Network Working Group
Request for Comments: 4840
Category: Informational

B. Aboba, Ed.
E. Davies
D. Thaler
Internet Architecture Board

April 2007

Multiple Encapsulation Methods Considered Harmful

Status of This Memo

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

Copyright Notice

Copyright (C) The IETF Trust (2007).

Abstract

This document describes architectural and operational issues that arise from link-layer protocols supporting multiple Internet Protocol encapsulation methods.

Table of Contents

1. Introduction

This document describes architectural and operational issues arising from the use of multiple ways of encapsulating IP packets on the same link.

While typically a link-layer protocol supports only a single Internet Protocol (IP) encapsulation method, this is not always the case. For example, on the same cable it is possible to encapsulate an IPv4 packet using Ethernet [DIX] encapsulation as defined in "A Standard for the Transmission of IP Datagrams over Ethernet Networks" [RFC894], the IEEE 802.2/802.3 LLC [IEEE-802.3.2002] Type 1 encapsulation defined in "Two Methods For The Transmission of IP Datagrams over IEEE 802.3 Networks" [RFC948], or the IEEE 802 [IEEE-802.1A.1990] encapsulation defined in "A Standard for the Transmission of IP Datagrams over IEEE 802 Networks" [RFC1042]. Historically, a further encapsulation method was used on some Ethernet systems as specified in "Trailer Encapsulations" [RFC893]. Similarly, ATM (e.g., see [RFC2684]), the Point-to-Point Protocol (PPP) [RFC1661], and IEEE 802.16 [IEEE-802.16e.2005] also support multiple encapsulation mechanisms.

1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Broadcast domain
: The set of all endpoints that receive broadcast frames sent by an endpoint in the set.

Classification
: As defined in [IEEE-802.16e.2005], the process by which a Medium Access Control (MAC) Service Data Unit (SDU) is mapped into a particular transport connection for transmission between MAC peers.

Connection Identifier (CID)
: In [IEEE-802.16e.2005] the connection identifier is a 16-bit value that identifies a transport connection or an uplink (UL)/downlink (DL) pair of associated management connections. A connection is a unidirectional mapping between base station (BS) and subscriber station (SS) MAC peers. Each transport connection has a particular set of associated parameters indicating characteristics such as the ciphersuite and quality-of-service.

Link
: A communication facility or medium over which nodes can communicate at the link layer, i.e., the layer immediately below IP.

Link Layer
: The conceptual layer of control or processing logic that is responsible for maintaining control of the link. The link-layer functions provide an interface between the higher-layer logic and the link. The link layer is the layer immediately below IP.

1.2. Ethernet Experience

The fundamental issues with multiple encapsulation methods on the same link are described in [RFC1042] and "Requirements for Internet Hosts -- Communication Layers" [RFC1122]. This section summarizes the concerns articulated in those documents and also describes the limitations of approaches suggested to mitigate the problems, including encapsulation negotiation and use of routers.

[RFC1042] described the potential issues resulting from contemporaneous use of Ethernet and IEEE 802.3 encapsulations on the same physical cable:

> Interoperation with Ethernet
>
> It is possible to use the Ethernet link level protocol [DIX] on the same physical cable with the IEEE 802.3 link level protocol. A computer interfaced to a physical cable used in this way could potentially read both Ethernet and 802.3 packets from the network. If a computer does read both types of packets, it must keep track of which link protocol was used with each other computer on the network and use the proper link protocol when sending packets.
>
> One should note that in such an environment, link level broadcast packets will not reach all the computers attached to the network, but only those using the link level protocol used for the broadcast.
>
> Since it must be assumed that most computers will read and send using only one type of link protocol, it is recommended that if such an environment (a network with both link protocols) is necessary, an IP gateway be used as if there were two distinct networks.
>
> Note that the MTU for the Ethernet allows a 1500 octet IP datagram, with the MTU for the 802.3 network allows only a 1492 octet IP datagram.

When multiple IP encapsulation methods were supported on a given link, all hosts could not be assumed to support the same set of encapsulation methods. This in turn implied that the broadcast domain might not include all hosts on the link. Where a single encapsulation does not reach all hosts on the link, a host needs to determine the appropriate encapsulation prior to sending. While a host supporting reception of multiple encapsulations could keep track of the encapsulations it receives, this does not enable initiation of communication; supporting initiation requires a host to support sending of multiple encapsulations in order to determine which one to use. However, requiring hosts to send and receive multiple encapsulations is a potentially onerous requirement. [RFC1122], Section 2.3.3, notes the difficulties with this approach:

> Furthermore, it is not useful or even possible for a dual-format host to discover automatically which format to send, because of the problem of link-layer broadcasts.

To enable hosts that only support sending and receiving of a single encapsulation to communicate with each other, a router can be utilized to segregate the hosts by encapsulation. Here only the router needs to support sending and receiving of multiple encapsulations. This requires assigning a separate unicast prefix to each encapsulation, or else all hosts in the broadcast domain would not be reachable with a single encapsulation.

[RFC1122], Section 2.3.3, provided guidance on encapsulation support:

> Every Internet host connected to a 10Mbps Ethernet cable:
>
> - MUST be able to send and receive packets using RFC-894 encapsulation;
> - SHOULD be able to receive RFC-1042 packets, intermixed with RFC-894 packets; and
> - MAY be able to send packets using RFC-1042 encapsulation.

