Using NetApp® SnapMirror® Async with Cisco Wide Area Application Services for Data Center Replication

Many enterprises use NetApp SnapMirror Async for data protection, replication, and business continuance over a wide range of distances and transports. By copying only changed blocks, SnapMirror optimizes the use of network resources. Data ONTAP® data deduplication can further reduce traffic by transmitting only nonredundant blocks in their entirety.

Despite these optimizations, latency, packet loss, and constrained bandwidth can still limit SnapMirror performance and thereby present a challenge for data center to data center replication, which often involves large volumes of business application data and time-sensitive service level agreements (SLAs).

Cisco Wide Area Application Services (WAAS) optimizes the performance of many different applications by transparently mitigating network constraints. With the release of software version 4.0.19, Cisco WAAS adds support for optimizing data center to data center replication using SnapMirror. With Cisco WAAS, SnapMirror can replicate data faster and more efficiently to meet SLAs without costly network upgrades.

The following sections are discussed:

Corporate Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Purpose

This document illustrates how Cisco WAAS optimizes NetApp SnapMirror Async by providing configuration examples and test results derived from experiments conducted with actual NetApp and Cisco hardware and software. Only wide area latency and packet loss rates are simulated. This paper does not cover SnapMirror Sync, which is not recommended when round trip times (RTTs) exceed 2 milliseconds, since Cisco WAAS is designed for RTTs greater than 4 milliseconds.

Product information and additional technical details outside the scope of this paper are covered in the references listed at the end.

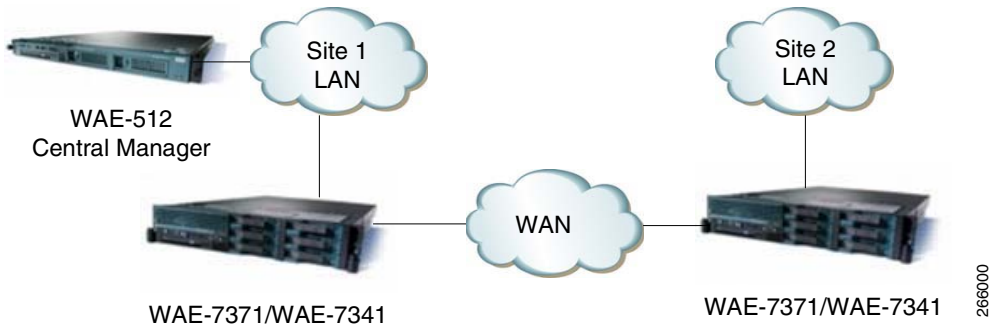
Intended Audience

This document contains technical information of interest primarily to enterprise storage and network administrators whose companies use SnapMirror Async to replicate data between geographically dispersed data centers. The paper assumes a basic familiarity with SnapMirror Async and no familiarity with Cisco WAAS on the reader's part.

What is Cisco WAAS?

The Cisco WAAS solution typically consists of at least three Wide Area Application Engines (WAEs) running Cisco WAAS software. Figure 1 depicts the solution at a high level and shows the High-Level WAAS Network Topology.

Figure 1 High-Level WAAS Network Topology



One WAE is configured in Central Manager (CM) device mode and provides a management GUI among other services. At least two additional WAEs, one in each site, examine traffic flowing through them and use built-in application policies to determine whether to optimize the traffic or allow it to pass through the network unoptimized. For high availability, a second CM device may be deployed (at Site 2 in Figure 1) as a standby.

The three basic optimizations include the following:

- Data redundancy elimination (DRE)
- Lempel-Ziv (LZ) compression
- Transport flow optimization (TFO)

DRE is an advanced form of network data redundancy elimination that reduces the amount of data to be transmitted. DRE maintains an application-independent history of previously-seen data from TCP byte streams. This enables transmission of a unique block of data only once; after that, only a reference to that block is transmitted.

LZ compression applies the well-known Lempel-Ziv lossless compression algorithm to the DRE-optimized data to shrink it even further. In particular, LZ effectively shrinks data which DRE was not able to eliminate due to seeing it for the first time.

Finally, TFO sends the twice-compressed data using a robust TCP proxy with advanced TCP window sizing and scaling, congestion management, and packet loss recovery techniques.

All of these optimizations occur at hardware speeds. The net effect of DRE, LZ, and TFO is a decrease in the amount of bandwidth used along with better utilization of that bandwidth with only a small increase in latency due to processing in the WAE.

Effect of Cisco WAAS on Network Traffic

Network traffic flow depends on whether WAEs are deployed inline or with various interception and redirection methods.

With the inline deployment, one WAE interface connects directly to a WAN-facing network device and another interface connects directly to a LAN-facing network device so that traffic can flow through the WAE. A "fail to wire" mechanism allows traffic to keep flowing natively if a WAE stops operating for some reason. The inline deployment option is not available for the WAE network module.

An alternative to inline deployment is an interception method like Web Cache Communication Protocol version 2 (WCCPv2) or policy-based routing (PBR), which can identify and redirect traffic through the WAEs. WCCPv2 uses Generic Route Encapsulation (GRE) tunnels for redirection by default in case the intercepting network device and the WAE are not in the same layer 2 domain. If they are in the same domain, WCCPv2 can redirect traffic by rewriting the frame header address fields in a process called Layer 2 redirection (L2-redirect). WCCPv2 redirection provides automatic load-balancing, fail-over, and fail-through operation when redundant WAEs are deployed.

