40 Gbps IPsec...
on Commodity Hardware

Jim Thompson
Founder & CTO
Netgate
REALITY BEFORE 2017

Packet Size

256
1024

Throughput (Gbps)

1
10
40

THE UNBREAKABLE BARRIER

Want small packets and/or high(er) throughput? Pay the Piper (Vendor)
Memory access “resource cost” has remained relatively constant
Oh, and one more thing...

Kernel changes are expensive, painful, and far-reaching. Writing your own stack is difficult, and expensive.
And then, a breakthrough...

And $DEITY said, “Let there be VPP and DPDK.”

Uhhh...say what?
VPP & DPDK

Vector Packet Processing
- VPP takes a vector of packets from NIC
- Stuffs them into RAM
- Runs set of functions against packets
- Packets move as a group
- Instruction cache remains hot
- First packet penalty, 2\textsuperscript{nd} packet is cheap
- Apache 2 licensed

Data Plane Development Kit
- Set of data plane libraries and network interface controller drivers for fast packet processing
- Programming framework for Intel and ARM processors
- Scales from Intel Atom to Xeon and more
- BSD licensed

The wall just fell down...
Is this one of those eccentric ideas?

Answer:  Nope. VPP was created and open sourced by...
Why would they do such a thing?

No choice. The world is changing fast.

Accept it, or accept your fate.
Test Harness

- Xeon E3-1275 v3
- 32GB RAM
- Intel XL710 Ethernet
- 1U server

- i7-6950X CPU, overclocked to 3.5GHz (10C)
- ASUS X99 board
- Water-cooled to avoid thermal throttling
- 32GB RAM
- XL710 40Gbps Ethernet
- CPIC (QAT) crypto offload card (Netgate)
# Test Results

