AROW Data Diode

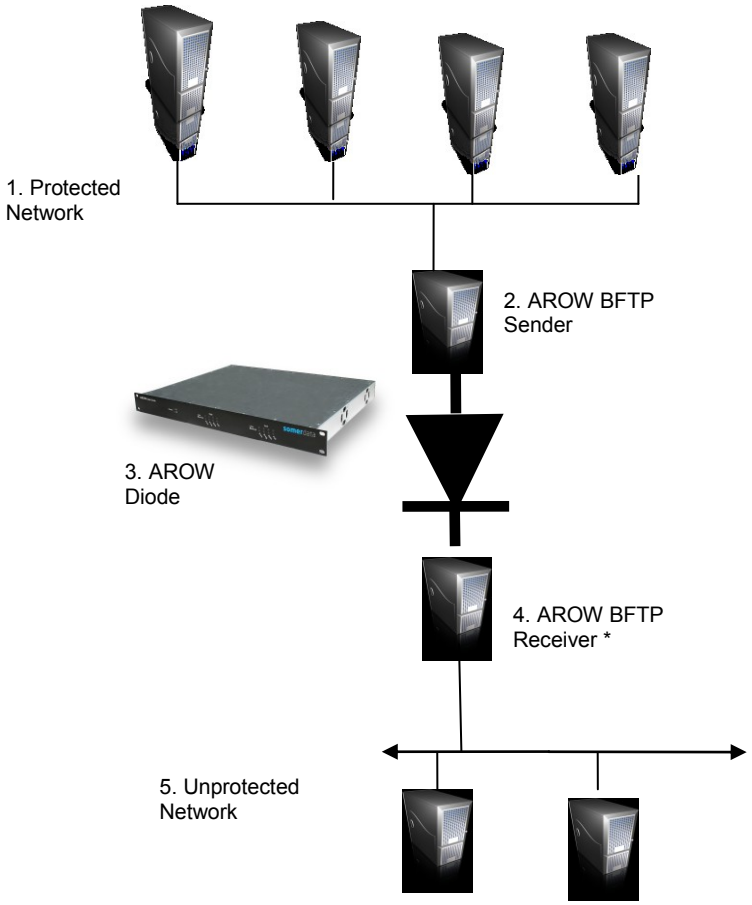
File Transfer Performance with AROWBftp

File and data transfer performance across Data Diodes is a complex process involving many factors and optimisations.

AROW Data Diode simplifies this process by operating as a TCP device, but there are still factors to consider.

Transfer speed and reliability depend on two factors – the speed and protocol support of the hardware device (Data Diode) and the speed and reliability of the supporting software and hardware.

Consider this common setup.



The requirement is to pass file data from the protected network to be made available to the unprotected network.

AROW File Transfer performance

There is no particular limit to the type, size or quantity of files, nether is there an assumption about the network types (Linux/Unix/Windows/Mac etc) involved. But file integrity is of the utmost importance followed swiftly by availability time. What do we mean by these? File integrity is simply ensuring that file data is not damaged, corrupted or lost during transfer, availability time means the elapsed time from a file appearing on the protected network to it's being available on the unprotected network.

The first of these is straightforward – corrupted data is useless, wastes bandwidth and leads to user dissatisfaction. Users expect data to be correct and when copied to appear as if transferred across a 'normal' network connection. Availability time is more complex and is composed of latency (the cumulative delay incurred at various points in the transfer chain) , and transmission time which is dependent on network equipment transfer speeds.

Each of these elements is to some extent influenced by the quantity and size of the files to be transferred – larger files take longer to transfer than smaller files, but large numbers of small files can also take a long time to transfer since they incur more overhead for the operating system, file transfer protocol and network protocol. Most networks therefore have a 'sweet spot' of file transfer size and quantity that optimises the file availability time and it is the job of the system administrator to optimise these parameters for their particular users.

When testing file transfer systems, including those using data diodes, and especially when comparing systems, it is a good idea to create a suite of files of various sizes, types and mixes and use these to stress-test the network. This is the method we are describing here and its use in evaluating the AROWBftp system.