Once the WAEs are in the data path, Cisco WAAS is transparent, meaning it does not use tunnels or require host or application configuration changes. When an end node in one site establishes a connection with an end node in the other site using the TCP three-way handshake, the WAE devices in each site automatically discover each other. The WAEs do this by marking the handshake packets with TCP option 0x21 to signal the peer WAE to establish an optimization session. Subsequent packets do not have this option added. The end nodes never see the option, since the WAEs remove it before forwarding packets to the destination node.

To handle cases where a peer WAE fails for some reason, each WAE increments the sequence number by 2,147,483,648 (0x8000000) on packets going to the peer WAE and decrements it by the same amount on packets going to the end node for each TCP flow. If a WAE fails, an end node sees a jump in the sequence number which causes it to reset the connection. In this case, the application has to reestablish a TCP connection, which then passes through the remaining WAE unoptimized. Optimization resumes for TCP connections established after the WAE is restored to service.

Table 1 shows an actual network trace between a source controller, with an IP address of 101.1.42.110, and a destination controller, with an IP address of 201.1.42.210, which illustrates the concepts above. The shaded lines are from a trace taken in the network between the two WAEs. The unshaded lines are from a trace taken using the **pktt** utility on the source controller.

Table 1 SnapMirror and Cisco WAAS Network Trace

No.	Source	Destination	Src Port	Dest Port	TCP Flags	Seq.	Ack.
1	101.1.42.110	201.1.42.210	33938	10565	SYN	0	N/A
*1	101.1.42.110	201.1.42.210	33938	10565	SYN	0	N/A
2	201.1.42.210	101.1.42.110	10565	33938	SYN, ACK	0	1
*2	201.1.42.210	101.1.42.110	10565	33938	SYN, ACK	0	1
3	101.1.42.110	201.1.42.210	33938	10565	ACK	1	1
*3	101.1.42.110	201.1.42.210	33938	10565	ACK	2147483649	2147483649
*4	101.1.42.110	201.1.42.210	33938	10565	PSH, ACK	2147483649	2147483649
4	101.1.42.110	201.1.42.210	33938	10565	PSH, ACK	1	1
5	101.1.42.110	201.1.42.210	33938	10565	PSH, ACK	2147483659	2147483649
5	201.1.42.210	101.1.42.110	10565	33938	ACK	1	57
6	201.1.42.210	101.1.42.110	10565	33938	ACK	2147483649	2147483659
6	201.1.42.210	101.1.42.110	10565	33938	PSH, ACK	1	57
7	201.1.42.210	101.1.42.110	10565	33938	PSH, ACK	2147483649	2147483659
7	101.1.42.110	201.1.42.210	33938	10565	PSH, ACK	57	57
8	101.1.42.110	201.1.42.210	33938	10565	ACK	2147483725	2147483659
8	201.1.42.210	101.1.42.110	10565	33938	ACK	57	89
9	201.1.42.210	101.1.42.110	10565	33938	PSH, ACK	2147483659	2147483725
9	201.1.42.210	101.1.42.110	10565	33938	PSH, ACK	57	89
10	101.1.42.110	201.1.42.210	33938	10565	ACK	2147483725	2147483747
10	101.1.42.110	201.1.42.210	33938	10565	PSH, ACK	89	93
11	101.1.42.110	201.1.42.210	33938	10565	PSH, ACK	2147483725	2147483747
11	201.1.42.210	101.1.42.110	10565	33938	ACK	93	129
12	201.1.42.210	101.1.42.110	10565	33938	ACK	2147483747	2147483775
12	101.1.42.110	201.1.42.210	33938	10565	ACK	129	93

Notice the sequence number jump in the third WAN packet which the controller never sees. The fourth WAN packet is not a retransmission; it contains ten bytes of control information that the peer WAE does not pass along to the destination controller.

How Does Cisco WAAS Optimize SnapMirror Async?

Before the release of Cisco WAAS software version 4.0.19, only two device modes, Central Manager (CM) and Application Accelerator (AA), were available. AA mode is primarily designed for a large number of single-user, file-mode connections over high latency and low bandwidth links such as those connecting remote branch office users to services in a central office. In contrast, replication applications like SnapMirror Async typically use a relatively small number of block-mode replication connections over the higher bandwidths and relatively lower WAN latencies usually found between data centers.

To accommodate replication applications, Cisco introduced the Replication Accelerator (RA) device mode with version 4.0.19. RA mode, which is supported on the high-end WAE-7341 and WAE-7371 models, must be configured on all WAEs handling SnapMirror Async traffic. In RA mode, the WAEs look for packets destined for TCP ports 10565 through 10569, classify it as SnapMirror traffic, and apply all possible optimizations to the traffic.

For optimization to occur, the WAEs must see the SYN packets at the start of each TCP session. This is no problem for SnapMirror Async, which by default starts a new session every minute if a previous session is not still transferring data. Each of these new sessions is a new TCP connection.

