Best Practices
with
StarWind Replication

©2012 NetEx Software
6420 Sycamore Lane N.
Suite 300
Maple Grove, MN 55369
http://www.netex.com

HyperIP in a Network with StarWind Replication

Implementation of HyperIP and StarWind Replication

The addition of HyperIP into the network using gateway mode does not require any changes to the StarWind Replication application. Changes to the host routing tables are necessary for HyperIP to get inserted in the data path. The routing changes are described later in this document.

Installation of HyperIP

Refer to the latest documentation, FAQ, and Updates at the HyperIP website to get the latest news regarding HyperIP releases: http://www.netex.com/support/products/hyperip.

HyperIP is a WAN Optimization Virtual Appliance and can be requested from this URL: http://www.netex.com/hyperip/evaluation-request.
Fill out the Evaluation Agreement, accept the terms, and a download link will be sent to you to download the installation package. HyperIP is keyed and instructions on how to obtain keys are included in the installation package.

Video tutorials are available to assist in stepping through the configuration of HyperIP: http://www.netex.com/support/product-support/hyperip-6-0-screencasts

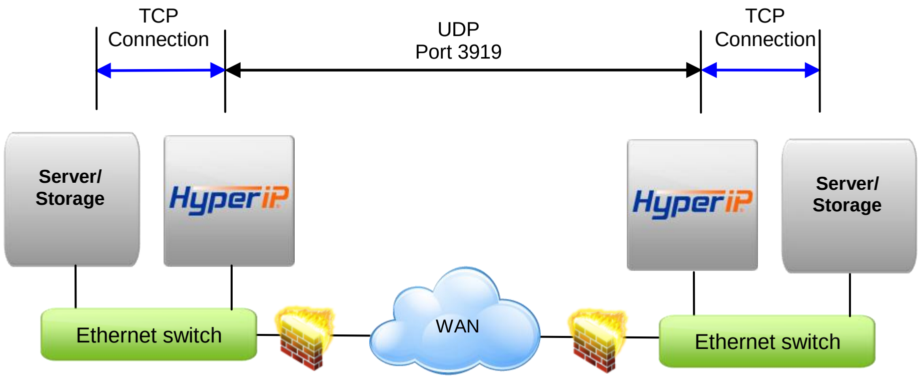
Adding HyperIP into your network

HyperIP improves the performance of backup and replication applications over your IP WAN. **HyperIP does not alter application protocols nor modify any file systems.** It efficiently moves block or file data over the IP WAN under any network conditions.

HyperIP also provides:

- support of WAN speeds scaling from 1-800 Mb/s
- virtual footprint
- adaptive lossless block level compression
- time of day rate controls for changing throughput requirements

HyperIP requires at least two appliances (virtual), one residing on each side of the WAN, as shown in the figure below. Multiple servers and storage at each site can utilize the HyperIP data path. HyperIP can also be deployed in a hub or mesh configuration.



HyperIP terminates TCP connections locally and tunnels the data between HyperIPs using UDP port 3919. **Network devices filtering IP traffic in the data path between the HyperIPs must be configured to allow UDP port 3919.**

HyperIP must be *<u>in</u>* the data path to optimize the movement of data. HyperIP connects to a (virtual) LAN switch with a single Gigabit Ethernet NIC and has two modes of operation to facilitate being inserted into the data path:

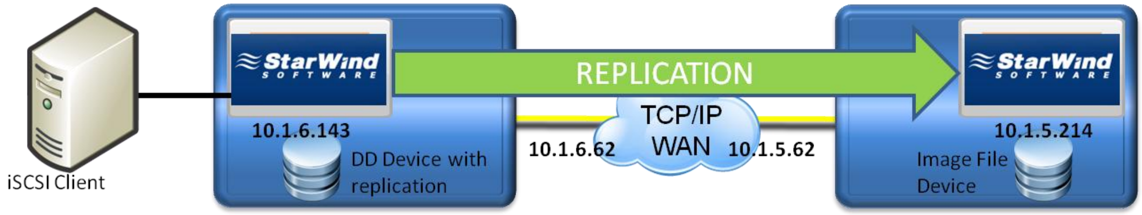
- **Gateway Mode: User must add route statements in the *data movers* (application servers, storage devices, etc.)** defining HyperIP as the IP gateway for the destination IP addresses or networks. Alternatively, these IP route statements or redirect filters, may be configured in a router. Gateway mode **requires users to define HyperIP intercepts** based on IP addresses, TCP ports and/or protocols to determine what traffic to act on.

