AMS-IX Port Configuration Hints

Steven Bakker, AMS-IX NOC

Amsterdam Internet Exchange, B.V.

E-mail: <noc@ams-ix.net>

This article gives some pointers towards setting up your device when connecting to the AMS-IX. AMS-IX rules restrict the type of traffic and number of source MAC addresses that any member is allowed to send to the exchange.

Table of Contents

1. Introduction **3**

1.1. Definition of Terms 3

2. General Configuration Recommendations **3**

2.1. IPv4 ARP / IPv6 Neighbor Timeout 3

2.2. Peering LAN Prefix 3

2.3. BGP Routing 4

3. Allowed Traffic Types and Configurations **4**

3.1. Physical L2 Topology 5

3.2. Commonly Seen Illegal Traffic and Set-Up 6

4. Cisco Configuration Hints **8**

4.1. Global Config 8

4.2. Interface Config 9

4.3. Layer 2 Config 9

5. Juniper Configuration Hints **10**

5.1. Unicast BGP Configuration 11

5.2. IPv4 ARP cache timeout 11

6. Extreme Networks Configuration Hints **12**

6.1. L2 Configuration 12

6.2. L3 Configuration 12

7. Foundry Configuration Hints **13**
8. Linux Configuration Hints **14**
8.1. ARP filtering and source routing 14
8.2. IPv4 ARP cache timeout 16
8.3. IPv6 neighbor cache timeout 17
8.4. RP filter setting 17
8.5. Running the “sysctl” commands at boot 17
9. Riverstone **18**
10. Acknowledgements **19**

1. Introduction

The Amsterdam Internet Exchange operates as a shared layer 2 (L2) Ethernet infrastructure. Large Ethernet LANs require that more or less everyone plays by the same set of rules. In other words, it can be quite sensitive to misbehaviour.

In order to improve the stability of the Exchange, AMS-IX has defined a set of rules to which every member's connnection *must* adhere, the *Technical Specifications* (http://www.ams-ix.net/members/techspec.html).

Not everybody immediately grasps the subtleties of configuring equipment to adhere to the rules, so this document tries to fill in some blanks and provide examples and hints for the most common equipment.

1.1. Definition of Terms

In this document we refer to terms like "L2 device", "L2/L3 hybrid", etc. It may be worthwile to explain what we mean by them here.

L2 Device

A device that functions as a *Layer 2 (Ethernet) Bridge* (a.k.a. "switch", "bridge", "hub", etc).

L3 Device

A device that functions as a L3 (IP) router only. This means it does not bridge any Ethernet frames between its interfaces. Such a device is typically called a "router".

L2/L3 Hybrid

A device that functions both as a L2 bridge and a L3 router. This means it can both bridge Ethernet frames between its interfaces as well as route IP traffic and participate in IP routing protocols. Foundry and Extreme are common examples of this type of device.

2. General Configuration Recommendations

2.1. IPv4 ARP / IPv6 Neighbor Timeout

Each equipment vendor implements his own maximum ages for the IPv4 ARP and IPv6 neighbor caches. The values vary widely and in at least one case (Linux) it is not a constant.

Low ARP timeouts can lead to excessive ARP traffic, especially if the values are lower than the BGP `HELLO` interval timers. On the other hand, long timeouts can theoretically lead to longer downtime if you change equipment (since your peers still have the old MAC address in their ARP cache); with BGP, however, this is unlikely to happen: as soon as your router is back up, it will start re-establishing its BGP peerings and this will cause your peers to update their ARP caches as well.

We recommend setting the ARP cache timeout to at least two hours, preferably four (240 minutes). See the sections on specific equipment vendors for examples.

2.2. Peering LAN Prefix

The IPv4 prefix for the AMS-IX peering LAN (`195.69.144.0/23`) is part of AS1200, and is not supposed to be globally routable. This means the following:

1. Do *not* configure "`network 195.69.144.0/23`" in your router's BGP configuration (seriously, we have seen this happen!).
2. Do *not* redistribute the route, a supernet, or a more specific outside of your AS. We (AS1200) announce it with a `no-export` attribute, please honour it.

In short, you can take the view that the Peering LAN is a link-local address range and you may decide to not even redistribute it internally (but in that case you may want to set a static for your NOC so you can troubleshoot peerings, etc.).