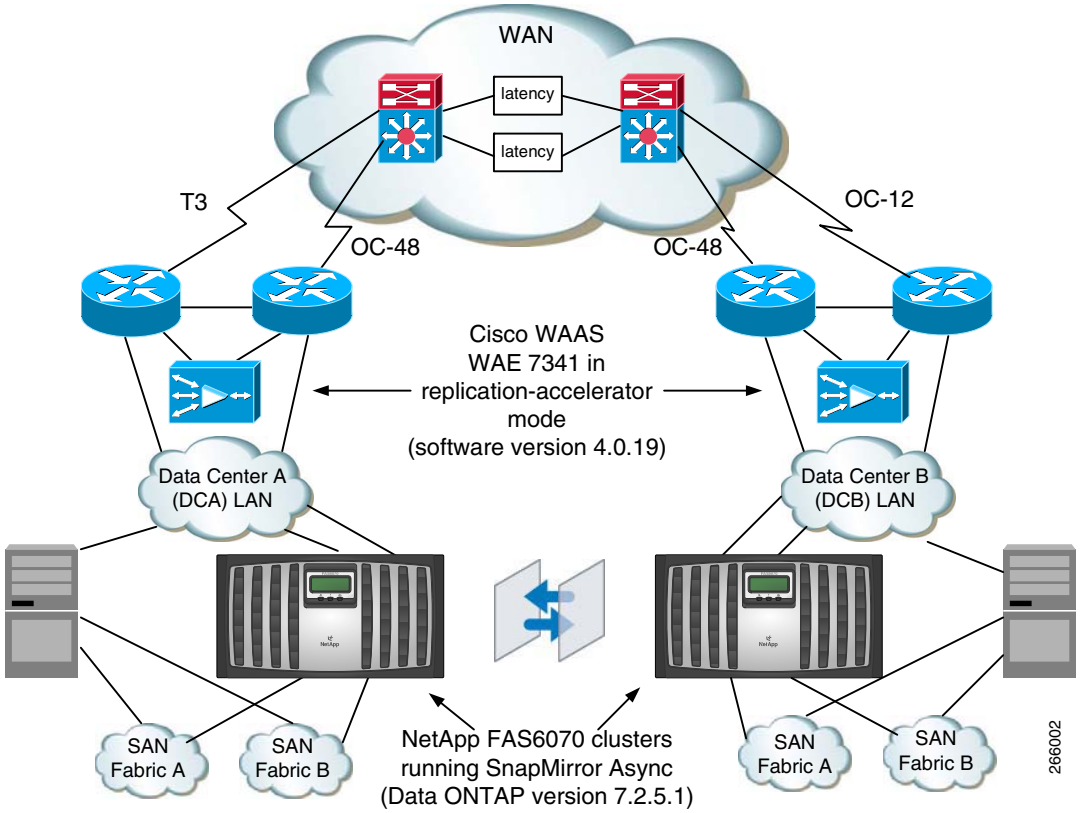
SnapMirror Async and Cisco WAAS Test Topology

To illustrate how Cisco WAAS optimizes SnapMirror Async traffic, NetApp and Cisco set up a joint test environment to emulate a production data center environment as closely as possible. The environment models two data centers connected by a wide area network (WAN). Each data center has the following equipment:

- NetApp FAS6070 cluster running Data ONTAP 7.2.5.1.
- Cisco WAE-7341 appliance running Cisco WAAS software version 4.0.19.
- Data center local area network (LAN) consisting of Cisco Catalyst 6500 switches.
- Data center storage area network (SAN) consisting of Cisco MDS 9500 switches.
- Inter-data center WAN consisting of Cisco Catalyst 6500s with T3 and OC-12 WAN interfaces as well as latency and packet loss generators using Linux Netem.
- RedHat Enterprise Linux version 4 update 4 hosts with NFS and fibre channel SAN access to the FAS6070s.

Figure 2 shows the network topology. Although all major components are pictured, the focus of this paper is on the NetApp controllers and the Cisco WAEs.

Figure 2 SnapMirror Async and Cisco WAAS Network Topology



Each controller has a SnapMirror Async configuration for replicating the following objects from the DCA controller to the DCB controller:

- FlexVolume named "async5" with no qtrees mounted via NFSv3 on the DCA Linux host and asynchronously replicated to DCB.
- FlexVolume named "qtree5" with three qtrees mounted via NFSv3 on the DCA Linux host and asynchronously replicated to DCB.

Most of the testing involves volume SnapMirror (VSM) with the async5 volume. Basic functionality tests for qtree SnapMirror (QSM) using the qtree5 volume and deduplication using the async5 volume round out the test coverage.

SnapMirror Configuration

The test configuration incorporates the following SnapMirror best practices:

- Multipathing in multiplex mode.
- Right-sized TCP window.

To achieve multipathing, each FAS6070 has a gigabit Ethernet interface connected to two different Catalyst 6500 LAN access switches (interfaces e0d and e0e). Each interface has an address on a different IP subnet.

Following is the relevant part of the host file (/etc/hosts) showing the IP addresses:

```
101.1.44.35 dca-e0d
```

```
201.1.44.35 dcb-e0d
101.1.42.110 dca-e0e
201.1.42.210 dcb-e0e
```

The relevant part of the SnapMirror configuration files (/etc/snapmirror.conf) for multiplex multipath is as follows:

```
waas = multi (dca-e0d, dcb-e0d) (dca-e0e, dcb-e0e)
waas:async5 dcb:async5 - * * * *
waas:/vol/qtree5/1 dcb:/vol/qtree5/1 - * * * *
waas:/vol/qtree5/2 dcb:/vol/qtree5/2 - * * * *
waas:/vol/qtree5/3 dcb:/vol/qtree5/3 - * * * *
```

The first asterisk (*) in the last four configuration lines means SnapMirror attempts update every minute. NetApp does not recommend this aggressive replication schedule for high RTTs, since the likelihood is high that a transfer of the changes made to a volume or qtree in one minute may not be finished before the next transfer is due (that is, one minute after the previous one began). This can result in the lag time for the volume or qtree growing until replication catches up. The tests documented in this paper use this configuration, however, to show the beneficial effects Cisco WAAS can have on replication throughput.

