

AROW Data Diode

Software Emulator

Using the emulator.

We have created an emulator for the AROW Data Diode that allows the demonstration of AROW Send, Receive and Control software suites.

Written in Python, the emulator allows connections from the client Send and Receive software normally supplied with the AROW product, enabling all of the same data transfer functions to be accomplished, albeit with some speed restrictions. There is also support for the control functions, that can be used to demonstrate the WinAROW GUI control application.

The emulator supports cross-network operation but can also be used on a single platform. It is necessary to have Python2.7 installed on each target platform to be used.

Most Linux distributions already include Python, Windows and Mac platforms will need to install this separately.

The software is supplied in compressed format (Zip or Tar) and should be decompressed before use to a suitable directory. Note that in the case of a new Windows application, it will be necessary to set the Path environment variable to point to the python installation, typically c:\python27.

The decompressed folders contain all of the Python modules necessary to run the emulator, the main module being AROWEmulator.py. As with the normal AROW software, it is necessary to pass parameters to the AROWEmulator .py script in order to set it up. This is best accomplished by using a shortcut on Windows or a link on Linux

The shortcut should include the absolute path to the AROWEmulator.py script and the parameters to be passed in as described below. The important information required is the network addresses of the platforms to be used and the ports that are free to be used.

For use on a single platform (eg Windows) the shortcut to AROWEmulator should look like:

```
C:\Somerdata\AROWEmulator\Src\AROWEmulator.py --lca 192.168.2.44 --hca 192.168.2.44 --la 192.168.2.44 --lp 9876 --ha 192.168.2.44 --hp 9875
```

where C:\Somerdata is the absolute path to the AROWEmulator folder and 192.168.2.44 is your local ip address.

Note that if your platform is not connected to a network and is set to automatic (DHCP) mode, you will have to assign it a static IP address temporarily. Choose something like 192.168.2.2 or 10.0.0.3

Setup Parameters

The parameters are :

--lca control address, the address of the control port associated with the low (sending) side of the diode

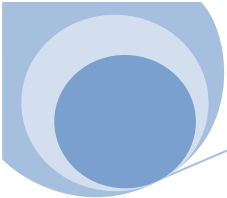
--hca control address, the address of the control port associated with the high(receiving) side of the diode

--ha high address , the address of the high (receive) software location

--hp high port, the port number of the high (receive) software

--la low address, the address of the low (send) software location

--lp low port, the port number of the low(send) software



AROW Data Diode

Software Emulator

Note that the emulator requires a different port number for send and receive, unlike AROW that will normally use two distinct IP addresses (probably different networks) but may use the same port numbers.

Clicking on the link should start the emulator . This needs to be done before starting the Send or Receive software, but once running, can be connected to at will. (You will need to allow communication through any firewall that may be running).

The emulator runs in a terminal window and has some status information, indicating when clients connect and disconnect. If it doesn't start, or starts and stops, then the network connection is most likely unavailable. Try another port setting in the shortcut.

The AROWSend and AROWReceive scripts can now be used as if connected to AROW.

Example connections and things to try.

For the simple case of a single platform running all 3 scripts (AROWEmulator, AROWSend and AROWReceive) on Windows.

Use these settings

AROWEmulator as above

AROWReceive

```
<path to AROWReceive folder>\AROWReceive\src\AROWReceive.py -r -f reception -a 192.168.2.44 -p 9875 -w 8080
```

This assumes a receiving folder (in the same location as the script) named reception and webserver port 8080.

AROWSend

```
<path to AROWSend folder>\AROWSend\src\AROWSend.py -b -S TestSend -a 192.168.2.44 -p 9876 -P 50 -l 80000 -w 8081
```

This assumes a sending folder (in the same location as the script) named TestSend, a repetition period (-P) of 50 seconds and a rate limit of 80000 kbps (-l) for use on a 10/100Mbps Ethernet network. For webserver access we are using port 8081

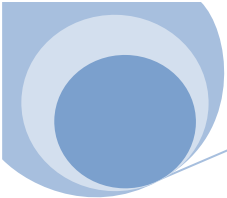
To see this working, start the AROWReceive shortcut, then the AROWSend shortcut. Each opens in a terminal window displaying the selected options. After a short time the receive window should show that a heartbeat has restarted. This is a small token that tells users on the normally isolated receive side that a sender is operational .

You can open a browser to see status on each script, <http://localhost:8080> and <http://localhost:8081> in this example. (or 127.0.0.1:8080 on Windows)

Now drag or copy a file into the TestSend folder. After a period not exceeding the repetition period (50 seconds in this example) the file should start to be transferred to the receive folder (reception). This will take time dependent on the size of the file and the flow rate limitation you set (80Mbps in this example), plus any delays due to general network congestion or platform activity.

You can also drag a folder tree to the send folder, it will also be transferred and the contents re-assembled in the original tree structure under the reception folder.

If file transfer activity is sufficiently prolonged, you will start to see a graphical display of data transfer rates on the browser.



AROW Data Diode

Software Emulator

The command line scripts can all be terminated by Ctrl-C. Note there will be a small delay on the ARWOSend script before closing to send a closedown message to the Receive side.

File Symlinks and Support Applications

Obviously it is not always practical or desirable to move or copy source or destination folders to the AROWBftp scanned directories. The answer to this is to use symlinks (Symbolic Links). These are links to files or folder structures that carry information about those files allowing the link to be treated as if it were the actual files/folders.

Symlink implementation varies on each operating system so you will need to consult documentation on your os to determine exact procedures.

