XORP MLD/IGMP Daemon

Version 1.2

XORP Project
International Computer Science Institute
Berkeley, CA 94704, USA
http://www.xorp.org/
feedback@xorp.org

March 8, 2006

1 Introduction

1.1 Overview

This document provides an overview of the XORP MLD/IGMP [2, 3] Routing Daemon. It is intended to provide a starting point for software developers who wish to modify this software.

A router running MLD/IGMP interacts with other MLD/IGMP routers and multicast group members, and keeps state with information about local multicast membership.

The chosen architecture for the XORP MLD/IGMP implementation emphasizes on correctness and extensibility rather than high performance or minimal memory footprint. Typically, the amount of state kept by MLD/IGMP is relatively small. Further, the multicast routing performance does not depend on the MLD/IGMP performance, therefore it is not necessary to optimize the implementation. Only if it turns out that there are performance issues with MLD/IGMP, we would try to optimize those parts of the implementation that should prove to be a bottleneck.

Currently (March 2006), the MLD and IGMP implementation is based on the specification in the following documents:

- RFC 2710: Multicast Listener Discovery for IPv6, Version 1
- RFC 2236: Internet Group Management Protocol, Version 2

In the future there will be support for MLDv2 and IGMPv3.

1.2 Acronyms

Acronyms used in this document:

- **MFEA**: **M**ulticast **F**orwarding **E**ngine **A**bstraction
- **MLD/IGMP**: **M**ulticast **L**istener **D**iscovery/**I**nternet **G**roup **M**anagement **P**rotocol
- **PIM-SM**: **P**rotocol **I**ndependent **M**ulticast–**S**parse **M**ode

1.3 MLD/IGMP Design Architecture Overview

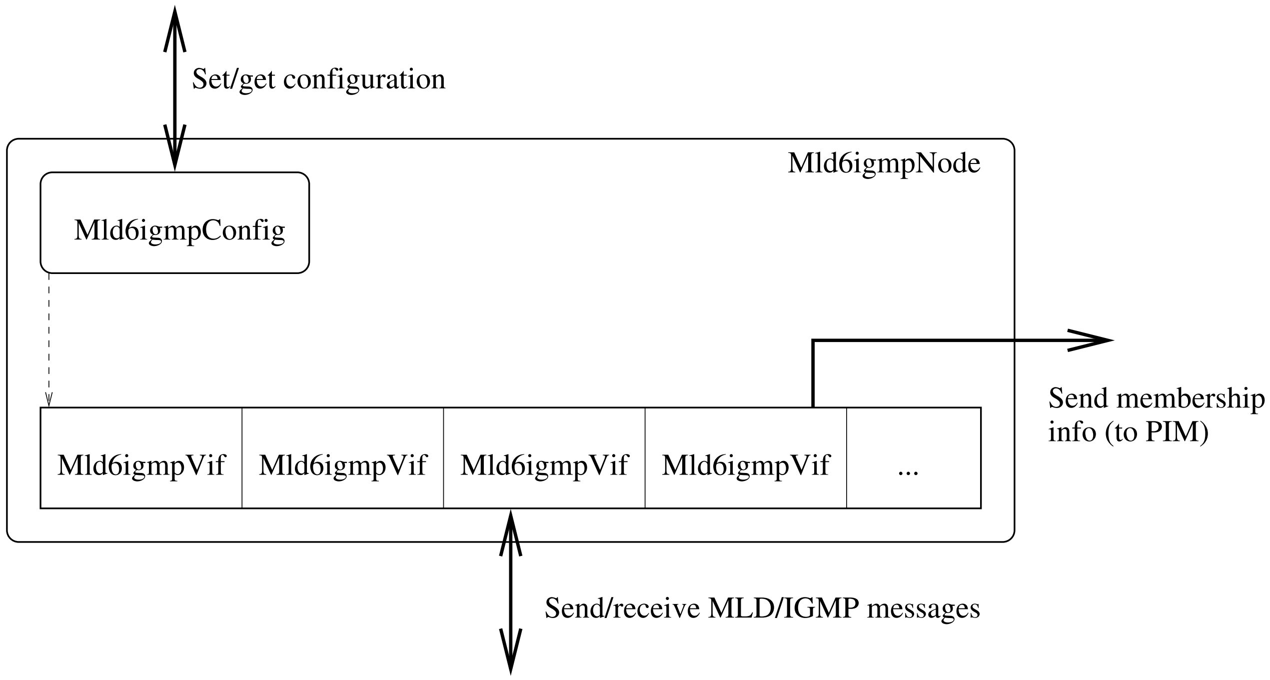

Figure 1: MLD/IGMP design overview

Figure 1 provides a general overview of the MLD/IGMP components. For each component there is a C++ class with exactly the same name. The main components are briefly described below:

- **Mld6igmpNode:** a representation of a single MLD/IGMP routing unit (*e.g.,* a virtual MLD/IGMP router). Typically, there would be a single Mld6igmpNode per multicast router.
- **Mld6igmpVif:** MLD/IGMP-specific virtual (network) interface that is used for sending and receiving MLD/IGMP control packets.
- **Mld6igmpConfig:** contains MLD/IGMP-specific configuration.

