Network Working Group
Request for Comments: 1683
Category: Informational
R. Clark
M. Ammar
K. Calvert
Georgia Institute of Technology
August 1994

# Multiprotocol Interoperability In IPng

## Status of this Memo

This memo provides information for the Internet community. This memo does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

## Abstract

This document was submitted to the IETF IPng area in response to RFC 1550. Publication of this document does not imply acceptance by the IPng area of any ideas expressed within. Comments should be submitted to the big-internet@munnari.oz.au mailing list.

## 1. Executive Summary

The two most commonly cited issues motivating the introduction of IPng are address depletion and routing table growth in IPv4. Further motivation is the fact that the Internet is witnessing an increasing diversity in the protocols and services found in the network. When evaluating alternatives for IPng, we should consider how well each alternative addresses the problems arising from this diversity. In this document, we identify several features that affect a protocol's ability to operate in a multiprotocol environment and propose the incorporation of these features into IPng.

Our thesis, succinctly stated, is: The next generation Internet Protocol should have features that support its use with a variety of protocol architectures.

## 2. Introduction

The Internet is not a single protocol network [4]. While TCP/IP remains the primary protocol suite, other protocols (e.g., IPX, AppleTalk, OSI) exist either natively or encapsulated as data within IP. As new protocols continue to be developed, we are likely to find that a significant portion of the traffic in future networks is not from single-protocol communications. It is important to recognize that multiprotocol networking is not just a transition issue. For instance, we will continue to see tunneling used to carry IPX traffic over the Internet between two Novell networks. Furthermore, the introduction of IPng is not going to result in a near term elimination of IPv4. Even when IPng becomes the primary protocol used in the Internet, there will still be IPv4 systems in use. We should consider such multiprotocol uses of the network as we design future protocols that can efficiently handle mixed protocol traffic.

We have identified several issues related to the way in which protocols operate in a multiprotocol environment. Many of these issues have traditionally been deemed "less important" by protocol designers since their goal was to optimize for the case where all systems supported the same protocol. With the increasing diversity of network protocols, this approach is no longer practical. By addressing the issues outlined in this paper, we can simplify the introduction of IPng to the Internet and reduce the risk for network managers faced with the prospect of supporting a new protocol. This will result in a faster, wider acceptance of IPng and increased interoperability between Internet hosts. In addition, by designing IPng to address these issues, we will make the introduction of future protocols (IPng2) even easier.

The outline for this document is as follows. In Section 3 we motivate the issues of multiprotocol networking with a discussion of an example system. In Section 4 we describe three main techniques for dealing with multiple protocols. This is followed in Section 5 by a description of the various protocol features that are important for implementing these three techniques. We conclude in Section 6 with a summary of the issues raised.

## 3. Multiprotocol Systems

Consider the multiprotocol architecture depicted in Figure 1. A system supporting this architecture provides a generic file-transfer service using either the Internet or OSI protocol stacks. The generic service presents the user with a consistent interface, regardless of the actual protocols used. The user can transfer files between this host and hosts supporting either of the single protocol stacks presented in Figures 2a and 2b. To carry out this file transfer, the user is not required to decide which protocols to use or to adjust between different application interfaces.

```
            +------------------------------------+
            |         File Transfer Service      |
            +------------+-----------------------+
            |            |          FTAM         |
            |            +-----------------------+
            |    FTP     |        ISO 8823       |
            |            +-----------------------+
            |            |        ISO 8327       |
            |            +-----------+-----------+
            |            |TP0/RFC1006|    TP4    |
            +------------+-----------+           |
            |          TCP           |           |
            +------------+-----------+-----------+
            |     IP     |         CLNP          |
            +------------+-----------------------+
```

Figure 1:  Multiprotocol architecture providing file-transfer service

```
  +------------+     +------------+     +------------+     +------------+
  |    FTP     |     |    FTAM    |     |    FTAM    |     |    FTP     |
  +------------+     +------------+     +------------+     +------------+
  |    TCP     |     | ISO 8823   |     | ISO 8823   |     |    TCP     |
  +------------+     +------------+     +------------+     +------------+
  |     IP     |     | ISO 8327   |     | ISO 8327   |     |    CLNP    |
  +------------+     +------------+     +------------+     +------------+
                     |    TP4     |     |TP0/RFC1006|
                     +------------+     +------------+
                     |    CLNP    |     |    TCP     |
                     +------------+     +------------+
                                        |     IP     |
                                        +------------+

    a) TCP/IP          b) OSI             c) RFC 1006       d) TUBA
```

Figure 2:  Protocol stacks providing file-transfer service.

