Network Working Group
Request for Comments: 1132

L. McLaughlin III
The Wollongong Group
November 1989

A Standard for the Transmission of 802.2 Packets over IPX Networks

Status of this Memo

This document specifies a standard method of encapsulating 802.2 [1] packets on networks supporting Novell's Internet Packet Exchange Protocol [2] (IPX). It obsoletes earlier documents detailing the transmission of Internet packets over IPX networks. It differs from these earlier documents in that it allows for the transmission of multiple network protocols over IPX and for the transmission of packets through IPX bridges. Distribution of this memo is unlimited.

Introduction

The goal of this specification is to allow compatible and interoperable implementations for transmitting Internet packets such as the Internet Protocol [3] (IP) and Address Resolution Protocol [4] (ARP) as well as the Connectionless-mode Network Protocol [5] (CLNP) over IPX networks.

IPX is a proprietary standard developed by Novell derived from Xerox's Internet Datagram Protocol [6] (IDP). Defining the encapsulation of the IEEE 802.2 Data Link Layer Standard over IPX in terms of yet another 802.X Physical Layer standard allows for the transmission of IP Datagrams as described in RFC 1042 [7]. This document will focus on the implementation of that RFC over IPX networks.

Description

In general, this specification allows IPX networks to be used to support any network protocol which can use the IEEE 802.2 Data Link Layer specification.

More specifically, IPX networks may be used to support IP networks and subnetworks of any class. By encapsulating IP datagrams within IPX datagrams and assigning IP numbers to the hosts on a IPX network, IP-based applications are supported on these hosts. The addition of an IP Gateway capable of encapsulating IP packets within 802.IPX datagrams would allow those hosts on an IPX network to communicate with the Internet.

Maximum Transmission Unit

The maximum data size of a IPX datagram is 546 bytes. As the combined size of the 802.2 LLC and SNAP headers is 8 bytes, this results in a Maximum Transmission Unit (MTU) of 538 bytes.

Address Mappings

The mapping of Internet Protocol addresses to 802.IPX addresses is done using the Address Resolution Protocol in the same fashion as with other IEEE 802.X physical addresses. However, the length of an 802.IPX physical address is 10 bytes rather than 2 or 6. This 10 byte physical address consists of the 4 bytes of the IPX network address followed by the 6 bytes of the IPX node address.

Byte Order

The byte transmission order is "big-endian" [8].

Broadcast Addresses

IPX packets may be broadcast by setting the IPX header Packet Type field to 0x14, the Destination Network field to the local network number, the the Destination Node field to 0xffffff, and the Immediate Address field of the IPX Event Control Block to 0xffffff.

Unicast Addresses

IPX packets may be unicast by setting the IPX header Packet Type field to 0x04, the Destination Network field and Destination Node field to those values found by address resolution, and the Immediate Address field of the IPX Event Control Block to the physical address of the destination node or the appropriate IPX bridge.

Checksum

Like most IPX applications, this specification does not use IPX checksum.

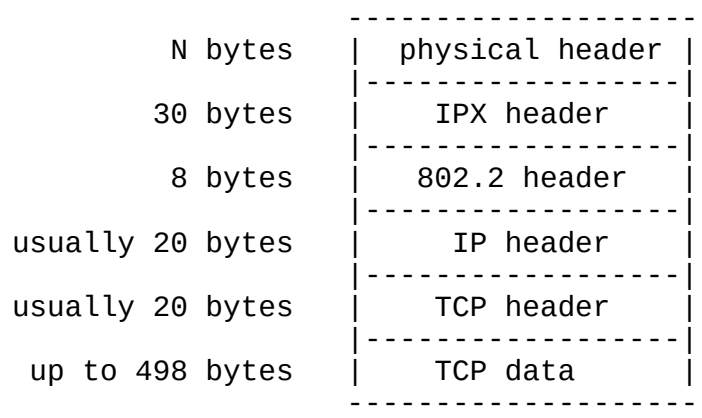
Reserved values

The IPX socket 0x8060 has been reserved by Novell for the implementation of this protocol.

Implementation

The encapsulation of Internet packets within IPX networks has proved to be quite useful. Because the IPX interface insulates knowledge of

the physical layer from an application, 802.2 over IPX networks work over any physical medium. A typical IP over IPX packet is shown below:



On workstations supporting an IPX programming interface, implementation of this specification has proved fairly straightforward. The only change which was done was to modify the existing address resolution protocol code to allow for cache entries larger than the hardware address length. This was done to allow room for the immediate address of a possible intervening IPX bridge in addition to the destination node and network addresses to be associated with a given IP address.

Thus far, no implementations have been attempted on systems which do not already support an IPX programming interface (e.g., a dedicated router) though a few implementation details can be noted. First, obviously any such implementation will have to distinguish IPX packets from other packets; this process will be media dependent. Second, note that no unicast packet is ever sent from host1 to host2 without a prior broadcast packet from host2 to host1. Thus, the immediate address of a possible intervening IPX bridge between host1 and host2 can be learned from the physical header of that prior broadcast packet. Third, any such implementation will need to discover the local IPX network number from a Novell bridge or file server. The mechanisms for doing this exist but documentation for their use is not commonly available.

References

[1] IEEE, "IEEE Standards for Local Area Networks: Logical Link Control", IEEE, New York, 1985.

[2] Novell, Inc., "Advanced NetWare V2.1 Internetwork Packet Exchange Protocol (IPX) with Asynchronous Event Scheduler (AES)", October

1986.

[3] Postel, J., "Internet Protocol", RFC-791, USC/Information Sciences Institute, September 1981.

[4] Plummer, D., "An Ethernet Address Resolution Protocol", RFC-826, November 1982.

[5] ISO DIS 8473: "Information Processing Systems - Data Communications - Protocol for Providing the Connectionless-mode Network Service".

[6] Xerox Corporation, "Xerox Network Systems Architecture", XNSG 068504, April 1985.

[7] Postel, J., and J. Reynolds, "A Standard for the Transmission of IP Datagrams over IEEE 802 Networks", RFC-1042, USC/Information Sciences Institute, February 1988.

[8] Cohen, D., "On Holy Wars and a Plea for Peace", Computer, IEEE, October 1981.

Security Considerations

Security issues are not addressed in this memo.

Author's Address:

Leo J. McLaughlin III
The Wollongong Group
1129 San Antonio Road
Palo Alto, CA 94303

Phone: (415) 962-7100

EMail: ljm@TWG.COM