
Lcore:
    0, RX-TX
               RX( 1): ( 0: 0)
               TX( 1): ( 0: 0)
    1, TX-Only
               TX( 1): ( 0: 1)
    2, TX-Only
               TX( 1): ( 0: 2)
    3, TX-Only
               TX( 1): ( 0: 3)
    4, TX-Only
               TX( 1): ( 0: 4)
    5, TX-Only
               TX( 1): ( 0: 5)
    6, TX-Only
               TX( 1): ( 0: 6)
    7, TX-Only
               TX( 1): ( 0: 7)

Port :
    0, nb_lcores  8, private 0x8f0690, lcores:  0  1  2  3  4  5  6  7
```

```
** Dev Info (rte_vmxnet3_pmd:0) **
  max_vfs       :   0 min_rx_bufsize    :1646 max_rx_pktlen : 16384 max_rx_queues         :  16 max_tx_queues:   8
  max_mac_addrs  :   1 max_hash_mac_addrs:   0 max_vmdq_pools:     0
  rx_offload_capa:  13 tx_offload_capa   :  45 reta_size     :     0 flow_type_rss_offloads:0000000000000514
  vmdq_queue_base:   0 vmdq_queue_num    :   0 vmdq_pool_base:     0
** RX Conf **
  pthreash      :   0 hthresh         :   0 wthresh       :     0
  Free Thresh    :   0 Drop Enable      :   0 Deferred Start :     0
** TX Conf **
  pthreash      :   0 hthresh         :   0 wthresh       :     0
  Free Thresh    :   0 RS Thresh        :   0 Deferred Start :     0 TXQ Flags:00000200

Initialize Port 0 -- TxQ 8, RxQ 1,  Src MAC 00:50:56:86:10:76

Pktgen > load synflood.txt

Pktgen> start 0

Pktgen> stop 0


-----BIGIP DUT:


net vlan esnet-1101 {
    if-index 896
    interfaces {
        1.1 {
            tagged
        }
    }
    tag 1101
}
net vlan esnet-1102 {
    if-index 912
    interfaces {
        1.1 {
            tagged
        }
    }
    tag 1102
}



net self 10.2.72.12 {
    address 10.2.72.12/16
    allow-service all
    traffic-group traffic-group-local-only
    vlan esnet-1102
}
net self 10.1.72.12 {
```

```
    address 10.1.72.12/16
    allow-service all
    traffic-group traffic-group-local-only
    vlan esnet-1101
}



ltm virtual /Common/vs_http {
    destination /Common/10.2.0.0:80
    ip-forward
    ip-protocol tcp
    mask 255.255.0.0
    profiles {
        /Common/fl4_swsyncookie { }
    }
    source 0.0.0.0/0
    source-address-translation {
        type automap
    }
    translate-address disabled
    translate-port disabled
}

ltm profile fastl4 /Common/fl4_swsyncookie {
    app-service none
    defaults-from /Common/fastL4
    hardware-syn-cookie enabled
    pva-dynamic-client-packets 2
    pva-dynamic-server-packets 2
    pva-offload-dynamic enabled
    software-syn-cookie enabled
}

net route /Common/default {
    gw 10.2.72.66
    network default
}

 -----BIGIP backend as server
net vlan /Common/esnet-1102 {
```

```
    interfaces {
        1.1 {
            tag-mode service
            tagged
        }
    }
    tag 1102
}

net self /Common/10.2.72.66 {
    address 10.2.72.66/16
    allow-service all
    traffic-group /Common/traffic-group-local-only
    vlan /Common/esnet-1102
}
net route /Common/route_to_10.1 {
    gw 10.2.72.12
    network 10.1.0.0/16
}
ltm virtual /Common/vs3_http {
    destination /Common/10.2.72.97:80
.............SKIP.........
}
ltm virtual /Common/vs98_http {
    destination /Common/10.2.72.98:80
.............SKIP............
}
```

4 Ported mTCP/DPDK to KVM VM using with VIRTIO PMD (mTCP/DPDK VM < ---> BIGIP VM)

Host hardware : Dell Desktop Optiplex 990

8 core: Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz

16GB RAM

NIC:

03:00.0 Ethernet controller: Broadcom Corporation NetXtreme BCM5720 Gigabit Ethernet PCIe

03:00.1 Ethernet controller: Broadcom Corporation NetXtreme BCM5720 Gigabit Ethernet PCIe

a: run mTCP DPDK app on load VM (generate ~340Mbps):