2.3. BGP Routing

Please exchange only unicast routes over your BGP sessions in the ISP peering LAN. Exchanging multicast routes is useless since multicast traffic is not allowed on the (unicast) ISP peering LAN.

3. Allowed Traffic Types and Configurations

The *Technical Specifications* (http://www.ams-ix.net/members/technical/index.html) state the following:

1. There are only three ethertypes allowed:

   a. `0x0800` - IPv4

   b. `0x0806` - ARP

   c. `0x86dd` - IPv6

   This implies IEEE 802.3 compliance, *not* 802.2, so no LLC encapsulation!

2. Only *one* MAC address allowed on a port, i.e. all frames sent towards the AMS-IX should have exactly one unique MAC address.
3. The only non-unicast traffic allowed is:
   - Broadcast ARP.
   - Multicast ICMPv6 Neighbour Discovery (ND) packets. (NOTE: this does *not* include Router Advertisement (ND-RA) packets!)
4. AMS-IX member equipment should only reply to ARP queries for IP addresses of their directly connected AMS-IX interface. In other words, proxy ARP is not allowed.
5. Traffic for link-local protocols is not allowed, except for ARP and IPv6 ND (see above).

6. IP packets addressed to AMS-IX peering LAN's directed broadcast address shall not be automatically forwarded to AMS-IX ports.
7. The speed and duplex setting of 10baseT and 100baseTX ports must be statically configured, i.e. auto-negotiation should be disabled.

3.1. Physical L2 Topology

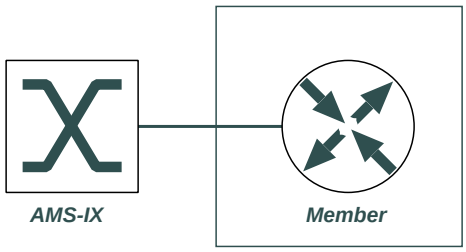
The AMS-IX rules dictate that only one MAC address is allowed behind a port. This means that you have to be extremely careful when connecting a device that can act as a L2 device. In general, we do not recommend using L2 devices between a member's router and the AMS-IX switch, except when used as a media converter.

The reason for allowing only one MAC address that we want no additional L2 network behind the AMS-IX ports. Extended L2 networks are not under the control of the AMS-IX, but instabilities in a L2 network behind the AMS-IX switches *can* and typically *do* have a negative impact on the whole exchange. Forwarding loops and spanning tree topology changes are good examples of this. By enforcing the one-MAC-address-per-port rule, we effectively prevent forwarding loops and STP traffic from intermediate L2 devices.

In short, an intermediate L2 device may only bridge frames from the member's router to the AMS-IX port (so we see only one MAC address) and should otherwise be completely invisible. No connected device should bridge frames from other devices onto the AMS-IX, or talk STP on its AMS-IX interface.

3.1.1. Connecting a L3 Device

The most preferred way of connecting to the AMS-IX is directly through a L3 device (router), see the diagram below.

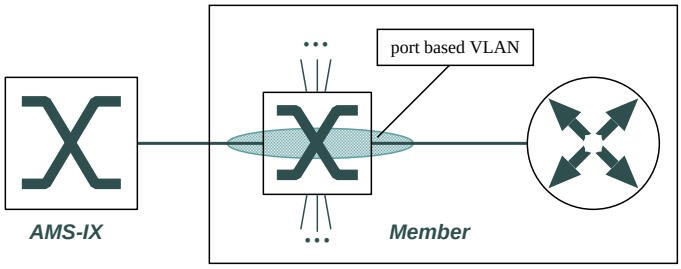


This is your best chance of not leaking MAC addresses or STP traffic and it greatly increases the stability of the network.

3.1.2. Connecting Through a L2 Device

We neither recommend nor encourage connecting your router through a L2 device, but if you do so, keep the following in mind:

- You *must* make *absolutely sure* that only traffic to/from your L3 router's interface goes to/from the AMS-IX port.
- You *must* disable Spanning Tree on your link to AMS-IX.