- **Proxy Mode: HyperIP requires additional local IP addresses (proxy) which represent remote IP addresses of the application servers or storage devices.** This local proxy IP address is then used to communicate with the remote application. HyperIP is configured with a 1:1 mapping in which each destination IP address requires an associated local proxy address. *Applications that do not support Network Address Translation (NAT) must use the HyperIP gateway mode.*

Each HyperIP requires its own key associated with the HyperIP serial number. You must connect to the user interface on each HyperIP to retrieve its serial number and complete the form at: http://www.netex.com/hyperip/hyperip-key-request to request the key.
For further explanation on the features/functionality of HyperIP see the HyperIP User guide at: http://www.netex.com/support/products/hyperip-docs

StarWind Replication before HyperIP

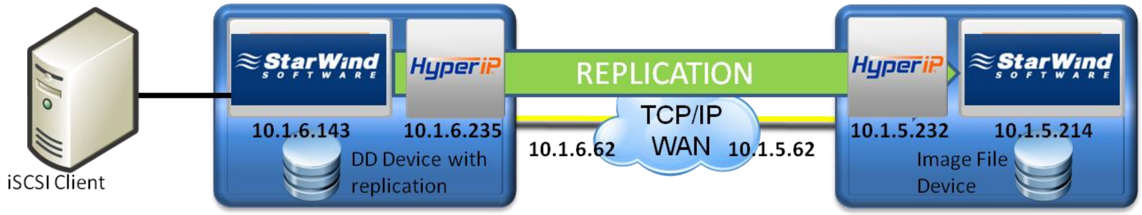
The drawing below shows a StarWind Replication configuration before installing HyperIP:



The StarWind Replication server will establish an iSCSI/TCP connection directly with the remote StarWind disk.

StarWind Replication with HyperIP (Gateway)

The drawing below shows a StarWind Replication configuration configuration after installing HyperIP:



When adding HyperIP, as in the drawing above, the StarWind Replication Server will use HyperIP as an IP gateway to the remote DR disk. HyperIP tunnels the iSCSI traffic across the WAN.

Route definitions and specific HyperIP information for the drawing is shown below:

Client site StarWind Server IP route to DR StarWind Server:
Route add 10.1.5.214/32 gateway 10.1.6.235

DR Starwind server IP route to Client site Starwind Server:
Route add 10.1.6.143/32 gateway 10.1.5.232

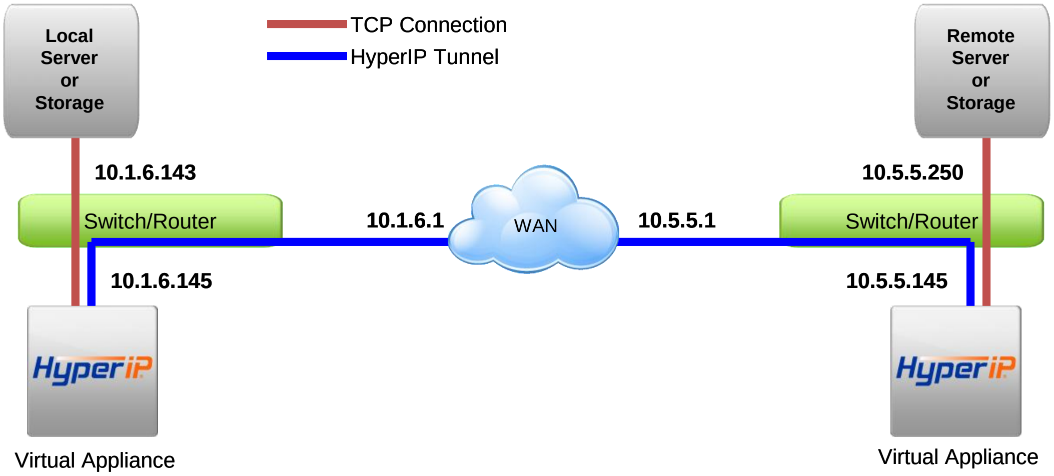
Client site HyperIP definitions:
NxN Sites:
Itself – 10.1.6.235
Peer HyperIP (DR StarWind site) – 10.1.5.232
Intercepts:
Source Address = 10.1.6.143 Destination Address = 10.1.5.214
Gateway: 10.1.6.62

