

Fast Prototyping DPDK Apps in Containernet

ANDREW WANG, PRINCIPAL SOFTWARE ENGINEER COMCAST



Who Am I

- Principal Software Engineer at Comcast
- Past projects
 - GDB
 - Content based routing in ad hoc networks
 - Key value stores
 - Load balancing services





Agenda

- Overview
 - · What happened when we started writing network functions
- Containernet
 - · What it is, what it is capable of and why should I care
- DPDK in Containernet
 - Setup
 - Running
- Short Demo



So we wanted to write fast network functions using DPDK



https://www.linkedin.com/in/andrewkewang

Photo by Mathew Schwartz on Unsplash.



And everyone was excited, including us!





Photo by Val Vesa on Unsplash

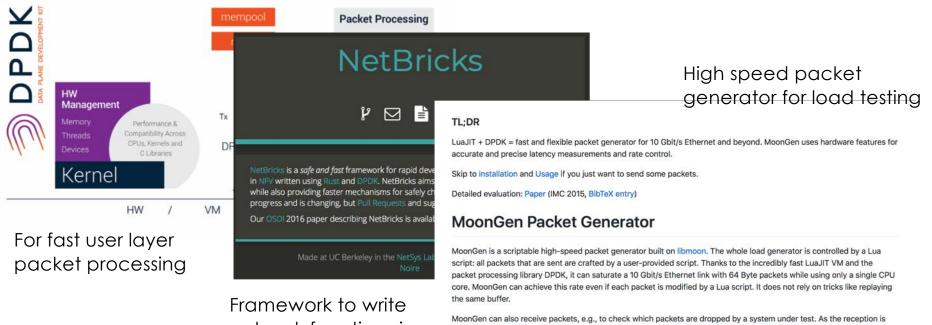




We set up everything we needed...



APPLICATION



network functions in Rust & DPDK

also fully under control of the user's Lua script, it can be used to implement advanced test scripts. E.g. one can use two instances of MoonGen that establish a connection with each other. This setup can be used to benchmark middle-boxes like firewalls.



https://www.linkedin.com/in/andrewkewang



But then we stopped and thought

- Where can we *run* this?
- How can we *verify* correct behavior?
- How easy is it to *debug* any issues found?





VMs How about VMs in a single host?

- Each host in the network is a VM
- Use host-only networking to connect multiple VMs
- Use tools like Vagrant and Virtualbox





VMs The problem with VMs

- High resource utilization
 - CPU
 - RAM
- Virtualization and Anycast
 - Same IP address bound to multiple interfaces creates problems





Lab

- Dedicated space
- Duplicate approximate network conditions where we want to deploy our functions





Lab The problem with Lab

- Not under our control
 - Hard to make any changes
 - Resources can be pre-empted for other higher priority projects
- Hard to debug
- Scheduling experiments







Production networks?

SSSUUUUREEE! Yeah, right!

https://www.linkedin.com/in/andrewkewang





What is Containernet?

• Fork of Mininet that supports using Docker containers as Mininet hosts

- <u>https://containernet.github.io</u>
- M. Peuster, H. Karl, and S. v. Rossem: <u>MeDICINE: Rapid Prototyping of</u> <u>Production-Ready Network Services in Multi-PoP Environments</u>. IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), Palo Alto, CA, USA, pp. 148-153. doi: 10.1109/NFV-SDN.2016.7919490. (2016)
- What is Mininet?
 - From http://mininet.org/
 - > "An Instant Virtual Network on your Laptop (or other PC)"
 - * "Mininet is a network emulator, or perhaps more precisely a network emulation orchestration system. It runs a collection of end-hosts, switches, routers, and links on a single Linux kernel. It uses lightweight virtualization to make a single system look like a complete network, running the same kernel, system, and user code."

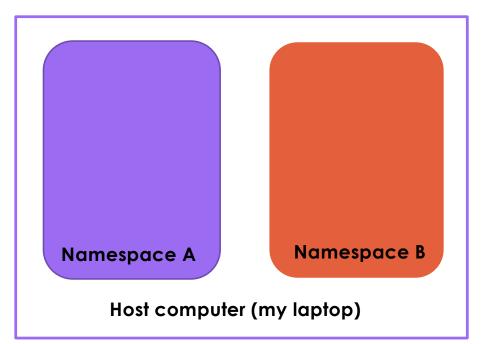




How does it work? Nodes

Network nodes are *network namespaces*

- Processes in a network namespace do not have access to the host's network interfaces
 - Think how chroot removes access to the host's root filesystem
 - In addition to network interfaces, the routing tables and iptables rules are different
 - Processes started in a network namespace inherit the view of the namespace
 - ip netns add <name>



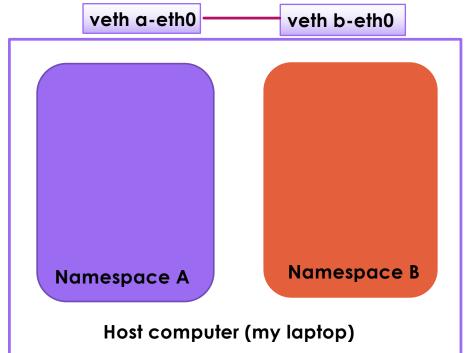




How does it work? Links

Network links are established via virtual Ethernet (veth) interface pairs

- "Act as tunnels between network namespaces to create a bridge to a physical network device in another namespace"
- "Created in interconnected pairs [...] packets transmitted on one device in the pair are immediately received on the other device"
 - ip link add <interface name> type veth peer name <interface pair name>
 - ip link set <interface name> netns
 <namespace>





