Securely separating your applications at the network layer

(MQ Appliance)

Anthony\_Beardsmore |Nov 30 2016 Updated

**Introduction - separating applications**

A common reason for deploying the MQ Appliance is as part of a consolidation of queue managers currently running across multiple hosts into one place (sometimes referred to as a ‘messaging hub’.)   The MQ Appliance is an attractive target for this for a number of reasons – ease of maintenance, a solid hardware/performance foundation, and the built in High Availability capabilities, for example.

However, when hosting a large number of applications on the same appliance, various new considerations around ‘multi-tenancy’ come into play.  One of these considerations is network separation, which I’ll address in this article.

You can argue that the basic unit of ‘tenancy’ in MQ has always been the queue manager – it makes sense to co-locate applications which work with the same set of queues/topics, are accessed by the same pool of users, etc. on the same queue manager. This minimises both runtime issues  (e.g. needless transmit queue/channel hops) and administration headaches (e.g. moving a queue manager from one host to another in future, or managing connections to LDAP).

**Using multiple network interfaces**

With that level of separation in place, a common requirement is next to control which networks these queue managers interact with.  For example, your organisation may have test/acceptance/production networks which traffic must not ‘hop’ between.  Or you may have queue managers dedicated to interactions with particular lines of business, by way of specific subnets.

The appliance provides a lot of flexibility in how you achieve this.  In the simplest form you can actually rely solely on two traditional MQ controls – the ‘IPADDR’ attribute on a listener definition, and the ‘LOCALADDR’ attribute on a sender channel.  The appliance is well provisioned with network interfaces, and you could simply physically connect a 1GB Ethernet cable for each network which you need to access, and then bind a queue manager to that IP address.  If you are using MQ clustering, the ‘MQ\_LCLADDR’ environment variable can be configured for each queue manager in place of setting LOCALADDR (for your auto defined sender channels).

When configuring multiple interfaces in this manner, **it is best practice to only assign a default gateway for one** **interface** (to provide a default route from the system via your preferred subnet – perhaps on your management network or used by a particular set of queue managers).  For other networks, you will need to assign static routes using the ‘[ip-route](http://www.ibm.com/support/knowledgecenter/SS5K6E_1.0.0/com.ibm.mqa.doc/reference/linkaggcmds/ip-route_aggregate.htm)’ command.  See this [DataPower technote](http://www-01.ibm.com/support/docview.wss?uid=swg21249208) for more discussion of this topic.

**Aggregated links and VLANs**

However, this obviously only works if you have one physical interface available for each network, and is limited in terms of dependency on that single connection.  Two concepts can help us improve on this model: **Link Aggregation Definitions**and**VLAN Interface definitions.**

A [Link Aggregation](http://www.ibm.com/support/knowledgecenter/SS5K6E_1.0.0/com.ibm.mqa.doc/configuring/co00106_.htm) (sometimes known as a ‘bonded’ interface) combines multiple physical interfaces into a single ‘virtual’ interface with one IP address.  The physical interfaces must NOT be assigned individual addresses (you must in fact mark the physical interfaces as ‘available for link aggregation’ which suppresses the option to define an individual IP).  There are various mechanisms for link aggregation (see KC), but generally LACP is used to allow the switch to spread packets to and from this aggregated interface across all physical connections – so for example 6 1GB connections can be treated as a single 6GB connection.

So, now we have one highly available, high performance link.  However, how can we treat this as a connection to multiple, ‘physically separate’ networks, and connect our queue managers only to the appropriate subnets?

The answer is that the appliance also supports native VLAN tagging (sometimes called ‘trunked’ or ‘trunk mode’).  A [VLAN interface](http://www.ibm.com/support/knowledgecenter/SS5K6E_1.0.0/com.ibm.mqa.doc/configuring/co00103_.htm) is configured on top of either a single physical interface or an aggregation, and packets sent through the VLAN interface are ‘tagged’ such that the switch knows which physical network they must be routed to – and vice versa for incoming data.  The VLAN interface has its own IP address, which can again be used in the IPADDR/LOCALADDR fields to bind a queue manager to a particular physical network.

One further thing to note is that **rather than explicitly using IP addresses in your MQ definitions, it is good practice to define ‘host aliases’**.  This allows you to modify the IP address/interface without changing your definitions (other than the alias), and is crucial in HA deployments where the same MQ definitions will be used on two systems with different interface addresses.

**Big Picture**

Bringing it all together, at a simplistic level the final picture might look something like this:

QM1 (Localaddr/IPaddr) <-> (VL1 Alias) ->  VLAN1 -------\                         [Subnet 1]

                                                         \                     /

QM2 (Localaddr/IPaddr) <-> (VL2 Alias) ->  VLAN2 ------  LinkAggr ====== **Switch** – [Subnet 2]

                                                         /                     \

QM3 (Localaddr/IPaddr) <-> (VL3 Alias) ->  VLAN3 -------/                         [Subnet 3]

There are many options in the configuration of this, and of course many more combinations of these features are possible, but hopefully this has given a flavour of how you can achieve clean separation between networks, applications and queue managers on the MQ appliance.

**Further reading**

For more on this and related topics see:

* [Planning Network connections (MQ Appliance Knowledge Center)](http://www.ibm.com/support/knowledgecenter/SS5K6E_1.0.0/com.ibm.mqa.doc/planning/pl00600_.htm)
* [The MQ Appliance Redbook](http://www.redbooks.ibm.com/abstracts/sg248283.html?Open)
* [Jamie's blog on using multiple TCP stacks with (software) MQ](https://community.ibm.com/HigherLogic/System/DownloadDocumentFile.ashx?DocumentFileKey=c98b09f5-6dcb-09eb-d631-4bd647a8e2aa&forceDialog=0)
* [Technote discussing managing multiple routes from appliances](http://www-01.ibm.com/support/docview.wss?uid=swg21249208) (though written for DataPower systems, advice also applies to MQ Appliance).