The Internet Protocol Journal

September 2006 Volume 9, Number 3

A Quarterly Technical Publication for Internet and Intranet Professionals

In This Issue

You can download IPJ back issues and find subscription information at: **www.cisco.com/ipj**

From The Editor

One of the most successful networking technologies of recent years has been IEEE 802.11 or, as it is commonly known, "Wi-Fi." Wireless networks have seen widespread deployment within organizations as well as in public "hotspots" all over the world. As a frequent traveler, I am very pleased with this development. It has been a long time since I had to resort to a modem and phone line in order to access e-mail or use the Web. Wireless networks have truly changed the way we use the Internet. Our first article, by T. Sridhar, explores the emerging use of *Wireless LAN Switches* in wireless access networks.

IPv6 is a technology that perhaps should have been widely deployed by now, but wide deployment has not happened yet, for numerous reasons. This journal has covered many aspects of IPv6. This time, Iljitsch van Beijnum looks at some of the details you need to be aware of when considering a move to IPv6. The article is adapted from his book *Running IPv6,* which was reviewed in our December 2005 issue.

In previous editions of IPJ we have pointed you to other sources of information, such as *The IETF Journal,* Geoff Huston's *ISP Column,* and other documents available from the Internet Society Website at **http://www.isoc.org**. This time I want to make you aware of an article that originally appeared in *Apster*, the newsletter of the *Asia Pacific Network Information Centre* (APNIC), one of the five *Regional Internet Registries* (RIRs). The article is entitled "IP Addressing in China and the Myth of Address Shortage," and you will find the URL for it in our "Fragments" section. If you want to further explore the work of the RIRs, you can start by visiting the *Number Resource Organization* (NRO) at **http://nro.net**.

You may have read that both of our sister publications, *Packet* and *IQ Magazine,* are publishing their final issues this September. Naturally, this has led to some of our readers asking what is in store for IPJ. We want to reassure you that we intend to continue publishing IPJ in both its paper and online forms. Plans are also under way to enhance our Website to provide you with more tools and resources. If you have suggestions for the Website, please send us a note at **ipj@cisco.com**.

—Ole J. Jacobsen, Editor and Publisher
ole@cisco.com

Wireless LAN Switches — Functions and Deployment

by T. Sridhar, Flextronics

Deployment of *Wireless LAN* (WLAN) switches is increasing in enterprise networks. These devices, which can be standalone switches or integrated into a blade on an enterprise class switch, are useful for the management and control of WLAN access points. Although their deployment is a relatively new phenomenon, such control and configuration functions have existed before in WLAN controller devices.

WLAN switches connect to the WLAN *access points* (APs) through wired connections (through a switch port). They also connect to the enterprise network through their other switch ports. The switches are the "gateway" to the wired enterprise—all frames from WLAN clients have to pass through the WLAN switches to the enterprise network.

To understand the motivation for WLAN switches and their operation in the network, it is useful to view the WLAN network architecture and the functions of the access points. We can view the WLAN switch as the control function and the APs as the wireless termination function.

This article presents the function of WLAN switches and controllers by detailing WLAN network architectures along with functions of the AP and controller. It also presents the various functions on the controller to AP interface. Subsequently, it outlines variables related to Layer 2/3 mobility in the centralized architecture and concludes by presenting some common myths and reality about these architectures.

This article uses the term *Wireless Termination Point* (WTP) to refer generically to APs and the term *Access Controller* (AC) to refer generically to the WLAN control function (whether implemented on a WLAN switch or standalone controller).

WLAN Network Architectures

Three types of WLAN network architectures are commonly deployed:

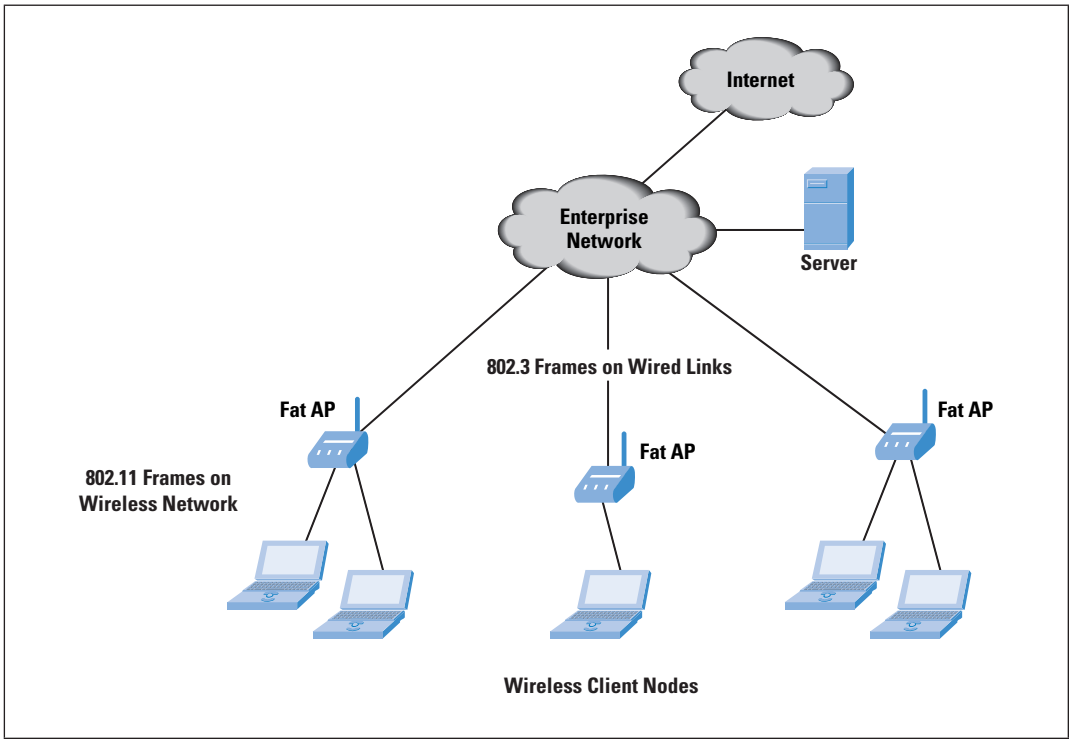
1. Autonomous Architecture
2. Centralized Architecture
3. Distributed Architecture

The following sections describe these architectures in greater detail.

Autonomous Architecture

In the autonomous architecture, the WTPs completely implement and terminate the 802.11 function so that frames on the wired LAN are 802.3 frames. Each WTP can be independently managed as a separate network entity on the network. The access point in such a network is often called a "Fat AP" (see Figure 1).