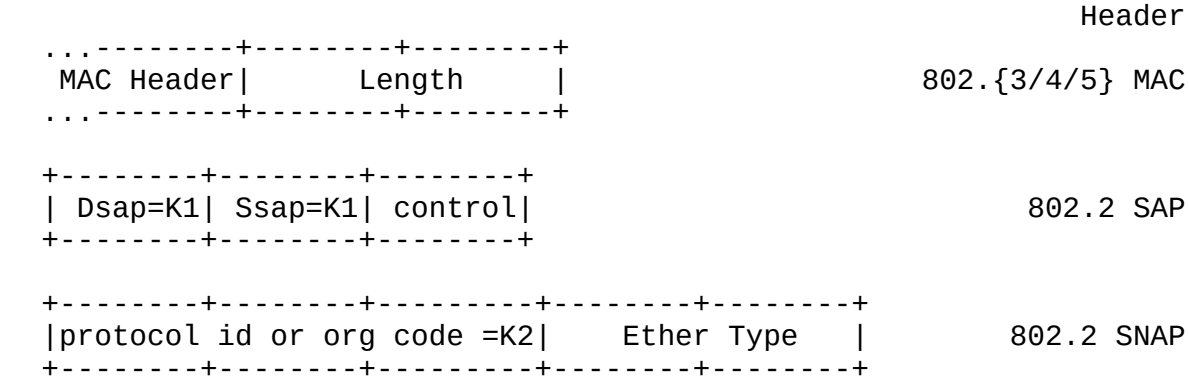
An Internet host that implements sending both the RFC-894 and the RFC-1042 encapsulation MUST provide a configuration switch to select which is sent, and this switch MUST default to RFC-894.

By making Ethernet encapsulation mandatory to implement for both send and receive, and also the default for sending, [RFC1122] recognized Ethernet as the predominant encapsulation, heading off potential interoperability problems.

1.2.1. IEEE 802.2/802.3 LLC Type 1 Encapsulation

Prior to standardization of the IEEE 802 encapsulation in [RFC1042], an IEEE 802.2/802.3 LLC Type 1 encapsulation was specified in [RFC948], utilizing 6 in the Source Service Access Point (SSAP) and Destination Service Access Point (DSAP) fields of the IEEE 802.2 header. However, since the SSAP and DSAP fields are each only a single octet, and the Ethertype values for IP, ARP [RFC826], and RARP [RFC903] are greater than 1500, these values cannot be represented in the SSAP and DSAP fields. As a result, the encapsulation described in [RFC948] did not support protocols requiring distinct Ethertypes such as ARP or RARP, and implementations typically included support for alternatives to ARP such as the Probe [PROBE] protocol. Support for ARP, RARP and other IP protocols utilizing distinct Ethertypes was addressed in [RFC1042], which obsoleted [RFC948]. [RFC1042] utilized the Sub-Network Access Protocol (SNAP) form of the IEEE 802.2 Logical Link Control (LLC) with the SSAP and DSAP fields set to 170, including support for the Ethertype field. As noted in "Assigned Numbers" [RFC1010]:

> At an ad hoc special session on "IEEE 802 Networks and ARP", held during the TCP Vendors Workshop (August 1986), an approach to a consistent way to send DoD-IP datagrams and other IP related protocols on 802 networks was developed.
>
> Due to some evolution of the IEEE 802.2 standards and the need to provide for a standard way to do additional DoD-IP related protocols (such as the Address Resolution Protocol (ARP) on IEEE 802 network, the following new policy is established, which will replace the old policy (see RFC 960 and RFC 948 [108]).
>
> The new policy is for the Internet community to use the IEEE 802.2 encapsulation on 802.3, 802.4, and 802.5 networks by using the SNAP with an organization code indicating that the following 16 bits specify the EtherType code (where IP = 2048 (0800 hex), see Ethernet Numbers of Interest).



The total length of the SAP Header and the SNAP header is 8-octets, making the 802.2 protocol overhead come out on a nice boundary.

K1 is 170. The IEEE likes to talk about things in little-endian bit transmission order and specifies this value as 01010101. In big-endian order, as used in Internet specifications, this becomes 10101010 binary, or AA hex, or 170 decimal.

K2 is 0 (zero).

The use of the IP LSAP (K1 = 6) is to be phased out as quickly as possible.

Many of the issues involved in coexistence of the [RFC948] and [RFC1042] encapsulations are similar to those described in Section 1.2. For example, due to use of different SSAP/DSAP values, the broadcast domain might not include all hosts on the link, and a host would need to determine the appropriate encapsulation prior to sending. However, the lack of support for ARP within the [RFC948] encapsulation created additional interoperability and implementation issues. For example, the lack of support for ARP in [RFC948] implied that implementations supporting both [RFC948] and [RFC894] or [RFC1042] encapsulations would need to implement both ARP and an alternative address resolution mechanism such as Probe. Also, since the address resolution mechanism for [RFC948] implementations was not standardized, interoperability problems would likely have arisen had [RFC948] been widely implemented.

1.2.2. Trailer Encapsulation

As noted in "Trailer Encapsulations" [RFC893], trailer encapsulation was an optimization developed to minimize memory-to-memory copies on reception. By placing variable-length IP and transport headers at the end of the packet, page alignment of data could be more easily

maintained. Trailers were implemented in 4.2 Berkeley System Distribution (BSD), among others. While, in theory, trailer encapsulation could have been applied to the Ethernet [RFC894] or IEEE 802 [RFC1042] encapsulations (creating four potential encapsulations of IP!), in practice, trailer encapsulation was only supported for Ethernet. A separate Ethertype was utilized in order to enable IP packets in trailer encapsulation to be distinguished from [RFC894] encapsulation. Since the [RFC948] encapsulation did not support the Ethertype field (or ARP), this mechanism could not have been used in [RFC948] implementations.