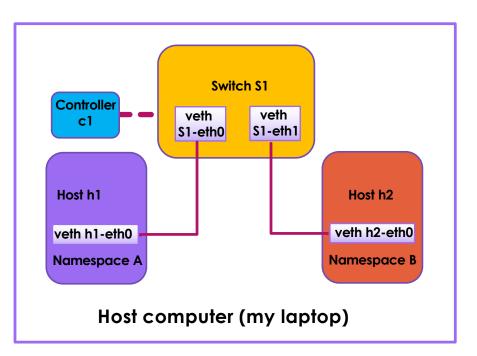




How does it work? Switches

The default switch is Open vSwitch running in kernel space, connected to the OpenFlow reference controller.

- Default behavior of the switch is the same as an Ethernet learning switch.
- Connect multiple hosts
- By default switches live in the host name space, but moving them to their own namespaces is supported





How does it work? Setup

Python API

1	#!/usr/bin/python	18	<pre>info('*** Adding switches\n')</pre>
2	000	19	<pre>s1 = net.addSwitch('s1')</pre>
3	This is the most simple example to showcase Containernet.	20	<pre>s2 = net.addSwitch('s2')</pre>
4		21	info('*** Creating links\n')
5	(from mininet.net import Containernet	22	net.addLink(d1, s1)
6	from mininet.node import Controller	23	<pre>net.addLink(s1, s2, cls=TCLink, delay='100ms', bw=1)</pre>
7	from mininet.cli import CLI	24	net.addLink(s2, d2)
8	from mininet.link import TCLink		
9	from mininet.log import info, setLogLevel	25	info('*** Starting network\n')
10	<pre>setLogLevel('info')</pre>	26	net.start()
11		27	<pre>info('*** Testing connectivity\n')</pre>
12	<pre>net = Containernet(controller=Controller)</pre>	28	net.ping([d1, d2])
13	info('*** Adding controller\n')	29	info('*** Running CLI\n')
14	<pre>net.addController('c0')</pre>	30	CLI(net)
15	<pre>info('*** Adding docker containers\n')</pre>	31	<pre>info('*** Stopping network')</pre>
16	<pre>d1 = net.addDocker('d1', ip='10.0.0.251', dimage="ubuntu:trusty")</pre>	32	net.stop()
17	<pre>d2 = net.addDocker('d2', ip='10.0.0.252', dimage="ubuntu:trusty")</pre>		

https://github.com/containernet/containernet/blob/master/examples/containernet_example.py







How does it work? Python API

- Python wraps around commands to create switches, veth, namespaces
- Can execute commands in each "host"





How does it work? Running

Containernet (Mininet) performs emulation

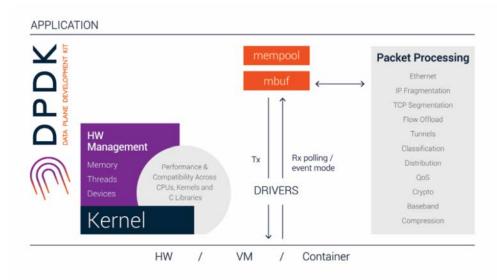
- Setup the topology desired, and start traffic sources, routing software, etc
- Software runs as-is, interacting with real network stack, at wall clock speed
 - When TCP BBR was released as a linux module, it could be run in Mininet
 - Shared resources limited by hardware speed (cannot emulate link speed faster than supported by underlying hardware)
- Linux only
- Ethernet links

COMCAST 19



DPDK

- Setup DPDK to run in containernet
- Aim to verify correct behavior, not performance
 - Environmental Abstraction Layer
 - Poll Mode Driver: AF_PACKET







DPDK in Containernet

AF_PACKET PMD

- AF_PACKET: "raw socket", bypasses normal in-kernel TCP/IP processing
- In MoonGen, we pass the following config parameters

```
"--file-prefix", "vdev",
```

```
"--no-pci",
```

- "--vdev=eth_af_packet,iface=<interface name>"
- In Netbricks, we pass the following config line for [[dpdk.port]]
 "dpdk:eth af packet,iface=<interface name>"
- Application name
 - Apps running on different containernet nodes must have different names





DPDK in Containernet

- Shared filesystem
 - Make sure configs have different paths
- TCP Checksum/segmentation offloading
 - Usually performed by hardware NIC
 - Must be disabled (in other nodes) for correct checksums
- Access to all hosts and links in the network, most tools
 - Can run wireshark (in the switches), tcpdump, dropwatch





Voilá!

X root@og west: /opt/router [Router] root@og west:/opt/router# /opt/router/target/debug/router -f /opt/router/containernet/occam-containernet-og west .toml Going to start DPDK with configuration Configuration: name: occam-router-og west mempool size: 512 core cache: 32 primary core: 0 Ports: Port: dpdk:eth af packet,iface=og west-eth0 RXQ Count: 1 RX Queues: [0] TXQ Count: 1 TX Queues: [0] RXD: 128 TXD: 128 Loopback: false Cores: 0 Failed to detect # of NUMA nodes from: /sys/devices/system/node/possible. Assuming a single-node system... EAL: Detected 4 lcore(s) EAL: Probing VFIO support... Running on node 0 Devname: "eth af packet,iface=og west-eth0" PMD: Initializing pmd af packet for eth af packet PMD: eth af packet: AF PACKET MMAP parameters: PMD: eth_af_packet: block size 4096 PMD: eth_af_packet: block count 256 PMD: eth af packet: frame size 2048 PMD: eth af packet: frame count 512 PMD: eth_af_packet: creating AF PACKET-backed ethdev on numa socket 0 Going to try and use port 0 Running on node 0 Receiving started Running 1 pipelines Starting scheduler on 0

https://www.linkedin.com/in/andrewkewang





Voilá!