Figure 1: FAT APs in Autonomous WLAN Network Architecture

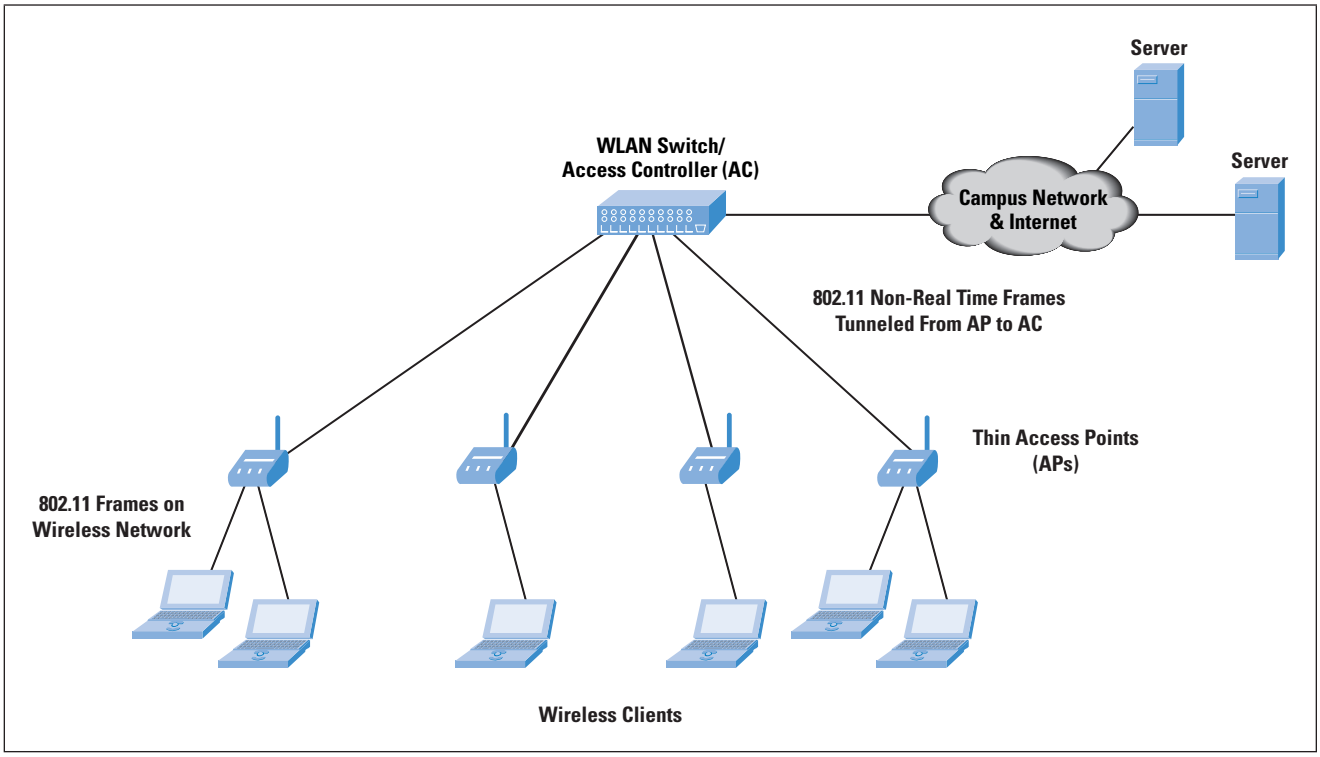


During the initial stages of WLAN deployment, most APs were autonomous APs, and manageable as independent entities in the network. During the past few years, centralized architectures (discussed next) with ACs and WTPs have gained popularity. The primary advantage of the centralized architecture is that it provides network administrators with a structured and hierarchical mode of control for multiple WTPs in the enterprise.

Centralized Architecture

The centralized architecture is a hierarchical architecture that involves a WLAN controller that is responsible for configuration, control, and management of several WTPs. The WLAN controller is also known as the *Access Controller* (AC). The 802.11 function is split between the WTP and the AC. Because the WTPs in this model have a reduced function as compared to the autonomous architecture, they are also known as "Thin APs." Some of the functions on the APs are variable, as discussed in the following section (see Figure 2).

Figure 2: Thin APs in Centralized WLAN Network Architecture



Distributed Architecture

In the distributed architecture, the various WTPs can form distributed networks with other WTPs through wired or wireless connections. A mesh network of WTPs is one example of such an architecture. The WTPs in the mesh can be linked with 802.11 links or wired 802.3 links. This architecture is often used in municipal networks and other deployments where an "outdoor" component is involved. This article does not address the distributed architecture.

WTP Functions – Fat, Thin, and Fit APs

To understand the autonomous and centralized architecture, it is useful to look at the functions performed by the APs. We start with the Fat APs, which form the core of the autonomous architecture, followed by the Thin APs, which were specified as part of the WLAN switch- or controller-based centralized architecture. The article will then outline the functions of a new variant called the "Fit AP," an optimized version of the AP for centralized architectures.

Fat Access Points

Figure 1 shows an example of an autonomous network with a fat access point. The AP is an addressable node in the network with its own IP address on its interfaces. It can forward traffic between the wired and wireless interfaces. It can also have more than one wired interface and can forward traffic between the wired interfaces—similar to a Layer 2 or Layer 3 switch. Connectivity to the wired enterprise can be through a Layer 2 or Layer 3 network.

It is important to understand that there is no "backhauling" of traffic from the Fat AP to another device through tunnels. This aspect is important and is addressed when discussing the other AP types. In addition, Fat APs can provide "router-like" functions such as the *Dynamic Host Configuration Protocol* (DHCP) server capabilities.

Management of the AP is done through a protocol such as the *Simple Network Management Protocol* (SNMP) or the *Hypertext Transfer Protocol* (HTTP) for Web-based management and a *Command-Line Interface* (CLI). To manage multiple APs, the network manager has to connect to each AP through one of these management schemes. Each AP shows up on the network map as a separate node. Any aggregation of the nodes for management and control has to be done at the *Network Management System* (NMS) level, which involves development of an NMS application.