```
root@dpdk-kvm:/home/dpdk/mtcp# ./apps/example/epwget 10.9.9.6 100000000 -N 1 -c 100000
[ ALL ] connect:   62890, read:    3 MB, write:    6 MB, completes:   46922 (resp_time avg: 9893, max: 2504881 us)
[CPU 0] dpdk0 flows: 100382, RX:  253506(pps) (err:     0),  0.23(Gbps), TX:  384867(pps),  0.34(Gbps)
[ ALL ] dpdk0 flows: 100382, RX:  253506(pps) (err:     0),  0.23(Gbps), TX:  384867(pps),  0.34(Gbps)
```

b: BIGIP VM on KVM

Ltm::Virtual Server: vs_http

Status

CMP : enabled

CMP Mode : all-cpus

Destination : 10.9.9.6:80

Traffic	ClientSide	Ephemeral	General
Bits In	233.3G	0	-
Bits Out	184.2G	0	-
Packets In	396.3M	0	-
Packets Out	316.2M	0	-
Current Connections	254.2K	0	-
Maximum Connections	268.3K	0	-

top - 13:57:14 up 1:13, 2 users, load average: 1.96, 1.16, 0.63

Tasks: 411 total, 3 running, 408 sleeping, 0 stopped, 0 zombie

Cpu0 : 31.9%us, 16.9%sy, 0.0%ni, 12.2%id, 0.0%wa, 0.4%hi, 36.6%si, 2.0%st

Cpu1 : 53.4%us, 22.0%sy, 0.0%ni, 21.1%id, 0.0%wa, 0.0%hi, 0.0%si, 3.5%st

Cpu2 : 52.4%us, 20.7%sy, 0.3%ni, 22.0%id, 0.0%wa, 0.0%hi, 0.0%si, 4.5%st

Cpu3 : 57.1%us, 21.1%sy, 0.0%ni, 19.2%id, 0.0%wa, 0.0%hi, 0.0%si, 2.6%st

Mem: 14402956k total, 14159080k used, 243876k free, 4720k buffers

Swap: 1048572k total, 81464k used, 967108k free, 341004k cached

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
13717	root	RT	0	12.1g	127m	102m	R	82.0	0.9	22:23.88	tmm.0 -T 4 --tmid 0 --npus 4 --platform Z100 -m -s 12088
13830	root	RT	0	12.1g	127m	102m	S	81.7	0.9	16:38.70	tmm.0 -T 4 --tmid 0 --npus 4 --platform Z100 -m -s 12088
13829	root	RT	0	12.1g	127m	102m	R	80.4	0.9	17:03.86	tmm.0 -T 4 --tmid 0 --npus 4 --platform Z100 -m -s 12088
13828	root	RT	0	12.1g	127m	102m	S	80.1	0.9	17:04.08	tmm.0 -T 4 --tmid 0 --npus 4 --platform Z100 -m -s 12088

5, Pktgen on SR-IOV

Moved Dell R710 with Intel 82599 10G NIC to ES lab, created two ubuntu VMs with NIC provisioned by Intel 82599 VF, one VM run pktgen syn flooding,

another VM run pktgen dns flooding, this is to simulate both tcp and udp flooding from same 10G NIC port

Dell R710 hypvervisor with Intel 82599 SR-IOV

5a, enable SR-IOV in bios and kernel cmdline

root@pktgen:/home/dpdk/mtcp# cat /proc/cmdline
BOOT_IMAGE=/vmlinuz-3.13.0-32-generic root=/dev/mapper/pktgen--vg-root ro intel_iommu=on ixgbe.max_vfs=2

p2p1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode DEFAULT group default qlen 1000

link/ether e8:ea:6a:06:1b:1a brd ff:ff:ff:ff:ff:ff

vf 0 MAC 52:54:00:4c:86:ba, spoof checking on, link-state auto

vf 1 MAC 52:54:00:8b:ea:5f, spoof checking on, link-state auto

5b Use the virsh nodedev-list command to get the PCI address of the VF you you want to assign and it's corresponding PF

```
root@pktgen:/home/dpdk/mtcp# virsh nodedev-list | grep 0000_04_
pci_0000_04_00_0
pci_0000_04_00_1
pci_0000_04_10_0
pci_0000_04_10_1
pci_0000_04_10_2
pci_0000_04_10_3
```

5c use virsh nodedev-dumpxml to dump the VF info

```
root@pktgen:/home/dpdk/mtcp# virsh nodedev-dumpxml pci_0000_04_10_0
<device>
  <name>pci_0000_04_10_0</name>
  <path>/sys/devices/pci0000:00/0000:00:05.0/0000:04:10.0</path>
  <parent>pci_0000_00_05_0</parent>
  <driver>
    <name>pci-stub</name>
  </driver>
  <capability type='pci'>
    <domain>0</domain>
    <bus>4</bus>
    <slot>16</slot>
    <function>0</function>
    <product id='0x10ed'>82599 Ethernet Controller Virtual Function</product>
    <vendor id='0x8086'>Intel Corporation</vendor>
    <capability type='phys_function'>
      <address domain='0x0000' bus='0x04' slot='0x00' function='0x0'/>
    </capability>
    <iommuGroup number='33'>
      <address domain='0x0000' bus='0x04' slot='0x10' function='0x0'/>
    </iommuGroup>
  </capability>
</device>
```

The following data is needed for the next step:

<domain>0</domain>

<bus>4</bus>

<slot>16</slot>

<function>0</function>

Create a temporary XML file (for example /tmp/vf-interface.xml containing the data necessary to add a VF network device to an existing VM Guest. The minimal content of the file needs to look like the following:

```
<interface type='hostdev'>
 <source>
  <address type='pci' domain='0' bus='4' slot='16' function='0'/>
 </source>
</interface>
```

5d Specify the data you acquired in the previous step here

In case a device is already attached to the host, it cannot be attached to a guest. To make it available for guests, detach it from the host first:

#virsh nodedev-detach pci_0000_04_10_0

Last, add the VF interface to an existing VM Guest:

#virsh attach-device GUEST /tmp/vf-interface.xml --OPTION

GUEST needs to be replace by the domain name, id or uuid of the VM Guest and --OPTION can be one of the following:

--persistent

This option will always add the device to the domain's persistent XML. In addition, if the domain is running, it will be hotplugged.

--config

This option will only affect the persistent XML, even if the domain is running., The device will only show up in the guest on next boot

--live

This option will only affect a running domain. If the domain is inactive, the operation will fail. The device is not persisted in the XML and won't be available in the guest on next boot.

--current

This option affects the current state of the domain. If the domain is inactive, the device is added to the persistent XML and will be available on next boot. If the domain is active, the device is hotplugged but not added to the persistent XML.

5e follow http://dpdk.org/browse/apps/pktgen-dpdk/tree/README.md to compile pktgen on VM and binding the ixgbevf to igb_uio

5f run patched pktgen with tcp syn flooding and dns flooding

Note pktgen use core 0 and core 1 for control and screen display, so do not assign logical core 0 and core 1 for packet tx/rx, otherwise, pktgen stops sending packets after a few minutes run

root@dpdk-sriov:/home/dpdk/pktgen-dpdk/dpdk/examples/pktgen-dpdk# ./app/app/x86_64-native-linuxapp-gcc/pktgen -c 0xf -- -m [2:3].0

Pktgen> load dnsflood.txt

Pktgen> start 0

Pktgen> page range

Pkten> page main

MBits/s Rx/Tx : 0/4195 0/4195 <==========~4Gbits/s

Src/Dest Port : 1234 / 5678

Pkt Type:VLAN ID: IPv4 / TCP:0001

Dst IP Address : 10.6.6.6

Src IP Address : 10.6.6.128/24

Dst MAC Address : 00:00:00:00:00:00

Src MAC Address 52:54:00:4c:86:ba

-- Pktgen Ver: 2.9.17 (DPDK 16.04.0-rc2) Powered by Intel® DPDK ----------

root@ubuntu-vm:/home/dpdk/pktgen-dpdk/dpdk/examples/pktgen-dpdk# ./app/app/x86_64-native-linuxapp-gcc/pktgen -c 0xf -- -m [2:3].0

Pktgen> load synflood.txt

Pktgen> start 0

Pktgen> page range

Pkten> page main

MBits/s Rx/Tx : 0/3739 0/3739 <==========~3Gbits/s

6 Pktgen with DPDK net bonding (link aggregation) to BIGIP trunking

NOTE:

6a: don't enable LACP on BIGIP trunk, it appears DPDK net bonding does not work well with LACP with BIGIP trunk

6b: when run Pktgen, make sure the port id is specified correctly to use the net_bonding0 port id, not individual DPDK net bonding member port id

6c: specify DPDK net bonding mode to non mode 4 (LACP) due to 6a

root@r710:/home/dpdk/dpdk-upstream/examples/pktgen-dpdk# ./app/app/x86_64-native-linuxapp-gcc/pktgen -c 0xff --vdev=net_bonding0,mode=0,xmit_policy=l34,slave=0000:04:00.1,slave=0000:04:00.0 -- -P -m [0:1-7].2

EAL: Initializing pmd_bond for net_bonding0

EAL: Create bonded device net_bonding0 on port 2 in mode 0 on socket 0. <=====here net_bonding0 is port id 2, thus reference this port id in Pktgen ~~-m[0:1-~~7].2

Pktgen> page stats

/ <Real Port Stats Page> Copyright (c) <2010-2016>, Intel Corporation

Port Name	Pkts Rx/Tx	Rx Errors/Missed	Rate Rx/Tx	MAC Address	
0-0000:04:00.0:	511/38430488961	0/5067921	0/8085293	E8:EA:6A:06:1B:1B	<====DPDK port 0
1-0000:04:00.1:	511/38430500852	0/5558243	0/8085340	E8:EA:6A:06:1B:1B	<=====DPDK port 1
2-net_bonding0:	1022/76860994839	0/10626165	0/16170614	E8:EA:6A:06:1B:1B	<====DPDK net bonding port 2.

-- Pktgen Ver: 3.1.0 (DPDK 17.02.0-rc0) Powered by Intel® DPDK ----------

References

http://highscalability.com/blog/2013/5/13/the-secret-to-10-million-concurrent-connections-the-kernel-i.html

http://www.dpdk.org/

http://www.ndsl.kaist.edu/~kyoungsoo/papers/mtcp.pdf

https://github.com/eunyoung14/mtcp

https://github.com/emmericp/MoonGen

http://git.es.f5net.com/index.cgi/codeshare/tree/vli/mtcp (git log --author="Vincent Li“)

http://git.es.f5net.com/index.cgi/codeshare/tree/vli/MoonGen

GS Operations - C10M