[RFC1122], Section 2.3.1, described the issues with trailer encapsulation:

> DISCUSSION
>
> The trailer protocol is a link-layer encapsulation technique that rearranges the data contents of packets sent on the physical network. In some cases, trailers improve the throughput of higher layer protocols by reducing the amount of data copying within the operating system. Higher layer protocols are unaware of trailer use, but both the sending and receiving host MUST understand the protocol if it is used. Improper use of trailers can result in very confusing symptoms. Only packets with specific size attributes are encapsulated using trailers, and typically only a small fraction of the packets being exchanged have these attributes. Thus, if a system using trailers exchanges packets with a system that does not, some packets disappear into a black hole while others are delivered successfully.
>
> IMPLEMENTATION:
>
> On an Ethernet, packets encapsulated with trailers use a distinct Ethernet type [RFC893], and trailer negotiation is performed at the time that ARP is used to discover the link-layer address of a destination system.
>
> Specifically, the ARP exchange is completed in the usual manner using the normal IP protocol type, but a host that wants to speak trailers will send an additional "trailer ARP reply" packet, i.e., an ARP reply that specifies the trailer encapsulation protocol type but otherwise has the format of a normal ARP reply. If a host configured to use trailers receives a trailer ARP reply message from a remote machine, it can add that machine to the list of machines that understand trailers, e.g., by marking the corresponding entry in the ARP cache.

> Hosts wishing to receive trailers send trailer ARP replies whenever they complete exchanges of normal ARP messages for IP. Thus, a host that received an ARP request for its IP protocol address would send a trailer ARP reply in addition to the normal IP ARP reply; a host that sent the IP ARP request would send a trailer ARP reply when it received the corresponding IP ARP reply. In this way, either the requesting or responding host in an IP ARP exchange may request that it receive trailers.
>
> This scheme, using extra trailer ARP reply packets rather than sending an ARP request for the trailer protocol type, was designed to avoid a continuous exchange of ARP packets with a misbehaving host that, contrary to any specification or common sense, responded to an ARP reply for trailers with another ARP reply for IP. This problem is avoided by sending a trailer ARP reply in response to an IP ARP reply only when the IP ARP reply answers an outstanding request; this is true when the hardware address for the host is still unknown when the IP ARP reply is received. A trailer ARP reply may always be sent along with an IP ARP reply responding to an IP ARP request.

Since trailer encapsulation negotiation depends on ARP, it can only be used where all hosts on the link are within the same broadcast domain. It was assumed that all hosts supported sending and receiving ARP packets in standard Ethernet encapsulation [RFC894], so that negotiation between Ethernet and IEEE 802 encapsulations was not required, only negotiation between standard Ethernet [RFC894] and trailer [RFC893] encapsulation. Had hosts supporting trailer encapsulation also supported one or more IEEE 802 framing mechanisms, the negotiation would have been complicated still further. For example, since [RFC948] implementations did not support the Ethertype field or ARP, the trailer negotiation mechanism could not have been utilized, and additional difficulty would have been encountered in distinguishing trailer encapsulated data frames from normally encapsulated frames.

[RFC1122], Section 2.3.1, provided the following guidance for use of trailer encapsulation:

> The trailer protocol for link-layer encapsulation MAY be used, but only when it has been verified that both systems (host or gateway) involved in the link-layer communication implement trailers. If the system does not dynamically negotiate use of the trailer protocol on a per-destination basis, the default configuration MUST disable the protocol.

4.2BSD did not support dynamic negotiation, only configuration of trailer encapsulation at boot time, and therefore [RFC1122] required that the trailer encapsulation be disabled by default on those systems.

1.3. PPP Experience

PPP can support both encapsulation of IEEE 802 frames as defined in [RFC3518], as well as IPv4 and IPv6 [RFC2472] packets. Multiple compression schemes are also supported.

In addition to PPP Data Link Layer (DLL) protocol numbers allocated for IPv4 (0x0021), IPv6 (0x0057), and Bridging PDU (0x0031), the following codepoints have been assigned:

- two for RObust Header Compression (ROHC) [RFC3095]: ROHC small-CID (0x0003) and ROHC large-CID (0x0005)
- two for Van Jacobson compression [RFC1144]: Compressed TCP/IP (0x002d) and Uncompressed TCP/IP (002f)
- one for IPv6 Header Compression [RFC2507]: (0x004f)
- nine for RTP IP Header Compression [RFC3544]: Full Header (0x0061), Compressed TCP (0x0063), Compressed Non TCP (0x0065), UDP 8 (0x0067), RTP 8 (0x0069), Compressed TCP No Delta (0x2063), Context State (0x2065), UDP 16 (0x2067), and RTP 16 (0x2069)

Although PPP can encapsulate IP packets in multiple ways, typically multiple encapsulation schemes are not operational on the same link, and therefore the issues described in this document rarely arise. For example, while PPP can support both encapsulation of IEEE 802 frames as defined in [RFC3518], as well as IPv4 and IPv6 [RFC2472] packets, in practice, multiple encapsulation mechanisms are not operational on the same link. Similarly, only a single compression scheme is typically negotiated for use on a link.

1.4. Potential Mitigations

In order to mitigate problems arising from multiple encapsulation methods, it may be possible to use switches [IEEE-802.1D.2004] or routers, or to attempt to negotiate the encapsulation method to be used. As described below, neither approach may be completely satisfactory.

The use of switches or routers to enable communication between hosts utilizing multiple encapsulation methods is not a panacea. If separate unicast prefixes are used for each encapsulation, then the choice of encapsulation can be determined from the routing table. If the same unicast prefix is used for each encapsulation method, it is necessary to keep state for each destination host. However, this may not work in situations where hosts using different encapsulations respond to the same anycast address.

In situations where multiple encapsulation methods are enabled on a single link, negotiation may be supported to allow hosts to determine how to encapsulate a packet for a particular destination host.