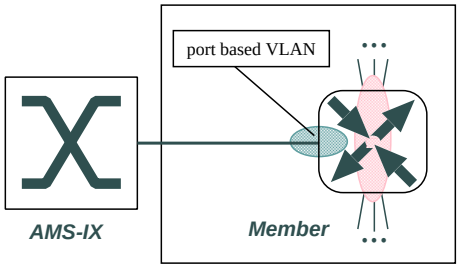


On all intermediate L2 devices, consider using explicitly defined port-based VLANs for production ports. It forces you to understand your topology and reduces the chances of a nasty surprise further down the road. In particular, we strongly recommend using a dedicated VLAN for the path from your router to the AMS-IX.

3.1.3. Connecting a L2/L3 Hybrid

The L2/L3 hybrid switch/router requires careful configuration in order to prevent unwanted traffic from leaking onto the exchange. As with intermediate L2 devices, you need to keep the following in mind:

- You *must* make *absolutely sure* that your AMS-IX port is configured as a “router only” port.
- You *must* disable Spanning Tree on your link to AMS-IX.



On a L2/L3 hybrid device, it is a good idea to put the AMS-IX connected interface (untagged) in a separate (non-default) port-based VLAN *without spanning tree* and with no other ports in it. This is the best way to ensure that no traffic from other ports will be bridged onto the AMS-IX port.

3.2. Commonly Seen Illegal Traffic and Set-Up

Any traffic other than the types mentioned in the previous section is deemed to be illegal traffic. In this section we will list some of the more common types of violations we see at the AMS-IX and give some arguments as to why it is considered unwanted.

3.2.1. Multiple MAC addresses

Since the AMS-IX operates on the principle of one router per port, there should be one MAC address visible behind each port. Some members connect through intermediate switches, or use a L2/L3 hybrid device. If these devices are not configured properly, they can cause forwarding loops, STP instabilites, and lots of unwanted traffic on the exchange. There is no excuse for these devices to leak traffic, and there is no necessity to talk STP on the link to the AMS-IX. Hence, by enforcing the one-MAC-address rule, we also enforce these issues. Beware that this rule is enforced automatically, so if you leak traffic from another MAC address, your bona fide traffic may be blocked (depending on which MAC address the switch sees first) or your port may be shut down for a few minutes.

3.2.2. Spanning Tree (STP)

This point is closely connected to the previous point. The device(s) connected to the AMS-IX port are not allowed to be visible as L2 bridges. This means that they should not speak STP (spanning tree) or any other (proprietary) L2 specific protocol.

3.2.3. Routing protocols: EIGRP, OSPF, RIP, IS-IS

The only routing protocol allowed on the AMS-IX is BGP. There is no valid reason for interior routing protocols to appear on the shared medium. These protocols only cause unnecessary multicast and broadcast traffic.

3.2.4. (Cisco) Keepalive

By default Cisco routers and switches periodically test their (Fast) Ethernet links by sending out Loopback frames (ethertype 0x9000) addressed to themselves. Call it a “L2 self-ping” if you will. In a switched environment it can be used to test the functionality of the switch and/or keep the router’s MAC address in the switch’s address table.

In the AMS-IX environment, this is not useful since we use MAC timeouts that are larger than the typical BGP and/or ARP timeouts. In fact, the keepalives a may actually cause port security violations if they are being sent by an intermediate switch.

3.2.5. Discovery protocols: CDP, EDP

Various vendors (e.g. Extreme, Cisco) tend to ship their boxes as gregarious devices: by default they announce their existence out of all their interfaces and try to find family members. `CDP` (Cisco) and `EDP` (Extreme) are examples of this, but there are others.

The only reason for running discovery protocols is to support certain types of autoconfiguration. Autoconfiguration on an Internet Exchange is a very bad idea. Hence, there is absolutely no reason to run discovery protocols on your AMS-IX interface. Discovery protocols typically cause unwanted broadcast or multicast traffic.

3.2.6. Non-unicast IPv4: IGMP, DHCP, TFTP

On the ISP peering LAN, the only non-unicast traffic that is allowed is the ARP query.

Sometimes we see equipment trying to get a configuration through broadcast TFTP, or configure themselves through DHCP. We will leave it to the reader to consider why this is a bad idea.

Other equipment has IGMP turned on by default (or by accident). The Peering LAN is for unicast IP traffic only, so there is no point in configuring multicast on the AMS-IX interface.

3.2.7. Proxy ARP

Since traffic over the AMS-IX is exchanged based on BGP routes, there is no reason to answer ARP queries for any other IP address(es) than those that are configured on your AMS-IX interface.

