IEEE P802.11
Wireless LANs

Integration Function Detailed Description				
Date: 2005-03-08				
Author(s):				
Name	**Company**	**Address**	**Phone**	**email**
Johnny Zweig	-	-	408-446-2248	Johnny@zweig.org
Darwin Engwer	Nortel Networks	4655 Great America Pkwy, Santa Clara, CA 95054	408-495-7099	dengwer@nortelnetworks.com

Abstract

A Portal in an 802.11 LAN performs the Integration Function to integrate the MAC SAPs within an 802.11 WLAN to the MAC SAPs within a non-802.11 LAN. This document gives details of the manipulations of MAC and LLC headers that a Portal performs when integrating an 802.11 LAN into an Ethernet/802.3 LAN. It should be noted that, as a result of these procedures, the format of **any** 802.11 MSDU carrying a network protocol whose format is defined on an Ethernet/802.3 LAN is precisely-defined (including Internet Protocol datagrams).

Introduction

In an 802.11 LAN, the Distribution System is the entity that distributes 802.11 MSDUs between MAC service endpoints. In an Infrastructure BSS, a Mobile Station will normally send and receive all MSDUs through the Access Point with which the STA is currently associated. All other MAC endpoints that are reachable on the LAN are accessed indirectly through the Access Point by means of the Distribution System. It should be noted that the Distribution System need not be a part of the 802.11 LAN itself. It may, for example, consist of a bridged 802.3 LAN that carries the 802.11 LAN's MSDUs in some appropriate format. An entity that integrates the 802.11 LAN into an external LAN is called a Portal. The Integration Function is responsible for making the services of the external LAN accessible to MAC endpoints in the 802.11 LAN (and vice versa).

The passage through a Portal onto the non-802.11 LAN, and subsequently through a peer Portal back onto its 802.11 LAN is transparent to MAC service users within the LAN. Therefore, all MAC endpoints on the 802.11 LAN(s) and external LAN(s) can reach one-another in the integrated WLAN system.

While there is typically a one-to-one relationship between an ESS and an (infrastructure) 802.11 LAN, it is important to keep the distinction in mind. The ESS defines a collection of Access Points that provide Mobile Stations access to the Distribution System. It is possible that multiple ESSs will be bridged together (i.e. connected to the same, possibly virtual, Distribution System), and it is also possible that the collection of MAC service endpoints reachable through different Access Points will vary from one part of the ESS to another. At any given moment, the LAN consists of all MAC service endpoints that could conceivably be reached by a given Mobile Station. The 802.11 LAN consists of those components of the LAN that are 802.11 components. The ESS consists of those parts of the 802.11 LAN that share the same SSID, and are thus meant to represent equivalent connectivity to the LAN.

Since the details of how higher-layer protocol endpoints are addressed differ between Ethernet/802.3 LANs and other IEEE 802-family LANs, it is necessary to look closely at how the Integration Function allows endpoints on an 802.11 LAN to communicate with peers on an integrated Ethernet/802.3 LAN.

Background

Two aspects of the development of 802.11 Wireless LAN products have influenced the format of MSDUs carrying IP packets on an 802.11 network. Firstly, many vendors sold Access Units that consisted of an 802.11 Access Point bridged to an Ethernet/802.3 LAN. Secondly, early device-driver software for 802.11 WLAN adapters sold for notebook computers (and other Mobile Units) often used an Ethernet Application Program Interface (API). That is, there was a substantial cohort of 802.11 device drivers that interfaced to the client's protocol stack via the same Ethernet-based interface that 802.3 LAN adapters had defined. Both of these practices necessitated bridges that would translate MSDUs that used an Ethernet Type Identifier (see below) into a format that preserved this type information on the non-Ethernet LAN.