<pre>Client: root@cliwest: copt/moongen/dpd/containernet/ Running: 'build/MoonGen dpd/containernet/J:-tcp-syn.ilwdpdk-config=dpdk/containernet/dpdk-conf-vdev-cliwest.lua-ifdaa:0:1cc iv:ll::-dol:001508:feed:method: WMRNI malloc() allocates objects >= 1 HiB from LualIT memory space. /WMRNI malloc() allocates objects >= 1 HiB from LualIT memory space. /WMRNI malloc() allocates objects >= 1 HiB from LualIT memory space. // MMRNI malloc() allocates objects >= 1 HiB from LualIT memory space. // MMRNI malloc() allocates objects >= 1 HiB from LualIT memory space. // MMRNI malloc() allocates objects >= 1 HiB from LualIT memory space. // MMRNI malloc() allocates objects >= 1 HiB from LualIT memory space. // MMRNI malloc() allocates objects >= 1 HiB from LualIT memory space. // MMRNI malloc() allocates objects >= 1 HiB from LualIT memory space. // MMRNI malloc() allocates objects >= 1 HiB from LualIT memory space. // MMRNI packet: from count Zoo PMD: eth of packet: frame size 2048 PMD: eth of packet: internet // from tog) // MMRNI Per-queue rate limit is not supported on this device, setting per-device rate limit to 10000 Mbit/s instead (note: this ma / fail as well if the NIC doesn't support any rate limiting). // MMRNI global rate limiting is not supported by the hardware or driver // TMFO bdst ip is 2001:558:feed:10:00:21: // MARNI global rate limiting is not supported by the hardware or driver // TMFO dst ip is 2001:558:feed:10:00:21: // MARNI global rate limiting is not supported by the hardware or driver // TMFO dst ip is 2001:558:feed:10:00:21: // MARNI global rate limit dis not support any rate limit foil not 6000 // ed000: // ed001:0000:0000:0000:0000:0000:0000:0000</pre>		
Rumning: 'build/Moondem dpdk/containernet/l3-tcp-syn.luadpdk/containernet/dpdk-conf-vdev-cli_west.lua -1 fdaa:0:1:c [0:11:: -d 2001;558:feed:l0:0;2:: [MKRN] malloc() allocates objects >= 1 MiB from LualIT memory space. [MKRN] malloc() allocates objects >= 1 MiB from LualIT memory space. [MKRN] malloc() allocates objects >= 1 MiB from LualIT memory space. [MKRN] malloc() allocates objects >= 1 MiB from LualIT memory space. [MKRN] malloc() allocates objects >= 1 MiB from LualIT memory space. [MKRN] malloc() allocates objects >= 1 MiB from LualIT memory space. [MKRN] finitializing DPM. This will take a few seconds EL: Detected 4 lcore(s) PMD: eth af packet: A PACKET MMP parameters: PMD: eth af packet: frame size 2048 PMD: eth af packet: set block count 512 PMD: eth af packet: frame size 2048 PMD: eth af packet: set block count 512 PMD: eth af packet: set block count 512 PMD: eth af packet: frame size 2040 PMD: eth af packet: block size 4000 MBit/s [MKN] dbal rate Limit is not supported on this device, setting per-device rate limit to 10000 Mbit/s instead (note: this ma y fail as well if the MIC doesn't support any rate limiting). [MKN] dbal rate Limit dis is not supported on this device or driver [IMFO] betected an IPV4 address: [IMFO] dbal rate Limit dis is not supported by the hardware or driver [IMFO] betected an IPV4 address. [IMFO] dbal rate Limit dis documentation flags 0x02 [- - - - SYN -] win 10 cksum 0x0000 urg 0 [] 0x0000: boil: 0x00: 0x00: 0x00: 0x00: 0x00 0x00 0x0	Image: Section of the section of	