Negotiating the encapsulation above the link layer is potentially problematic since the negotiation itself may need to be carried out using multiple encapsulations. In theory, it is possible to negotiate an encapsulation method by sending negotiation packets over all encapsulation methods supported, and keeping state for each destination host. However, if the encapsulation method must be dynamically negotiated for each new on-link destination, communication to new destinations may be delayed. If most communication is short, and the negotiation requires an extra round trip beyond link-layer address resolution, this can become a noticeable factor in performance. Also, the negotiation may result in consumption of additional bandwidth.

2. Evaluation of Arguments for Multiple Encapsulations

There are several reasons often given in support of multiple encapsulation methods. We discuss each in turn, below.

2.1. Efficiency

Claim: Multiple encapsulation methods allow for greater efficiency. For example, it has been argued that IEEE 802 or Ethernet encapsulation of IP results in excessive overhead due to the size of the data frame headers, and that this can adversely affect performance on wireless networks, particularly in situations where support of Voice over IP (VoIP) is required.

Discussion: Even where these performance concerns are valid, solutions exist that do not require defining multiple IP encapsulation methods. For example, links may support Ethernet frame compression so that Ethernet Source and Destination Address fields are not sent with every packet.

It is possible for link layers to negotiate compression without requiring higher-layer awareness; the Point-to-Point Protocol (PPP)

[RFC1661] is an example. "The PPP Compression Control Protocol (CCP)" [RFC1962] enables negotiation of data compression mechanisms, and "Robust Header Compression (ROHC) over PPP" [RFC3241] and "IP Header Compression over PPP" [RFC3544] enable negotiation of header compression, without Internet-layer awareness. Any frame can be "decompressed" based on the content of the frame, and prior state based on previous control messages or data frames. Use of compression is a good way to solve the efficiency problem without introducing problems at higher layers.

There are also situations in which use of multiple encapsulations can degrade performance or result in packet loss. The use of multiple encapsulation methods with differing Maximum Transfer Units (MTUs) can result in differing MTUs for on-link destinations. If the link-layer protocol does not provide per-destination MTUs to the IP layer, it will need to use a default MTU; to avoid fragmentation, this must be less than or equal to the minimum MTU of on-link destinations. If the default MTU is too low, the full bandwidth may not be achievable. If the default MTU is too high, packet loss will result unless or until IP Path MTU Discovery is used to discover the correct MTU.

Recommendation: Where encapsulation is an efficiency issue, use header compression. Where the encapsulation method or the use of compression must be negotiated, negotiation should either be part of bringing up the link, or be piggybacked in the link-layer address resolution exchange; only a single compression scheme should be negotiated on a link. Where the MTU may vary among destinations on the same link, the link-layer protocol should provide a per-destination MTU to IP.

2.2. Multicast/Broadcast

Claim: Support for Ethernet encapsulation requires layer 2 support for distribution of IP multicast/broadcast packets. In situations where this is difficult, support for Ethernet is problematic and other encapsulations are necessary.

Discussion: Irrespective of the encapsulation used, IP packets sent to multicast (IPv4/IPv6) or broadcast (IPv4) addresses need to reach all potential on-link receivers. Use of alternative encapsulations cannot remove this requirement, although there is considerable flexibility in how it can be met. Non-Broadcast Multiple Access (NBMA) networks can still support the broadcast/multicast service via replication of unicast frames.

Techniques are also available for improving the efficiency of IP multicast/broadcast delivery in wireless networks. In order to be receivable by any host within listening range, an IP

multicast/broadcast packet sent as link-layer multicast/broadcast over a wireless link needs to be sent at the lowest rate supported by listeners. If the sender does not keep track of the rates negotiated by group listeners, by default, multicast/broadcast traffic is sent at the lowest supported rate, resulting in increased overhead. However, a sender can also deliver an IP multicast/broadcast packet using unicast frame(s) where this would be more efficient. For example, in IEEE 802.11, multicast/broadcast traffic sent from the Station (STA) to the Access Point (AP) is always sent as unicast, and the AP tracks the negotiated rate for each STA, so that it can send unicast frames at a rate appropriate for each station.

In order to limit the propagation of link-scope multicast or broadcast traffic, it is possible to assign a separate prefix to each host.

Unlike broadcasts, which are received by all hosts on the link regardless of the protocol they are running, multicasts only need be received by those hosts belonging to the multicast group. In wired networks, it is possible to avoid forwarding multicast traffic on switch ports without group members, by snooping of Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) traffic as described in "Considerations for IGMP and MLD Snooping Switches" [RFC4541].

In wireless media where data rates to specific destinations are negotiated and may vary over a wide range, it may be more efficient to send multiple frames via link-layer unicast than to send a single multicast/broadcast frame. For example, in [IEEE-802.11.2003] multicast/broadcast traffic from the client station (STA) to the Access Point (AP) is sent via link-layer unicast.

Recommendation: Where support for link-layer multicast/broadcast is problematic, limit the propagation of link-scope multicast and broadcast traffic by assignment of separate prefixes to hosts. In some circumstances, it may be more efficient to distribute multicast/broadcast traffic as multiple link-layer unicast frames.

2.3. Multiple Uses

Claim: No single encapsulation is optimal for all purposes. Therefore, where a link layer is utilized in disparate scenarios (such as both fixed and mobile deployments), multiple encapsulations are a practical requirement.

Discussion: "Architectural Principles of the Internet" [RFC1958], point 3.2, states:

> If there are several ways of doing the same thing, choose one. If a previous design, in the Internet context or elsewhere, has successfully solved the same problem, choose the same solution unless there is a good technical reason not to. Duplication of the same protocol functionality should be avoided as far as possible, without of course using this argument to reject improvements.