Fat APs also have enhanced capabilities such as *Access Control Lists* (ACLs), which permit filtering of traffic for specific WLAN clients. Another significant capability of these devices is configuration and enforcement of *Quality of Service* (QoS)-related functions. For example, traffic from specific mobile stations might need to have a higher priority than others. Or, you might need to insert and enforce IEEE 802.1p priority or *Differentiated Services Code Point* (DSCP) for traffic from mobile stations. In summary, these APs act like a switch or router in that they provide many of the functions of such devices.

The downside of such APs is complexity. Fat APs tend to be built on powerful hardware and require complex software. These devices are expensive to install and maintain because of the complexity. Nevertheless, the devices have uses in smaller network installations.

Some Fat AP installations still use a controller at the back end for control and management functions. These controllers lead to a slightly scaled-down version of the Fat AP, called, not surprisingly, a Fit AP, discussed later.

Thin Access Points

As their name indicates, Thin APs are intended to reduce the complexity of APs. An important motivation for this reduction is the location of APs. In several enterprises, APs are plenum-mounted (and thus in hard-to-reach areas) so that they can provide optimum radio connectivity for end stations. In environments like warehouses, this is even more evident. For such reasons, network managers prefer to install APs just once and not have to perform complex maintenance on them.

Thin APs are often known as "intelligent antennas," in that their primary function is to receive and transmit wireless traffic. They backhaul the wireless frames to a controller where the frames are processed before being switched to the wired LAN (see Figure 2).

The APs use a (typically secure) tunnel to backhaul the wireless traffic to the controller. In their most basic form, Thin APs do not even perform WLAN encryption such as *Wired Equivalence Privacy* (WEP) or *WiFi Protected Access* (WPA/WPA2). This encryption is done at the controller—the APs just transmit or receive the encrypted wireless frames, thereby keeping the APs simple and avoiding the necessity to upgrade their hardware or software.

The introduction of WPA2 necessitated encryption on the controller. Although WPA was hardware-compatible with WEP and required only a firmware upgrade, WPA2 was not backward-compatible. Instead of replacing APs across the enterprise, network managers could just backhaul the wireless traffic to the controller where the WPA2 decryption was done, and the frames were sent on the wired LAN.

The protocol between the AP and the controller for carrying the control and data traffic was proprietary. Also, there is no capability to manage the AP as a single entity on the Layer 2/3 network—it can be managed only through the controller, to which the NMS can communicate through HTTP, SNMP, or CLI/Telnet. A controller can manage and control multiple APs, implying that the controller should be based on powerful hardware and often be able to perform switching and routing functions. Another important requirement is that the connectivity and tunnel between the AP and the AC should ensure low delay for packets between those two entities.

With Thin APs, QoS enforcement and ACL-based filtering are handled at the controller—not a problem because all the frames from the AP have to pass through the controller anyway. Centralized control functions for ACLs and QoS are not new—they were implemented in networks with Fat APs too. Such installations have controllers that act as the gateway for managing traffic from APs to the wired network. However, the controller function takes on a new dimension with Thin APs, especially with respect to the data plane and forwarding functions. The controller function subsequently was integrated into Ethernet switches that connected the wireless and wired LANs—the motivation for the family of devices known as WLAN switches.

The Wireless MAC architecture in this scenario is known as the *Remote MAC* architecture. The entire set of 802.11 MAC functions is offloaded to the WLAN controller, including the delay-sensitive MAC functions.

Fit Access Points

Fit APs are gaining in popularity in that they try to take advantage of the best of both worlds—that is, the Fat APs and the Thin APs. A Fit AP provides the wireless encryption while using the AC for the actual key exchange. This approach is used for newer APs that use the latest wireless chipsets supporting WPA2. The management and policy functions reside on the controller that connects to multiple APs through tunnels.

Also, Fit APs provide additional functions such as DHCP relay for the station to obtain an IP address through DHCP. In addition, Fit APs can perform functions such as VLAN tagging based on the *Service Set Identifier* (SSID) that the client uses to associate with the AP (when the AP supports multiple SSIDs).

Two types of MAC implementations are possible with Fit APs, known as the *Local MAC* and the *Split MAC* architectures. Local MAC is where all the wireless MAC functions are performed at the AP. The complete 802.11 MAC functions, including management and control frame processing, are resident on the APs. These functions include time-sensitive functions (also known as *Real Time MAC* functions).