An Ethernet/802.3 LAN has two distinct ways of addressing higher-layer protocol service access points (SAPs). An MSDU in 802.3 format carries a number between 0 and 1500 (decimal) encoded in the 13th and 14th octets of the MAC header. This field represents a length, and is normally followed by a Logical Link Control (LLC) header that identifies the SAP to which the MSDU should be delivered. An MSDU in Ethernet format carries a larger number in these two octets, and this field (called an Ethernet Type Identifier, or sometimes a Protocol Identifier) addresses a Network-layer SAP directly. Examples of this MSDU format are the Internet Protocol (which uses the value 0x0800) and IP Address Resolution Protocol (ARP) (which uses the value 0x0806).

The IEEE 802.1H recommended-practice document suggests a way of carrying Ethernet MSDUs on other 802 LAN technologies (without losing the higher-layer protocol identification), and is designed to solve a problem that resulted from having a number of LAN protocol vendors (in particular, AppleTalk) use the same encapsulation format that was used by RFC1042 bridges under certain circumstances.

Integrating Ethernet

The problem faced by a bridge between an Ethernet/802.3 LAN and an 802 LAN of a different variety is that Ethernet MSDUs contain a "protocol type" field that is widely used. In particular, the Internet Protocol and its Address Resolution Protocol both use Ethernet type fields when running on an 802.3 LAN. Since other 802 LANs do not have a type field, a method of encapsulating this type information in an 802.2 LLC header was developed.

The Sub-Network Access Protocol (SNAP) defines a header for LLC that contains a 3-octet OUI and a 2-octet OUI-specific identifier, to be used for selecting a (Layer 3) Service Access Point (don't confuse LLC's uses of "Access Point" with 802.11's) for further demultiplexing.

RFC1042 defines a technique for carrying the 2-octet Ethernet type field information inside a SNAP header using the OUI 00-00-00. This leads to a non-802.3 LAN MSDU that is 8 octets longer than the original 802.3 MSDU (if and only if the 802.3 MSDU used the type/length field as a protocol type rather than a frame-length indicator).

Unfortunately, two protocol vendors (Apple Computer and Novell Inc.) released protocol implementations (respectively) of the AppleTalk Phase 2 Address Resolution Protocol (AARP) and the Novell Internetwork Packet eXchange (IPX) protocol that used RFC1042-style SNAP headers on the 802.3 LAN. This created a problem for bridges that received what appeared to be RFC1042-encapsulated MSDUs from a non-802.3 LAN, since they would erroneously decapsulate and thus change the format of these AARP and IPX packets[1]. That is, an AARP packet that was sent on an 802.3 LAN would have its format changed if it passed through an RFC1042-style bridge onto a non-802.3 LAN and subsequently back onto an 802.3 LAN.

In essence, the RFC1042 SNAP header became ambiguous, since the bridge that is placing the non-802.3 MSDU onto the Ethernet/802.3 LAN has no way of knowing whether an MSDU with such a header is carrying an AppleTalk or IPX packet whose EtherType was encapsulated by the original bridge, or carrying an 802.3 MSDU that should be passed (with its SNAP header intact) onto the 802.3 LAN.

The Solution

IEEE recommended practice 802.1H defines a new encapsulation format for the Bridge Tunnel Encapsulation Protocol (BTEP). This protocol creates a way to tunnel encapsulated Ethernet MSDUs from one Ethernet LAN to another, across a non-802.3 LAN, without changing their format. That is, it creates a new format that specifically alleviates the ambiguity associated with RFC1042-encapsulation. A BTEP-encapsulated MSDU can carry the AARP/IPX Ethernet Type Identifier values in a different format, leaving the RFC1042-style headers to be left intact on both sides of the bridge.

Extending 802.1H

To keep implementations as simple as possible, 802.1H recommends only performing the BTEP when an AppleTalk Address Resolution Protocol packet is being bridged. That is, the Selective Translation Table (STT) contains only a single entry. The reason is that IPX equipment can be configured in such a way that the format change is benign. That is, an IPX client can generate frames with the SNAP header (which looks like, but is not, an RFC1042 header) and the IPX server on the bridged LAN can be configured to accept Ethernet-style headers.