Unfortunately, some vendors (e.g. Cisco) ship their products with proxy ARP enabled by default.

Proxy ARP is not only sloppy, it can lead to unwanted traffic on your network. Consider that if you have it enabled at the AMS-IX, it's likely to be enabled at other peering points, allowing parties on both sides to use you as a transit.

Proxy ARP is not allowed.

3.2.8. Non-unicast IPv6: IPv6 ND-RA

IPv6 router advertisements are not allowed: they generate a lot of unnecessary traffic, since IPv6 hosts on the AMS-IX are not autoconfigured and besides, you don't want to be the default router for the whole AMS-IX.

3.2.9. Miscellaneous non-IP: DEC mop, etc.

Some vendors enable protocols other than IP by default. Cisco, for example ships certain versions of IOS with DEC MOP enabled by default. This is non-IP traffic and has no place on the AMS-IX.

4. Cisco Configuration Hints

Cisco's philosophy seems to be similar to that of some PC OS vendors: enable as many protocols and features as possible by default, so the device works out-of-the-box in most situations. Unfortunately, this means that a lot of unnecessary features are turned on that, while harmless in LAN or corporate environments, can cause undesired traffic on an Internet exchange.

Typical things that need to be disabled are: autoconfiguration protocols (DHCP, BOOTP, TFTP config download over the AMS-IX interface), CDP, DEC MOP, IP redirects, IP directed broadcasts, proxy ARP, IPv6 Router Advertisements, keepalive.

Intermediate switches or hybrid devices will also need to disable VTP, STP, etc.

4.1. Global Config

```
! Do not run a DHCP server/relay agent
no service dhcp

! Older IOS versions require this instead of the above.
no ip bootp server
```

```
! Do not download configs through TFTP
no service config

! Do not run CDP
no cdp run
```

4.2. Interface Config

```
! Don’t do redirects -- if they don’t know
! how to route properly, tough luck!
no ip redirects

! Don’t run proxy ARP on your AMS-IX interface
no ip proxy-arp

! Don’t run CDP on your AMS-IX interface
no cdp enable

! Directed broadcasts are evil.
no ip directed-broadcast

! v6 ND-RA is unnecessary and undesired
ipv6 nd suppress-ra

! Disable the DEC drek if you haven’t done so globally yet.
no mop enable

! For (Fast)Ethernet: no auto-negotiation on your connection.
! no negotiation auto
! duplex half
duplex full

! L2 keepalives are useless on the AMS-IX
no keepalive
```

4.3. Layer 2 Config

It is difficult to give a complete guide for Cisco products, because of the many different types of devices and (IOS) software versions. When in doubt, consult your documentation.

4.3.1. 29xx and 35xx series

If you use a Cisco Layer 2 device (such as the 2900 and 3500 series), you have to turn off VTP (VLAN Trunking Protocol), DTP (Dynamic Trunking Protocol) and UDLD.

In global config mode:

```
vtp mode transparent
!
```

```
no spanning-tree vlan 1200
!
vlan 1200
 name AMS-IX
!
interface IfIdent
 description Interface to AMS-IX
 switchport access vlan 1200
 switchport mode access
 switchport nonegotiate
 no keepalive
 speed nonegotiate
 no cdp enable
 no udld enable
 ! If you do not want to shut off STP for some bizarre reason...
 spanning-tree bpdufilter enable
end
```

4.3.2. Catalyst devices

CatOS and IOS are different beasts, so for Catalyst switches, the following applies:

```
set vtp mode off
set port name IfIdent My AMS-IX Port
set cdp disable IfIdent
set udld disable IfIdent
set trunk IfIdent off dot1q
set spantree bpdu-filter IfIdent enable
set vlan 1200 name My_AMS-IX_Vlan
set vlan 1200 IfIdent
```

If, for some reason, you cannot afford to turn off VTP globally, the only way to turn it off on individual ports seems to be by using l2pt:

```
set port l2protocol-tunnel IfIdent vtp enable
```

Depending on your CatOS platform, you may or may not be able to do this.

4.3.3. Other devices

For other devices, some or all of the above may apply. Check your documentation for details.

5. Juniper Configuration Hints

For Juniper routers, there isn't much to disable. The *Juniper Documents* (http://www.qorbit.net/documents.html) from *qOrbit Technologies* (http://www.qorbit.net) contain useful hints on how to set up your Juniper router.