<table>
<thead>
<tr>
<th>Context</th>
<th>Crypto processing</th>
<th>Crypto/AES/AD algorithm</th>
<th>Integrity algorithm</th>
<th># of SAs</th>
<th>Total # of streams</th>
<th>Throughput TCP 1500</th>
<th># of samples</th>
<th>pktgen UDP 512</th>
<th>pktgen UDP 1500</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kernel</td>
<td>AES-NI</td>
<td>AES-CBC-128</td>
<td>SHA1</td>
<td>1</td>
<td>2.03 Gbps</td>
<td>16.51 kbps</td>
<td>4</td>
<td>267 kbps</td>
<td>749 kbps</td>
</tr>
<tr>
<td>Kernel</td>
<td>AES-NI</td>
<td>AES-CBC-128</td>
<td>SHA1</td>
<td>1</td>
<td>2.03 Gbps</td>
<td>16.51 kbps</td>
<td>4</td>
<td>267 kbps</td>
<td>749 kbps</td>
</tr>
<tr>
<td>Kernel</td>
<td>AES-NI</td>
<td>AES-CBC-128</td>
<td>SHA1</td>
<td>6</td>
<td>25.25 Gbps</td>
<td>16.07 kbps</td>
<td>6</td>
<td>246 kbps</td>
<td>246 kbps</td>
</tr>
<tr>
<td>Kernel</td>
<td>AES-NI</td>
<td>AES-GCM-128</td>
<td>SHA1</td>
<td>1</td>
<td>4.84 Gbps</td>
<td>16.07 kbps</td>
<td>6</td>
<td>976 kbps</td>
<td>976 kbps</td>
</tr>
<tr>
<td>Kernel</td>
<td>QAT</td>
<td>AES-OBC-123</td>
<td>SHA1</td>
<td>1</td>
<td>2.03 Gbps</td>
<td>16.51 kbps</td>
<td>4</td>
<td>267 kbps</td>
<td>749 kbps</td>
</tr>
<tr>
<td>Kernel</td>
<td>QAT</td>
<td>AES-OBC-123</td>
<td>SHA1</td>
<td>1</td>
<td>2.03 Gbps</td>
<td>16.51 kbps</td>
<td>4</td>
<td>267 kbps</td>
<td>749 kbps</td>
</tr>
<tr>
<td>Kernel</td>
<td>QAT</td>
<td>AES-OBC-123</td>
<td>SHA1</td>
<td>6</td>
<td>25.25 Gbps</td>
<td>16.07 kbps</td>
<td>6</td>
<td>246 kbps</td>
<td>246 kbps</td>
</tr>
<tr>
<td>User</td>
<td>VPP native [OpenSSL 1.0.1]</td>
<td>AES-GCM-128</td>
<td>SHA1</td>
<td>1</td>
<td>2.03 Gbps</td>
<td>16.51 kbps</td>
<td>4</td>
<td>267 kbps</td>
<td>749 kbps</td>
</tr>
<tr>
<td>User</td>
<td>VPP native [OpenSSL 1.0.1]</td>
<td>AES-CBC-128</td>
<td>SHA1</td>
<td>1</td>
<td>2.03 Gbps</td>
<td>16.51 kbps</td>
<td>4</td>
<td>267 kbps</td>
<td>749 kbps</td>
</tr>
<tr>
<td>User</td>
<td>VPP native [OpenSSL 1.0.1]</td>
<td>AES-CBC-128</td>
<td>SHA1</td>
<td>6</td>
<td>25.25 Gbps</td>
<td>16.07 kbps</td>
<td>6</td>
<td>246 kbps</td>
<td>246 kbps</td>
</tr>
<tr>
<td>User</td>
<td>VPP AESNI MB cryptodev</td>
<td>AES-GCM-128</td>
<td>SHA1</td>
<td>1</td>
<td>2.03 Gbps</td>
<td>16.51 kbps</td>
<td>4</td>
<td>267 kbps</td>
<td>749 kbps</td>
</tr>
<tr>
<td>User</td>
<td>VPP AESNI MB cryptodev</td>
<td>AES-CBC-128</td>
<td>SHA1</td>
<td>1</td>
<td>2.03 Gbps</td>
<td>16.51 kbps</td>
<td>4</td>
<td>267 kbps</td>
<td>749 kbps</td>
</tr>
<tr>
<td>User</td>
<td>VPP AESNI MB cryptodev</td>
<td>AES-CBC-128</td>
<td>SHA1</td>
<td>6</td>
<td>25.25 Gbps</td>
<td>16.07 kbps</td>
<td>6</td>
<td>246 kbps</td>
<td>246 kbps</td>
</tr>
<tr>
<td>User</td>
<td>VPP AESNI MB cryptodev</td>
<td>AES-GCM-128</td>
<td>SHA1</td>
<td>1</td>
<td>2.03 Gbps</td>
<td>16.51 kbps</td>
<td>4</td>
<td>267 kbps</td>
<td>749 kbps</td>
</tr>
<tr>
<td>User</td>
<td>VPP AESNI MB cryptodev</td>
<td>AES-CBC-128</td>
<td>SHA1</td>
<td>6</td>
<td>25.25 Gbps</td>
<td>16.07 kbps</td>
<td>6</td>
<td>246 kbps</td>
<td>246 kbps</td>
</tr>
<tr>
<td>User</td>
<td>VPP QAT cryptodev</td>
<td>AES-CBC-123</td>
<td>SHA1</td>
<td>1</td>
<td>2.03 Gbps</td>
<td>16.51 kbps</td>
<td>4</td>
<td>267 kbps</td>
<td>749 kbps</td>
</tr>
<tr>
<td>User</td>
<td>VPP QAT cryptodev</td>
<td>AES-CBC-123</td>
<td>SHA1</td>
<td>6</td>
<td>25.25 Gbps</td>
<td>16.07 kbps</td>
<td>6</td>
<td>246 kbps</td>
<td>246 kbps</td>
</tr>
<tr>
<td>User</td>
<td>VPP QAT cryptodev</td>
<td>AES-CBC-123</td>
<td>SHA1</td>
<td>1</td>
<td>2.03 Gbps</td>
<td>16.51 kbps</td>
<td>4</td>
<td>267 kbps</td>
<td>749 kbps</td>
</tr>
<tr>
<td>User</td>
<td>VPP QAT cryptodev</td>
<td>AES-CBC-123</td>
<td>SHA1</td>
<td>6</td>
<td>25.25 Gbps</td>
<td>16.07 kbps</td>
<td>6</td>
<td>246 kbps</td>
<td>246 kbps</td>
</tr>
<tr>
<td>User</td>
<td>None</td>
<td>None</td>
<td>None</td>
<td>1</td>
<td>2.03 Gbps</td>
<td>16.51 kbps</td>
<td>4</td>
<td>267 kbps</td>
<td>749 kbps</td>
</tr>
<tr>
<td>User</td>
<td>None</td>
<td>None</td>
<td>None</td>
<td>1</td>
<td>2.03 Gbps</td>
<td>16.51 kbps</td>
<td>4</td>
<td>267 kbps</td>
<td>749 kbps</td>
</tr>
<tr>
<td>User</td>
<td>None</td>
<td>None</td>
<td>None</td>
<td>6</td>
<td>25.25 Gbps</td>
<td>16.07 kbps</td>
<td>6</td>
<td>246 kbps</td>
<td>246 kbps</td>
</tr>
<tr>
<td>User</td>
<td>None</td>
<td>None</td>
<td>None</td>
<td>1</td>
<td>2.03 Gbps</td>
<td>16.51 kbps</td>
<td>4</td>
<td>267 kbps</td>
<td>749 kbps</td>
</tr>
<tr>
<td>User</td>
<td>None</td>
<td>None</td>
<td>None</td>
<td>6</td>
<td>25.25 Gbps</td>
<td>16.07 kbps</td>
<td>6</td>
<td>246 kbps</td>
<td>246 kbps</td>
</tr>
</tbody>
</table>