Existing encapsulations have proven themselves capable of supporting disparate usage scenarios. For example, the Point-to-Point Protocol (PPP) has been utilized by wireless link layers such as General Packet Radio Service (GPRS), as well as in wired networks in applications such as "PPP over SONET/SDH" [RFC2615]. PPP can even support bridging, as described in "Point-to-Point Protocol (PPP) Bridging Control Protocol (BCP)" [RFC3518].

Similarly, Ethernet encapsulation has been used in wired networks as well as Wireless Local Area Networks (WLANs) such as IEEE 802.11 [IEEE-802.11.2003]. Ethernet can also support Virtual LANs (VLANs) and Quality of Service (QoS) [IEEE-802.1Q.2003].

Therefore, disparate usage scenarios can be addressed by choosing a single encapsulation, rather than multiple encapsulations. Where an existing encapsulation is suitable, this is preferable to creating a new encapsulation.

Where encapsulations other than IP over Point-to-Point Protocol (PPP) [RFC1661], Ethernet, or IEEE 802 are supported, difficulties in operating system integration can lead to interoperability problems.

In order to take advantage of operating system support for IP encapsulation over PPP, Ethernet, or IEEE 802, it may be tempting for a driver supporting an alternative encapsulation to emulate PPP, Ethernet, or IEEE 802 support. Typically, PPP emulation requires that the driver implement PPP, enabling translation of PPP control and data frames to the equivalent native facilities. Similarly, Ethernet or IEEE 802 emulation typically requires that the driver implement Dynamic Host Configuration Protocol (DHCP) v4 or v6, Router Solicitation/Router Advertisement (RS/RA), Address Resolution Protocol (ARP), or IPv6 Neighbor Discovery (ND) in order to enable translation of these frames to and from native facilities.

Where drivers are implemented in kernel mode, the work required to provide faithful emulation may be substantial. This creates the temptation to cut corners, potentially resulting in interoperability problems.

For example, it might be tempting for driver implementations to neglect IPv6 support. A driver emulating PPP might support only IP Control Protocol (IPCP), but not IPCPv6; a driver emulating Ethernet or IEEE 802 might support only DHCPv4 and ARP, but not DHCPv6, RS/RA, or ND. As a result, an IPv6 host connecting to a network supporting IPv6 might find itself unable to use IPv6 due to lack of driver support.

Recommendation: Support a single existing encapsulation where possible. Emulation of PPP, Ethernet, or IEEE 802 on top of alternative encapsulations should be avoided.

3. Additional Issues

There are a number of additional issues arising from use of multiple encapsulation methods, as hinted at in Section 1. We discuss each of these below.

3.1. Generality

Link-layer protocols such as [IEEE-802.1A.1990] and [DIX] inherently support the ability to add support for a new packet type without modification to the link-layer protocol.

IEEE 802.16 [IEEE-802.16.2004] splits the Media Access Control (MAC) layer into a number of sublayers. For the uppermost of these, the standard defines the concept of a service-specific Convergence Sublayer (CS). The two underlying sublayers (the MAC Common Part Sublayer and the Security Sublayer) provide common services for all instantiations of the CS.

While [IEEE-802.16.2004] defined support for the Asynchronous Transfer Mode (ATM) CS and the Packet CS for raw IPv4, raw IPv6, and Ethernet with a choice of six different classifiers, [IEEE-802.16e.2005] added support for raw and Ethernet-framed ROHC Enhanced Compressed RTP (ECRTP) compressed packets. As a result, [IEEE-802.16e.2005] defines the ATM CS and multiple versions of the Packet CS for the transmission of raw IPv4, raw IPv6, 802.3/Ethernet, 802.1Q VLAN, IPv4 over 802.3/Ethernet, IPv6 over 802.3/Ethernet, IPv4 over 802.1Q VLAN, IPv6 over 802.1Q VLAN, raw ROHC-compressed packets, raw ECRTP-compressed packets, ROHC-compressed packets over 802.3/Ethernet. and ECRTP-compressed packets over 802.3/Ethernet.

As noted in [Generic], [IEEE-802.16.2004] appears to imply that the standard will need to be modified to support new packet types:

> We are concerned that the 802.16 protocol cannot easily be extendable to transport new protocols over the 802.16 air interface. It would appear that a Convergence Sublayer is needed for every type of protocol transported over the 802.16 MAC. Every time a new protocol type needs to be transported over the 802.16 air interface, the 802.16 standard needs to be modified to define a new CS type. We need to have a generic Packet Convergence Sublayer that can support multi-protocols and which does not require further modification to the 802.16 standard to support new protocols. We believe that this was the original intention of the Packet CS. Furthermore, we believe it is difficult for the industry to agree on a set of CS's that all devices must implement to claim "compliance".

The use of IP and/or upper-layer protocol specific classification and encapsulation methods, rather than a 'neutral' general purpose encapsulation, may give rise to a number of undesirable effects explored in the following subsections.

If the link layer does not provide a general purpose encapsulation method, deployment of new IP and/or upper-layer protocols will be dependent on deployment of the corresponding new encapsulation support in the link layer.

Even if a single encapsulation method is used, problems can still occur if demultiplexing of ARP, IPv4, IPv6, and any other protocols in use, is not supported at the link layer. While it is possible to demultiplex such packets based on the Version field (first four bits on the packet), this assumes that IPv4-only implementations will be able to properly handle IPv6 packets. As a result, a more robust design is to demultiplex protocols in the link layer, such as by assigning a different protocol type, as is done in IEEE 802 media where a Type of 0x0800 is used for IPv4, and 0x86DD for IPv6.

Recommendations: Link-layer protocols should enable network packets (IPv4, IPv6, ARP, etc.) to be demultiplexed in the link layer.