Server site HyperIP site definition:
NxN Sites
Itself – 10.1.5.232
Peer HyperIP (Client site) – 10.1.6.235
Intercept:
Source Address = 10.1.5.214 Destination Address = 10.1.6.143
Gateway: 10.1.5.62

The following sections include additional information about HyperIP configurations and tuning.

Gateway Mode Configuration

The drawing below shows HyperIP added to the network. Below the drawing is information to clarify the server/storage configuration to utilize HyperIP and the pertinent HyperIP configuration for the example.



When adding HyperIP, as in the drawing above, the server or storage IP nodes will use HyperIP as an IP gateway to the remote server or storage. HyperIP tunnels the TCP traffic across the WAN. To accept TCP connections for acceleration, HyperIP uses intercepts as a filter on the traffic.

When configuring intercepts, the "source IP" is always an IP address or network on the same side of the WAN as the HyperIP being configured. The "Destination IP" then will always be an IP address or network on the other side of the WAN.

Note: Intercept networks may be defined on a byte boundary by using an asterisk wildcard.
(Using 10.5.5.* will match IP addresses 10.5.5.0-10.5.5.255)

Intercepts are defined in the "Proxies and Intercepts" section on the HyperIP config page

Route definitions and specific HyperIP information for the drawing is shown below:

Local server or Storage Storage IP route to remote server or storage:
Route add 10.5.5.250/32 gateway 10.1.6.145

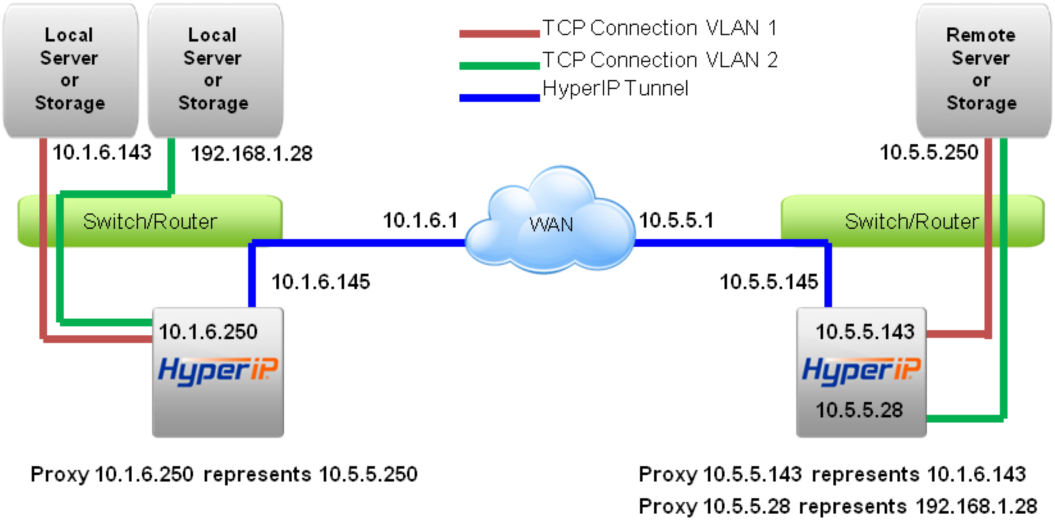
Remote server or storage IP route to local server or storage:
Route add 10.1.6.143/32 gateway 10.5.5.145

