Network Working Group
Request for Comments: 1234

D. Provan
Novell, Inc.
June 1991

Tunneling IPX Traffic through IP Networks

Status of this Memo

This memo describes a method of encapsulating IPX datagrams within UDP packets so that IPX traffic can travel across an IP internet. This RFC specifies an IAB standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "IAB Official Protocol Standards" for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Introduction

Internet Packet eXchange protocol (IPX) is the internetwork protocol used by Novell's NetWare protocol suite. For the purposes of this paper, IPX is functionally equivalent to the Internet Datagram Protocol (IDP) from the Xerox Network Systems (XNS) protocol suite [1]. This memo describes a method of encapsulating IPX datagrams within UDP packets [2] so that IPX traffic can travel across an IP internet [3].

This RFC allows an IPX implementation to view an IP internet as a single IPX network. An implementation of this memo will encapsulate IPX datagrams in UDP packets in the same way any hardware implementation might encapsulate IPX datagrams in that hardware's frames. IPX networks can be connected thusly across internets that carry only IP traffic.

Packet Format

Each IPX datagram is carried in the data portion of a UDP packet. All IP and UDP fields are set normally. Both the source and the destination ports in the UDP packet should be set to the UDP port value allocated by the Internet Assigned Numbers Authority for the implementation of this encapsulation method.

As with any UDP application, the transmitting party has the option of avoiding the overhead of the checksum by setting the UDP checksum to zero. Since IPX implementations never use the IPX checksum to guard IPX packets from damage, UDP checksumming is highly recommended for IPX encapsulation.

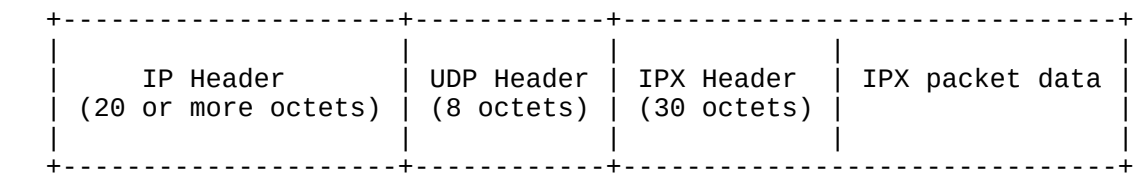

Figure 1: An IPX packet carried as data in a UDP packet.

Reserved Packets

The first two octets of the IPX header contain the IPX checksum. IPX packets are never sent with a checksum, so every IPX header begins with two octets of FF hex. Implementations of this encapsulation scheme should ignore packets with any other value in the first two octets immediately following the UDP header. Other values are reserved for possible future enhancements to this encapsulation protocol.

Unicast Address Mappings

IPX addresses consist of a four octet network number and a six octet host number. IPX uses the network number to route each packet through the IPX internet to the destination network. Once the packet arrives at the destination network, IPX uses the six octet host number as the hardware address on that network.

Host numbers are also exchanged in the IPX headers of packets of IPX's Routing Information Protocol (RIP). This supplies end nodes and routers alike with the hardware address information required for forwarding packets across intermediate networks on the way towards the destination networks.

For implementations of this memo, the first two octets of the host number will always be zero and the last four octets will be the node's four octet IP address. This makes address mapping trivial for unicast transmissions: the first two octets of the host number are discarded, leaving the normal four octet IP address. The encapsulation code should use this IP address as the destination address of the UDP/IP tunnel packet.

Broadcasts between Peer Servers

IPX requires broadcast facilities so that NetWare servers and IPX routers sharing a network can find one another. Since internet-wide IP broadcast is neither appropriate nor available, some other mechanism is required. For this memo, each server and router should maintain a list of the IP addresses of the other IPX servers and

routers on the IP internet. I will refer to this list as the "peer list", to individual members as "peers", and to all the peers taken together, including the local node, as the "peer group". When IPX requests a broadcast, the encapsulation implementation simulates the broadcast by transmitting a separate unicast packet to each peer in the peer list.

Because each peer list is constructed by hand, several groups of peers can share the same IP internet without knowing about one another. This differs from a normal IPX network in which all peers would find each other automatically by using the hardware's broadcast facility.

The list of peers at each node should contain all other peers in the peer group. In most cases, connectivity will suffer if broadcasts from one peer consistently fail to reach some other peer in the group.