3.2. Layer Interdependence

Within IEEE 802.16, the process by which frames are selected for transmission on a connection identifier (CID) is known as "classification". Fields in the Ethernet, IP, and UDP/TCP headers can be used for classification; for a particular CS, a defined subset of header fields may be applied for that purpose.

Utilizing IP and/or upper layer headers in link-layer classification will almost inevitably lead to interdependencies between link-layer and upper-layer specifications. Although this might appear to be

desirable in terms of providing a highly specific (and hence interoperable) mapping between the capabilities provided by the link layer (e.g., quality-of-service support) and those that are needed by upper layers, this sort of capability is probably better provided by a more comprehensive service interface (Application Programming Interface) in conjunction with a single encapsulation mechanism.

IPv6, in particular, provides an extensible header system. An upper-layer-specific classification scheme would still have to provide a degree of generality in order to cope with future extensions of IPv6 that might wish to make use of some of the link layer services already provided.

Recommendations: Upper-layer-specific classification schemes should be avoided.

3.3. Inspection of Payload Contents

If a classification scheme utilizing higher-layer headers proposes to inspect the contents of the packet being encapsulated (e.g., IEEE 802.16 IP CS mechanisms for determining the connection identifier (CID) to use to transmit a packet), the fields available for inspection may be limited if the packet is compressed or encrypted before passing to the link layer. This may prevent the link layer from utilizing existing compression mechanisms, such as Van Jacobson Compression [RFC1144], ROHC [RFC3095][RFC3759], Compressed RTP (CRTP) [RFC2508], Enhanced Compressed RTP (ECRTP) [RFC3545], or IP Header Compression [RFC2507].

Recommendations: Link-layer classification schemes should not rely on the contents of higher-layer headers.

3.4. Interoperability Guidance

In situations where multiple encapsulation methods are operational and capable of carrying IP traffic, interoperability problems are possible in the absence of clear implementation guidelines. For example, there is no guarantee that other hosts on the link will support the same set of encapsulation methods, or that if they do, that their routing tables will result in identical preferences.

In IEEE 802.16, the Subscriber Station (SS) indicates the Convergence Sublayers it supports to the Base Station (BS), which selects from the list one or more that it will support on the link. Therefore, it is possible for multiple CSes to be operational.

Note that IEEE 802.16 does not provide multiple encapsulation methods for the same kind of data payload; it defines exactly one

encapsulation scheme for each data payload. For example, there is one way to encapsulate a raw IPv4 packet into an IEEE 802.16 MAC frame, one encapsulation scheme for a raw IPv6 packet, etc. There is also one way to encapsulate an Ethernet frame, even when there are multiple possibilities for classifying an Ethernet frame for forwarding over a connection identifier (CID). Since support for multiple CSes enables IEEE 802.16 to encapsulate layer 2 frames as well as layer 3 packets, IP packets may be directly encapsulated in IEEE 802.16 MAC frames as well as framed with Ethernet headers in IEEE 802.16 MAC frames. Where CSes supporting both layer 2 frames as well as layer 3 packets are operational on the same link, a number of issues may arise, including:

Use of Address Resolution Protocol (ARP)
: Where both IPv4 CS and Ethernet CS are operational on the same link, it may not be obvious how address resolution should be implemented. For example, should an ARP frame be encapsulated over the Ethernet CS, or should alternative mechanisms be used for address resolution, utilizing the IPv4 CS?

Data Frame Encapsulation
: When sending an IP packet, which CS should be used? Where multiple encapsulations are operational, multiple connection identifiers (CIDs) will also be present. The issue can therefore be treated as a multi-homing problem, with each CID constituting its own interface. Since a given CID may have associated bandwidth or quality-of-service constraints, routing metrics could be adjusted to take this into account, allowing the routing layer to choose based on which CID (and encapsulation) appears more attractive.

This could lead to interoperability problems or routing asymmetry. For example, consider the effects on IPv6 Neighbor Discovery:

(a) If hosts choose to send IPv6 Neighbor Discovery traffic on different CSes, it is possible that a host sending an IPv6 Neighbor Discovery packet will not receive a reply, even though the target host is reachable over another CS.

(b) Where hosts all support the same set of CSes, but have different routing preferences, it is possible for a host to send an IPv6 Neighbor Discovery packet over one CS and receive a reply over another CS.

Recommendations: Given these issues, it is strongly recommended that only a single kind of CS supporting a single encapsulation method should be usable on a particular link.

3.5. Service Consistency

If a link-layer protocol provides multiple encapsulation methods, the services offered to the IP-layer and upper-layer protocols may differ qualitatively between the different encapsulation methods. For example, the 802.16 [IEEE-802.16.2004] link-layer protocol offers both 'native' encapsulation for raw IPv4 and IPv6 packets, and Ethernet encapsulation. In the raw case, the IP layer can be directly mapped to the quality-of-service (QoS) capabilities of the IEEE 802.16 transmission channels, whereas using the Ethernet encapsulation, an IP-over-Ethernet CS has to be deployed to circumvent the mapping of the IP QoS to the Ethernet header fields to avoid the limitations of Ethernet QoS. Consequently, the service offered to an application depends on the classification method employed and may be inconsistent between sessions. This may be confusing for the user and the application.

Recommendations: If multiple encapsulation methods for IP packets on a single link-layer technology are deemed to be necessary, care should be taken to match the services available between encapsulation methods as closely as possible.

3.6. Implementation Complexity

Support of multiple encapsulation methods results in additional implementation complexity. Lack of uniform encapsulation support also results in potential interoperability problems. To avoid interoperability issues, devices with limited resources may be required to implement multiple encapsulation mechanisms, which may not be practical.

