# XORP Configuration Guide
# Part 2: Configuration
# Version 1.0

XORP Project
International Computer Science Institute
Berkeley, CA 94704, USA
*feedback@xorp.org*

July 8, 2004

## 1 Introduction

A XORP router must be configured to perform the desired operations. The configuration information can be provided in one of the two ways:

- Use a configuration file when the rtrmgr is started. By default, the rtrmgr will load the configuration from file "config.boot" in the XORP installation directory. This file can be specified by the "-b <filename>" command line option:

  ```
  xorp_rtrmgr -b my_config.boot
  ```

  See "rtrmgr/config.boot.sample" for an example of a configuration file (note that this file MUST be modified before using it).

- Use the xorpsh command line interface after the rtrmgr is started. It should be noted that command line completion in the xorpsh does greatly simplify configuration.

A mixture of both methods is permissible. For example, a configuration file can also be loaded from within the xorpsh.

At the very least, a router's interfaces must be configured (see Section 2). Typically, the FEA needs to be configured (*e.g.,* to enable unicast forwarding); the FEA configuration is described in Section 3. All protocol configuration is described in Section 4.

## 2 Network Interfaces

A XORP router will only use interfaces that it has been explicitly configured to use. Even for protocols such as BGP that are agnostic to interfaces, if the next-hop router for a routing entry is not through a configured interface the route will not be installed. For protocols that are explicitly aware of interfaces only configured interfaces will be used.

Every physical network device in the system is considered to be an "interface". Every interface can contain a number of virtual interfaces ("vif"s). In the majority of cases the interface name and vif name will be identical and will map to the name given to the interface by the operating system. A virtual interface is configured with the address or addresses that should be used. At each level in the configuration hierarchy (`interface`, `vif` and `address`) it is necessary to enable this part of the configuration.

```
interfaces {
  interface dc0 {
    description: "data interface"
    enabled: true
    /* default-system-config */
    vif dc0 {
      enabled: true
      address 10.10.10.10 {
        prefix-length: 24
        broadcast: 10.10.10.255
        enabled: true
      }
      /*
      address 10:10:10:10:10:10:10:10 {
        prefix-length: 64
        enabled: true
      }
      */
    }
  }
}
```

We recommend that you select the interfaces that you want to use on your system and configure them as above. If you are configuring an interface that is currently being used by the the system make sure that there is no mismatch in the `address`, `prefix-length` and `broadcast` arguments. If the `default-system-config` statement is used, it instructs the FEA that the interface should be configured by using the existing interface information from the underlying system. In which case, the `vif` and `address` sections must not be configured.

# 3 Forwarding Engine Abstraction

It is a requirement to explicitly enable forwarding for each protocol family.

```
fea {
  enable-unicast-forwarding4: true
  /* enable-unicast-forwarding6: true */
}
```

If IPv4 forwarding is required you will require the configuration above. If the system supports IPv6 and IPv6 forwarding is required, then `enable-unicast-forwarding6` must be set to `true`.

# 4 Protocols

An unicast router typically needs to be configured with one or several of the following protocols: StaticRoutes (Section 4.1), RIP (Section 4.2) or BGP (Section 4.3).

A multicast router must have the MFEA configured (Section 4.4). Typically, a multicast router should have IGMP/MLD configured (Section 4.5). Currently, PIM-SM is the only multicast routing protocol implemented (Section 4.6). If some multicast-specific static routes need to be installed in the MRIB (for computing the reverse-path forwarding information), those can be specified in the StaticRoutes configuration (Section 4.1). Eventually (*e.g.,* if there are no unicast routing protocols configured), the FIB2MRIB module may need to be configured too (Section 4.7).

### 4.1 Static Routes

This is the simplest routing protocol in XORP. It allows the installation of unicast or multicast static routes (either IPv4 or IPv6). Note that in case of multicast the routes are installed only in the user-level Multicast Routing Information Base and are used for multicast-specific reverse-path forwarding information by multicast routing protocols such as PIM-SM.

```
protocols {
  static {
    route4 10.20.0.0/16 {
      nexthop: 10.10.10.20
      metric: 1
    }
    mrib-route4 10.20.0.0/16 {
      nexthop: 10.10.10.30
      metric: 1
    }
    /*
    route6 20:20:20:20::/64 {
      nexthop: 10:10:10:10:10:10:10:20
      metric: 1
    }
    mrib-route6 20:20:20:20::/64 {
      nexthop: 10:10:10:10:10:10:10:30
      metric: 1
    }
    */
  }
}
```

Static routes can be redistributed through the RIP protocol using RIP's `export` statement.

### 4.2 Routing Information Protocol

In order to run RIP it is sufficient to specify the set of interfaces, vifs and addresses (`interface`, `vif` and `address`) on which RIP is enabled. Remember that each `address` must be explicitly enabled.

If you wish to announce routes then it is necessary to `export` the routes that are to be announced. For example, `connected`, `static` and `rip`.