IGMP Bug (PR/20343) in JunOS versions 5.3R4

There's a bug in JunOS versions up to 5.3R4, that will cause a Juniper router to emit IGMP packets on all its interfaces, even when IGMP is disabled. The only way to stop your router from transmitting IGMP is to configure outgoing packet filters on your AMS-IX interface(s).

5.1. Unicast BGP Configuration

Make sure to exchange only unicast routes in the unicast ISP peering LAN by explicitly adding the following statement to *all* neighbors, groups and prefix-limits:

```
set family inet unicast
```

Be thorough with `family inet unicast`

If even one of the neighbors, groups or prefix-limits is defined with a family inet "any", you'll enable multicast and turn on MBGP.

5.2. IPv4 ARP cache timeout

Juniper's default ARP cache timeout is 20 minutes (by comparision: Cisco's default ARP cache timeout is 4 hours which fits AMS-IX's relatively static environment much better).

To reduce the amount of unnecessary broadcast traffic, we recommend setting the ARP cache timeout on Juniper routers to 4 hours. A recipe for this follows:

```
> configure
Entering configuration mode

[edit]
you@juniper# edit system arp

[edit system arp]
you@juniper# set aging-timer 240

[edit system arp]
you@juniper# show | compare
[edit system arp]
+ aging-timer 240;

[edit system arp]
you@juniper# commit and-quit
commit complete
Exiting configuration mode
```

6. Extreme Networks Configuration Hints

Updating Firmware in an EAPS Environment

When updating firmware in an Extreme Networks EAPS environment, be sure to temporarily disable your AMS-IX port(s). TFTP file transfers may cause EAPS instabilities resulting in bogus traffic. This is likely to trip the port security on the AMS-IX switches, which may result in 10 minutes downtime.

Most people who use Extreme equipment do not have problems with their AMS-IX connections, some do. We would appreciate feedback from people running Extreme equipment on how they configure their AMS-IX facing side.

6.1. L2 Configuration

The configuration fragment below shows how to configure an intermediate *L2 switch*, which is also part of an EAPS ring. Port 1 is connected to the AMS-IX switch. Ports 2 and 3 are in the ring. The router is somewhere in that ring, in the "`amsix`" VLAN.

```
create vlan "ring"
configure vlan "ring" tag 1200     # VLAN-ID=0x4b0  Global Tag 3
configure vlan "ring" qosprofile "QP8"
configure vlan "ring" add port 2 tagged
configure vlan "ring" add port 3 tagged

create vlan "amsix"
configure vlan "amsix" tag 1700     # VLAN-ID=0x6a4  Global Tag 9
configure vlan "amsix" add port 1 untagged
configure vlan "amsix" add port 2 tagged
configure vlan "amsix" add port 3 tagged

configure port 1 auto off speed 1000 duplex full
configure port 2 auto off speed 1000 duplex full
configure port 3 auto off speed 1000 duplex full

disable edp port 1
disable igmp snooping
disable igmp snooping with-proxy

create eaps "ring-eaps"
configure eaps "ring-eaps" mode transit
configure eaps "ring-eaps" primary port 2
configure eaps "ring-eaps" secondary port 3
configure eaps "ring-eaps" add control vlan "ring"
configure eaps "ring-eaps" add protect vlan "amsix"
enable eaps "ring-eaps"
```

6.2. L3 Configuration

The configuration fragment below shows the relevant configuration information for a L3-only device. As in the previous example, port 1 is connected to the AMS-IX and is configured in the "`amsix`" VLAN (untagged).

```
#
# Config information for VLAN amsix.
#
create vlan "amsix"
configure vlan "amsix" tag 1200
configure vlan "amsix" protocol "IP"
configure vlan "amsix" ipaddress 195.69.14X.Y 255.255.254.0
configure vlan "amsix" add port 1 untagged
#
configure port 1 display-string "AMS-IX"
disable edp port 1
#
enable ipforwarding vlan "amsix"
disable ipforwarding broadcast vlan "amsix"
disable ipforwarding fast-direct-broadcast vlan "amsix"
disable ipforwarding ignore-broadcast vlan "amsix"
disable ipforwarding lpm-routing vlan "amsix"
disable isq vlan "amsix"
disable irdp vlan "amsix"
disable icmp unreachable vlan "amsix"
disable icmp redirects vlan "amsix"
disable icmp port-unreachables vlan "amsix"
disable icmp time-exceeded vlan "amsix"
disable icmp parameter-problem vlan "amsix"
disable icmp timestamp vlan "amsix"
disable icmp address-mask vlan "amsix"
disable subvlan-proxy-arp "amsix"
configure ip-mtu 1500 vlan "amsix"
#
# IP Route Configuration
#
configure iproute add blackhole default
disable icmpforwarding vlan "amsix"
disable igmp vlan "amsix"
```