Figure 2c depicts a mixed stack architecture that provides the upper layer OSI services using the Internet protocols.  This is an example of a "transition architecture" for providing OSI applications without requiring a full OSI implementation.  Figure 2d depicts a mixed stack architecture that provides the upper layer Internet applications using the OSI network protocol.  In addition to communicating with the two previous simple protocol stacks, the multiprotocol system of Figure 1 includes all the protocols necessary to communicate with these two new, mixed protocol stacks.

It is likely that many future network systems will be configured to support multiple protocols including IPng. As the IPng protocol is deployed, it is unreasonable to expect that users will be willing to give up any aspect of their current connectivity for the promise of a better future. In reality, most IPng installations will be made "in addition to" the current protocols. The resulting systems will resemble Figure 1 in that they will be able to communicate with systems supporting several different protocols.

Unfortunately, in most current examples, the architecture of Figure 1 is implemented as independent protocol stacks. This means that even though both TCP and CLNP exist on the system, there is no way to use TCP and CLNP in the same communication. The problem with current implementations of architectures like Figure 1 is that they are designed as co-existence architectures and are not integrated interoperability systems. We believe future systems should include mechanisms to overcome this traditional limitation. By integrating the components of multiple protocol stacks in a systematic way, we can interoperate with hosts supporting any of the individual stacks as well as those supporting various combinations of the stacks.

In order to effectively use multiple protocols, a system must identify which of the available protocols to use for a given communication task. We call this the Protocol Determination [2] task. In performing this task, a system determines the combination of protocols necessary to provide the needed service. For achieving interoperability, protocols are selected from the intersection of those supported on the systems that must communicate.

## 4. Multiprotocol Techniques

In this section we identify three main techniques to dealing with multiprotocol networks that are in use today and will continue to be used in the Internet. The first two techniques, tunneling and conversion, are categorized as intermediate-system techniques in that they are designed to achieve multiprotocol support without changing the end-systems. The third technique explicitly calls for the support of multiple protocols in end-systems. By describing these techniques here, we can motivate the need for the specific protocol features described in Section 5.

### 4.1 Encapsulation/Tunneling

Encapsulation or tunneling is commonly used when two networks that support a common protocol must be connected using a third intermediate network running a different protocol. Protocol packets from the two end networks are carried as data within the protocol of the intermediate network. This technique is only appropriate when both end-systems support the same protocol stack. It does not provide interoperability between these end systems and systems that only support the protocol stack in the intermediate network. Some examples of this technique are: a mechanism for providing the OSI transport services on top of the Internet protocols [13], encapsulating IEEE 802.2 frames in IPX network packets [5], tunneling IPX [10] and AppleTalk traffic over the Internet backbone. We expect IPng to be used for tunneling other network protocols over IPng and to be encapsulated.

## 4.2 Translation/Conversion

Despite their known limitations [8], translation or conversion gateways are another technique for handling multiple protocols [11, 12]. These gateways perform direct conversion of network traffic from one protocol to another. The most common examples of conversion gateways are the many electronic mail gateways now in use in the Internet. In certain cases it may also be feasible to perform conversion of lower layer protocols such as the network layer. This technique has been suggested as part of the transition plan for some of the current IPng proposals [3, 15].

## 4.3 Multiprotocol End-Systems

We expect that IPng will be introduced as an additional protocol in many network systems. This means that IPng should be able to coexist with other protocols on both end- and intermediate-systems. Specifically, IPng should be designed to support the Protocol Determination task described in Section 3.

One technique that we consider for solving the Protocol Determination problem is to employ a directory service in distributing system protocol configuration information. We have developed and implemented mechanism for using the Internet Domain Name System (DNS) [6, 7] to distribute this protocol information [2]. Using this mechanism, a multiprotocol host can determine the protocol configuration of a desired host when it retrieves the network address for that host. Then the multiprotocol host can match the configuration of the desired host to its own configuration and determine which protocols should be used to carry out the requested communication service.

Another alternative to determining protocol information about another host is Protocol Discovery. Using this approach, a host determines which protocols to use by trial-and-error with the protocols currently available. The initiating host monitors successive attempts to communicate and uses the information gained from that monitoring to build a knowledge base of the possible protocols of the remote system.

This knowledge is used to determine whether or not a communication link can be established and if it can, which protocol should be used.

An important aspect of the Protocol Discovery approach is that it requires an error and control feedback system similar to ICMP [9], but with additional functionality (See Section 5).

## 5. Protocol Features

In this section we identify features that affect a protocol’s ability to support the multiprotocol techniques described in the previous section. These features indicate specific areas that should be considered when comparing proposed protocols. We present two different types of protocol features: those that should be included as part of the IPng protocol standard, and those that should be considered as part of the implementation and deployment requirements for IPng.

