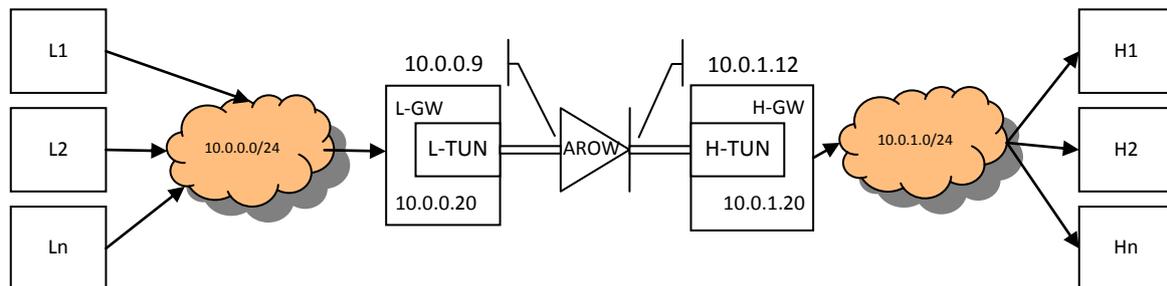


Tunnelling UDP through AROW using SOCAT

Multiple parallel UDP streams of data can be sent through AROW using socat to create virtual network interface ports using tcp socket connections to AROW as the "wire".

It is often required to send UDP data through Data Diodes, for example streaming process data, cctv, low-latency data streams. AROW's unique TCP connectivity means that it is possible to send multiple data streams of UDP data using the SOCAT application.



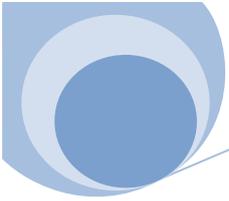
Socat is a Linux application that includes tunnelling a network interface over a TCP socket connection.

Socat can be found here: <http://www.dest-unreach.org/socat> and is available on most Linux distributions from the standard repository.

Tunnelling is a technique that allows data of different protocols to be embedded in an overall carrier protocol. Since the data diode is unidirectional, of course this is only appropriate for unidirectional protocols, that do not require responses from the receiving end in order to operate. UDP is such a protocol and widely used for sending low-latency data that can tolerate some data corruption, usually by incorporating a higher level layer that deals with errors and misalignments. CCTV often uses this approach, with a MPEG transport layer to deal with lower level link issues.

Socat enables multiple simultaneous UDP streams to be combined together into a tunnel. this can then be connected to a TCP stream, such as available in AROW. Another instance of Socat running on the receiving side restores the streams to their original or even re-mapped destinations.

An example serves to illustrate this.



Example

Ensure socat is installed on each side of the network on the platforms that will attach to AROW.

Assuming the low side gateway (L-GW) has a connection to the 10.0.0.0/24 network, create the low side of the tunnel (L-TUN) (the tunnel address is arbitrary but should not exist anywhere else on the network):

```
# socat -d -d TCP:10.0.0.9:9876 TUN:192.168.254.1/24,up
```

Assuming the high side gateway has a connection to the 10.0.1.0/24 network, create the high side of the tunnel(H-TUN):

```
# socat -d -d TCP:10.0.1.12:9876 TUN:192.168.254.2/24,up
```

(Note, -d -d is optional and is useful for confirming a network connection to AROW.)

Now we should have a "tun" interface in "ip addr" or "ifconfig" of each of the hosts.

On the low side gateway platform (L-GW) add a line to the routing table that send packets for the 10.0.1.0/24 network over the tunnel:

```
# route add -net 10.0.1.0/24 gw 10.0.0.20
```

Sending UDP over that tunnel is simple

For example:

On a high side host (for example H1 having the ip address 10.0.1.5):

```
$ netcat -u -s 10.0.1.5 -p 50002 10.0.0.5 50001
```

And on a low side host (for example L1 having the ip address 10.0.0.5):

```
$ netcat -u -s 10.0.0.5 -p 50001 10.0.1.5 50002
```

Now any packets sent from the low side netcat instance will be received by the netcat instance on the high side. Packets sent from the high side will be silently dropped.