When encapsulation methods require hardware support, implementations may choose to support different encapsulation sets, resulting in market fragmentation. This can prevent users from benefiting from economies of scale, precluding some uses of the technology entirely.

Recommendations: Choose a single encapsulation mechanism that is mandatory to implement for both sending and receiving, and make that encapsulation mechanism the default for sending.

3.7. Negotiation

The complexity of negotiation within ARP or IP can be reduced by performing encapsulation negotiation within the link layer.

However, unless the link layer allows the negotiation of the encapsulation between any two hosts, interoperability problems can still result if more than one encapsulation is possible on a given

link. In general, a host cannot assume that all other hosts on a link support the same set of encapsulation methods, so that unless a link-layer protocol only supports point-to-point communication, negotiation of multiple potential encapsulation methods will be problematic. To avoid this problem, it is desirable for link-layer encapsulation negotiation to determine a single IP encapsulation, not merely to indicate which encapsulation methods are possible.

Recommendations: Encapsulation negotiation is best handled in the link layer. In order to avoid dependencies on the data frame encapsulation mechanism, it is preferable for the negotiation to be carried out using management frames, if they are supported. If multiple encapsulations are required and negotiation is provided, then the negotiation should result in a single encapsulation method being negotiated on the link.

3.8. Roaming

Where a mobile node roams between base stations or to a fixed infrastructure, and the base stations and fixed infrastructure do not all support the same set of encapsulations, then it may be necessary to alter the encapsulation method, potentially in mid-conversation. Even if the change can be handled seamlessly at the link and IP layer so that applications are not affected, unless the services offered over the different encapsulations are equivalent (see Section 3.5), the service experienced by the application may change as the mobile node crosses boundaries. If the service is significantly different, it might even require 'in-flight' renegotiation, which most applications are not equipped to manage.

Recommendations: Ensure uniformity of the encapsulation set (preferably only a single encapsulation) within a given mobile domain, between mobile domains, and between mobile domains and fixed infrastructure. If a link layer protocol offers multiple encapsulation methods for IP packets, it is strongly recommended that only one of these encapsulation methods should be in use on any given link or within a single wireless transmission domain.

4. Security Considerations

The use of multiple encapsulation methods does not appear to have significant security implications.

An attacker might be able to utilize an encapsulation method that was not in normal use on a link to cause a denial-of-service attack, which would exhaust the processing resources of interfaces if packets utilizing this encapsulation were passed up the stack to any significant degree before being discarded.

An attacker might be able to force a more cumbersome encapsulation method between two endpoints, even when a lighter weight one is available, hence forcing higher resource consumption on the link and within those endpoints, or causing fragmentation. Since IP fragments are more difficult to classify than non-fragments, this may result in packet loss or may even expose security vulnerabilities [WEP].

If different methods have different security properties, an attacker might be able to force a less secure method as an elevation path to get access to some other resource or data. Similarly, if one method is rarely used, that method is potentially more likely to have exploitable implementation bugs.

Since lower-layer classification methods may need to inspect fields in the packet being encapsulated, this might deter the deployment of end-to-end security, which is undesirable. Where encryption of upper layer headers (e.g., IPsec tunnel mode) is required, this may obscure headers required for classification. As a result, it may be necessary for all encrypted traffic to flow over a single connection.

5. Conclusion

The use of multiple encapsulation methods on the same link is problematic, as discussed above.

Although multiple IP encapsulation methods were defined on Ethernet cabling, recent implementations support only the Ethernet encapsulation of IPv4 defined in [RFC894]. In order to avoid a repeat of the experience with IPv4, for operation of IPv6 on IEEE 802.3 media, only the Ethernet encapsulation was defined in "A Method for the Transmission of IPv6 Packets over Ethernet Networks" [RFC1972], later updated in [RFC2464].

In addition to the recommendations given earlier, we give the following general recommendations to avoid problems resulting from use of multiple IP encapsulation methods:

When developing standards for encapsulating IP packets on a link-layer technology, it is desirable that only a single encapsulation method should be standardized for each link-layer technology.

If a link-layer protocol offers multiple encapsulation methods for IP packets, it is strongly recommended that only one of these encapsulation methods should be in use within any given link.

Where multiple encapsulation methods are supported on a link, a single encapsulation should be mandatory to implement for send and receive.

6. References

6.1. Normative Reference

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

6.2. Informative References

[DIX] Digital Equipment Corporation, Intel Corporation, and Xerox Corporation, "The Ethernet -- A Local Area Network: Data Link Layer and Physical Layer (Version 2.0)", November 1982.

[Generic] Wang, L. et al, "A Generic Packet Convergence Sublayer (GPCS) for Supporting Multiple Protocols over 802.16 Air Interface", Submission to IEEE 802.16g: CB0216g_05_025r4.pdf, November 2005, <http://www.ieee802.org/16/netman/contrib/C80216g-05_025r4.pdf>.

[IEEE-802.1A.1990] Institute of Electrical and Electronics Engineers, "Local Area Networks and Metropolitan Area Networks: Overview and Architecture of Network Standards", IEEE Standard 802.1A, 1990.

[IEEE-802.1D.2004] Institute of Electrical and Electronics Engineers, "Information technology - Telecommunications and information exchange between systems - Local area networks - Media access control (MAC) bridges", IEEE Standard 802.1D, 2004.

[IEEE-802.1Q.2003] IEEE Standards for Local and Metropolitan Area Networks: Draft Standard for Virtual Bridged Local Area Networks, P802.1Q-2003, January 2003.

[IEEE-802.3.2002] Institute of Electrical and Electronics Engineers, "Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications", IEEE Standard 802.3, 2002.

[IEEE-802.11.2003] Institute of Electrical and Electronics Engineers, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", IEEE Standard 802.11, 2003.