The Split MAC architecture divides the implementation of the MAC functions between the AP and the controller. The real-time MAC functions include functions such as beacon generation, probe transmission and response, control frame processing (for example *Request to Send* and *Clear to Send*—RTS and CTS), retransmission, and so on. The non-real time functions include authentication and deauthentication; association and reassociation; bridging between Ethernet and Wireless LAN; fragmentation; and so on.

Vendors differ in the type of functions that are split between the AP and the controller, and in some cases, even about what constitutes real time. One common implementation of a Fit AP involves local MAC at the AP and control and management functions at the AP.

Access Controller and Control Functions

The next critical component of the Centralized WLAN Architecture is the *Access Controller* (AC). For the following discussion, we consider the controller function to be implemented on a WLAN switch and call the function an AC. We also use the term "WTP" to refer to APs (fat, thin, or fit).

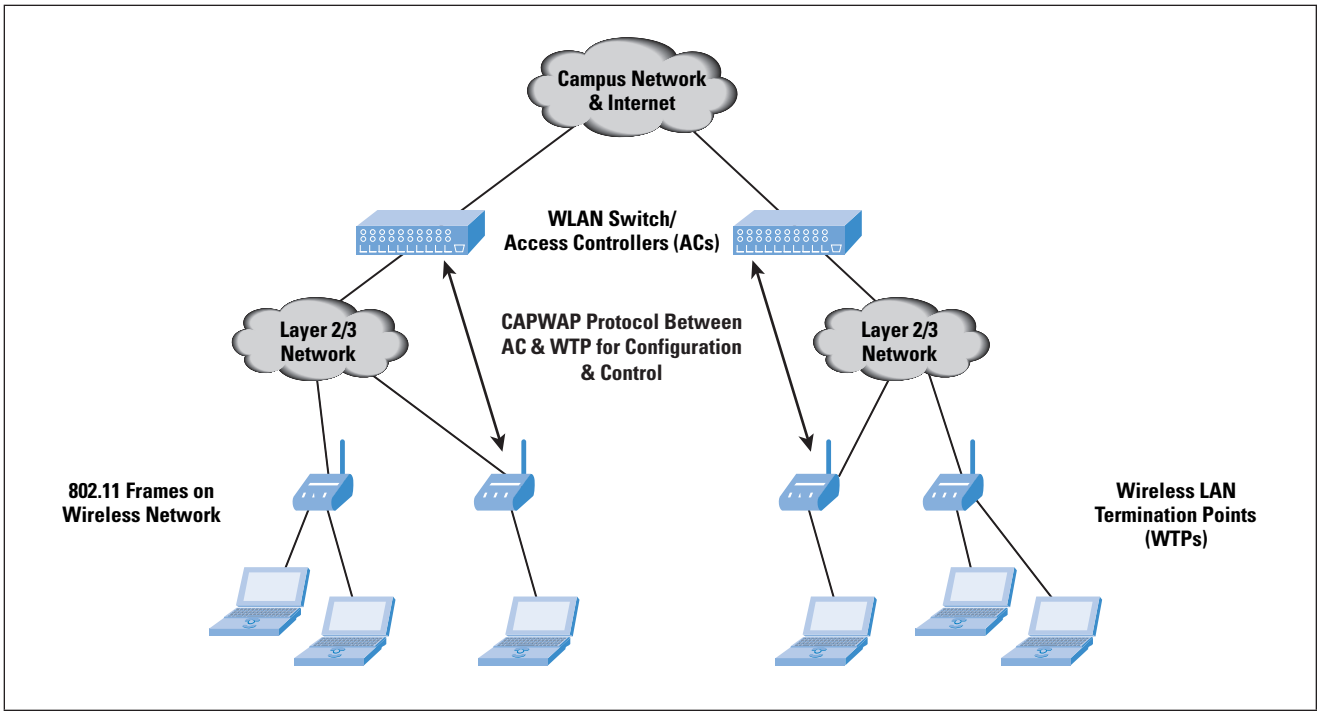
The *Control and Provisioning of Wireless Access Points* (CAPWAP) Working Group in the IETF is working on defining the interface and protocol between an AP and its controlled WTP. This section uses the CAPWAP framework to detail the interface between the AC and the WTP. [3,4,5]

