

Re: arp-proxy

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/net/2005-11/0176.html>

From: Brian Candler (*B.Candler_at_pobox.com*)

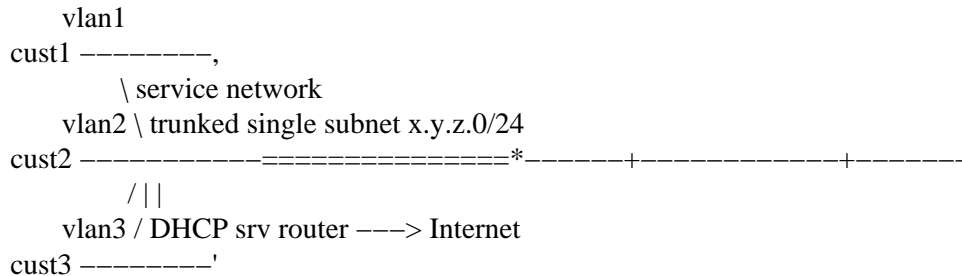
Date: 11/21/05

Date: Mon, 21 Nov 2005 11:28:56 +0000

To: Jon Otterholm <jon.otterholm@ide.resurscentrum.se>

> > *On Thu, Nov 17, 2005 at 04:52:03PM +0100, Jon Otterholm wrote:*
> > > *Scenario#1:*
> > > *-I have a range of ip's, for example 215.10.10.0 - 215.10.10.255.*
> > > *-I want to distribute these ip's to my customers via DHCP.*
> > > *-They are all attached to me via a VLAN-trunk on a unique VID*
> > > *-I have 200+ customers.*

Let me see if I can summarise your requirements:



The constraints are:

- cust1, cust2, cust3 are all on the same subnet x.y.z.0/24 and get an IP address allocated by DHCP
- However, the MAC address for cust1 must never appear as a source MAC address on vlan2 or vlan3, as this would confuse the provider's trunked switching infrastructure

So you can't just bridge all the VLANs together. Rather:

1. a broadcast from custX must appear on the service network, but not on any of the other customer VLANs. More strongly, no packet from custX may appear on any other VLAN apart from the service network. [*]
2. a broadcast from the service subnet should appear on all customer VLANs (e.g. ARP from DHCP server or router)

freebsd-net: Re: arp-proxy

3. an ARP request for custY from custX must be proxy-ARP responded to by a device on the service network, otherwise customers wouldn't be able to communicate with each other.
4. a unicast packet from the service network to a customer must be forwarded to the correct customer VLAN

Is that a reasonable summary? Taking that as the base:

Point (4) implies that a bridge forwarding table must still be built, because the device performing this function (labelled '*' in the above diagram) needs to associate a MAC address with a VLAN, and do so by learning rather than static configuration.

Point (1) says that you can't just bridge all the VLANs together, because otherwise a packet to a broadcast address or to an unknown MAC address would be forwarded to all the other VLANs. We want this to happen for packets originating from the service network (point (2)), but not for packets which originate from the customer networks. So, some sort of L2 forwarding filter should do the trick: configure it so that packets may only be forwarded to customer VLANs if they originate from the service network. If this filter is applied, you guarantee that a customer's MAC address will never appear as a source on any other VLAN.

Point (3), proxy ARP, is easy enough. You know all the possible customer IPs – they are exactly the range assigned to the DHCP server to allocate – and therefore you can proxy-ARP respond for any IP address within the DHCP range, as long as the request originated from a customer VLAN (which can also be determined by the ARP source IP). The router itself could perform this proxy ARP function, or else any server on the service network running something like choparp (which can give out the router's MAC address in the ARP responses). IP datagrams from custX to custY then go custX->router->custY.

So actually, when thought about like this, the L2 masquerading requirement vanishes, and what you really need is bridging plus some L2 filtering based on ingress and egress interfaces.

Unfortunately, I don't know if FreeBSD has this level of L2 filtering (I note that the bridge(4) documentation says that ipf/ipfw filtering only works for IP datagrams). However a frob on the bridging code should be possible; call the first interface 'master' and the rest 'slaves', and have a rule so that packets to a 'slave' interface are only forwarded if they originated from the 'master' interface.

Aside: the network as designed above has an obvious flaw that any customer can DHCP for as many different machines as they wish, and therefore exhaust your DHCP pool. You could have a separate mini DHCP server listening on each VLAN and only handing out a single IP, but that doesn't stop customers stealing other customer's IPs through static configuration. So actually, I think you need anti-spoofing filters on each VLAN too.

Re: arp-proxy

freebsd-net: Re: arp-proxy

Doing that, you end up statically routing a separate IP down each VLAN, in which case what you *really* want is to be able to configure each VLAN subinterface as if it were a point-to-point interface. But I don't think FreeBSD supports that on broadcast media.

Regards,

Brian.

[*] Or if it did appear on the other customer VLANs, it would have to be masqueraded to appear as if it came from a MAC address on the service network; however I believe this isn't actually necessary, as the only broadcasts we really care about here are ARP requests. All others can be dropped, and indeed probably should be dropped so that all your customers don't get drowned in each other's broadcast traffic.

freebsd-net@freebsd.org mailing list

<http://lists.freebsd.org/mailman/listinfo/freebsd-net>

To unsubscribe, send any mail to "freebsd-net-unsubscribe@freebsd.org"