The optimized window size is the product of the RTT and available bandwidth. For example, in the case of a 50 ms one-way delay (100 ms RTT) over a T3 connection (45 Mbps), the window size is 562,500 bytes (45,000,000/8 * 0.1).

The following command sets the window size in the NetApp Data ONTAP command line interface (CLI):

```
options snapmirror.window_size 562500
```

Notice the window size is in bytes.

Cisco WAAS Configuration

The test topology also features WAE-7341 devices in each data center. WCCPv2 with L2-redirect and generic router encapsulation (GRE) return perform traffic interception and redirection.

To support WCCPv2 interception and redirection, the network devices to which the WAEs are connected must be set up for WCCPv2. In the test topology, each WAN edge router has a configuration similar to the following extract:

```
ip wccp 61 redirect-list waas
ip wccp 62 redirect-list waas
!
ip access-list extended waas
 permit ip 101.1.42.0 0.0.0.255 any
 permit ip any 101.1.42.0 0.0.0.255
 permit ip 101.1.44.0 0.0.0.255 any
 permit ip any 101.1.44.0 0.0.0.255
 permit ip 201.1.42.0 0.0.0.255 any
 permit ip any 201.1.42.0 0.0.0.255
 permit ip 201.1.44.0 0.0.0.255 any
 permit ip any 201.1.44.0 0.0.0.255
!
interface POS2/2/0
 description WAN INTERFACE
 ip wccp 62 redirect in
end
!
interface TenGigabitEthernet1/1
 description LAN INTERFACE
```

```
 ip wccp 61 redirect in
end
!
interface Vlan81
 ip address 10.0.81.3 255.255.255.0
 standby timers 1 3
 standby 1 ip 10.0.81.1
 standby 1 priority 170
 standby 1 preempt delay minimum 1
```

The extended access list allows diverting traffic from only selected devices. In the test topology, the only devices on the IP networks in the permit statements are the NetApp controllers. The **ip wccp 62 redirect in** statement redirects all TCP traffic coming in on the WAN interface, and the **ip wccp 61 redirect in** redirects all TCP traffic coming in on the LAN interface. Each WAN edge router has a VLAN such as VLAN 81 in the configuration extract to communicate with the WAE.

The WAEs must be in RA mode to support SnapMirror. To verify the current mode, enter the following CLI command:

```
dca-wae-7341-1#show device-mode current
Current device mode: application-accelerator
```

The above output shows the WAE is in AA mode. To put it in RA mode, enter the following CLI commands:

```
dca-wae-7341-1#conf t
dca-wae-7341-1(config)#device mode replication-accelerator
The new configuration will take effect after a reload
dca-wae-7341-1(config)#end
dca-wae-7341-1#reload
```

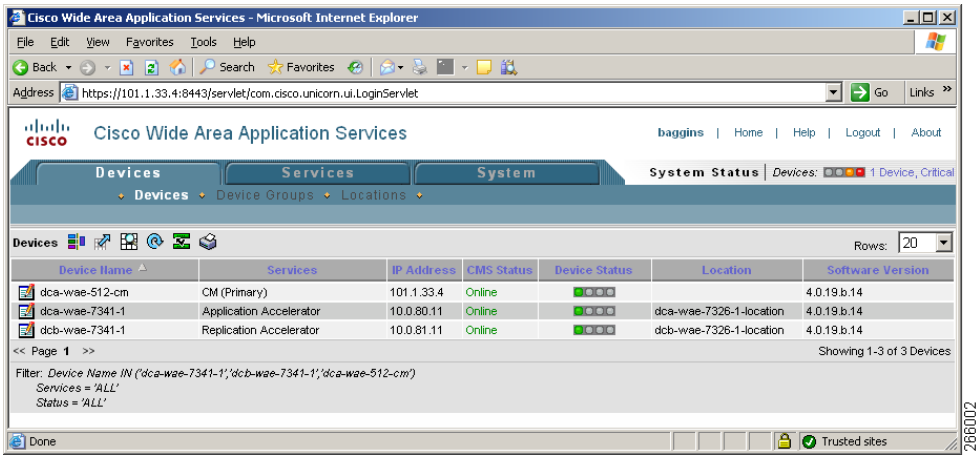
Note that a reload is required for the WAAS software to provision memory and disk resources appropriately for RA mode. Once the WAE is up in RA mode, configuration for WCCPv2 is next. Following are the relevant portions of the WAE configuration once everything is set up:

```
device mode replication-accelerator
!
ip default-gateway 10.0.80.1
!
wccp router-list 1 10.0.81.2 10.0.81.3
!
wccp tcp-promiscuous router-list-num 1 l2-redirect mask-assign assign-method-strict
!
wccp version 2
!
policy-engine application
 name Replication
 classifier NetApp-SnapMirror
 match dst port range 10565 10569
 exit
 map basic
 name Replication classifier NetApp-SnapMirror action optimize full
 exit
exit
```

The **ip default-gateway** statement references the IP address of the hot standby routing protocol (HSRP) address shared by the WAN edge routers. The **wccp router-list** statement references the IP addresses of both local WAN edge routers. Together, the WCCPv2 configurations on the routers and WAE allow the devices to negotiate the appropriate interception and redirection parameters.