<pre>PMD: Initializing pmd af packet for eth af packet PMD: eth af packet: AF PACKET MWAP parameters: PMD: eth af packet: block size 4096 PMD: eth af packet: frame size 2048 PMD: eth af packet: frame size 2048 PMD: eth af packet: frame count 512 PMD: eth af packet: creating AF PACKET-backed ethdev on numa socket 0 [INFO] Found 1 usable devices: Device 0: BA:DC:AF:EB:01:A1 (unknown NIC (PCI ID 0:0)) [INFO] Device 0: BA:DC:AF:EB:01:A1 is not supported on this device, setting per-device rate limit to 10000 Mbit/s instead (note: this ma y fail as well if the NIC doesn't support any rate limiting). [INFO] Device 0: BA:DC:AF:EB:01:A1 (unknown NIC (PCI ID 0:0)) [INFO] Device 0: BA:DC:AF:EB:01:A1 (unknown NIC (PCI ID 0:0)) [INFO] Device 0: BA:DC:AF:EB:01:A1 (unknown NIC (PCI ID 0:0)) [INFO] Device 0: BA:DC:AF:EB:01:A1 (unknown NIC (PCI ID 0:0)] [INFO] Device 0: BA:DC:AF:EB:01:A1 (unknown NIC (PCI ID 0:0)] [INFO] Device 0: BA:DC:AF:EB:01:A1 (unknown NIC (PCI ID 0:0)] [INFO] Device 0: BA:DC:AF:EB:01:A1 (unknown NIC (PCI ID 0:0)] [INFO] Device 0: BA:DC:AF:EB:01:A1 (unknown NIC (PCI ID 0:0)] [INFO] Device 0: BA:DC:AF:EB:01:A1 (unknown NIC (PCI ID 0:0)] [INFO] Device 0: BA:DC:AF:EB:01:A1 (unknown NIC (PCI ID 0:0)] [INFO] Device 0: BA:DC:AF:EB:01:A1 (unknown NIC (PCI ID 0:0)] [INFO] Device 0: BA:DC:AF:EB:01:A1 (unknown NIC (PCI ID 0:0)] [INFO] Device 0: BA:DC:AF:EB:01:A1 (unknown NIC (PCI ID 0:0)] [INFO] Device 0: BA:DC:AF:EB:01:A1 (unknown NIC (PCI ID 0:0)] [INFO] Device 0: BA:DC:AF:EB:01:AD:AF</pre>	Running: 'build/MoonGen dpdk/containernet/l3-tcp-syn.luadpdk-config=dpdk/containernet/dpdk-conf-vdev-cli_west.lua -i fdaa:0:1:c :0:11:: -d 2001:558:feed:10:0:2:: -m BA:DC:AF:EB:EE:F2 0' [WARN] malloc() allocates objects >= 1 MiB from LuaJIT memory space. [WARN] Install libjemalloc if you encounter out of memory errors. [INFO] Initializing DPDK. This will take a few seconds	
<pre>PMD: eth_af_packet: AF_PACKET_MMAP parameters: PMD: eth_af_packet: block count 256 PMD: eth_af_packet: frame count 512 PMD: eth_af_packet: frame count 512 PMD: eth_af_packet: creating AF_PACKET-backed ethdev on numa socket 0 [INFO] Device 0: BA:DC:AF:EB:01:All (unknown NIC (PCI ID 0:0)) [INFO] dotal rate limiting is not supported on this device, setting per-device rate limit to 10000 Mbit/s instead (note: this ma y fail as well if the NIC doesn't support any rate limiting). [INFO] dotal rate limiting is not supported by the hardware or driver [INFO] dot jn is 20001:5001:500:1001:0000:0000:20000:0000 ver 6 tc 0 fl 0 len 20 next 0x06 (unknown) t tl 64 TCP 1025 > 80 seq# 1 ack# 0 offset 0x5 reserved 0x80 flags 0x02 [- - - - SYN -] win 10 cksum 0x0000 urg 0 [] 0x0000: undc afeb cer2 badc afed 0101:0000 0x0000: 0udc 0000 2001 0558 feed 0000 0x0010: 0000 0000 10558 feed 0000 0000 0x0010: 0000 0000 10558 feed 0000 0000 0x0010: 0000 0000 0000 0000 0000 0x0002: 0010 0000 0000 0000 0000 0x0000: 0000 5002 0000 0000 0000 0x00000 000 5002 0000 0000 0000 0x0000: 0000 5002 0000 0000 0000 0x00000 5002 0000 0000 0000 0x00000 5002 0000 00</pre>	PMD: Initializing pmd af packet for eth af packet	
<pre>PMD: eth_af_packet: block count 256 PMD: eth_af_packet: frame size 2048 PMD: eth_af_packet: frame count 512 PMD: eth_af_packet: creating AF_PACKET-backed ethdev on numa socket 0 [INF0] found 1 usable devices: Device 0: BA:DC:AF:EB:01:A1 (unknown NIC (PCT ID 0:0)) [INF0] Device 0: (BA:DC:AF:EB:01:A1 (unknown NIC (PCT ID 0:0)) [INF0] Device 0: (BA:DC:AF:EB:01:A1 (unknown NIC (PCT ID 0:0)) [INF0] Device 0: (BA:DC:AF:EB:01:A1 (unknown NIC (PCT ID 0:0)) [INF0] Device 0: (BA:DC:AF:EB:01:A1 (unknown NIC (PCT ID 0:0)) [INF0] Device 0: (BA:DC:AF:EB:01:A1 (unknown NIC (PCT ID 0:0)) [INF0] Device 0: (BA:DC:AF:EB:01:A1 (unknown NIC (PCT ID 0:0)) [INF0] Device 0: (BA:DC:AF:EB:01:A1 (unknown NIC (PCT ID 0:0)) [INF0] Detected an IPv4 address. [INF0] Detected an IPv4 address. [INF0] Detected an IPv4 address. [INF0] backesteriolial > backesteriolice: [2 iypo 0x06dd (IPP) IP6 fdaa:0000:0001:000c:0000:0000 > 2001:0558:feed:0010:0000:0000 ver 6 tc 0 fl 0 len 20 next 0x06 (unknown) t t 64 CPC 1025 > 80 seg# 1 ack# 0 offset 0x5 reserved 0x00 flags 0x02 [- - - - SYN -] win 10 cksum 0x00000 urg 0 [] 0x0000: back afec cet2 back afed 0010 000c 0000 0x0001: 0000 20000 2000 0001 0000 0000 0x0002: 0001 0000 2000 0001 0000 0000 0x0002: 0000 2000 0001 0005 0000 0000 0x0002: 0000 2000 0000 0000 0000 0x0002: 0000 2000 0000 0000 0000 0x00040: 0000 0000 0000 0000 0x00040: 0000 0010 005 0000 0x00040: 0000 0000 0000 0x0040: 0000 0000 0000 0x04</pre>	PMD: eth af packet: AF PACKET MMAP parameters:	