Figure 3 shows an enterprise network with multiple ACs and WTPs. The WTPs can be connected to the ACs through a Layer 2 (switched) or Layer 3 (routed) network. The interface between the WTP and the AC is responsible for the following:

- Discovery and selection of an AC by WTP
- Firmware download to the WTP by the AC—upon startup and upon triggering by the WTP
- Capabilities negotiation between the WTP and the AC

- Mutual authentication between the WTP and the AC
- Configuration, status, and statistics exchange between the WTP and the AC
- QoS mapping across the wired and wireless segments

Figure 3: Centralized WLAN Architecture with Multiple ACs, WTPs and CAPWAP Protocol Context



In addition, although CAPWAP does not explicitly define all the details, the AC performs functions such as *Radio Resource Management* (RRM) and rogue AP detection based on configuration and monitoring of the various access points in its domain of control. The extent of these functions varies according to the vendor implementation. Another important function provided by ACs is mobility management. The following sections provide more detail about these functions, with specific reference to CAPWAP. Note that the CAPWAP protocol, which is based on the Cisco *Lightweight Access Point Protocol* (LWAPP), is still under development in the IETF, as of the writing this article (March 2006).

Discovery and AC Selection

A WTP discovers an AC to connect to through discovery request messages, to which one or more ACs can respond (depending on the network topology). Communication between the AC and the WTP is through the *User Datagram Protocol* (UDP). The WTP determines which AC to connect to and then tries to establish a secure session with the AC. Subsequent CAWAP packets are sent over the secure session.

Subsequently a configuration exchange takes place between the AC and WTP. This exchange includes:

- IEEE SSID
- Security parameters (for WEP, WPA, and WPA2)
- Data rate that is to be advertised (11 or 54 Mbps)
- Radio channels to be used

CAPWAP Functions

CAPWAP control messages include the following message types:

- Discovery
- WTP configuration—used to push a specific configuration to the WTP and also to retrieve statistics from a WTP; statistics includes information such as:
  - Number of fragmented frames, multicast frames transmitted and received
  - Number of transmit retries, excessive retries (failed count)
  - Number of successfully transmitted and failed *Requests to Sends* (RTS)
  - Number of errored frames: duplicate frames, failed acks, decryption errors, *frame-check-sequence* (FCS) error count, etc.

Configuration includes information such as beacon period, maximum transmit power level, *Orthogonal Frequency Division Multiplexing* (OFDM) control, antenna control, supported rates, QoS, encryption, and so on.

- *Mobile session management*—to push specific mobile policies to the WTP

  ACs can add policy information about specific mobile devices that can include security parameters that the WTP should apply for that mobile device. It can indicate whether the WTP should forward or discard traffic for that mobile device.

- *Firmware management*—used to push a specific firmware image to the WTP

AC and WTP Interaction

The WTP provides information such as hardware, software, or boot version; maximum number of radios; radios in use; encryption capabilities; type of radio (802.11b/g/a/n); type of MAC (local, split, or both); tunneling modes; and frame type between AC and WTP (for example, local bridging or native bridging—that is, encapsulating all user payloads as native wireless frames).

The AC information includes hardware or software version, number of mobile stations currently associated with the AC, number of WTPs currently attached to the AC, maximum numbers for each of these, security parameters (authentication credentials) between AC and WTP, control IPv4 or IPv6 address, and so on.

Because the WTPs fall under the category of "Fit APs," they can also be configured with an IP address from the AC. Another parameter that can be configured is ACLs at the MAC address level.

Rebooting (reset) of the WTP can be done by the AC at any time. Independently, the WTP can request a new image through an *Image Data Request*, which is followed by an *Image Data Response* and the image data itself.

Events are sent by the WTP when it determines that it has important information to send to the AC. Such information can include data transfer messages that can be used to deliver debug information from the WTP to the AC.

Radio Resource Management

Radio resource management is a generic term used to describe the control and configuration of radios on the AP. The type of control includes reducing and increasing the strength automatically or on user input—for example, if two WTPs controlled by an AC are interfering with each other, the AC can send a signal to one of the APs to reduce its strength. It can also do this based on user configuration.