At this point, only SnapMirror traffic is flowing through the WAEs at each site. Figure 3 shows the Cisco WAAS Central Manager GUI and the three WAE devices in the test topology once they're up and running. The WAE-7341 devices are called "dca-wae-7341-1" and "dcb-wae-7341-1." The Central Manager is called "dca-wae-512-cm."

Figure 3 Cisco WAAS Central Manager GUI



As soon as a SnapMirror Async replication session starts up, run the following CLI command on one of the WAEs to verify it's optimizing the traffic:

```
dca-wae-7341-1#show tfo conn summ

Optimized Connection List
Policy summary order: Our's, Peer's, Negotiated, Applied
F: Full optimization, D: DRE only, L: LZ Compression, T: TCP Optimization
Local-IP:Port Remote-IP:Port ConId PeerId Policy
201.1.44.35:40857 101.1.44.35:10566 31 00:1a:64:26:e4:a0 F,F,F,F
101.1.44.35:48981 201.1.44.35:10565 32 00:1a:64:26:e4:a0 F,F,F,F
101.1.42.110:1154 201.1.42.210:10565 33 00:1a:64:26:e4:a0 F,F,F,F
```

The "F,F,F,F" means the WAE is fully optimizing SnapMirror traffic at ports 10565 and 10566 for both paths, from IP addresses 101.1.44.35 and 101.1.42.110 on the DCA controller to 201.1.44.35 and 201.1.42.210 on the DCB controller.

Test Methodology

Each test consists of replicating application data using SnapMirror Async with and without Cisco WAAS, and then comparing the replication throughput, which is the key metric.

Following is an example of the statements gathered the SnapMirror log file (/etc/log/snapmirror) on the destination controller to determine throughput:

```
dst Wed Sep 3 13:31:03 EDT waas:async5 dcb-controller:async5 Start
dst Wed Sep 3 13:32:33 EDT waas:async5 dcb-controller:async5 End (127164 KB)
```

In the above example, the amount of data is 127,164 KB and the time is 90 seconds, for a rate of 1413 KB/sec. Network utilization data (primarily the number of packets and bytes transmitted between the WAEs) are additional metrics collected.

For each test, a simulated RTT of 50 ms and packet loss rate of 0.1% emulates a distance of about 3000 km between data centers (based on a one-way propagation delay of light in optical fiber of 0.5 ms/100 km and about 10 ms for one-way device processing and queuing delays).

The application data consisted of the following types:

- Microsoft Exchange 2003 database files.
- Oracle 10gR2 database files.
- Oracle 10gR2 archive log files.

A given test may copy one or more data types from a local directory on the Linux host to a directory using storage provided by the source NetApp controller. The types of storage tested include the following:

- NAS volume: a mount point called /NAS on the server corresponds to a 30 GB FlexVolume called "async5" which is replicated by SnapMirror Async.
- Qtree volume: a mount points called "/Q1" corresponds to a qtree called "/vol/qtree/1" on a 75 GB FlexVolume called "qtree5" which is replicated by SnapMirror Async. Two other qtrees, "/vol/qtree/2" and "/vol/qtree/3" are also on the qtree5 volume but are not directly used in testing.

The host accesses all these devices on networks separate from the SnapMirror networks. Following are the mount options for each mount point as displayed by the Linux **mount** command:

```
# mount
10.0.5.160:/vol/async5 on /NAS type nfs
(rw,nfsvers=3,tcp,hard,intr,rsize=32768,wsize=32768,addr=10.0.5.160)
10.0.5.160:/vol/qtree5/1 on /Q1 type nfs
(rw,nfsvers=3,tcp,hard,intr,rsize=32768,wsize=32768,addr=10.0.5.160)
```

The copy for the tests was done by a shell script similar to the following:

```
# date; time { cd /var/ftp/pub/snapmirror/data; tar cf - exchange oracle | ( cd /NAS/data;
tar xpf - ) }; date
Wed Sep 3 13:30:18 EDT 2008
real 0m12.328s
user 0m0.017s
sys 0m0.492s
Wed Sep 3 13:30:30 EDT 2008
```

In the above example, the script copies Microsoft Exchange and Oracle database files to the /NAS mount point. The time taken for this is not relevant, since it has nothing to do with the replication time, but it helps to ensure uniformity among different tests.

Before any copy starts, the volume on the controller is in "snapmirrored" state and no data from previous copies is still being replication.

