Copy of Poor Network Performance Between CDP Server and Host on a XEN Virtual Machine, or Connection Closed due to (ClosedChannelException)

Copy of Poor Network Performance Between CDP Server and Host on a XEN Virtual Machine, or Connection Closed due to (ClosedChannelException)

This issue is related to the Large Send Offload setting on the network adapter, and is most commonly experienced when using Broadcom NICs.

More information is available here regarding disabling this setting on a host with a physical NIC: http://wiki.r1soft.com/display/kb/Back+Up+Stopped

Symptom

Windows and Linux Server hosts stopped backing up due to this error. The error description from buagent:

> **Network timeout occurred after (3600) seconds closing connection to (XXX.XXX.XXX.XXX). (ClosedChannelException)**

Alternatively, you may notice poor network performance between the CDP Server and host as if they are both running on a XEN Virtual Machine.

Cause

On a physical machine, the overhead associated with packet receipt verification (ie: computing checksums), is usually handed off to be calculated by the physical network card itself, effectively relieving the Operating System of these duties. In a Virtual Machine, there is no real physical Network Adapter with a hardware chipset that the Operating System can use to pass off the task of calculating the checksums.

Resolution

Linux Virtual Machine

To check the settings (on each domU), you can use ethtool.

```
# ethtool \-k eth0
Offload parameters for eth0:
Cannot get device rx csum settings: Operation not supported
Cannot get device udp large send offload settings: Operation not supported
rx-checksumming: off
tx-checksumming: on
scatter-gather: on
tcp segmentation offload: off
udp fragmentation offload: off
generic segmentation offload: off
```

If TX-checksumming is "on", you can turn it off with the following command:

```
# ethtool \-K eth0 tx off
```

You can use ethtool again to verify the change:

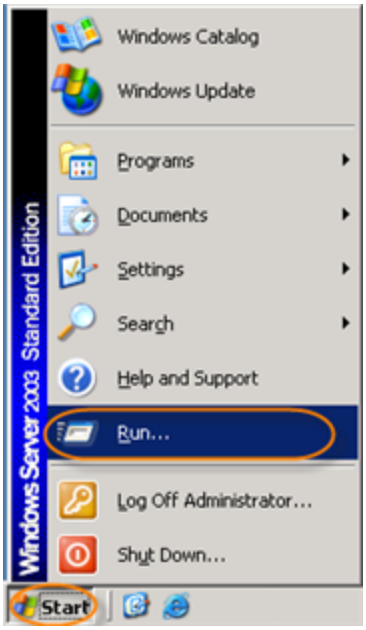
```
# ethtool \-k eth0
```

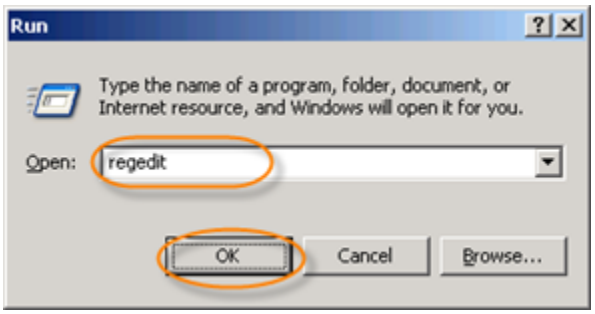
Keep in mind that these changes will be lost following a reboot, so you may want to add them to a preferred script that will run after the network is up. (You can add them to /etc/rc.local which starts after networking is up.)

Windows Virtual Machine

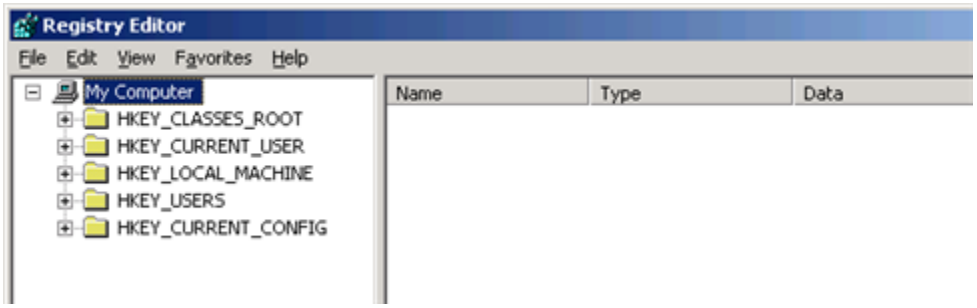
To disable the TCP task offload option in Windows, you will need to create a registry DWORD.