Referring to our network diagram above, the points of performance we need to consider are
1. the source data – network speed, storage and application speed, request and action latency
2. the serialisation node performance. The Data Diode is a single node that is required to gather all input data, select and serialise for transfer. This requires a dedicated server network node for the diode, encapsulation software for identifying the various sources so they can be de-serialised and re-constructed, error prevention, timestamping etc.
3. the physical speed of the diode device and its latency due to internal buffering
4. the de-serialisation node performance, including reconstruction and error checking
5. the receiving network performance

It turns out that 3, the physical speed of the AROW diode, is the least bottleneck. The all-hardware design of the diode means that it performs at full GBE network speed with low latency, and due to its inherent TCP capability, introduces no data errors. We have shown elsewhere the intrinsic high-speed and durable performance of the AROW diode, so it can be ignored for this discussion.

The discussion thus turns to the performance of the supporting software and hardware – the serialisation and de-serialisation components and the attached networks.

The serialisation process incurs a data overhead – data is framed in a manner to make de-serialisation possible and this adds extra data to the user data while it is being transferred. De-serialisation removes this extra data, rendering the final file data transparent to the process.

Somerdata Ltd
1 Riverside Business Park
St Annes Road
Bristol
BS4 4ED
UK

Phone: +44 (0)1179 634050
E-mail: sales@somerdata.com
Website: www.somerdata.com

AROW File Transfer performance

But this overhead data does incur a processing penalty. In particular the de-serialisation process requires good processing capability – a fast processor some dedicated memory and most importantly high speed write capability on the receiving storage.

Disk storage systems should be capable of sustained write transfer speeds. For example, a full rate GBE payload operates at about 105MBps, so sending and receiving disk systems must be able to support this, and some. Our experience shows that most disk systems (including flash) do not reach their specified transfer rates when faced with sustained writing of data. Consequently we recommend specifying at least twice the rated speed (in this case say 200+ Mbytes/S) on any receiving storage.

We also need to recognise the different mixes of file sizes and structures – is transfer performance the same for tens of millions of small files and a few gigantic files? What about the mix of folders and files?

Our tests involve the creation and measurement of different mixes of files according to this table

Profile of Files Transferred Across the Data Diode
100 % very small files (10KB)
100% small files (60KB),
100% medium files (100MB)
100% big files (500MB)
70% very small files, 20% medium files, 10% big files

The files profile is arranged as folders containing files and sub-folders containing files and sub-subfolders also containing files.
Very small files – 124 subfolders each containing 64000 files, plus one sub-subfolder containing 41 subfolders of 64000 files each.
Small files – 28 subfolders each containing 64000 files
Medium files – one folder containing 1024 files
Big files – one folder containing 204 files
Mixed files – 26000 files,43 subfolders each containing 37000 files, one subfolder containing 2600 files plus74 sub-subfolders containing 37000 files

And in order to be meaningful we choose a source file capacity of 100GB .

We create files so that we can make consistent and repeatable measurements across different diode systems. We also can stress-test the system with extreme edge cases. Our aim is to replicate the exact file structure of the source data to the receive data, to measure data integrity of the transfer and to measure the availability time of any specified file.

AROW File Transfer performance

We also need to specify the type and performance of the test hardware and software – for these tests we use the Linux operating system (Debian) and platforms capable of sustaining high-speed data transfers – a disk system with sustained throughput of 140MB/s and sufficient RAM to accommodate large in-memory storage (a minimum of 8GB free for these tests.) together with a 4-core processor. The system is assumed to be 'settled' after switch on with its internal housekeeping up to date.

With all this in place we can make some measurements. AROWBftp includes log files with event timestamps, we can use these to time transfers and file availability.
viz.
From the sender log

```
19/10/2015 09:41:50 INFO Synchronising Directory TestSend
```

From the receiver log

```
19/10/2015 09:41:50 INFO 19/10/2015  09:41:50 Sender says: testslaveiii
started 10.0.0.22
19/10/2015 10:05:52 INFO Sender says:scan complete 1024 files sent
(Redundancy factor 1)
19/10/2015 10:05:52 INFO New or changed files sent 1024 Lost 0
```

So examining this log, we can see that 1024 files were transmitted, all were received satisfactorily and the total time taken was 1442 seconds. We know that these were 100MB files, so a total of approximately 100GB transferred, at a rate of just under 600Mbps.