7. Foundry Configuration Hints

The following fragment of configuration gives an idea of how to configure a Foundry (BigIron) device. Depending on the actual role of the device (router or switch between router and AMS-IX) and the type of code loaded into the device you may need to mix and match a little here.

```
! Define a single-port VLAN for the AMS-IX port
vlan number name "AMS-IX" by port
no spanning-tree
untagged ethernet i/f
```

```
! Configure the AMS-IX interface
interface ethernet i/f
 port-name "AMS-IX"

! Behave as a router.
 route-only
 no spanning-tree

! Don't do IPv6 ND-RA (Router Advertisements)
 ipv6 nd suppress-ra

! No weird discovery proto, please.
 no vlan-dynamic-discovery

! IP address
 ip address 195.69.14X.Y 255.255.254.0

! No redirects
 no ip redirect

! AMS-IX recommends 2 hour ARP timeouts
 ip arp-age 120

! For fast-ethernet: no autoconfig.
 speed-duplex 100-full
```

8. Linux Configuration Hints

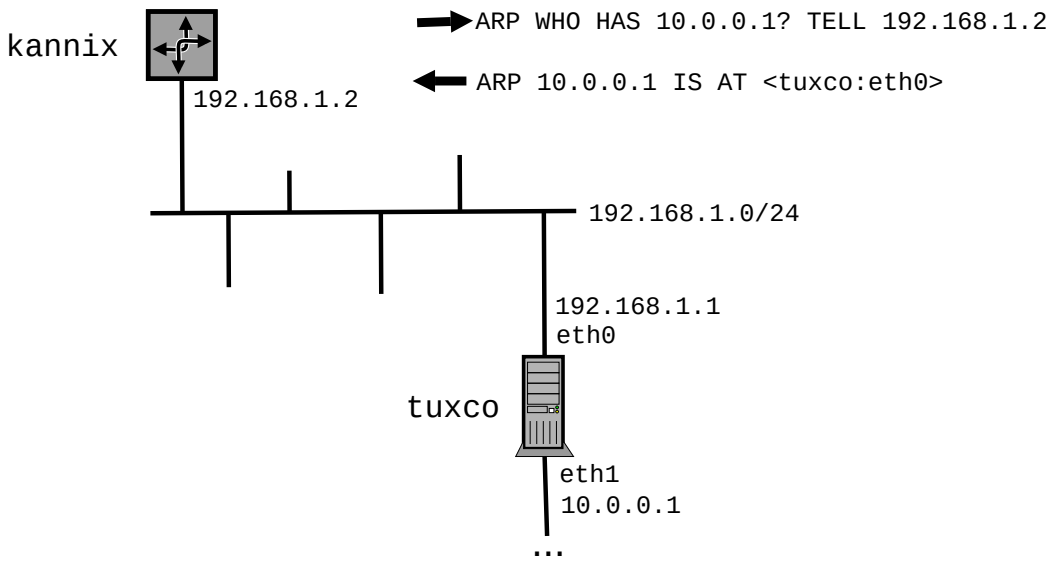
We are not aware of any major issues with Linux boxes used as routers, and they seem to be pretty rare on the Exchange. Having said that, there are a few parameters that can (and usually should) be tuned:

1. ARP filtering & source routing
2. ARP cache timeout
3. Reverse Path (RP) filter

For more information on tuning your Linux system for routing, see the *Linux Advanced Routing & Traffic Control HOWTO* (http://www.tldp.org/HOWTO/Adv-Routing-HOWTO/index.html).

8.1. ARP filtering and source routing

The Linux approach to IP addresses is that they belong to the system, not any single interface. As a result, Linux hosts have a default behaviour that is different from most other systems: interfaces semi-promiscuously answer for *all* IP addresses of all other interfaces. Example:



In this example, host `tuxco` is a Linux box with a peering connection on `eth0` (`192.168.1.1/24`) and a backbone link on `eth1` (`10.0.0.1/24`).