Note: In release candidate 1.0-RC, the `import` keyword is used in place of `export`.

```
protocols {
  rip {
  /* Redistribute routes for connected interfaces */
    /*
    export connected {
      metric: 0
      tag: 0
    }
    */
  /* Redistribute static routes */
    /*
    export static {
      metric: 1
      tag: 0
    }
    */
  /* Run on specified network interface addresses */
  /*
    interface dc0 {
      vif dc0 {
        address 10.10.10.10 {
          enabled: true
        }
      }
    }
  */
  }
}
```

### 4.3 Border Gateway Protocol

In order to run BGP the `bgp-id` (BGP Identifier) and `local-as` (Autonomous System number) must be specified.

The `peer` statement specifies a peering. The argument to the peer statement is the IP address of the peer. The `local-ip` is the IP address that TCP should use. The `as` is the Autonomous System Number of the peer.

```
protocols {
  bgp {
    bgp-id: 10.10.10.10
    local-as: 65002

    peer 10.30.30.30 {
      local-ip: 10.10.10.10
      as: 65000
      next-hop: 10.10.10.20
      /*
      local-port: 179
      peer-port: 179
      */
      /* holdtime: 120 */
      /* enabled: true */

      /* Optionally enable other AFI/SAFI combinations */
      /* enable-ipv4-multicast */

      /* enable-ipv6-unicast */
      /* enable-ipv6-multicast */
    }

    /* Originate IPv4 Routes */
    /*
    network4 10.10.10.0/24 {
      next-hop: 10.10.10.10
      unicast: true
      multicast: true
    }
    */

    /* Originate IPv6 Routes */
    /*
    network6 10:10:10:10::/64 {
      next-hop: 10:10:10:10:10:10:10:10
      unicast: true
      multicast: true
    }
    */
  }
}
```

Currently BGP is not able to `import` routes from other routing protocols such as `static`. It is however possible to originate routes using `network4` and `network6` statements such as in the above example.

### 4.4 Multicast Forwarding Engine Abstraction

The MFEA must be configured if the XORP router is to be used for multicast routing. The MFEA for IPv4 and IPv6 are configured separately.

In the configuration we must explicitly enable the entity itself, and each `vif`. The `traceoptions` section is used to explicitly enable log information that can be used for debugging purpose.

```
plumbing {
  mfea4 {
    enabled: true
    interface dc0 {
      vif dc0 {
        enabled: true
      }
    }
    interface register_vif {
      vif register_vif {
        /* Note: this vif should be always enabled */
        enabled: true
      }
    }
    traceoptions {
      flag all {
        enabled: true
      }
    }
  }
}

plumbing {
  mfea6 {
    enabled: true
    interface dc0 {
      vif dc0 {
        enabled: true
      }
    }
    interface register_vif {
      vif register_vif {
        /* Note: this vif should be always enabled */
        enabled: true
      }
    }
    traceoptions {
      flag all {
        enabled: true
      }
    }
  }
}
```

Note that the interface/vif named `register_vif` is special. If PIM-SM is configured, then `register_vif` must be enabled in the MFEA.

### 4.5 Internet Group Management Protocol/Multicast Listener Discovery

IGMP/MLD should be configured if the XORP router is to be used for multicast routing and if we want to track multicast group membership for directly connected subnets. Typically this is the case for a multicast router, therefore it should be enabled. IGMP and MLD are configured separately: IGMP is used for tracking IPv4 multicast members; MLD is used for tracking IPv6 multicast members.

In the configuration we must explicitly enable each entity and each `vif`. The `traceoptions` section is used to explicitly enable log information that can be used for debugging purpose.

```
protocols {
  igmp {
    enabled: true
    interface dc0 {
      vif dc0 {
        enabled: true
      }
    }
    traceoptions {
      flag all {
        enabled: true
      }
    }
  }
}

protocols {
  mld {
    enabled: true
    interface dc0 {
      vif dc0 {
        enabled: true
      }
    }
    traceoptions {
      flag all {
        enabled: true
      }
    }
  }
}
```