1. Click Start › Search and type "regedit" (Start › Run in Windows XP/2003/2000).



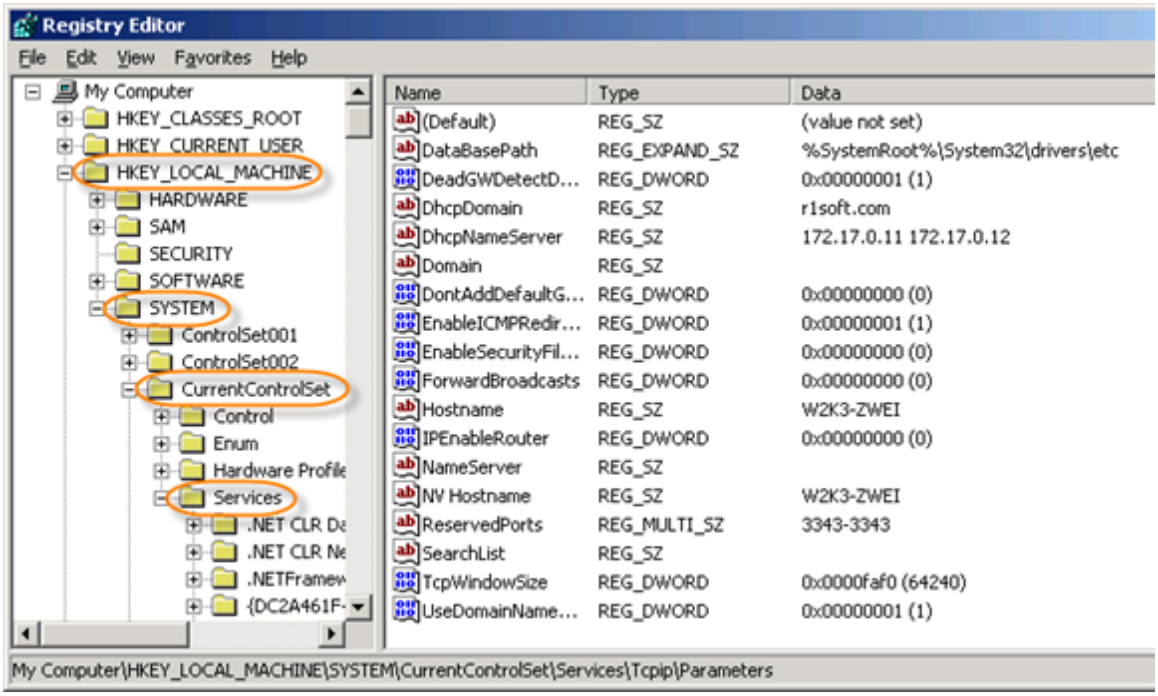


This launches the Windows Registry Editor.

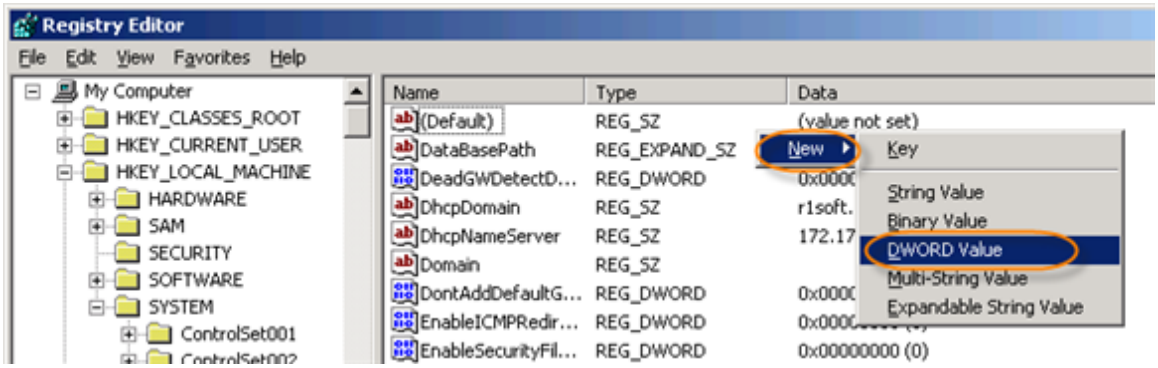


2. Browse to the following registry key:

HKEY_LOCAL_MACHINE › SYSTEM › CurrentControlSet › Services › Tcpip › Parameters

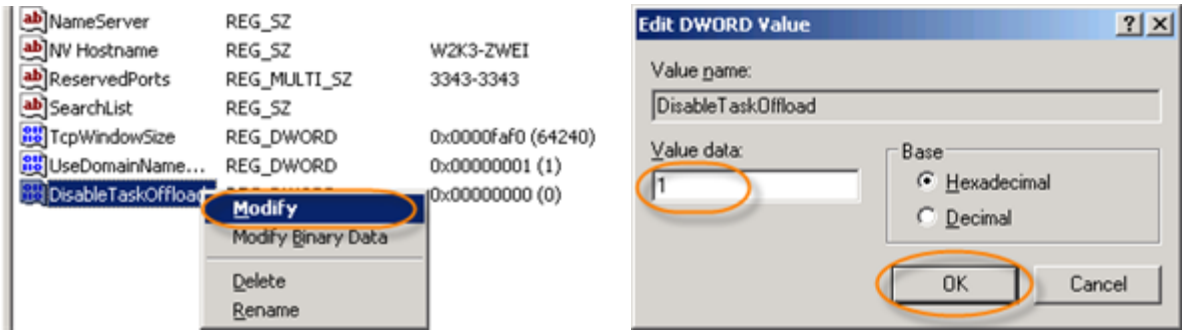


3. Right-click in the pane on the right and choose New › DWORD (32-bit value).



4. Name it "DisableTaskOffload".

And set its value to "1" (Either hex or decimal).



Setting it to"1" will disable TCP Task Offloading.

If you wish to re-enable it at a later time, set it "0" or simply delete the the "DisableTaskOffload" DWORD.