<pre>PMD: eth_af_packet: frame size 2048 PMD: eth_af_packet: frame count 512 PMD: eth_af_packet: creating AF_PACKET-backed ethdev on numa socket 0 [IMFO] Found 1 usable devices: Device 0: BA:DC:AF:EB:01:A1 (unknown NIC (PCI ID 0:0)) [IMFO] Device 0 (BA:DC:AF:EB:01:A1 is up: 10000 MBit/s [WARN] Per-queue rate limit is not supported on this device, setting per-device rate limit to 10000 Mbit/s instead (note: this ma y fail as well if the NIC doesn't support any rate limiting). [IMFO] Device 0 (BA:DC:AF:EB:01:A1) is up: 10000 MBit/s [IMFO] dotat rate limiting is not supported by the hardware or driver [IMFO] dotat jp is 2001:558:feed:10:00:21: 21:11:18.030636 FTM monder of neuronal is build: af beinder of brows 12 type 0x86dd (IPF0) IP6 fdaa:0000:0001:000c:0000:0010:0000:0000 > 2001:0558:feed:010:0000:0000:0000 ver 6 tc 0 fl 0 len 20 next 0x06 (unknown) t t 64 TCP 1025 > 80 seq# 1 ack# 0 offset 0x5 reserved 0x00 flags 0x02 [-]-]-]-[SYN]-] win 10 cksum 0x00000 urg 0 [] 0x00000: bdds afde 0afd bed 10800 0000 0000 0x0003: 0000 0001 0000 0000 0000 0000 0000</pre>		
<pre>PMD: eth af packet: frame count 512 PMD: eth af packet: creating AF_PACKET-backed ethdev on numa socket 0 [INFO] Found 1 usable devices: Device 0: BA:DC:AF:EB:01:A1 (unknown NIC (PCI ID 0:0)) [INFO] Device 0 (BA:DC:AF:EB:01:A1) is up: 10000 MBit/s [WARN] Per-queue rate limit is not supported on this device, setting per-device rate limit to 10000 Mbit/s instead (note: this ma y fail as well if the NIC doesn't support any rate limiting). [WARN] global rate limiting is not supported by the hardware or driver [INFO] Detected an IPV4 address. [INFO] Detected an IPV4 address. [INFO] dst_ip is 2001:558:feed:10:0:2:: 21:11:18.030636 [M Londor:Battering of the distribution of the distres of the distribution of the distribution of the di</pre>		
<pre>PMD: eth af_packet: creating AF_PACKET-backed ethdev on numa socket 0 [INFO] Found 1 usable devices: Device 0 BA:DC:AF:EB:01:A1 (unknown NIC (PCI ID 0:0)) [INFO] Device 0 (BA:DC:AF:EB:01:A1) is up: 10000 Mbit/s [WARN] Per-queue rate limit is not supported on this device, setting per-device rate limit to 10000 Mbit/s instead (note: this ma y fail as well if the NIC doesn't support any rate limiting). [WARN] global rate limiting is not supported by the hardware or driver [INFO] Detected an IPv4 address. [INFO] dst_ip is 2001:558:feed:10:0:2:: 21:11:18.030636 IN backets of the order of the</pre>		
<pre>[INFO] Found 1 usable devices: Device 0: BA:DC:AF:EB:01:A1 (unknown NIC (PCI ID 0:0)) [INFO] Device 0 (BA:DC:AF:EB:01:A1) is up: 10000 MBit/s [WARN] Per-queue rate limit is not supported on this device, setting per-device rate limit to 10000 Mbit/s instead (note: this ma y fail as well if the NIC doesn't support any rate limiting). [WARN] global rate limiting is not supported by the hardware or driver [INFO] Detected an IPv4 address. [INFO] Detected an IPv4 address. [INFO] det_ip is 2001:558:feed:10:0:2:: 21:11:18.030636 "The backers devided as > backers devided to beer f2 type 0x86dd (IP6) IP6 fdaa:0000:0001:000c:0000:0011:0000:0000 > 2001:0558:feed:0010:0000:0002:0000:0000 ver 6 tc 0 fl 0 len 20 next 0x06 (unknown) t tl 64 ICP 1025 > 80 seq# 1 ack# 0 offset 0x5 reserved 