Local Site HyperIP definitions:
NxN Sites:
Itself – 10.1.6.145
Peer HyperIP (Remote site) – 10.5.5.145
Intercepts:
Source Address = 10.1.6.143 Destination Address = 10.5.5.250

Remote site HyperIP definitions:
NxN Sites
Itself – 10.5.5.145
Peer HyperIP (Client site) – 10.1.6.145
Intercept:
Source Address = 10.5.5.250 Destination Address = 10.1.6.143

HyperIP Proxy Mode Configuration

In situations where the clients and servers can not configure HyperIP as their gateway to remote networks, HyperIP proxies may be an easier implementation. Proxies are a virtual IP address on the HyperIP that will deliver data to the defined remote address. Each proxy represents a one-to-one mapping to a remote IP address (i.e 3 hosts in a site would require 3 remote site proxies). No routing changes to the clients or servers are required for this configuration. The application will use the appropriate local proxy address as its destination.
Proxy mode can typically be used with applications that work through NAT without hard addressing in the NAT device.



Local site HyperIP definitions:
NxN Sites:
    Itself – 10.1.6.145
    Peer HyperIP (Remote site) – 10.5.5.145
Proxies:
    Proxy Address = 10.1.6.250 Destination Address = 10.5.5.250

Remote site HyperIP site definition:
NxN Sites
    Itself – 10.5.5.145
    Peer HyperIP (Local site) – 10.1.6.145
Proxies:
    Proxy Address = 10.5.5.143 Destination Address = 10.1.6.143
    Proxy Address = 10.5.5.28   Destination Address = 192.168.1.28

The storage clients or servers use the proxy address where the remote client or server would be set in their configuration. For example, server 192.168.1.28, in the local site, transfers a file to the server at the remote site of using ftp with a command similar to the one below:

```
ftp 10.1.6.250
```

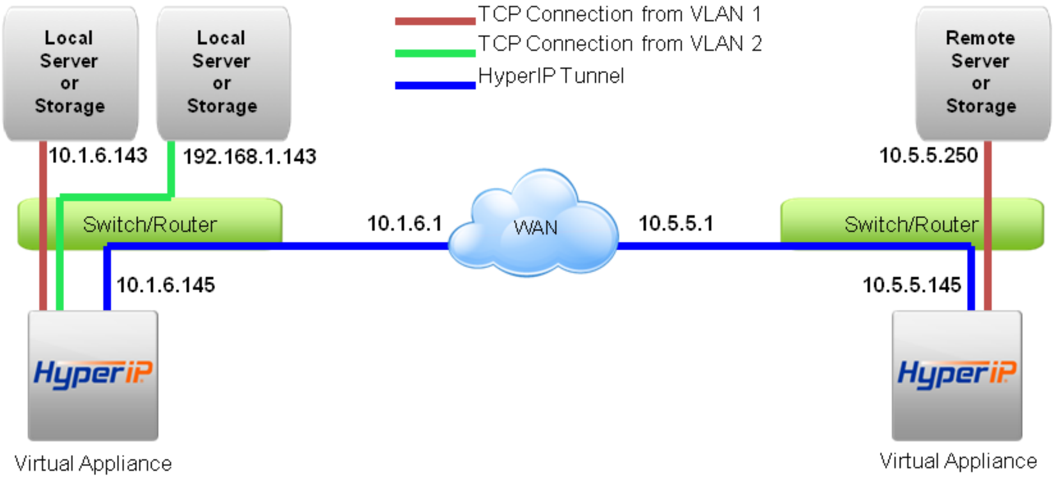
The return address to complete the TCP connection will be 10.5.5.28 which is the proxy address used to communicate with 192.168.1.28.

HyperIP Configuration with Multiple Subnets

When servers and storage located multiple networks, or the HyperIP is not installed on the same network, a router must direct traffic to the HyperIP interface. When using proxy mode, the proxy address is the destination and routing happens automatically between the IP node and the proxy IP address. See the "HyperIP Proxy Mode Configuration" section for information on this configuration. When using HyperIP gateway mode, the router configuration must be modified to send traffic to the HyperIP.