### 4.6 Protocol Independent Multicast - Sparse Mode

PIM-SM should be configured if the XORP router is to be used for multicast routing in PIM-SM domain. PIM-SM for IPv4 and IPv6 are configured separately. At minimum, the entity itself and the virtual interfaces should be enabled, and the mechanism for obtaining the Candidate-RP set (either the Bootstrap mechanism, or a static-RP set).

```
protocols {
  pimsm4 {
    enabled: true
    interface dc0 {
      vif dc0 {
        enabled: true
        /* dr-priority: 1 */
        /* alternative-subnet 10.40.0.0/16 */
      }
    }
    interface register_vif {
      vif register_vif {
        /* Note: this vif should be always enabled */
        enabled: true
      }
    }

    static-rps {
      rp 10.60.0.1 {
        group-prefix 224.0.0.0/4 {
          /* rp-priority: 192 */
          /* hash-mask-len: 30 */
        }
      }
    }

    bootstrap {
      enabled: true
      cand-bsr {
        scope-zone 224.0.0.0/4 {
          /* is-scope-zone: false */
          cand-bsr-by-vif-name: "dc0"
          /* bsr-priority: 1 */
          /* hash-mask-len: 30 */
        }
      }

      cand-rp {
        group-prefix 224.0.0.0/4 {
          /* is-scope-zone: false */
          cand-rp-by-vif-name: "dc0"
          /* rp-priority: 192 */
          /* rp-holdtime: 150 */
        }
      }
    }

    switch-to-spt-threshold {
      /* approx. 1K bytes/s (10Kbps) threshold */
      enabled: true
      interval-sec: 100
      bytes: 102400
    }

    traceoptions {
      flag all {
        enabled: true
      }
    }
  }
}
```

```
protocols {
  pimsm6 {
    enabled: true
    interface dc0 {
      vif dc0 {
        enabled: true
        /* dr-priority: 1 */
        /* alternative-subnet 40:40:40:40::/64 */
      }
    }
    interface register_vif {
      vif register_vif {
        /* Note: this vif should be always enabled */
        enabled: true
      }
    }

    static-rps {
      rp 50:50:50:50:50:50:50:50 {
        group-prefix ff00::/8 {
          /* rp-priority: 192 */
          /* hash-mask-len: 126 */
        }
      }
    }

    bootstrap {
      enabled: true
      cand-bsr {
        scope-zone ff00::/8 {
          /* is-scope-zone: false */
          cand-bsr-by-vif-name: "dc0"
          /* bsr-priority: 1 */
          /* hash-mask-len: 30 */
        }
      }

      cand-rp {
        group-prefix ff00::/8 {
          /* is-scope-zone: false */
          cand-rp-by-vif-name: "dc0"
          /* rp-priority: 192 */
          /* rp-holdtime: 150 */
        }
      }
    }

    switch-to-spt-threshold {
      /* approx. 1K bytes/s (10Kbps) threshold */
      enabled: true
      interval-sec: 100
      bytes: 102400
    }

    traceoptions {
      flag all {
        enabled: true
      }
    }
  }
}
```

A number of parameters have default values, therefore they don't have to be configured (those parameters are commented-out in the above sample configuration).

Note that the interface/vif named `register_vif` is special If PIM-SM is configured, then `register_vif` must be enabled.

The dr-priority parameter is used to configure the Designated Router priority per virtual interface (note that in case of register_vif it is not used).

The alternative-subnet statement is used to add alternative subnets to a network interface. For example, if you want to make incoming traffic with a non-local source address appear as it is coming from a local subnet, then alternative-subnet can be used. Typically, this is needed as a work-around solution when we use uni-directional interfaces for receiving traffic (e.g., satellite links). Note: use alternative-subnet with extreme care, only if you know what you are really doing!

If PIM-SM uses static RPs, those can be configured within the static-rps section. For each RP, an rp section is needed, and each section should contain the multicast prefix address the static RP is configured with. The RP priority can be modified with the rp-priority parameter.

If PIM-SM uses the Bootstrap mechanism to obtain the Candidate-RP set, that can be configured in the bootstrap section. If the XORP router is to be used as a Candidate-BSR, this should be specified in the cand-bsr section. For a router to be a Candidate-BSR it must advertise for each zone (scoped or non-scoped) the associated multicast prefix address. The cand-bsr section should contain scope-zone statements for each multicast prefix address. The vif name with the address that is to be used as the Candidate-BSR is specified by the cand-bsr-by-vif-name statement. The Candidate-BSR priority can be modified with the bsr-priority parameter.

If the XORP router is to be a Candidate-RP, this should be specified in the cand-rp section. For a router to be a Candidate-RP it must advertise for each zone (scoped or non-scoped) the associated multicast prefix address. The cand-rp section should contain group-prefix statements for each multicast prefix address. The vif name with the address that is to be used as the Candidate-RP is specified by the cand-rp-by-vif-name statement. The Candidate-RP priority can be modified with the rp-priority parameter; the Candidate-RP holdtime can be modified with the rp-holdtime parameter.

The is-scope-zone parameter is used to specify whether a Candidate-BSR scope-zone or a Candidate-RP group-prefix is scoped. Currently, scoped zones are not well tested, hence it is recommended scope-zone is always set to false. Note that typically the hash-mask-len should not be modified; if you don't know what hash-mask-len is used for, don't modify it!

The switch-to-spt-threshold section can be used to specify the multicast data bandwidth threshold used by the last-hop PIM-SM routers and the RPs to initiate shortest-path switch toward the multicast source. Parameter interval-sec is used to specify the periodic measurement interval; parameter bytes is used to specify the threshold in number of bytes within the measurement interval. It is recommended that the measurement interval is not too small, and should be on the order of tens of seconds.

The traceoptions section is used to explicitly enable log information that can be used for debugging purpose.

### 4.7 FIB2MRIB

The FIB2MRIB module is used to obtain the Forwarding Information Base information from the underlying system (via the FEA), and to propagate it to the MRIB, so it can be used by multicast routing protocols such as PIM-SM. Typically, it is needed only if the unicast routing protocols (if any) on that router do not inject routes into the MRIB. Note that FIB2MRIB is disabled by default, therefore if it is needed it must be explicitly enabled.

```
protocols {
  fib2mrib {
    enabled: true
  }
}
```