For the deduplication tests, data deduplication is applied by applying the following commands in the order shown:

Step 1 Stop data deduplication:

```
dca-controller> sis stop /vol/async5
Operation is currently idle: /vol/async5
dca-controller> sis off /vol/async5
SIS for "/vol/async5" is disabled.
dca-controller> sis undo /vol/async5
```

Step 2 Wait for a minute, then verify deduplication is complete.

```
dca-controller> df -s async5
Filesystem used saved %saved
```

```
/vol/async5/ 592804 0 0%
```

Step 3 Wait until snapmirror data transmission is back to the baseline amount, which is the number of kilobytes transferred when no user data changes have occurred:

```
dst Fri Aug 8 15:28:01 EDT waas:async5 dcb-controller:async5 Start
dst Fri Aug 8 15:28:03 EDT waas:async5 dcb-controller:async5 End (120 KB)
```

Step 4 Start deduplication:

```
dca-controller> sis start -s /vol/async5
```

Step 5 Wait until status shows idle, and then verify status:

```
dca-controller> sis status -l /vol/async5
Path: /vol/async5
State: Enabled
Status: Idle
Progress: Idle for 00:00:10
Type: Regular
Schedule: sun-sat@0
Last Operation Begin: Fri Aug 8 15:37:31 EDT 2008
Last Operation End: Fri Aug 8 15:38:59 EDT 2008
Last Operation Size: 457 MB
Last Operation Error: -
```

Step 6 Check savings:

```
dca-controller> df -s async5
Filesystem used saved %saved
/vol/async5/ 387064 155440 29%
```

Step 7 Monitor snapmirror throughput until it returns to baseline:

```
dst Fri Aug 8 15:38:01 EDT waas:async5 dcb-controller:async5 Start
dst Fri Aug 8 15:38:04 EDT waas:async5 dcb-controller:async5 End (1756 KB)
dst Fri Aug 8 15:39:01 EDT waas:async5 dcb-controller:async5 Start
dst Fri Aug 8 15:39:04 EDT waas:async5 dcb-controller:async5 End (6088 KB)
dst Fri Aug 8 15:40:01 EDT waas:async5 dcb-controller:async5 Start
dst Fri Aug 8 15:40:03 EDT waas:async5 dcb-controller:async5 End (200 KB)
```

Step 8 Collect metrics. In the above example, data deduplication resulted in 7844 KB of data (1756 + 6088) transmitted in 6 seconds (3 + 3) for a throughput of 1307 KB/sec. Note the baseline amounts are excluded.

For the WAAS tests, the copy is done first with a cold cache and then with a warm cache The following is the command sequence to clear the cache (commands entered are boldfaced):

```
dca-wae-7341-1#clear cache dre
TFO application needs to be restarted (all existing
connections will be reset, alarms may be raised and system may reboot if required).
Do you want to Continue? [yes/no]yes
Restarting processes
Clearing DRE cache
Done. No reboot was required.
```

Test Results

In all tests, with an RTT of 50 ms and packet loss rate of 0.1%, WAAS dramatically increased throughput and reduced the number of packets and amount of data sent over the network. This included both when the WAAS cache was cold (that is, the cache was just cleared) and when it was warm (that is, when the same data was copied without clearing the cache). Table 2 summarizes the results of all the tests.

Table 2 Test Result Summary

	Throughput Improvement		Packet Savings	
Test	**cold**	**warm**	**cold**	**warm**
VSM Async (NAS) - small data set	600%	748%	83%	89%
VSM Async (NAS) - large data set	941%	-	-	-
QSM Async (NAS)	520%	557%	87%	89%
VSM Async (NAS deduplication)	-	366%	-	-

The following charts illustrate the findings graphically. Figure 4 and Figure 5 show throughput improvements and packet reduction for volume SnapMirror Async with a small data set.

Figure 4 VSM Async Throughput for Small Data Set

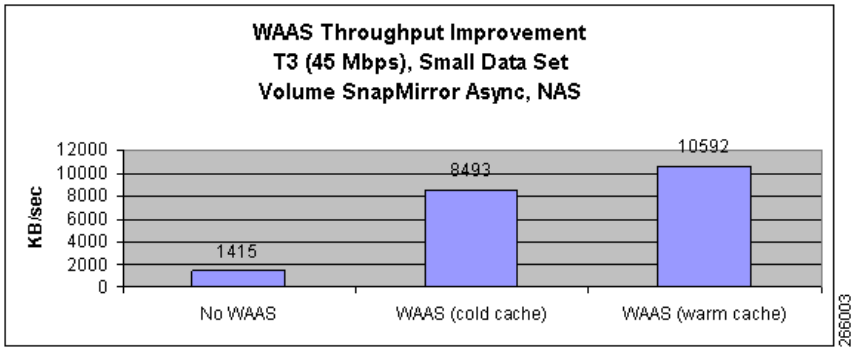


Figure 5 VSM Async Packets

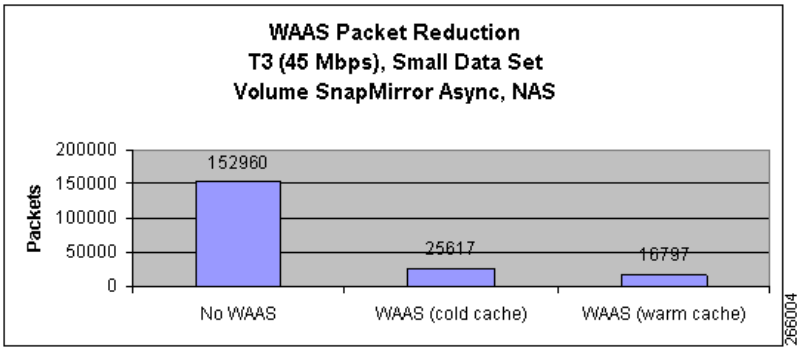


Figure 6 shows throughput improvement for volume SnapMirror Async with a data set much larger than the cache size in the WAE-7341.