Configure static routes for the destination IP addresses
This option is acceptable when all traffic to the remote IP address should be optimized. If there is a mix of traffic (i.e. management traffic from another IP node), TCP connections for the non-optimized traffic may not complete due to the "destination only" nature of IP routing decisions.

Configure Router Filter Mechanism
Router filters (i.e. Cisco access lists) may be used to redirect traffic to HyperIP.
This option may provide control of the source and destination IP addresses and ports.



In the drawing above, servers on two networks in the left site need to communicate with the server in the right site. An example adding a static route in the router is described below:

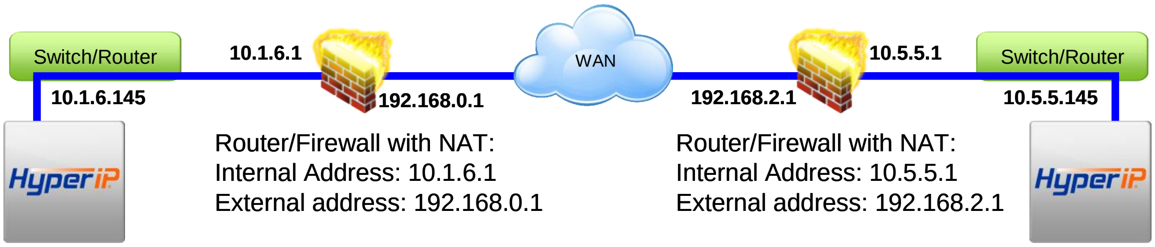
Static Routes:
Local site server 10.1.6.143 will add the route in its routing table
    Route add 10.5.5.250/32 10.1.6.145
Local site server 192.168.1.143 will use the router as its default gateway
The router will add a static route to the HyperIP
    Ip route 10.5.5.250 255.255.255.255 10.1.6.145

Remote site server 10.5.5.250 will use HyperIP as gateway to Local site servers
    Route add 10.1.6.143/32 10.5.5.145
    Route add 192.168.1.143/32 10.5.5.145

Network Address Translation (NAT) with HyperIP

Network Address Translation (NAT) is a method of connecting multiple computers to the Internet (or any other IP network) using one IP address. When internal users and servers communicate with computers on the internet, the NAT IP address is used to return traffic from the internet to the originating NAT device which maps the connection to the internal IP address. When a connection is incoming on the NAT IP address the destination port must be mapped in the NAT device for the connection to complete.

When adding HyperIP, as in the drawing below, server and storage nodes will use HyperIP gateway or proxy mode as per normal operation with HyperIP. The variation in this configuration is in the HyperIP NxN configuration to communicate through the NAT device. HyperIP uses UDP port 3919 to communicate across the WAN. When using HyperIP with NAT, the NAT device must direct incoming UDP port 3919 traffic to the HyperIP. HyperIP will tunnel the server and storage traffic across the NAT configuration. Since HyperIP tunnels across the NAT environment, the application will use the “real” address or the proxy address of the remote application node to communicate.


Switch/Router
10.1.6.145
10.1.6.1
192.168.0.1
WAN
192.168.2.1
10.5.5.1
Switch/Router
10.5.5.145
HyperIP
Router/Firewall with NAT:
Internal Address: 10.1.6.1
External address: 192.168.0.1
Router/Firewall with NAT:
Internal Address: 10.5.5.1
External address: 192.168.2.1
HyperIP

Local site HyperIP definitions:
NxN Sites:
Itself – 10.1.6.145 (data port address)
Peer HyperIP (external address mapped to HyperIP) – 192.168.2.1
Intercept: Source Address = 10.1.6.* Destination Address = 10.5.5.*

Remote site HyperIP site definition:
Itself – 10.5.5.145
Peer HyperIP (external address mapped to HyperIP) – 192.168.0.1
Intercept: Source Address = 10.5.5.* Destination Address = 10.1.6.*