When host `kannix` (`192.168.1.2`) sends an ARP query for `10.0.0.1` it will get a reply from `tuxco`'s `eth0` interface!

In other words, a Linux host will answer to ARP queries coming in on any interface if the queried address is configured on *any* of its interfaces. The idea behind this is that an IP address belongs to the system, not just a single interface. Although this may work well for server or desktop systems, it is not desirable behaviour in a router system. One reason is that it is a limited version of proxy-arp, which is forbidden on the AMS-IX peering LAN. Another reason is that two separate routers could potentially answer ARP queries for the same RFC1918 address.

8.1.1. Fixing ARP

The ARP behaviour can be fixed by using `arp_ignore` and `arp_announce` on the WAN interface:

```
tuxco# sysctl -w net/ipv4/conf/eth0/arp_ignore=1
tuxco# sysctl -w net/ipv4/conf/eth0/arp_announce=1
```

8.1.2. Multiple interfaces on one subnet

If you have multiple interfaces on the same subnet, you may also want to enable `arp_filter`:

```
tuxco# sysctl -w net/ipv4/conf/eth0/arp_filter=1
```

This prevents the ARP entry for an interface to fluctuate between two or more MAC addresses. However, you need to use source routing to make this work correctly. From the `Documentation/networking/ip-sysctl-2.6.txt` file in the kernel source:

> [. . .]

arp_filter - BOOLEAN

1 - Allows you to have multiple network interfaces on the same subnet, and have the ARPs for each interface be answered based on whether or not the kernel would route a packet from the ARP'd IP out that interface (therefore you must use source based routing for this to work). In other words it allows control of which cards (usually 1) will respond to an arp request.

[. . .]

8.2. IPv4 ARP cache timeout

The ARP cache timeout on Linux-based routers should be changed from the default, especially if you have a large number of peers. This parameter can be tuned by setting the appropriate `procfs` variable through the **sysctl** interface. The Linux `arp`(7) manual says:

[. . .]

SYSCTLS

ARP supports a sysctl interface to configure parameters on a global or per-interface basis. The sysctls can be accessed by reading or writing the `/proc/sys/net/ipv4/neigh/*/*` files or with the **sysctl**(2) interface. Each interface in the system has its own directory in `/proc/sys/net/ipv4/neigh/`. The setting in the '`default`' directory is used for all newly created devices. Unless otherwise specified time related sysctls are specified in seconds.

[. . .]

`base_reachable_time`

Once a neighbour has been found, the entry is considered to be valid for at least a random value between base_reachable_time/2 and 3*base_reachable_time/2. An entry's validity will be extended if it receives positive feedback from higher level protocols. Defaults to 30 seconds.

This means that Linux systems keep ARP entries in their cache for some time between 15 and 45 seconds (and yes, the average works out to 30 seconds). This is not very high. In fact, it is lower than the typical BGP HELLO interval and may thus result in excessive ARPs.

We suggest a timeout of at least two hours for ARP entries on your AMS-IX interface, so you'd have to set the `base_reachable_time` to 2 x 2hrs = 4 hours.

```
tuxco1# sysctl net.ipv4.neigh.ifname.base_reachable_time
net.ipv4.neigh.ifname.base_reachable_time = 30
```

The above command tells you that the ARP cache timeout is 30 seconds average. To change it so it's between 2 and 6 hours, use the following command:

```
tuxco1# sysctl -w net.ipv4.neighifname.base_reachable_time=14400
net.ipv4.neigh.ifname.base_reachable_time = 14400
```

Here ***ifname*** is the name of the interface that connects to AMS-IX. You can also use "`default`" here, but that may have undesired side-effects for your other interfaces.

8.3. IPv6 neighbor cache timeout

As with the IPv4 ARP cache, Linux systems tend to set the lifetime of the IPv6 neighbor cache quite short as well. The lifetime is controlled in a similar way as for IPv4 ARP:

```
tuxco1# sysctl net.ipv6.neighifname.base_reachable_time
net.ipv6.neigh.ifname.base_reachable_time = 30

tuxco1# sysctl -w net.ipv6.neighifname.base_reachable_time=14400
net.ipv6.neigh.ifname.base_reachable_time = 14400
```

8.4. RP filter setting

You may need to turn off the Reverse Path Filter (`rp_filter`) functionality on a Linux-based router to allow asymmetric routing, particularly on your WAN interface.