0x00 flags 0x02 [- - - - SYN -] win 10 cksum 0x0000 urg 0 [] 0x00000: back afeb offset 0x5 reserved 0x00 flags 0x02 [- - - - SYN -] win 10 cksum 0x0000 urg 0 [] 0x00000: back afeb offset 0x5 reserved 0x00 flags 0x02 [- - - - SYN -] win 10 cksum 0x0000 urg 0 [] 0x00000: back afeb offset 0x5 reserved 0x00 flags 0x02 [- - - - SYN -] win 10 cksum 0x0000 urg 0 [] 0x00000: back afeb offset 0x5 reserved 0x00 flags 0x02 [- - - - SYN -] win 10 cksum 0x0000 urg 0 [] 0x00000: 0010 0000 0000 0001 000c 0000 0x00020: 0010 0000 0000 0001 0000 0000 000</pre>	PMD: etn al_packet: Trame count 512	
Device 0: BA:DC:AF:EB:01:A1 (unknown NIC (PCT ID 0:0)) [INFO] Device 0 (BA:DC:AF:EB:01:A1) is up: 10000 MBit/s [WARN] Per-queue rate limit is not supported on this device, setting per-device rate limit to 10000 Mbit/s instead (note: this may fail as well if the NIC doesn't support any rate limiting). [WARN] global rate limiting is not supported by the hardware or driver [INFO] Detected an IPV4 address. [INFO] dst_ip is 2001:558:feed:10:0:2:: 21:11:18.030636 [TM mode:drived of this baidc:nete:f2 type 0x86dd [IP6) IP6 fdaa:0000:0001:0000:0000:0000 > 2001:0558:feed:0010:0000:0002:0000:0000 ver 6 tc 0 fl 0 len 20 next 0x06 (unknown) t tl 64 ICCP 1025 > 80 seq# 1 ack# 0 offset 0x5 reserved 0x00 flags 0x02 [- - - - SYN -] win 10 cksum 0x0000 urg 0 [] 0x00000: Bacc afeb ecf2 badc afeb 0ial Edu 6000 0x0010: 0000 0001 40640 fdaa 0000 0001 0000 0000 0x0020: 0011 0000 0000 0001 0058 feed 0010 0000 0x0020: 0010 0000 0000 0000 0000 0000 0x0020: 0000 0000 0000 0000 0000 0000 0x0020: 0000 0000 0000 0000 0000 0000 0x0040: 0000 5002 0000 0000 0000 (Device: id=0] IX: nan (StdDev 0.00 Mps, nan (StdDev 0) Mbit/s (nan Mbit/s with framing), total 1 packets with 74 bytes (incl. C RC) [Client] root@cli_west:/opt/moongen/dpdk/containernet#		
<pre>[INFO] Device 0 (BA:DC:AF:EB:01:A1) is up: 10000 MBit/s [WARN] Per-queue rate limit is not supported on this device, setting per-device rate limit to 10000 Mbit/s instead (note: this may fail as well if the NIC doesn't support any rate limiting). [WARN] global rate limiting is not supported by the hardware or driver [INFO] detected an IPv4 address. [INFO] dst_ip is 2001:558:feed:10:0:2:: 21:11:18.030636 TH bardcrifteducter 2 type 0x86dd [IP6] IP6 fdaa:0000:0001:000c:0000:0001:0000:0000 > 2001:0558:feed:0010:0000:0002:0000:0000 ver 6 tc 0 fl 0 len 20 next 0x06 (unknown) t tl 64 ICP 1025 > 80 seg# 1 ack# 0 offset 0x5 reserved 0x00 flags 0x02 [- - - - SYN -] win 10 cksum 0x0000 urg 0 [] 0x0000: Dade afeb eef2 hade afeb 01al 8600 6000 0x001: D000 0000 2001 0058 feed 01al 8600 6000 0x002: 0011 0000 2000 0210 00558 feed 0010 0000 0x0030: D002 0000 0000 0401 0050 0000 0001 0000 0x0030: D002 0000 0000 0401 0050 0000 0000 0x0040: 0000 5002 0000 0000 0401 0050 0000 0000 0x0040: 0000 5002 0000 0000 0401 0050 0000 0000 0x0040: 0000 5002 0000 0000 0001 0000 0000 0x0040: 0000 5002 0000 0000 0000 0001 0000 0x0040: 0000 5002 0000 0000 0000 0000 0000 0x0040: 0000 0000 0000 0000 0000 0000 00</pre>		
<pre>y fail as well if the NIC doesn't support any rate limiting). [WARN] global rate limiting is not supported by the hardware or driver [INF0] betected an IPV4 address. [INF0] dst_ip is 2001:558:feed:10:0:2:: 21:11:18.030636 TH bardcaftebiole1 > bardcaftebioe:f2 type 0x06dd (TP6) IP6 fdaa:0000:0001:000c:0000:0011:0000:0000 > 2001:0558:feed:0010:0000:0002:0000:0000 ver 6 tc 0 fl 0 len 20 next 0x06 (unknown) t tl 64 TCP 1025 > 80 seq# 1 ack# 0 offset 0x5 reserved 0x00 flags 0x02 [- - - - SYN -] win 10 cksum 0x00000 urg 0 [] 0x00000: bard afeb cef2 bardc afeb 010 0000 0000 0x0020: 0011 0000 0000 2001 0558 feed 0010 0000 0x0030: 0002 0000 0000 0440 fdaa 0000 0001 0000 0x0030: 0002 0000 0000 0440 1055 0000 0001 0000 0x0040: 0000 5002 0000 0000 0440 0050 0000 0000</pre>	[INF0] Device 0 (BA:DC:AF:EB:01:A1) is up: 10000 MBit/s	