The peer list could be implemented using IP multicast [4], but since multicast facilities are not widely available at this time, no well-known multicast address has been assigned and no implementations using multicast exist. As IP multicast is deployed in IP implementations, it can be used by simply including in the peer list an IP multicast address for IPX servers and routers. The IP multicast address would replace the IP addresses of all peers which will receive IP multicast packets sent from this peer.

Broadcasts by Clients

Typically, NetWare client nodes do not need to receive broadcasts, so normally NetWare client nodes on the IP internet would not need to be included in the peer lists at the servers.

On the other hand, clients on an IPX network need to send broadcasts in order to locate servers and to discover routes. A client implementation of UDP encapsulation can handle this by having a configured list of the IP addresses of all servers and routers in the peer group running on the IP internetwork. As with the peer list on a server, the client implementation would simulate the broadcast by sending a copy of the packet to each IP address in its list of IPX servers and routers. One of the IP addresses in the list, perhaps the only one, could be a broadcast address or, when available, a multicast address. This allows the client to communicate with members of the peer group without knowing their specific IP addresses.

It's important to realize that broadcast packets sent from an IPX client must be able to reach all servers and routers in the server

peer group. Unlike IP, which has a unicast redirect mechanism, IPX end systems are responsible for discovering routing information by broadcasting a packet requesting a router that can forward packets to the desired destination. If such packets do not tend to reach the entire server peer group, resources in the IPX internet may be visible to an end system, yet unreachable by it.

Maximum Transmission Unit

Although larger IPX packets are possible, the standard maximum transmission unit for IPX is 576 octets. Consequently, 576 octets is the recommended default maximum transmission unit for IPX packets being sent with this encapsulation technique. With the eight octet UDP header and the 20 octet IP header, the resulting IP packets will be 604 octets long. Note that this is larger than the 576 octet maximum size IP implementations are required to accept [3]. Any IP implementation supporting this encapsulation technique must be capable of receiving 604 octet IP packets.

As improvements in protocols and hardware allow for larger, unfragmented IP transmission units, the 576 octet maximum IPX packet size may become a liability. For this reason, it is recommended that the IPX maximum transmission unit size be configurable in implementations of this memo.

Security Issues

Using a wide-area, general purpose network such as an IP internet in a position normally occupied by physical cabling introduces some security problems not normally encountered in IPX internetworks. Normal media are typically protected physically from outside access; IP internets typically invite outside access.

The general effect is that the security of the entire IPX internetwork is only as good as the security of the entire IP internet through which it tunnels. The following broad classes of attacks are possible:

1) Unauthorized IPX clients can gain access to resources through normal access control attacks such as password cracking.

2) Unauthorized IPX gateways can divert IPX traffic to unintended routes.

3) Unauthorized agents can monitor and manipulate IPX traffic flowing over physical media used by the IP internet and under control of the agent.

To a large extent, these security risks are typical of the risks facing any other application using an IP internet. They are mentioned here only because IPX is not normally suspicious of its media. IPX network administrators will need to be aware of these additional security risks.

Assigned Numbers

The Internet Assigned Numbers Authority assigns well-known UDP port numbers. It has assigned port number 213 decimal to the IPX encapsulation technique described in this memo [5].

Acknowledgements

This encapsulation technique was developed independently by Schneider & Koch and by Novell. I'd like to thank Thomas Ruf of Schneider & Koch for reviewing this memo to confirm its agreement with the Schneider & Koch implementation and also for his other valuable suggestions.

References

[1] Xerox, Corp., "Internet Transport Protocols", XSIS 028112, Xerox Corporation, December 1981.

[2] Postel, J., "User Datagram Protocol", RFC 768, USC/Information Sciences Institute, August 1980.

[3] Postel, J., "Internet Protocol", RFC 791, DARPA, September 1981.

[4] Deering, S., "Host Extensions for IP Multicasting", RFC 1112, Stanford University, August 1989.

[5] Reynolds, J., and J. Postel, "Assigned Numbers", RFC-1060, USC/Information Sciences Institute, March 1990.

Security Considerations

See the "Security Issues" section above.

Author's Address

Don Provan
Novell, Inc.
2180 Fortune Drive
San Jose, California, 95131

Phone: (408)473-8440

EMail: donp@Novell.Com