A number of vendors of 802.11 WLAN systems found the onus of telling customers who use IPX that they need to reconfigure their networks in order to use 802.11 too burdensome. So these vendors used a 2-entry BTEP Selective Translation Table, rather than the 1-entry (AppleTalk only) table suggested in 802.1H. The result is that all protocols ran without modification on a LAN containing an arbitrary mix of Ethernet/802.3 and 802.11 LAN components, as long as all the bridges were using the same encapsulation protocols.

Protocol	Ethertype
AppleTalk ARP	0x80F3

[1] I use the term "packet" to refer to a Layer-3 protocol such as IP, IPX or AppleTalk. A packet is sent as an MSDU, so there is a one-to-one correspondence. But they are not, technically, the same thing.

Novell IPX	0x8137

Table 1 – NTI STT

Bridges and Formats

Because many 802.11 mobile adapters used Ethernet-based interfaces to their client device driver software, it was necessary to put a nearly-trivial Ethernet/802.3 to 802.11 bridge implementation inside the device driver. The bridging is nearly trivial because there is no decision necessary as to which bridge port an MSDU should be forwarded. However, an Ethernet-hosted protocol stack running on the mobile client contained such a bridge.

This resulted in 802.11 LANs all of whose Internet Protocol packets were in the RFC1042-style SNAP encapsulated format. This created a *de facto* standard format for carrying IP (or any other protocol that uses an Ethernet type value, such as ARP) packets on an 802.11 WLAN.

In fact, another way to think about the Integration Function is in terms of correspondence rather than translation. That is, any protocol (such as IP) whose format is defined for Ethernet has a corresponding format for 802.11 (and other IEEE 802) LANs. The translation rules define an unambiguous mapping from Ethernet to "802" MSDUs. So, for example, the format of an IP packet on an 802.11 LAN uses an LLC header that carries a SNAP subheader with an OUI of 00-00-00 and a SNAP identifier of 08-00. This results from the definition of IP over Ethernet.

The Specifics

The implementation of an RFC1042/802.1H-compliant bridge (using the NTI[2] Selective Translation Table shown above) uses a set of translation rules for examining Ethernet/802.3 MSDU headers and creating corresponding LLC headers for use on the non-802.3 LAN side of the bridge. There is a corresponding set of translation rules for bridging non-802.3 LAN (i.e. 802.11) MSDUs back onto the integrated Ethernet/802.3 LAN.

Ethernet/802.3 LAN to non-Ethernet LAN Encapsulation Rules

Frames whose Type/Length field has a value between 0x0000 and 0x05DC are interpreted as 802.3 frames, and are passed to the bridged LAN intact. (Note that this will include AppleTalk Phase 2 and certain Novell IPX/SPX frames.)

Other frames whose Type field is not in the NTI STT (that is, values other than 0x80F3 and 0x8137) are encapsulated using an RFC1042 SNAP header of the form 0xAA-AA-03-00-00-00-nn-mm, where "nn-mm" is the Ethernet Type field contents. (The RFC1042 OUI is 00-00-00.)

Other frames (namely those whose Ethernet Type field contains 0x80F3 or 0x8137) are encapsulated using a BTEP header of the form 0xAA-AA-03-00-00-F8-nn-mm where nn-mm is the value from the Ethernet frame's Type field. (The BTEP OUI is 00-00-F8.)

Non-Ethernet LAN to Ethernet/802.3 LAN Decapsulation Rules

Any frame whose SNAP header is a BTEP header (i.e. it begins with 0xAA-AA-03-00-00-F8) will be decapsulated into an Ethernet frame whose Type field is taken from the last two octets of the BTEP header.

A frame whose SNAP header is an RFC1042 header (i.e. it begins with 0xAA-AA-03-00-00-00), whose last two octets are <u>not</u> in the STT (i.e. any value other than 0x80F3 or 0x8137) is decapsulated into an Ethernet frame whose Type field is taken from the last two octets of the RFC1042 header.