Local server or Storage Storage IP route to remote server or storage:
Route add 10.5.5.0/24 gateway 10.1.6.145

Remote server or storage IP route to local server or storage:
Route add 10.1.6.0/24 gateway 10.5.5.145

Tuning HyperIP Segment Size to the Network

HyperIP uses a UDP packet, called a segment, for communication between HyperIPs. This allows the HyperIP transport to keep more data in the pipe than TCP would allow. Segment sizes up to 32768 (default) are allowed. If the segment plus overhead is larger than the media maximum transmission unit (MTU), the packet will be fragmented when leaving the HyperIP interface. If any fragment of the packet is dropped, it can't be put together at the receiving HyperIP and the entire segment will be negatively acknowledged by the HyperIP and retransmitted. Larger segments provide the best efficiency and highest transfer rates with HyperIP. Smaller segment sizes allow less traffic to be resent when there is packet loss on the network. Segment size test is available on the diagnostics page to assist in tuning the segment size to the network. This test examines a variety of segment sizes by running traffic between the HyperIPs and determining which segment size provides the highest throughput for this network.

Before running segment size test, complete the NxN site configuration using the default segment size. The HyperIP sites must be configured and communicating before the test can be run properly.

To run segment size test, go to the diagnostics commands page. In the segment size test area, select the target site in the drop-down menu. The test should be run twice; the first pass will determine whether all segment sizes will pass through the network and the second pass will determine the best performing segment size.

Pass 1:
Leave start, end and increment parameters at defaults (1300, 32000, 4000 respectively)
Change the MegaBtyes per pass to one
Click Start SegTest

Results can be viewed on the right hand page by clicking the "Retrieve Seg Results" button. At the completion of the test, the best pass segment size will be suggested. Ignore this for now, we just wanted to be sure all test segment sizes would pass through the network, and move on to the second pass of the test;
If the test appears to be stopped or taking an extremely long time to complete, clicking on 'Kill Segtest' will stop the test. If the test needed to be stopped, view the results and note the highest block size to complete. Move on to the second pass of the test.

Pass 2:
Leave start and increment parameters at defaults (1300, 4000 respectively)
Set end to the highest block size that completed in pass 1
Set Megabytes per pass so each pass runs at least one minute.

> A handy guide is to use the HyperIP key or the network rate limit, whichever is lower, and multiply it by 10. For example, on a 10Mb/s network, the number in the 'MegaBytes per pass' would be 100.

This test runs in the background and will take approximately 15 minutes to complete.
When the test completes the results will suggest a segment size to use. Use this value (or leave at default if the value was 32000) in the NxN site configuration for the remote site to which the segment size test was run.

Additional notes on segment size:
To change the segment size on a currently configured site, change the value in the blue box at the bottom of the NxN config page and click the <new config> button. Confirm the change.
The site will need to be started, and the restart from "show pending restarts" executed for the change to be implemented.

On networks below 100 Mb/s or utilizing a VPN between HyperIPs, segment size test can be skipped and the segment size set to 1300 bytes in the NxN configuration. (If using 1300 byte segments modify the “bufolimit” parameter in the site tuning area under the advanced configuration page.

To calculate a full IP packet without fragmentation, use your interface MTU size minus 116 bytes for your segsize. For example with a 1500 byte MTU set the segsize to 1384 bytes.

Advanced Site Tuning

Individual site tuning is accessed on the advanced configuration page by choosing the remote site from Show/Set Tuning Parameters and click SiteTune. We have found these parameters to be beneficial in particular situations.
The parameters, how they work together and why they may be set are described below:

The first five parameters; maxmtowait, minbtosend, compalg, compadapt and compapercnt are used to tune compression situations.

