

Network Technical Specifications

Ports, IPv4 and IPv6

All ports on the BNIX public exchange infrastructure are Ethernet and are available at the following speeds:

- 1 Gbps (GE) - 1000BASE-LX
- 1 Gbps (GE) – 1000BASE-SX
- 1 Gbps (GE) – 1000BASE-T
- 10 Gbps (10GE) - 10GBASE-LR
- 10 Gbps (10GE) - 10GBASE-SR
- 100 Gbps (100GE) – 100GBASE-LR4

Additionally, BNIX is able to aggregate GE, 10GE or 100GE ports to provide higher bandwidth trunks. Also, using ratelimits you can connect physically to a certain line rate but only pay for the amount of bandwidth you are using.

Members can peer with IPv4, IPv6 or both. BNIX strongly recommends peering with both IPv4 and IPv6 address families.

Routeserver

peering details:

rs.bruzav.bnix.net 194.53172.1, 2001:7f8:26::a500:5406:1, AS5406

rs.brueve.bnix.net 194.53172.2, 2001:7f8:26::a500:5406:2, AS5406

It is encouraged to setup a peering with both servers in order to guarantee redundancy.

Routeserver is running on the routing daemon Birdv2 and uses IRRDB and RPKI filtering to determine the prefix list. It is advised to check if your route object are correctly configured. Community strings are permitted to decide to who you wish to announce your prefixes via the routeserver.

MAC Layer

Ethernet framing

The BNIX infrastructure is based on the Ethernet II (or “DIX Ethernet”) standard. This means that LLC/SNAP encapsulation (802.2) is not permitted.

Ethertypes

Frames forwarded to BNIX ports must have one of the following Ether types:

- 0x0800 - IPv4
- 0x0806 - ARP
- 0x86dd - IPv6

One MAC address per port

Frames forwarded to an individual BNIX port shall all have the same destination MAC address.

Exceptions are of those ports that belong to a reseller.

No proxy ARP

Use of proxy ARP on the router's interface to the Exchange is not allowed.

Unicast only

Frames forwarded to BNIX ports shall not be addressed to a multicast or broadcast MAC destination address except as follows:

- broadcast ARP packets
- multicast ICMPv6 Neighbour Discovery packets. This **DOES NOT** include Router Discovery packets.

No link-local traffic

Traffic for link-local protocols shall not be forwarded to BNIX ports. Link-local protocols include, but are not limited to, the following list:

- IRDP
- ICMP redirects
- IEEE 802 Spanning Tree

- Vendor proprietary protocols. These include, but are not limited to:
 - Discovery protocols: CDP, EDP - VLAN/trunking protocols: VTP, DTP
 - Interior routing protocol broadcasts (e.g. OSPF, ISIS, IGRP, EIGRP)
- BOOTP/DHCP
- PIM-SM
- PIM-DM
- DVMRP
- ICMPv6 ND-RA
- UDLD
- L2 Keepalives

The following link-local protocols are exceptions and are allowed:

- ARP
- IPv6 ND

BFD (Bidirectional Forwarding Detection)

BNIX recommends configuring BFD as a failure detection mechanism on all BGP peering sessions, be it peer-to-peer or with the route servers.

BFD is a generic keepalive and failure detection protocol that runs over almost any communication media. It allows sub-second failure detection, if participating systems are fast enough. It is designed as a lightweight protocol that can run autonomously in the forwarding engine of network devices, independent of the control plane.

In our case, BFD can be configured as an additional failure detection mechanism for BGP. Each BGP session will be doubled by a dedicated BFD session that runs on UDP ports 3784 and 3785, on the same IPv4 or IPv6 addresses as the BGP session itself. When BFD detects the failure of a neighbor, it informs the BGP process which triggers immediately the withdrawal of the neighbor's routes, shortcutting the overly generous BGP timeouts and enabling the immediate use of an alternate route.

On BNIX, we now encourage members to use BFD on their peer-to-peer peering sessions. This allows for extremely fast failover in case of data path outages for whatever reason. In addition, we will start offering BFD support on the route servers. Thus, BGP sessions with the route servers can also be protected with BFD.

The following are our current recommended timer values for BFD:

- interval 500ms
- Multiplier 10