To disable the RP filter:

```
tuxco1# sysctl -w net.ipv4.confifname.rp_filter=0
```

8.5. Running the "sysctl" commands at boot

The various system parameters discussed above can be set at boot time by adding it to a file such as `/etc/sysctl.conf`. The exact name, location and very existence of this file typically depends on the Linux distribution in use, but both Debian and RedHat/Fedora use `/etc/sysctl.conf`:

```
# file: /etc/sysctl.conf
# These settings should be duplicated for all interfaces that are
# on a peering LAN.

### Typical stuff you really want on a router

# Fix the "promiscuous ARP" thing...
net/ipv4/conf/ifname/arp_ignore=1
net/ipv4/conf/ifname/arp_announce=1

# Turn off RP filtering to allow asymmetric routing:
net/ipv4/conf/ifname/rp_filter=0
```

```
# Multiple (non-aggregated) interfaces on the same peering LAN.
# READ THE MANUAL FIRST!
#net/ipv4/conf/ifname/arp_filter=1

### Keep the AMS-IX ARP Police happy. :-)

net/ipv4/neigh/ifname/base_reachable_time=14400
net/ipv6/neigh/ifname/base_reachable_time=14400
```

Modules must be loaded before `sysctl` is executed

On Debian systems, kernel modules for some network interfaces (e.g. 10GE cards) are not loaded before the **init** process executes the script that runs the **sysctl** commands. In those cases, it is necessary to force the module to be loaded earlier. The same goes for the IPv6 settings; the `ipv6` module is usually not loaded until the network interfaces are brought up, which is typically *after* the **sysctl** variables are set by the `procps.sh` script.

(On RedHat/Fedora systems no action needs to be taken; the `/etc/init.d/network` script automatically (re-)sets the **sysctl** variables before and after bringing up the interfaces.)

There are a few ways around this:

1. Re-run the **sysctl** directives after the interfaces are brought up (and the appropriate modules are loaded). This method is probably the only option available to you if your system does no autoloading of modules.

   On Debian-based systems, this can be done by creating a symbolic link in `/etc/rc2.d` to re-run `procps.sh` after the network is brought up:

   ```
   root@tuxco# ln -s ../init.d/procps.sh /etc/rc2.d/S20procps.sh
   ```

2. Pre-load the appropriate modules before the **sysctl** settings are applied.

   On Debian-based systems, the necessary modules can be pre-loaded by listing the appropriate modules in `/etc/modules`. The **module-init-tools** script (or **modutils** on older systems) will load the modules before the `sysctl.conf` entries are executed:

   ```
   # file: /etc/modules
   # load the kernel module for "mycard".
   mycard
   # load the ipv6 stack
   ipv6
   ```

   (As a curiosity, on RedHat/Fedora systems this would be accomplished by creating one or more executable scripts in `/etc/sysconfig/modules` with names ending in `.modules`. The scripts should be proper shell scripts executing the appropriate commands to load and initialise the modules).

3. Modify `/etc/modprobe.conf` (or the appropriate file in `/etc/modprobe.d`) and use the **`install`** directive to execute the relevant **sysctl** directives after loading the module. Although this is possible, we recommend against it, as it is far easier and clearer to use one of the alternative methods above.

9. Riverstone

On Riverstone equipment, proxy ARP seems to be enabled by default, so you will need to disable it:

```
ip disable proxy-arp interface ifname
```

Here, *ifname* refers to your interface towards AMS-IX, or the string "**all**"

10. Acknowledgements

Various AMS-IX members contributed to this document. We received configuration info from:

Vincent Bourgonjen (Open Peering)
Kevin Day (Your.org)
Thijs Eilander (Cobweb)
Ronald Esveld (Equant)
Santi Mercado (SARENET)
Niels Raijer (Demon)
Najam Saquib (Mediaways)
Jesper Skriver (TDC)
Miquel van Smoorenburg (Cistron)
Andree Toonk (SARA)
Blake Willis (Neo Telecoms)
Martijn Bakker (Support Net)
Lucas van Schouwen (Eweka)

Thanks to all those who contributed.