### 5.1 Protocol Standard Features

o Addressing

A significant problem in dealing with multiprotocol networks is that most of the popular network protocols use different addressing mechanisms. The problem is not just with different lengths but also with different semantics (e.g., hierarchical vs. flat addresses). In order to accommodate these multiple formats, IPng should have the flexibility to incorporate many address formats within its addressing mechanism.

A specific example might be for IPng to have the ability to include an IPv4 or IPX address as a subfield of the IPng address. This would reduce the complexity of performing address conversion by limiting the number of external mechanisms (e.g., lookup tables) needed to convert an address. This reduction in complexity would facilitate both tunneling and conversion. It would also simplify the task of using IPng with legacy applications which rely on a particular address format.

o Header Option Handling

In any widely used protocol, it is advantageous to define option mechanisms for including header information that is not required in all packets or is not yet defined. This is especially true in multiprotocol networks where there is wide variation in the requirements of protocol users. IPng should provide efficient, flexible support for future header options. This will better accommodate the different user needs and will facilitate conversion between IPng and other protocols with different standard features.

As part of the support for protocol options, IPng should include a mechanism for specifying how a system should handle unsupported options. If a network system adds an option header, it should be able to specify whether another system that does not support the option should drop the packet, drop the packet and return an error, forward it as is, or forward it without the option header. The ability to request the "forward as is" option is important when conversion is used. When two protocols have different features, a converter may introduce an option header that is not understood by an intermediate node but may be required for interpretation of the packet at the ultimate destination. On the other hand, consider the case where a source is using IPng with a critical option like encryption. In this situation the user would not want a conversion to be performed where the option was not understood by the converter. The "drop the packet" or "drop and return error" options would likely be used in this scenario.

o Multiplexing

The future Internet protocol should support the ability to distinguish between multiple users of the network. This includes the ability to handle traditional "transport layer" protocols like TCP and UDP, as well as other payload types such as encapsulated AppleTalk packets or future real-time protocols. This kind of protocol multiplexing can be supported with an explicit header field as in IPv4 or by reserving part of the address format as is done with OSI NSEL’s.

In a multiprotocol network there will likely be a large number of different protocols running atop IPng. It should not be necessary to use a transport layer protocol for the sole purpose of providing multiplexing for the various network users. The cost of this additional multiplexing is prohibitive for future high-speed networks [14]. In order to avoid the need for an additional level of multiplexing, the IPng should either use a payload selector larger than the 8-bits used in IPv4 or provide an option for including additional payload type information within the header.

o Status/Control Feedback

With multiple protocols, the correct transmission of a packet might include encapsulation in another protocol and/or multiple conversions to different protocols before the packet finally reaches its destination. This means that there are many different places the transmission can fail and determining what went wrong will be a challenge.

In order to handle this situation, a critical protocol feature in multiprotocol networks is a powerful error reporting mechanism.

In addition to reporting traditional network level errors, such as those reported by ICMP [9], the IPng error mechanism should include feedback on tunneling and conversion failures. Also, since it is impossible to know exactly which part of a packet is an encapsulated header, it is important that the feedback mechanism include as much of the failed packet as possible in the returned error message.

In addition to providing new types of feedback, this mechanism should support variable resolution such that a transmitting system can request limited feedback or complete information about the communication process. This level of control would greatly facilitate the Protocol Discovery process described in Section 4.3. For example, a multiprotocol system could request maximal feedback when it sends packets to a destination it has not communicated with for some time. After the first few packets to this "new" destination, the system would revert back to limited feedback, freeing up the resources used by the network feedback mechanisms.

Finally, it is important that the information provided by the feedback mechanism be available outside the IPng implementation. In multiprotocol networks it is often the case that the solution to a communication problem requires an adjustment in one of the protocols outside the network layer. In order for this to happen, the other protocols must be able to access and interpret these feedback messages.

o MTU Discovery or Fragmentation

A form of multiprotocol support that has long been a part of networking is the use of diverse data link and physical layers. One aspect of this support that affects the network layer is the different Maximum Transmission Units (MTU) used by various media formats. For efficiency, many protocols will attempt to avoid fragmentation at intermediate nodes by using the largest packet size possible, without exceeding the minimum MTU along the route. To achieve this, a network protocol performs MTU discovery to find the smallest MTU on a path.

The choice of mechanism for dealing with differing MTUs is also important when doing conversion or tunneling with multiple protocols. When tunneling is performed by an intermediate node, the resulting packets may be too large to meet the MTU requirements. Similarly, if conversion at an intermediate node results in a larger protocol header, the new packets may also be too large. In both cases, it may be desirable to have the source host reduce the transmission size used in order to prevent the need for additional fragmentation. This information could be sent to the source host as part of the previously described feedback mechanism or as an additional MTU discovery message.