Figure 6 VSM Async Throughput for Large Data Set

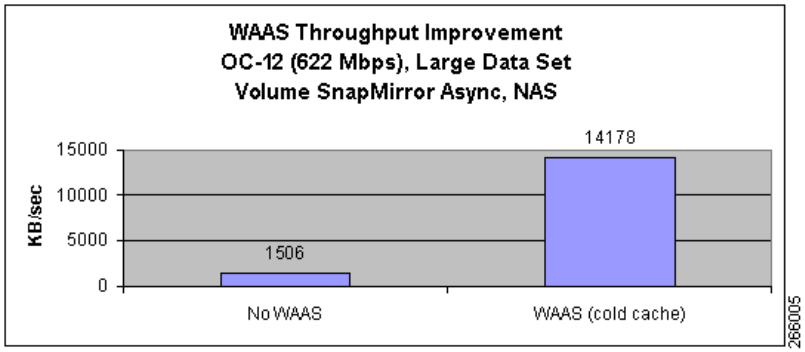


Figure 6 doesn't tell the whole story, though. Not only is overall throughput much higher with WAAS, but also the lag between the source and destination volumes is much smaller with WAAS. Following is the section of the SnapMirror log file showing the test without WAAS:

```
dst Wed Sep 3 13:39:10 EDT waas:async5 dcb-controller:async5 Start
dst Wed Sep 3 13:43:58 EDT waas:async5 dcb-controller:async5 End (435964 KB)
dst Wed Sep 3 13:44:00 EDT waas:async5 dcb-controller:async5 Request (Scheduled)
dst Wed Sep 3 13:44:01 EDT waas:async5 dcb-controller:async5 Abort (update from source not
possible; snapmirror may be misconfigured, the source volume may be busy or unavailable)
dst Wed Sep 3 13:45:03 EDT waas:async5 dcb-controller:async5 Start
dst Wed Sep 3 14:20:05 EDT waas:async5 dcb-controller:async5 End (3111636 KB)
dst Wed Sep 3 14:21:09 EDT waas:async5 dcb-controller:async5 Start
dst Wed Sep 3 17:51:53 EDT waas:async5 dcb-controller:async5 End (19144652 KB)
dst Wed Sep 3 17:52:03 EDT waas:async5 dcb-controller:async5 Start
dst Wed Sep 3 19:23:07 EDT waas:async5 dcb-controller:async5 End (8185004 KB)
```

Notice at one point the lag is more than three and a half hours (between 14:21:09 and 17:51:53). In contrast, with WAAS, as the following extract from the SnapMirror log file shows, the flow of data is much more even:

```
dst Thu Sep 4 10:12:02 EDT waas:async5 dcb-controller:async5 Start
dst Thu Sep 4 10:12:34 EDT waas:async5 dcb-controller:async5 End (506516 KB)
dst Thu Sep 4 10:13:02 EDT waas:async5 dcb-controller:async5 Start
dst Thu Sep 4 10:13:39 EDT waas:async5 dcb-controller:async5 End (506492 KB)
```

<intermediate entries deleted>

```
dst Thu Sep 4 11:08:06 EDT waas:async5 dcb-controller:async5 Start
dst Thu Sep 4 11:08:47 EDT waas:async5 dcb-controller:async5 End (579452 KB)
dst Thu Sep 4 11:09:02 EDT waas:async5 dcb-controller:async5 Start
dst Thu Sep 4 11:09:18 EDT waas:async5 dcb-controller:async5 End (184604 KB)
```

SnapMirror starts up a new replication session every minute and is able to transmit the amount of data collected since the last session in 35 to 45 seconds the entire time the copy is running.

Figure 7 and Figure 8 show similar throughput improvements and packet reduction for qtree SnapMirror Async with a small data set.

