**[XenServer Network Tuning](%22%20%5Co%20%22Permalink%20to%20XenServer%20Network%20Tuning)**

**Date** Mon 02 July 2012

**Improving performance by disabling TCP Offloading**

In XenServer 5.6+ there are several issues that can come up when you begin virtualizing a lot of instances. One of these issues that will become more apparent as your Virtualized infrastructure grows is checksumming and overall network throughput. Depending on the Hardware that you have installed in your Server, performance of network devices can vary. One of the very best features of XenServer is the ability to efficiently handle network throughput in software.

When leveraged properly the virtualized infrastructure can perform just as well as physical hardware. To take advantage of the software created by XenServer you will have ensure that the system does not attempt to handle networking by means of hardware offloading.

**TCP Offloading**

*(TCP offload) TCP Offload Engine or TOE is a technology used in network interface cards (NIC) to offload processing of the entire TCP/IP stack to the network controller.*

**An Indication of issues**

If you dump traffic on your XenServer host and see that there are issues with checksumming not matching, causing checksum errors, and or if you notice a drop in network performance, you can disable TCP offloading. This offloading needs to be done on the host's Network interfaces, as well as the Instances' VIF and PIF. Here is an example of how you can find these checksum errors as well as how you can manually disable TCP offloading for the host's Network Adapter.

tcpdump -i eth0 -v -nn | grep incorrect

Wth this command you will see some output that looks similar to this :

tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes

16:38:07.676943 IP (tos 0x0, ttl 64, id 60844, offset 0, flags [DF], proto TCP (6), length 60) xxx.xxx.xxx.xxx.46455 > yyy.yyy.yyy.yyy.80: S, cksum 0x4b8f (incorrect (-> 0x1f20), XXXXXXXXXX:YYYYYYYYYY(0) win 5840 <mss 1460,sackOK,timestamp 731623207 0,nop,wscale 4>

16:38:07.711402 IP (tos 0x0, ttl 64, id 28467, offset 0, flags [DF], proto TCP (6), length 52) xxx.xxx.xxx.xxx.41269 > yyy.yyy.yyy.yyy.80: ., cksum 0x6a18 (incorrect (-> 0xa1f1), 1:1(0) ack 773 win 552 <nop,nop,timestamp XXXXXXXX YYYYYYYY>

16:38:07.726013 IP (tos 0x0, ttl 64, id 60845, offset 0, flags [DF], proto TCP (6), length 40) xxx.xxx.xxx.xxx.46455 > yyy.yyy.yyy.yyy.80: ., cksum 0x4b7b (incorrect (-> 0x328c), 1:1(0) ack 1 win 5840

The output that you are really looking for is something like **incorrect (-> 0x6e35)** this is showing that there are checksums that are failing to be received correctly.

**How you can improve Network Performance**

With the previously found information we can see that the host is having TCP offloading issues. To begin fixing these issues we need to check to see what has been been marked as active on the Network Interfaces for offloading. Using the ethtool command we can see what is enabled on a select device. Please note, if you have multiple network devices, you will need to run this command for all network devices.

ethtool -k eth0

 Offload parameters **for** eth0:

 rx-checksumming: on

 tx-checksumming: on

 scatter-gather: on

 tcp-segmentation-offload: on

 udp-fragmentation-offload: off

 generic-segmentation-offload: on

 generic-receive-offload: off

 large-receive-offload: off

Being that we can see that there are several Offloading parameters turned ON, we need to turn them OFF. To do this we use the same ethtool command with the -K flag.

ethtool -K eth0 rx off tx off sg off tso off ufo off gso off gro off lro off;

**Just like the check command, you will need to run this for all network devices.**

* If you want to disable ALL Offloading for ALL interfaces in one command you can. Here is a simple for loop that will disable all offloading for all network devices.

**for** ETHADP in `ip addr | awk '/eth[0-9]/ {print $2}' | sed 's/://g'`

 **do**

ethtool -K **${**ETHADP**}** rx off tx off sg off tso off ufo off gso off gro off lro off

**done**

Once you have disabled TCP offloading on the Network Interface I would also recommend that you disable offloading for the instance, which can be set in the "Other-Config" parameters.

To disable the TCP offloading on a specific instance you have to first get the VM UUID number followed by the VIF UUID number. To do this you will be using the xe command.

**Here is how you Disable TCP Offloading as a PIF Parameter**

1. The first step is to get all of the PIF Devices.

xe pif-list | awk '/uuid/ {print $5}'| sed '/^$/d'

**This will provide you with ALL of the UUID's for ALL of the network devices. If you want to have TCP offloading completely off you will have to disable it for all network devices.**

1. Now that you have the Devices you will have to set the parameters for all of the devices.

xe pif-param-set uuid=$PIFUUID other-config:ethtool-gso="off";

xe pif-param-set uuid=$PIFUUID other-config:ethtool-ufo="off";

xe pif-param-set uuid=$PIFUUID other-config:ethtool-tso="off";

xe pif-param-set uuid=$PIFUUID other-config:ethtool-sg="off";

xe pif-param-set uuid=$PIFUUID other-config:ethtool-tx="off";

xe pif-param-set uuid=$PIFUUID other-config:ethtool-rx="off";

**Remember to replace "$PIFUUID" with the PIF UUID number that you got in the previous command.**

* If you want to do this for ALL of the devices you can run this as a simple for loop which will disable offloading for ALL PIF Devices.