[IEEE-802.16.2004] Institute of Electrical and Electronics Engineers, "Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks, Part 16: Air Interface for Fixed Broadband Wireless Access Systems", IEEE Standard 802.16-2004, October 2004.

[IEEE-802.16e.2005] Institute of Electrical and Electronics Engineers, "Information technology - Telecommunications and information exchange between systems - Local and Metropolitan Area Networks - Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems, Amendment for Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands", IEEE P802.16e, September 2005.

[PROBE] Hewlett Packard, "A Primer on HP Probe", http://www.hp.com/rnd/support/manuals/pdf/hp_probe.pdf, July 1993.

[RFC826] Plummer, D., "Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48.bit Ethernet address for transmission on Ethernet hardware", STD 37, RFC 826, November 1982.

[RFC893] Leffler, S. and M. Karels, "Trailer encapsulations", RFC 893, April 1984.

[RFC894] Hornig, C., "A Standard for the Transmission of IP Datagrams over Ethernet Networks", STD 41, RFC 894, April 1984.

[RFC903] Finlayson, R., Mann, T., Mogul, J., and M. Theimer, "A Reverse Address Resolution Protocol", STD 38, RFC 903, June 1984.

[RFC948] Winston, I., "Two Methods for the Transmission of IP Datagrams over IEEE 802.3 Networks", RFC 948, June 1985.

[RFC1010] Reynolds, J. and J. Postel, "Assigned Numbers", RFC 1010, May 1987.

[RFC1042] Postel, J. and J. Reynolds, "Standard for the transmission of IP datagrams over IEEE 802 networks", STD 43, RFC 1042, February 1988.

[RFC1122] Braden, R., "Requirements for Internet Hosts -- Communication Layers", STD 3, RFC 1122, October 1989.

[RFC1144] Jacobson, V., "Compressing TCP/IP Headers for Low-Speed Serial Links", RFC 1144, February 1990.

[RFC1661] Simpson, W., "The Point-to-Point Protocol (PPP)", STD 51, RFC 1661, July 1994.

[RFC1958] Carpenter, B., "Architectural Principles of the Internet", RFC 1958, June 1996.

[RFC1962] Rand, D., "The PPP Compression Control Protocol (CCP)", RFC 1962, June 1996.

[RFC1972] Crawford, M., "A Method for the Transmission of IPv6 Packets over Ethernet Networks", RFC 1972, August 1996.

[RFC2472] Haskin, D. and E. Allen, "IP Version 6 over PPP", RFC 2472, December 1998.

[RFC2464] Crawford, M., "Transmission of IPv6 Packets over Ethernet Networks", RFC 2464, December 1998.

[RFC2507] Degermark, M., Nordgren, B., and S. Pink, "IP Header Compression", RFC 2507, February 1999.

[RFC2508] Casner, S. and V. Jacobson, "Compressing IP/UDP/RTP Headers for Low-Speed Serial Links", RFC 2508, February 1999.

[RFC2615] Malis, A. and W. Simpson, "PPP over SONET/SDH", RFC 2615, June 1999.

[RFC2684] Grossman, D. and J. Heinanen, "Multiprotocol Encapsulation over ATM Adaptation Layer 5", RFC 2684, September 1999.

[RFC3095] Bormann, C., Burmeister, C., Degermark, M., Fukushima, H., Hannu, H., Jonsson, L-E., Hakenberg, R., Koren, T., Le, K., Liu, Z., Martensson, A., Miyazaki, A., Svanbro, K.,

Wiebke, T., Yoshimura, T., and H. Zheng, "RObust Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed", RFC 3095, July 2001.

[RFC3241] Bormann, C., "Robust Header Compression (ROHC) over PPP", RFC 3241, April 2002.

[RFC3518] Higashiyama, M., Baker, F., and T. Liao, "Point-to-Point Protocol (PPP) Bridging Control Protocol (BCP)", RFC 3518, April 2003.

[RFC3544] Koren, T., Casner, S., and C. Bormann, "IP Header Compression over PPP", RFC 3544, July 2003.

[RFC3545] Koren, T., Casner, S., Geevarghese, J., Thompson, B., and P. Ruddy, "Enhanced Compressed RTP (CRTP) for Links with High Delay, Packet Loss and Reordering", RFC 3545, July 2003.

[RFC3759] Jonsson, L-E., "RObust Header Compression (ROHC): Terminology and Channel Mapping Examples", RFC 3759, April 2004.

[RFC4541] Christensen, M., Kimball, K., and F. Solensky, "Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches", RFC 4541, May 2006.

[WEP] Bittau, A., Handley, M., and J. Lackey, "The Final Nail in WEP's Coffin", Proceedings of the 2006 IEEE Symposium on Security and Privacy, pp. 386-400.

7. Acknowledgments

The authors would like to acknowledge Jeff Mandin, Bob Hinden, Jari Arkko, Max Riegel, Alfred Hoenes, and Phil Roberts for contributions to this document.

Appendix A. IAB Members at the Time of This Writing

Bernard Aboba
Loa Andersson
Brian Carpenter
Leslie Daigle
Elwyn Davies
Kevin Fall
Olaf Kolkman
Kurtis Lindqvist
David Meyer
David Oran
Eric Rescorla
Dave Thaler
Lixia Zhang

Authors' Addresses

Bernard Aboba
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

EMail: bernarda@microsoft.com
Phone: +1 425 706 6605
Fax: +1 425 936 7329

Elwyn B. Davies
Consultant
Soham, Cambs
UK

EMail: elwynd@dial.pipex.com
Phone: +44 7889 488 335

Dave Thaler
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

EMail: dthaler@microsoft.com
Phone: +1 425 703 8835

Full Copyright Statement

Copyright (C) The IETF Trust (2007).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.