Figure 7 QSM Async Throughput for Small Data Set

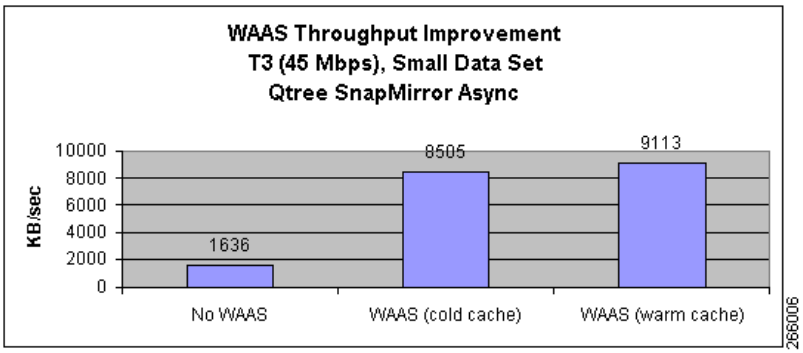
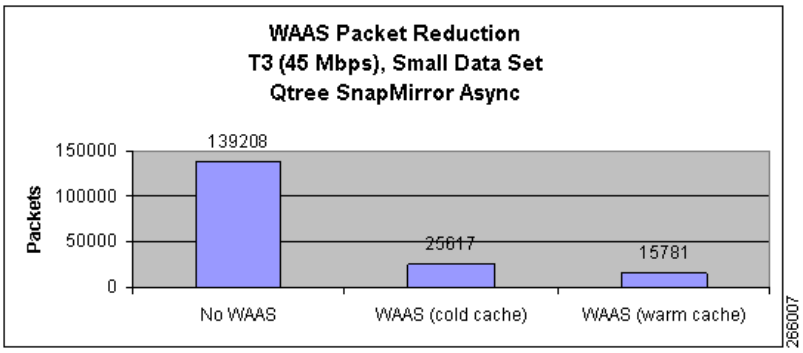
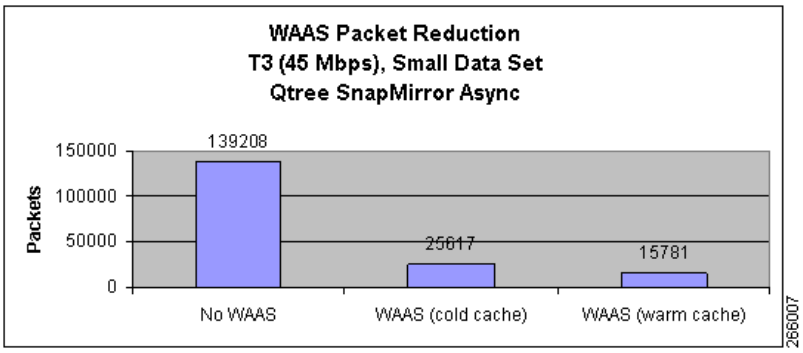


Figure 8 QSM Async Packets



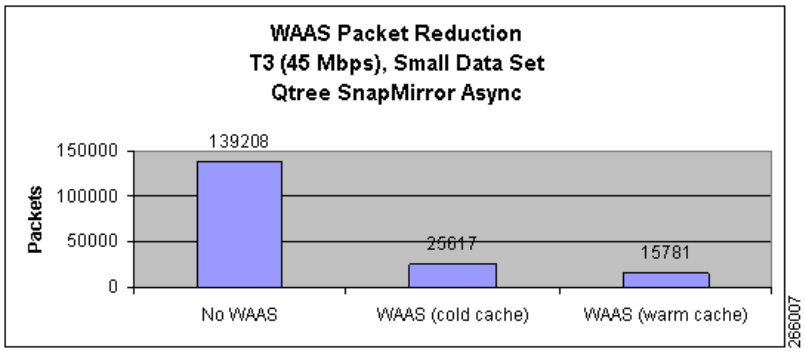
Finally, Figure 9 shows a somewhat smaller but still substantial throughput improvement for volume SnapMirror Async for data sent after a volume is deduplicated. This shows the NetApp and Cisco WAAS data redundancy algorithms are compatible.

Figure 9 VSM Async Throughput for Small Data Set with NetApp Deduplication



From the WAAS point of view, many statistics are available for the network administrator to track the above statistics. For example, Figure 10 shows a portion of the CM GUI showing Optimization Statistics screen.

Figure 10 Cisco WAAS CM Optimization Statistics



Extensive commands to show how Cisco WAAS is operating are also available from the command line interface.

Summary

The test results reported in this white paper demonstrate how well NetApp SnapMirror Async and Cisco WAAS in Replication Accelerator mode can work together to improve replication performance and reduce network resource requirements. Once configured, Cisco WAAS is completely transparent to SnapMirror. Together, SnapMirror Async and WAAS ensure customers get the best replication experience possible out of their network.

References

The following NetApp and Cisco reference material is available online.

NetApp Reference Documents

1. SnapMirror Async Overview and Best Practices Guide (NetApp Technical Report TR-3446) - requires NOW™ (NetApp on the Web, now.netapp.com) account

   http://www.netapp.com/us/library/technical-reports/tr-3446.html

2. NetApp Deduplication for FAS Deployment and Implementation Guide (NetApp Technical Report TR-3505)

   http://media.netapp.com/documents/tr-3505.pdf

3. SvnapMirror Sync and SnapMirror Semi-SyncOverview and Design Considerations (NetApp Technical Report TR-3326) - requires NOW™ (NetApp on the Web, now.netapp.com) account

   http://www.netapp.com/us/library/technical-reports/tr-3326.html

Cisco Reference Documents

1. Enterprise Data Center Wide Area Application Services (WAAS) Design Guide

   http://www.cisco.com/application/pdf/en/us/guest/netsol/ns377/c649/ccmigration_09186a008081c7da.pdf

2. Cisco Wide Area Application Services Configuration Guide (Software Version 4.0.19)

http://www.cisco.com/en/US/docs/app_ntwk_services/waas/waas/v4019/configuration/guide/waas4cfg.html

3. Release Note for Cisco Wide Area Application Services (Software Version 4.0.19)

   http://www.cisco.com/en/US/docs/app_ntwk_services/waas/waas/v4019/release/notes/ws4019rn.html

4. Cisco WAAS Optimizations for Data Protection Applications

   http://www.cisco.com/en/US/prod/collateral/contnetw/ps5680/ps6870/prod_white_paper0900aecd8051c0a6.pdf

5. Cisco Wide Area Application Services (WAAS) V4.0 Technical Overview

   http://www.cisco.com/en/US/prod/collateral/contnetw/ps5680/ps6870/prod_white_paper0900aecd8051d5b2.html