Those components are described in details in Section 2. For information about the interaction between the MLD/IGMP and other modules see [1].

2 Components Description

2.1 Mld6igmpNode Description

Mld6igmpNode is a representation of a single MLD/IGMP routing unit (*e.g.,* a virtual MLD/IGMP router). Typically, there would be a single Mld6igmpNode per multicast router. However, in some cases a multicast router may have more than one routing unit. For example, it could have one Mld6igmpNode for IPv4, and another one for IPv6 multicast routing. Further, if we want to run MLD/IGMP in a simulation environment, each multicast router within that simulation will be represented by a single Mld6igmpNode.

From a developer's point of view, Mld6igmpNode contains all the state related to the MLD/IGMP routing unit, and exports the front-end interface to interact with that unit. For example, Mld6igmpNode contains the methods to start/stop or configure MLD/IGMP, or to send/receive MLD/IGMP control packets to/from the routing unit. Those methods are described in the following files:

- `mld6igmp/mld6igmp_node.hh`
- `libproto/proto_node.hh`
- `libproto/proto_unit.hh`

Mld6igmpNode itself does not implement the mechanisms to communicate with other routing units (*e.g.,* to send or receive control packets to/from the network), or to perform other MLD/IGMP-independent operations such as joining a multicast routing group. Those mechanisms are outside the scope of Mld6igmpNode, and must be implemented separately.

Mld6igmpNode contains several pure virtual methods (*e.g.,* `join_multicast_group()` is used to join a multicast group on an interface) that must be implemented by a class that inherits Mld6igmpNode. For example, XrlMld6igmpNode is a class that uses Mld6igmpNode as a base class; XrlMld6igmpNode uses XRL-based communication mechanisms between Mld6igmpNode and other XORP components such as the MFEA and PIM modules.

By default, Mld6igmpNode is disabled; therefore, on startup it must be enabled explicitly.

2.2 Mld6igmpVif Description

Mld6igmpVif is a MLD/IGMP-specific virtual (network) interface that is used for sending and receiving MLD/IGMP control packets. It includes the methods for processing and composing MLD/IGMP control messages, as well as various state per interface (*e.g.,* the multicast group membership on an interface). It contains various state per interface ((*e.g.,* the multicast group membership on an interface), as well as the methods for processing and composing MLD/IGMP control messages.

Typically, there would be one Mld6igmpVif per network interface such as physical interface, tunnel, or the loopback interface. Not all virtual interfaces are used by MLD/IGMP; for example, all interfaces that are not multicast capable, and the loopback interface are ignored for multicast routing.

Typically, from developer's point of view, all interaction with Mld6igmpNode would be through Mld6igmpNode [1].

The public interface for Mld6igmpVif contains the methods to manipulate a virtual (network) interface. Those methods are to start/stop/enable/disable a virtual interface, and to configure it. The methods are described in the following files:

- `mld6igmp/mld6igmp_vif.hh`
- `libxorp/vif.hh`
- `libproto/proto_unit.hh`

[1] For simplicity, currently (March 2006) there are few occasions when XrlMld6igmpNode uses direct access to Mld6igmpVif.

Mld6igmpVif contains state such as group membership information about local members (a list of objects of class MemberQuery), and information about the current MLD/IGMP querier. Also, all the MLD/IGMP-specific methods for parsing or constructing MLD/IGMP control messages when a MLD/IGMP packet is received or sent are implemented as methods in Mld6igmpVif. The parsing or construction of each message type is implemented in files `mld6_proto.cc` and `igmp_proto.cc` for MLD and IGMP respectively.

By default, each Mld6igmpVif is disabled; therefore, on startup it must be enabled explicitly.

2.3 Mld6igmpConfig Description

Mld6igmpConfig handles the MLD/IGMP-specific configuration [2]. This configuration is used to configure the following units:

- Mld6igmpVif: protocol version, membership query related options and timer values, etc.

A Modification History

- December 11, 2002: Version 0.1 completed.
- March 10, 2003: Updated to match XORP version 0.2 release code; cleanup.
- June 9, 2003: Bump-up the version to 0.3, and the date.
- August 28, 2003: Bump-up the version to 0.4, and the date.
- November 6, 2003: Bump-up the version to 0.5, and the date.
- July 8, 2004: Bump-up the version to 1.0, and the date.
- April 13, 2005: Bump-up the version to 1.1, and the date.
- March 8, 2006: Bump-up the version to 1.2, and the date.

References

[1] XORP Multicast Routing Design Architecture. XORP technical document. http://www.xorp.org/.

[2] Steve Deering, Bill Fenner, and Brian Haberman. Multicast Listener Discovery (MLD) for IPv6. Request For Comments (RFC) 2710, October 1999. http://www.ietf.org/rfc.html.

[3] W. Fenner. Internet Group Management Protocol, Version 2. Request For Comments (RFC) 2236, November 1997. http://www.ietf.org/rfc.html.

[2] Currently (March 2006), Mld6igmpConfig is not implemented; rather, all state is kept inside Mld6igmpNode instead.