

[https://github.com/DVSwitch/MMDVM\\_Bridge](https://github.com/DVSwitch/MMDVM_Bridge)

MMDVM\_Bridge (MB) does most of the heavy lifting required to build a bridge. In fact it IS the bridge. The job of MB is to connect to a foreign network and produce a data stream that can be sent to another program like Analog\_Bridge or even back to MB for transmission to another foreign network. Let's consider a simple YSF to DMR bridge.

YSF Reflector <-> MMDVM\_Bridge <-> FreeDMR

It does not get any simpler than that. MB connects to both the YSF Reflector and FreeDMR. No gateways needed, very simple. This is a single instance of MB. What do I mean by single instance? One copy of MB is running.

MB uses 2 configuration files. One file configures MB for connecting to the foreign network(s) The second file configures MB to send the data to another program or back to MB itself. Let's start with configuring MB to connect to the foreign networks. The file is MMDVM\_Bridge.ini I will only show the [stanzas] that really matter and only the fields.

### **MMDVM\_Bridge.ini**

```
[DMR]
Enable=1
```

```
[System Fusion]
Enable=1
```

The above 2 stanzas simply enable the DMR and YSF modes in MB

```
[DMR Network]
Enable=1
Address=3102.repeater.net
Port=62031
Password=passw0rd
```

```
[System Fusion Network]
Enable=1
GatewayAddress=ysfreflector.dvswitch.org
GatewayPort=42166
```

The above 2 stanzas enable and configure the foreign network connections required for the bridge. Now let's look at **DVSwitch.ini**

```
[DMR]
Address = 127.0.0.1      ; Address to send AMBE TLV frames to (export)
TXPort = 31100         ; Port to send AMBE TLV frames to (export)
RXPort = 31103        ; Port to listen on (import)
```

```
[YSF]
Address = 127.0.0.1      ; Address to send AMBE TLV frames to (export)
TXPort = 35100         ; Port to send AMBE TLV frames to (export)
RXPort = 35103        ; Port to listen on (import)
```

In the DMR stanza above, traffic coming from the DMR network, in this case, FreeDMR is processed by MB and sent out UDP port 31100 to IP address 127.0.0.1 (local host or the machine running MB) Traffic received on UDP port 31103 is processed and sent out to the DMR network. In the YSF stanza above, traffic coming from the YSF reflector is processed by MB and sent out UDP port 35100 to IP address 127.0.0.1 Traffic received on UDP port 35103 is processed and sent out to the YSF Reflector.

OK, so how do we bridge DMR to YSF? Do we need another program? Are we finished? No to all. What we want is when traffic is received from FreeDMR it is sent to YSF and vice versa. What we need to do, is send the traffic right back into MB.

DVSwitch.ini

[DMR]

Address = 127.0.0.1 ; Address to send AMBE TLV frames to (export)

TXPort = 31100 ; Port to send AMBE TLV frames to (export)

RXPort = 31103 ; Port to listen on (import)

[YSF]

Address = 127.0.0.1 ; Address to send AMBE TLV frames to (export)

TXPort = 31103 ; Port to send AMBE TLV frames to (export)

RXPort = 31100 ; Port to listen on (import)

Again,

Traffic coming in from FreeDMR is received by MB processed and sent out UDP port 31100 to IP address 127.0.0.1 (local host or the machine running MB) If you look at the TXPort for DMR it matches the RXPort for YSF. The TXPort for YSF matches the RXPort for DMR.

Let me present it a different way:

[DMR]

YSF

Address = 127.0.0.1 <---> Address = 127.0.0.1

TXPort = 31100 <-----> RXPort = 31100

RXPort = 31103 <-----> TXPort = 31103

That is all there is to it. If there is any magic, it's in what happens to the data received from the foreign network. MB takes apart the data and "normalizes" it so that another DVSwitch program can use it. That DVSwitch program can be Analog\_Bridge or even MB itself. MB can process 5 foreign networks at the same time. What MB can NOT do is to process 2 of the same foreign networks at the same time. For that, you need a second instance of MB.

Hope this helps.

73, Steve N4IRS

<https://dvswitch.groups.io/g/main>

<https://dvswitch.groups.io/g/main/subgroups>