- **Full Encryption**: 
  - 36 Gbps (which is not 40 Gbps...)
Where is the other 4 gbps?

- Original packet + padding has to fit in 1500 - 56 = 1444 bytes
- Length must be multiple of 16. 1444 mod 16 = 4, so 1440 is largest size for original packet + padding
- 40 bytes of original packet are headers, at least 2 bytes of padding added for pad length/next header
- So 1398 is max size of data that can be sent in the original packet without overflowing the 1500 bytes
- Therefore: ratio of payload to data transmitted = 1398/1538 = 0.90897
- Which equates to a maximum throughput of 36.36 Gbps (no tuning!)
- The highest result we saw was 36.3 Gbps
Ready for production?

VPP Lab testing + NETCONF/RESTCONF/CLI control plane

Almost…
And it’s all free & open source?

Free

Roll your own control plane

Free

$ for Enterprise Support Subscription

Programmers need to eat, too…
And why do I care?

The “middle man” cost is collapsed, yet again
What’s Next?

- **100 Gbps**
- **KVM**
- **ARM v8 (NXP, Cavium, Marvell)**
- **Intel Communications SKUs (C3000, etc)**
- **OpenVPN re-write**
References

- DPDK
  - http://www.dpdk.org
- VPP
  - https://wiki.fd.io/view/VP/What_is_VPP
- CPIC
- CLIXON - NETCONF/RESTCONF/CLI
  - http://www.clicon.org
- “Building a Behemoth Router”
  - https://www.netgate.com/blog/building-a-behemoth-router.html
Thank You

Jim Thompson
Founder & CTO
Netgate
1. Up until 2 years ago, 40 Gbps from off the shelf hardware was unthinkable
   – Processors vs. memory - Role Reversal
   – Packet Size - Big trouble comes in small package

2. Why unthinkable?
   – Processor cost
   – Moving data in and out of memory cost
   – Memory cost

3. But we have 40 Gbps boxes. How?
   – Specialized processors + offload processor
   – Memory has gotten much cheaper
   – Could not unhinge from specialty silicon because of kernel constraints
   – $$ - High 5 figure box prices and up

4. What has changed?
   – Processors vs. memory - cost Role Reversal
   – But we needed innovation in the kernel for h/w to truly get there
   – But kernels are slow to change and for a good reason
     • It’s a rip and replace game

5. So what innovation has made it possible?
   – VPP
   – What is that?

6. Is this a “Wankel engine”?
   – Nope. It came from Cisco

7. Why in the world would they let it go open?
   – Docker pressure

8. OK, and it’s ready for prime time?
   – Yes and no.
   – Yes - look at this packet throughout data...
   – No - control plane sw still needed

9. And this is all open source?
   – Yes and no
   – There will be a version of pfSense that is a rocket - very soon
   – And you can of course build your own control plane with puppet, open stack et al
   – But’s it really tricky
   – Ours will be easier to use and integrate, but it won’t be free as it’s our value add
   – And we’ll have eprise service subscriptions to boot

10. Got it. Now, why do I care about all this?
    – The network infra game is changing fast
    – The middle box is gone
    – Security is shifting
    – Network infra price collapse is coming
    – Budget dollars will shift further to the edges
    – Network perimeter and zone fw and routing will not go away, but it will go hard to the hoop on high throughput for any biz app (think small packets) and ACLs will only increase
    – Open source is key ingredient in that movement
    – And in the end, OS can meet the challenge AND it will save beaucoup $$ for e-prise, education and gov