The first file sent, file number 1, was available 1.4 seconds after start of transmission, so here we can see ordering of files is important – if you need the last file you have to wait some 24 minutes, if you need the first file just a couple of seconds! We can repeat these measurements for different file sizes and fill in our table.

Files Profile	Total transfer time (Seconds)	Setup time	First file available (Seconds	Last file available (Seconds)	Average net transfer rate(Mbps)	Memory Used	No.of Files
100 % very small files (10240B)	2361	441	441	2361	342	15GB	8396800
100% small files (61440B)	1725	98	98	1725	472	3.5GB	1811684
100% medium files (104857600B)	1442	5	1.4	1442	595	0.4GB	1024
100% big files (524288000B)	1490	1	5	1582	540	0.5GB	204
70% very small files, 20% medium files, 10% big files	1966	328	328-333	1966-1973	413	9.2GB	5154512

A notional 100GB transferred for each row. Flow rate limit -600Mbps

somerdata

Somerdata Ltd
1 Riverside Business Park
St Annes Road
Bristol
BS4 4ED
UK

Phone: +44 (0)1179 634050
E-mail: sales@somerdata.com
Website: www.somerdata.com

AROW File Transfer performance

We can see that there is a 'sweet spot' of transfer with files up to 10-100 MB being the overall fastest. Lots of small files take longer, due to the serialisation/de-serialisation overhead and also consume more RAM due to the process of file description for integrity checking. (Note that this setup overhead only occurs once per data set – additions to or deletions from the set are handled incrementally.)

Large files incur processing overhead of CRC calculation, (for very large files in excess of 1GB this can take many tens of seconds)

Of course this is very dependent on the hardware being used and the software options chosen. In fact, since the Python software is entirely open, you can readily optimise it for any particular set of circumstances, or even create profiles to suit changing circumstances.

Memory usage is dependent on the number and nature of files to be sent –scanning the directory tree in one go and then storing the results takes a long time and occupies more memory if the depth of scan is high (lots of nested folders) and the number of files per folder is high.

For this scenario, AROWBftp offers an option of changing the depth of scan, thereby creating sets of scans that individually require less resource. However this adds complication to any restoration process, since there are now multiple restoration files (in fact, one per folder for every depth level).

If we invoke this option, and set the depth level to 3 for the above table and test data, we obtain the following results:

Files Profile*	Total transfer time (Seconds)	Setup time	First file available (Seconds	Last file available (Seconds)	Average net transfer rate(Mbps)	Memory Used	No.of Files	Total Transfer (GB)
100 % very small files (10240B)	3355	120	120	3355	256	430MB	10,485,924	107
100% small files (61440B)	1946	21	21	1946	402	250MB	1,747,654	98G
100% medium files (104,857,600B)	1465	5	5	1465	550	130MB	1024	101
100% big files (524,288,000B)	1431	1	5	1436	560	240MB	204	100
70% very small files, 20% medium files, 10% big files	2230	115	115-120	2115-2120	353	256MB	5,354,089	100

Flow rate limit -600Mbps

somerdata

Somerdata Ltd
1 Riverside Business Park
St Annes Road
Bristol
BS4 4ED
UK

Phone: +44 (0)1179 634050
E-mail: sales@somerdata.com
Website: www.somerdata.com

The resulting overall transfer time is increased for nested sub-folders, the resources required are much-reduced but there are now multiple recovery files to be examined and acted upon.

Taking the mixed files profile, there is an average 15% reduction in transfer rate, but a 97% reduction in resources used.

Conclusions

The clear conclusion from this is that we have to understand the type and topography of data being transferred in order to optimise system resources, transfer rates and latency.

Most practical systems contain a mix of files, directories, sub-directories etc., which once established do not tend to change very much in structure but can change a lot in content, for example email servers, document servers and so on.

So it is incumbent on the system administrator to understand this profile, to monitor performance and to adjust the transfer system appropriately.

There may be a certain amount of trial and error involved, but the AROWBftp software is flexible enough to accommodate most scenarios, and is readily customisable for scenarios that fall outside the norm.