<pre>[WARN] global rate limiting is not supported by the hardware or driver [INFO] Detected an IPv4 address. [INFO] dst_ip is 2001:558:feed:10:0:2:: 21:11:18.036636 ETM bm contention = bm dc an eb (eff2 type 0x06dd (TP6) IP6 fdaa:0000:0001:000c:0000:0011:0000:0000 > 2001:0558:feed:0010:0000:0002:0000:0000 ver 6 tc 0 fl 0 len 20 next 0x06 (unknown) t tl 64 TCP 1025 > 80 seq# 1 ack# 0 offset 0x5 reserved 0x00 flags 0x02 [- - - - SYN -] win 10 cksum 0x0000 urg 0 [] 0x00000: bm c afeb ef2 bm c afeb 010 End 6000 0x0010: 0000 0014 0640 fdaa 0000 0001 000c 0000 0x0020: 0011 0000 0000 2001 0558 feed 0010 0000 0x0020: 0011 0000 0000 2001 0558 feed 0010 0000 0x0030: 0002 0000 0000 0000 0000 0000 0000</pre>		
<pre>[INF0] Detected an IPv4 address. [INF0] dst_ip is 2001:558:feed:10:0:2:: 21:11:18.030636 TH herdcrafteb101al > herdcrafteb.eerf2 type 0x86dd (IP6) IP6 fdaa:0000:0001:000c:0000:0011:0000:0000 > 2001:0558:feed:0010:0000:0002:0000:0000 ver 6 tc 0 fl 0 len 20 next 0x06 (unknown) t tl 64 TCP 1025 > 80 seq# 1 ack# 0 offset 0x5 reserved 0x00 flags 0x02 [- - - - SYN -] win 10 cksum 0x0000 urg 0 [] 0x0000: here afeb eef2 hade afeb flag 6000 0x0010: 0000 0014 0640 fdaa 0000 0001 0000 0000 0x0020: 0011 0000 0000 2001 0558 feed 0010 0000 0x0020: 0002 0000 0000 2001 0558 feed 0010 0000 0x0040: 0000 5002 000a 0000 0000 0001 0000 0x0040: 0000 5002 000a 0000 0000 0001 0000 0x0040: 0000 5002 000a 0000 0000 0001 0000 0x0040: 0000 5002 000a 0000 0000 0000 0000 0000</pre>	y fail as well if the NIC doesn't support any rate limiting).	
<pre>[INF0] dst_ip is 2001:558:feed:10:0:2:: 21:11:18.039636 ETH backs feed:10:0 > backs feed:0:0:0:0000 > 2001:0558:feed:0010:0000:0002:0000 ver 6 tc 0 fl 0 len 20 next 0x06 (unknown) t 164 TCP 1025 > 80 seq# 1 ack# 0 offset 0x5 reserved 0x00 flags 0x02 [- - - -]SYN -] win 10 cksum 0x0000 urg 0 [] 0x0000: back afeb cer2 back afeb cer2 back afeb 01a 86dd 6000 0x0010: 0000 0014 0640 fdaa 0000 0001 000c 0000 0x0020: 0011 0000 0000 2001 0558 feed 0010 0000 0x0020: 0011 0000 0000 2001 0558 feed 0010 0000 0x0030: 0002 0000 0401 0056 0000 0001 0000 0x0040: 0000 5002 0000 0401 0056 0000 0001 0000 0x0040: 0000 5002 0000 0401 0058 feed 0010 0000 0x0040: 0000 5002 0000 0401 0050 0000 0001 0000 0x0040: 0000 5002 0000 0000 0001 0000 0x0040: 0000 5002 0000 0000 0401 0058 feed 010 0000 0x0040: 0000 5002 0000 0400 0000 000 000 0000 0x0040: 0000 5002 0000 0400 0000 0000 0000 0x0040: 0000 5002 0000 0400 0000 0000 0000 0000</pre>	[WARN] global rate limiting is not supported by the hardware or driver	
<pre>21:11:18.030636 ETM ba:dc:sf:eb:001;a1 > ba:dc:sf:eb:001;2 type 0x86dd (TP6) TP6 fdaa:0000:0001:000c:0000:0011:0000:0000 > 2001:0558:feed:0010:0000:0002:0000:0000 ver 6 tc 0 fl 0 len 20 next 0x06 (unknown) t tl 64 TCP 1025 > 80 seq# 1 ack# 0 offset 0x5 reserved 0x00 flags 0x02 [- - - SYN -] win 10 cksum 0x0000 urg 0 [] 0x00000: bacc afeb cef2 bacc afeb 01al 86dd 6000 0x0010: 0000 0014 0640 fdaa 0000 0001 000c 0000 0x0020: 0011 0000 0000 2001 0558 feed 0010 0000 0x0020: 0011 0000 0000 2001 0558 feed 0010 0000 0x0020: 0011 0000 0000 2001 0558 feed 0010 0000 0x0020: 0011 0000 0000 2001 0558 feed 0010 0000 0x0040: 0000 5002 000a 0000 0001 0000 0x0040: 0000 5002 000a 0000 0001 0000 0x0040: 0000 5002 000a 0000 0001 (Device: id=0) TX: nan (StdDev 0.00) Mpps, nan (StdDev 0) Mbit/s (nan Mbit/s with framing), total 1 packets with 74 bytes (incl. C RC) [Client] root@cli_west:/opt/moongen/dpdk/containernet#</pre>		