## 5.2 Implementation/Deployment Features

- o Switching

We define switching in a protocol as the capability to simultaneously use more than one different underlying protocol [1]. In network layer protocols, this implies using different datalink layers. For example, it may be necessary to select between the 802.3 LLC and traditional Ethernet interfaces when connecting a host to an "ethernet" network. Additionally, in some systems IPng will not be used directly over a datalink layer but will be encapsulated within another network protocol before being transmitted. It is important that IPng be designed to support different underlying datalink services and that it provide mechanisms allowing IPng users to specify which of the available services should be used.

- o Directory Service Requirements

While not specifically a part of the IPng protocol, it is clear that the future Internet will include a directory service for obtaining address information for IPng. In light of this, there are some features of the directory service that should be considered vis-a-vis their support for multiple protocols.

First, the directory service should be able to distribute address formats for several different protocol families, not just IPng and IPv4. This is necessary for the use of tunneling, conversion, and the support of multiprotocol systems. Second, the directory service should include support for distributing protocol configuration information in addition to addressing information for the network hosts. This feature will support the protocol determination task to be carried out by multiprotocol systems [2].

## 6. Conclusion

Future networks will incorporate multiple protocols to meet diverse user requirements. Because of this, we are likely to find that a significant portion of the traffic in the Internet will not be from single-protocol communications (e.g., TCPng/IPng). This will not just be true of near term, transitional networks but will remain as a reality for most of the Internet. As we pursue the selection of IPng, we should consider the special needs of multiprotocol networks. In particular, IPng should include mechanisms to handle mixed protocol traffic that includes tunneling, conversion, and multiprotocol end-systems.

## 7. Acknowledgments

The authors would like to acknowledge the support for this work by a grant from the National Science Foundation (NCR-9305115) and the TRANSOPEN project of the Army Research Lab (formerly AIRMICS) under contract number DAKF11-91-D-0004.

## 8. References

[1] Clark, R., Ammar, M., and K. Calvert, "Multi-protocol architectures as a paradigm for achieving inter-operability", In Proceedings of IEEE INFOCOM, April 1993.

[2] Clark, R., Calvert, K. and M. Ammar, "On the use of directory services to support multiprotocol interoperability", To appear in proceedings of IEEE INFOCOM, 1994. Technical Report GIT-CC-93/56, College of Computing, Georgia Institute of Technology, ATLANTA, GA 30332-0280, August 1993.

[3] Gilligan, R., Nordmark, E., and B. Hinden, "IPAE: the SIPP Interoperability and Transition Mechanism, Work in Progress, November 1993.

[4] Leiner, B., and Y. Rekhter, "The Multiprotocol Internet", RFC 1560, USRA, IBM, December 1993.

[5] McLaughlin, L., "Standard for the Transmission of 802.2 Packets over IPX Networks", RFC 1132, The Wollongong Group, November 1989.

[6] Mockapetris, P., "Domain Names - Concepts and Facilities", STD 13, RFC 1034, USC/Information Sciences Institute, November 1987.

[7] Mockapetris, P., "Domain Names - Implementation and Specification. STD 13, RFC 1035, USC/Information Sciences Institute, November 1987.

[8] Padlipsky, M., Gateways, Architectures, and Heffalumps", RFC 875, MITRE, September 1982.

[9] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, USC/Information Sciences Institute, September 1981.

[10] Provan, D., "Tunneling IPX Traffic Through IP Networks", RFC 1234, Novell, Inc., June 1991.

[11] Rose, M., "The Open Book", Prentice-Hall, Englewood Cliffs, New Jersey, 1990.

[12] Rose, M., "The ISO Development Environment User's Manual - Version 7.0.", Performance Systems International, July 1991.

[13] Rose, M., and D. Cass, "ISO Transport Services on top of the TCP", STD 35, RFC 1006, Northrop Research and Technology Center, May 1987.

[14] Tennenhouse, D., "Layered multiplexing considered harmful", In IFIP Workshop on Protocols for High-Speed Networks. Elsevier, May 1989.

[15] Ullmann, R., "CATNIP: Common architecture technology for next-generation internet protocol", Work in Progress, October 1993.

## 9. Security Considerations

Security issues are not discussed in this memo.

## 10. Authors' Addresses

Russell J. Clark
College of Computing Georgia Institute of Technology
Atlanta, GA 30332-0280

EMail: rjc@cc.gatech.edu

Mostafa H. Ammar
College of Computing Georgia Institute of Technology
Atlanta, GA 30332-0280

EMail: ammar@cc.gatech.edu

Kenneth L. Calvert
College of Computing Georgia Institute of Technology
Atlanta, GA 30332-0280

EMail: calvert@cc.gatech.edu