Maxmtowait defines milliseconds to hold data before being sent
Minbtosend defines how much data to hold before being sent

Whichever of these two parameters hits first triggers the data to be sent to the remote HyperIP either through the compression engine or straight to the HyperIP transport. By holding the data for a short period of time, throughput may be improved in situations where there are large swings in data received or compression ratio. If setting these parameters the suggested settings are:

Maxmtowait – 2
Minbtosend - 65400

Compalg defines the compression algorithm to use. Setting this parameter to 0 disables compression; a setting of 1 uses LZO compression.

Compadapt defines whether adaptive compression will be used. 1=yes 0=no
Compapercnt defines the percent smaller the data must be to continue compression.

When using adaptive compression, HyperIP compares the block size sent to compression with the block size exiting compression to verify the compression achieved is equal to or greater than the compapercent value. (i.e. the default value is 80 so the exiting block can not be more than 80% of the original block size) If the comparison determines the configured compression value is not being achieved HyperIP will not perform compression for a number of data blocks. A future data block will be compressed and compared and if it meets the compression percent, we return to compression, if not we will skip more data.
Adaptive compression may be turned off or the compapercent set to 99 on links where compression is always desired.

userexmitq defines whether HyperIP will hold data a period of time before it is retransmitted
rexmwblks defines the number of segments to wait when using rexmitq

In situations where retransmits are due to out of order packets seen by the receiver (Receiving HyperIP “Display HyperIP State” output shows “dupes”) holding packets temporarily before retransmitting may reduce the number of packets retransmitted. Time on queue is determined by the formula (CurrKBS / (segment size * rexmwblks)) or in english the amount of time it would take to send <rexmwblks> of <segmentsize> at the rate we are currently trying to send data.

rcvdataqhb defines the number of bytes on dataQ over which received data is discarded
rcvdataqlb defines the number of bytes on dataQ where data is again accepted after discarding

There are a few reasons the receiving HyperIP dataQ may be over-run:
The HyperIP is having difficulty sending to the target server or storage

Lost packets require the HyperIP to hold data before it can be sent to the server

When the receiving HyperIP's dataQ is full it will tell the sender to stop sending and begin to drop packets. To identify this situation, the transport log will contain messages like this:

```
Jan 20 23:03:06 hphv1b hyperip: (29) setting proceed to 25414170: tubilrn=25413549
Jan 20 23:03:06 hphv1b hyperip: (29) clearing proceed: tubilrn=25413863
```

Contact NetEx support to assist in why HyperIP is holding data and in setting these parameters.

bufolim defines maximum number of segments allowed to be in the air between HyperIPs

Once bufolim is reached NetEx stops sending segments and the 'Display HyperIP State" counter "OLimCnt" will increment.
When using small segments, this limit may need to be increased to achieve desired performance.

Usercvgapq holds data received out of order and should not be used at this time.

Global Tuning Parameters

The advanced configuration page contains additional global tuning parameters described here:

OKtoDec defines whether the HyperIP transport is allowed to reduce its sending speed.
OKtoDec maybe set to 0 (off) in situations where the TCP connections are established and not sending traffic for a long period of time or where HyperIP traffic is not in contention with other traffic on the WAN.

Deadto defines the length of time HyperIP will wait before declaring the remote HyperIP unreachable and drop any TCP connections. The default is 60 seconds.
Deadto may be increased on links where extended drop-outs occur.

Diagnostic Dump Files

Support may request dumps of the HyperIPs to assist in troubleshooting or reviewing a configuration. Here are the instructions for creating and sending a dump file:

1. From the left frame drop down menu, select **Diagnostic Commands**.
2. In the "Diagnostic Information" section, enter reason for this dump.
3. Push **CreateDump** button. (It may take several minutes to complete.)
4. Select **File Downloads/Uploads** from left frame drop down menu.
5. Under the **Diagnostic Dump Files** you should see the dump name.
6. Right click on the dump file and save the file to your workstation. Do not change the filename.
7. Upload the file from your workstation here: https://ftp.netex.com

Additional Information

For further explanation on the features/functionality of HyperIP see the HyperIP User guide at: http://www.netex.com/support/products/hyperip-docs

Video Tutorials, FAQs, and Updates

are available at: http://www.netex.com/support/hyperip-support/hyperip