IP Encapsulator 3000 ™

Multicast Status API

Version 1.3

LOGIC INNOVATIONS

Copyrights and Notices

Published by Logic Innovations, Inc.
9980 Carroll Canyon Road, Suite 100
San Diego, California USA 92131

Support
Phone: 858-805-6300
fax: 858-805-6340
e-mail: info@logici.com
website: www.logici.com

Table of Contents

1. Revision History

Revision Number	Description	Author	Date
1.0	Initial release.	Anthony Massa	12/18/2000
1.1	Only IP Address and Mask are in network byte order	Leon Havin	02/01/2001
1.2	Minor cosmetic changes	Theo Aukerman	01/03/2006
1.3	Changed legal text to allow publication. Added some implementation notes.	Theo Aukerman	2/26/2007

2. Introduction

This document details the Application Programming Interface (API) structure for the status multicast packets transmitted by the Logic Innovations IP Encapsulator Data Gateway (IPEDG). The status datagram packet is sent out of the IPEDG in a User Datagram Protocol (UDP) multicast packet. The header information in this packet is configurable via the user interface on the IPEDG. Figure 1.1 shows the overall packet structure sent on the network.

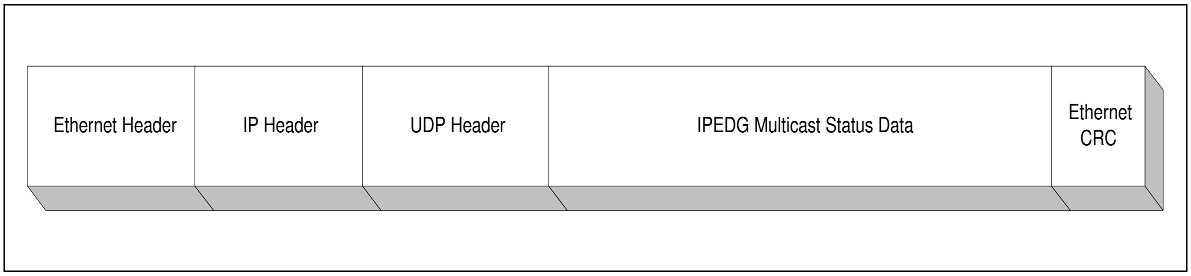

Figure 1.1 - Overall Multicast Packet Structure.

3. Multicast Configuration

The header information in the UDP multicast packet is configured from the user interface on the IPEDG. The fields that are configurable are:

- IP Address – Multicast address that the UDP packets are transmitted.
- Port – UDP port to transmit the data from.
- TTL – Time-to-live field for the UDP packets.
- Transmit Interval – Interval (in milliseconds) that the UDP packets are sent out.

4. Multicast Datagram Structure

The multicast datagram has a limit dictated by UDP, which is 65535 bytes. The amount of data sent in each packet is dependent upon the number of routes currently active on the IPEDG. The syntax of the multicast datagram is defined in Table 3.1.

Syntax	Number of Bytes	Comments
`status_datagram() {`		
`    total_routes`	4	
`    status_length`	4	
`    for (i=0; i < total_routes; i++) {`		
`        ip_address`	4	Network byte order (MSB first)
`        ip_mask`	4	Network byte order (MSB first)
`        buffer_length`	4	
`        buffer_level`	4	
`    }`		
`}`		

`total_routes` : this is a 4-byte field set to the number of active routes currently running.

`status_length` : this is a 4-byte field set to the total number of bytes following this field.

`ip_address` : this is a 4-byte field set to the IP address of the active route.

`ip_mask` : this is a 4-byte field set to the mask of the active route.

`buffer_length` : this is a 4-byte field set to the total buffer length, in bytes, available for the active route.

`buffer_level` : this is a 4-byte field set to the current buffer level, in bytes, of the active route.

5. Implementation Notes

- Since the IPE uses an entire packet buffer for each IP datagram processed, transmitting x bytes of data to the IPE is likely to cause the buffer level to be reduced by a value that is greater than x. For this reason, a simplistic implementation that transmitted “buffer_length – buffer_level” bytes of data to the IPE will not always function properly. Selecting or allowing configuration of a target buffer fullness percentage is one means of dealing with this.
- The configured transmit interval will have an impact on the theoretical maximum bit rate as follows: RMax = buffer_length * 1000/TxInt. Where RMax is the theoretical max byte rate (Bytes per second) and TxInt is the transmit interval of the UDP status datagrams in milliseconds. This formula produces a theoretical max only, and specifically does not take into consideration the note above regarding buffer consumption.