**for** PIFUUID in`xe pif-list | awk '/uuid/ {print $5}' | sed '/^$/d'`

 **do**

xe pif-param-set uuid=$PIFUUID other-config:ethtool-gso="off";

 xe pif-param-set uuid=$PIFUUID other-config:ethtool-ufo="off";

 xe pif-param-set uuid=$PIFUUID other-config:ethtool-tso="off";

 xe pif-param-set uuid=$PIFUUID other-config:ethtool-sg="off";

 xe pif-param-set uuid=$PIFUUID other-config:ethtool-tx="off";

 xe pif-param-set uuid=$PIFUUID other-config:ethtool-rx="off";

**done**

* To verify that the setting have taken effect on the PIF, use the xe command and check the network settings for the specified network device.

xe pif-param-list uuid=$PIFUUID | grep "other-config"

The Output will look something similar to :

other-config (MRW): ethtool-rx: "off"; ethtool-tx: "off"; ethtool-sg: "off"; ethtool-tso: "off"; ethtool-ufo: "off"; ethtool-gso: "off"

**Here is how you Disable TCP Offloading as a VIF Parameter**

1. Here is how to get the VM UUID

xe vm-list name-label=$INSTANCEID | awk '/uuid/ {print $5}'

**Remember to replace "$INSTANCEID" with the Instance Name.**

1. Here is how to get the VIF UUID

xe vif-list vm-uuid=$VMUUID | awk '/uuid/ {print $5}' | sed '/^$/d'

**Remember to replace "$VMUUID" with the VM UUID number that you got in the previous command.**

1. Now that you have both the VM UUID and the VIF UUID you can now set the parameters to disable TCP Offloading on a particular instance.

xe vif-param-set uuid=$VIFUUID other-config:ethtool-gso="off";

xe vif-param-set uuid=$VIFUUID other-config:ethtool-ufo="off";

xe vif-param-set uuid=$VIFUUID other-config:ethtool-tso="off";

xe vif-param-set uuid=$VIFUUID other-config:ethtool-sg="off";

xe vif-param-set uuid=$VIFUUID other-config:ethtool-tx="off";

xe vif-param-set uuid=$VIFUUID other-config:ethtool-rx="off"

**Remember to replcae the "$VIFUUID" with the VIF UUID that you had gotten in the previous command.**

* If you want to do this for ALL instances on ALL of the devices you can run this as a simple for loop which will disable offloading for ALL VIF Devices for ALL registered Instances.

**for** INSTANCEID in`xe vm-list | awk '/name-label/' | grep -v "Control domain" | awk '{print $4}'`

 **do**

VMUUID=`xe vm-list name-label=**${**INSTANCEID**}** | awk '/uuid/ {print $5}'`

 **for** VIFUUID in`xe vif-list vm-uuid=$VMUUID | awk '/uuid/ {print $5}' | sed '/^$/d'`

 **do**

xe vif-param-set uuid=$VIFUUID other-config:ethtool-gso="off"

 xe vif-param-set uuid=$VIFUUID other-config:ethtool-ufo="off"

 xe vif-param-set uuid=$VIFUUID other-config:ethtool-tso="off"

 xe vif-param-set uuid=$VIFUUID other-config:ethtool-sg="off"

 xe vif-param-set uuid=$VIFUUID other-config:ethtool-tx="off"

 xe vif-param-set uuid=$VIFUUID other-config:ethtool-rx="off"

 **done**

**done**

* To verify that the setting have taken effect on the VIF, use the xe command and check the network settings for the specified VIF.

xe vif-param-list uuid=023247ab-e2e5-6ac6-ca5d-8d2f323665e0 | grep "other-config"

The output will look similar to this :

other-config (MRW): ethtool-rx: "off"; ethtool-tx: "off"; ethtool-sg: "off"; ethtool-tso: "off"; ethtool-ufo: "off"; ethtool-gso: "off"

**After you have set the parameters you will have to restart the instance before the setting take effect.**

**Conclusion**

As you can see, the output shows that Network Offloading has been disabled in the PIF and VIF and is represented by the "ethtool" syntax present in the other-config parameters.

Now that you have disabled ALL of the network offloading on ALL devices and ALL instances, and have restarted the instances to ensure that it is in fact not using offloading, your network performance with regard to overall throughput should increase. Secondly, issues where networking is failing TCP checksumming should be resolved.