Windows NTFS has had symlink support since Windows Vista, and this has been improved and expanded from Windows 7. Linux has always used symlinks.

A type of symlink is also available on Windows XP, called junctions. These can be tricky and quite dangerous to data since direct deletion will not only delete the link but also the files that were pointed to. Use with care, and manage the creation and deletion of junctions with a tool like Winbolic.

Placing symlinks in your send and receive folders means that AROWBftp can scan and reconstruct folders and files as if they were actually in these folders.

For example, if you have a mail database on your platform, perhaps outlook.pst or a thunderbird profile tree, simply create a symlink to the containing folder and drag it to the TestSend folder. The mail database will be transferred to the reception folder and can be accessed through your normal mail client. Be careful though, don't change your mail client settings unless you know what you are doing and disable automatic send/receive, or you will find your automatic send/receive has sent your mail to the receive location.

There are other examples and settings for Linux in the manual

Website viewing

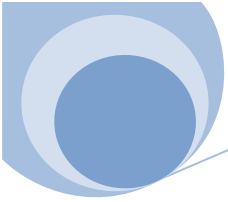
This is taken from the AROW manual

(Websites are by design interactive and require the user to ask for data and information via their browser. Clearly this is not allowed or available from a network protected by AROW, so a different strategy must be employed.

For websites with mostly static content, offline web cacheing tools are appropriate. Tools such as HTTrack, available for most operating systems, allow websites to be downloaded as a tree structure, creating files with all the appropriate links on a local storage system, thus creating a browseable offline cache.

These tools necessarily require some administration, to set up the levels of browsing, provide regular refreshes and so on, but when used with AROW enable users on the protected network to browse most of the functionality of a website, although of course no interaction that requires contacting the original site can take place.

Simply create a link to the cached files, and AROW will transfer them to the protected network. They can then be used to reconstruct and populate a proxy browser cache or simply browsed directly with the browser in offline or file mode.)



AROW Data Diode

Software Emulator

The above is more difficult to accomplish on a single pc with the emulator running and normal web access. You will have to set your browser to offline mode and point it at the transferred file structure.

TCP and UDP Streaming

AROWBftp includes the ability to connect to a TCP and/or UDP stream and mix-in stream data with file data.

AROWBftp acts as a server on each side of the data diode, allowing client streams to connect directly.

By default, AROWSend uses port 10002 to accept data from TCP sources and AROWReceive uses port 10000 to accept clients to output TCP data.

For UDP, AROWSend uses port 8002 to accept data from UDP sources and AROWReceive uses port 8001 to accept clients to output UDP data.

A system of priority queueing is used to give preference to stream frames over file and heartbeat frames, however processing overhead necessarily means that TCP stream rates are more restricted than in Direct Streaming mode.

File transfer is thus delayed when TCP/UDP streaming is used.

The next example needs a transcoder and media player. On windows, download and install ffmpeg, and VLC from Videolan, a free media player with lots of codecs.

Using a webcam

If you have a webcam running under Direct Show, install VLC, ffmpeg and Ncat for windows, and then try this from a command window:

```
'ffmpeg -list_devices true -f dshow -i dummy'
```

to get a list of the direct show names for your video and audio devices.

Run AROWSend in the normal way.

From a command line on the send server:

```
'ffmpeg -f dshow -i video="<yourcameraname>" -f mpegts tcp://127.0.0.1:10002'
```

Run AROWReceive in the normal way.

Run VLC and select network Streaming as the input. Type tcp:127.0.0.1:10000 in the dialog box then click play.

After a short delay (<10secs) the webcam image should appear on VLC..

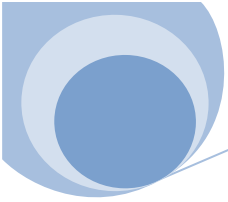
Note that for the purposes of this example we have shown a Windows server as the sending host and receiver, with appropriate syntax. (Under Linux simply find the appropriate camera device under /dev and change the ffmpeg syntax appropriately).

You can also send the video data over UDP, using the following :

```
ffmpeg -f dshow -i video="<yourcameraname>" -f mpegts udp://127.0.0.1:8002'
```

and udp://127.0.0.1:8001 in the media player.

If changed files are detected by AROWSend, these will be transmitted in the usual way, but any Stream frames detected will be given priority, thus maintaining a 'real-time' data flow.



AROW Data Diode

Software Emulator

Stream data can be disconnected or interrupted at any time without interrupting AROWBftp's file transfer operation.

However, high volumes of Stream data will delay file transfers so causing them to take longer.

Direct Data Streaming

It is possible to stream data directly through the emulator without using AROWBftp. Simply connect as a client to each server socket. Data received on the unprotected low side will be made available on the protected high side as soon as the high-side client connects.

eg `ffmpeg -f dshow -i video="<yourcameraname>" -f mpegts tcp://192.168.2.44:9876'`

and `tcp://192.168.2.44:9875` in the media player

This is useful to demonstrate the use of WinAROW.

WinAROW, the windows control program

Install and start WinAROW . This is a Windows only program that puts a GUI on the control port. The program will start requesting to open the setup window. Click OK and then enter the AROWEmulator network address . The default port of 10001 should be used for the low side control and 10002 for the high side control if using the same platform. Click connect and the program should communicate with the emulator. If there is heavy traffic going through the emulator you will see the buffer level indicator changing, and a high-tide marker indicating how full the buffer has been.

The buffer in AROW is very large, so there may be no obvious activity for small data transfers. Open the Info window to see the actual buffer level as an integer.