A frame whose SNAP header is an RFC1042 header (i.e. it begins with 0xAA-AA-03-00-00-00), whose last two octets <u>are</u> in the STT (i.e. either 0x80F3 or 0x8137) is not decapsulated, but rather passed intact as an 802.3 frame (with appropriate length information filled in to the Type/Length field).

[2] The 2-entry STT was originally developed by Netwave Technologies, Inc. and was called the NTI STT for short.

All other frames are passed intact as 802.3 frames onto the integrated Ethernet/802.3 LAN.

Some Examples

Let's consider the examples of various types of packets traveling between an 802.11 LAN and a bridged Ethernet/802.3 LAN. Parts of the MAC headers, the LLC headers, and a few extra octets are shown in the following tables.

Ethernet/802.3 LAN to 802.11 LAN Encapsulation

Protocol	**Type/Length**	**LLC Header**		**802.11 LLC Header**
IP	08-00	--		AA-AA-03-00-00-00-08-00
IP 802.3[3]	length	AA-AA-03-00-00-00-08-00		AA-AA-03-00-00-00-08-00
IP ARP	08-06	--		AA-AA-03-00-00-00-08-06
AppleTalk (1)	80-9B	--		AA-AA-03-00-00-00-80-9B
AppleTalk (2)	length	AA-AA-03-08-00-07-80-9B		AA-AA-03-08-00-07-80-9B
AppleTalk AARP (1)	80-F3	--		AA-AA-03-00-00-F8-80-F3
AppleTalk AARP (2)	length	AA-AA-03-00-00-00-80-F3		AA-AA-03-00-00-00-80-F3
IPX Ethernet II	81-37	--		AA-AA-03-00-00-F8-81-37
IPX SNAP	length	AA-AA-03-00-00-00-81-37		AA-AA-03-00-00-00-81-37
IPX 802.2	length	E0-E0-03		E0-E0-03
IPX 802.3[4]	length	FF-FF		FF-FF

Table 2 – Ethernet/802.3 to 802.11 Translation

[3] This format of IP packet over 802.3 is denigrated, and the change (see next table) to the canonical Ethernet IP format is not considered harmful.

[4] The use of this format happens to work with these rules, even though the FF-FF is not actually a valid LLC header value. (The broadcast LSAP is not valid as a source SAP in LLC. See 802.2.)

802.11 LAN to Ethernet/802.3 LAN Decapsulation

Protocol	802.11 LLC Header		Type/Length	802.3 LLC Header
IP	AA-AA-03-00-00-00-08-00		08-00	--
IP 802.3[5]	AA-AA-03-00-00-00-08-00		08-00	--
IP ARP	AA-AA-03-00-00-00-08-06		08-06	--
AppleTalk (1)	AA-AA-03-00-00-00-80-9B		80-9B	--
AppleTalk (2)	AA-AA-03-08-00-07-80-9B		length	AA-AA-03-08-00-07-80-9B
AppleTalk AARP (1)	AA-AA-03-00-00-F8-80-F3		80-F3	--
AppleTalk AARP (2)	AA-AA-03-00-00-00-80-F3		length	AA-AA-03-00-00-00-80-F3
IPX Ethernet II	AA-AA-03-00-00-F8-81-37		81-37	--
IPX SNAP	AA-AA-03-00-00-00-81-37		length	AA-AA-03-00-00-00-81-37
IPX 802.2	E0-E0-03		length	E0-E0-03
IPX 802.3	FF-FF		length	FF-FF

Table 3 – 802.11 to Ethernet/802.3 Translation

References:

1. IEEE Std 802.1H, 1997 Edition.
2. Inside AppleTalk®, 2nd Edition. Gursharan S. Sidhu, Richard F. Andrews, Alan B. Oppenheimer. Addison-Weslet Publishing Company, 1990. ISBN 0-201-55021-0.
3. Novell's Guide to NetWare® LAN Analysis, 2nd Edition. Laura A. Chappell, Dan E. Hakes. Novell Press, 1994. ISBN 0-7821-1362-1.

[5] Note that this format of IP packet does not survive the trip across the non-802.3 LAN intact.