<pre>IP6 fdaa:0000:0001:000c:0000:0011:0000:0000 > 2001:0558:feed:0010:0000:0002:0000:0000 ver 6 tc 0 fl 0 len 20 next 0x06 (unknown) t tl 64 ICP 1025 > 80 seq# 1 ack# 0 offset 0x5 reserved 0x00 flags 0x02 [-]-]-]-SYN]-] win 10 cksum 0x0000 urg 0 [] 0x0000: badc afeb eef2 badc afeb 010 860d 6000 0x0010: 0000 0014 0640 fdaa 0000 0001 000c 0000 0x0020: 0011 0000 0000 2001 0558 feed 0010 0000 0x0030: 0002 0000 0000 0401 0050 0000 0001 0000 0x0040: 0000 5002 0000 0000 0001 0000 (Device: id=0] TX: nan (StdDev 0.00) Mpps, nan (StdDev 0) Mbit/s (nan Mbit/s with framing), total 1 packets with 74 bytes (incl. C RC) [Client] root@cli_west:/opt/moongen/dpdk/containernet#</pre>		
<pre>tl 64 TCP 1025 > 80 seq# 1 ack# 0 offset 0x5 reserved 0x00 flags 0x02 [- - - SYN -] win 10 cksum 0x0000 urg 0 [] 0x0000: badc afeb 0012 badc afeb 01al 06d0 6000 0x0010: 0000 0014 0640 fdaa 0000 0001 000c 0000 0x0020: 0011 0000 0000 2001 0558 feed 0010 0000 0x0040: 0002 0000 0000 0401 0050 0000 0001 0000 0x0040: 0000 5002 000a 0000 0000 (Device: id=0] TX: nan (StdDev 0.00) Mpps, nan (StdDev 0) Mbit/s (nan Mbit/s with framing), total 1 packets with 74 bytes (incl. C RC) [Client] root@cli_west:/opt/moongen/dpdk/containernet#</pre>		
0x0000: badc afeb cef2 badc afeb 01a1 86dd 6000 0x0010: 0000 0014 0640 fdaa 0000 0001 000c 0000 0x0020: 0011 0000 0000 2001 0558 feed 0010 0000 0x0030: 0002 0000 0000 0401 0050 0000 0001 0000 0x0040: 0000 5002 000a 0000 0000 (Device: id=0) TX: nan (StdDev 0.00) Mpps, nan (StdDev 0) Mbit/s (nan Mbit/s with framing), total 1 packets with 74 bytes (incl. C RC) [Client] root@cli_west:/opt/moongen/dpdk/containernet#		
0x0010: 0000 0014 0640 fdaa 0000 0000 0000 0x0020: 0011 0000 0000 2001 0558 feed 0010 0000 0x0030: 0002 0000 0401 0050 0000 0001 0000 0x0040: 0000 5002 0000 0000 0000 0000 (pevice: id=0 TX: nan (StdDev 0.00) Mpps, nan (StdDev 0) Mbit/s (nan Mbit/s with framing), total 1 packets with 74 bytes (incl. C RC) [Client] root@cli_west:/opt/moongen/dpdk/containernet#		
0x0020: 0011 0000 0000 2001 0558 feed 0010 0000 0x0030: 0002 0000 0000 0401 0050 0000 0001 0000 0x0040: 0000 5002 000a 0000 0000 [Device: id=0] 1X: nan (StdDev 0.00) Mpps, nan (StdDev 0) Mbit/s (nan Mbit/s with framing), total 1 packets with 74 bytes (incl. C RC) [Client] root@cli_west:/opt/moongen/dpdk/containernet#		
0x0030: 0002 0000 0401 0000 0000 0x0040: 0000 5002 0000 0000 [Device: id=0] 1X: nan (StdDev 0.00) Mbit/s (nan Mbit/s with framing), total 1 packets with 74 bytes (incl. C RC) [Client] root@cli_west:/opt/moongen/dpdk/containernet#		
0x0040: 0000 5002 000a 0000 0000 [Device: id=0] TX: nan (StdDev 0.00) Mpps, nan (StdDev 0) Mbit/s (nan Mbit/s with framing), total 1 packets with 74 bytes (incl. C RC) [Client] root@cli_west:/opt/moongen/dpdk/containernet#		
<pre>[Device: id=0] TX: nan (StdDev 0.00) Mpps, nan (StdDev 0) Mbit/s (nan Mbit/s with framing), total 1 packets with 74 bytes (incl. C RC) [Client] root@cli_west:/opt/moongen/dpdk/containernet#</pre>		
RC) [Client] root@cli_west:/opt/moongen/dpdk/containernet#		
RC) [Client] root@cli_west:/opt/moongen/dpdk/containernet#	[Device: id=0] IX: nan (StdDev 0.00) Mpps, nan (StdDev 0) Mbit/s (nan Mbit/s with framing), total 1 packets with 74 bytes (incl. C	
	RC)	
Linkedin.com/in/andrewkewang	[Client] root@cli_west:/opt/moongen/dpdk/containernet#	
COMCASI 24		24
	COMCASI	24

https://www.linkedin.com/in/andrewkewang



Summary

- We wanted to write network functions
 - DPDK, Netbricks, MoonGen
- Environment to develop
 - VMs, Labs, Production networks
- Containernet
 - Scalable
 - Configurable
 - Emulation: run real software stack
 - Access to all hosts in the network





Thank you

https://www.linkedin.com/in/andrewkewang