Several WTPs are designed to also be used as "Air Monitors;" that is, they can monitor channels when not transmitting. Opinion is still divided on whether this mode of using WTPs is efficient—some vendors use dedicated air monitors instead of having their WTPs do double duty. With dedicated air monitors, it is much easier to scan and monitor all channels without having to worry about degrading the service for client stations.

Air monitors can forward information about other access points to the AC. The AC can determine if the information is for a valid WTP (that is, one that is supposed to be on the network and has, in fact, registered with the AC) or for a "rogue" access point. If it is for a rogue access point, the AC can perform multiple steps to prevent clients from attaching to this AP—for example, it can instruct the air monitor to "jam" this rogue AP by increasing the transmit power on the same channel.

Mobility Management

Mobility management can take two forms—Layer 2 and Layer 3 mobility. Consider a client moving from one WTP to another, a scenario that can happen when a user with a laptop moves between two conference rooms within the same building. The client station reassociates with the new WTP, after which authentication is performed. Note that the association with the previous AP is "broken" before the association with the new AP is "made;" thus handoff in WLANs is known as "break before make." Although this approach can lead to potential traffic disruption (and retransmissions), it is chosen over "make before break" (used in cellular telecommunications) to keep the client radio simple and less expensive.

One way to envision Layer 2 and Layer 3 mobility is to treat Layer 2 mobility as movement between APs under the control of the same AC (that is, Layer 3 network), whereas Layer 3 mobility is movement between APs under the control of different ACs.

Layer 2 Mobility

Layer 2 mobility means that when the station moves from one WTP to another, there is no impact on the IP addressability, effectively meaning that all the APs are on the same Layer 2 network and implying that they are connected to the same AC (see Figure 4). To prevent loss of data destined to the Layer 2 client, the WLAN switch must now forward client data to the new WTP. After the client association, the new WTP sends out an Ethernet frame to the AC with the client's MAC address as the source address. The switch now associates the client's MAC address to the port on which the new WTP is connected.

Although this process works well with Layer 2 (switched network) connectivity between the APs and the AC, it requires a slightly different approach when tunnels are used between them. The AC moves the mapping of the client to a different tunnel (that is, a virtual port) when it receives the MAC frame from the new WTP.

Another concern to be addressed with Layer 2 handoff is the buffering of data at the WTP. In normal circumstances, the switch or AC is not aware of the handoff until it hears from the new WTP. However, with enhanced statistics available at the WTP, it can determine that the specific client has moved away from the old WTP and stop forwarding data to the old WTP. These statistics can include maximum retry attempts on the *Carrier Sense Multiple Access/Collision Avoidance* (CSMA/CA) MAC layer protocol on the wireless link. The switch does not need to buffer the data because it is not clear when the handoff to the new WTP will occur. This approach helps avoid wasteful traffic on the link between the old WTP and the AC.

Some vendors have approached this problem differently with Fat APs. There, the APs might buffer the traffic until they see a frame from the switch indicating that the client is now on a different switch port. These APs then send the buffered traffic to the switch, which forwards that to the new WTP. Because our intent is to lower the complexity of the WTPs, this approach is not a preferred one in the Centralized AC + WTP architecture.

Another important feature of Layer 2 roaming is preauthentication that needs to be done on the new WTP. Through 802.11i, clients can preauthenticate with neighboring WTPs so that roaming to a different WTP does not involve the lengthy authentication process of *Pairwise Master Keys* (PMKs) being sent to the new WTP. (The *Pairwise Transient Keys* (PTKs) still need to be derived.)

When the AC maintains the PMK for a specific client (through interaction with a RADIUS server), this process is automatic—that is, the AC can send the client-specific PMK to the new WTP. The encryption of 802.11 frames is still done by the old and new WTPs with the new PTKs.

Layer 3 Mobility

Layer 3 mobility involves the client retaining the same IP address while moving across multiple APs. This often happens when the client has published its IP address to multiple nodes. Such a scenario is likely in peer-to-peer communications and when the mobile station needs to act as a server for some function. It is desirable that the correspondent nodes communicating with the mobile node not have to change their configuration whenever the mobile node moves to a new Layer 3 network.

This problem of Layer 3 mobility is solved by *Mobile IP*[6]. We do not discuss the details of Mobile IP here except to indicate that it has three distinct components. The *Home Agent* (HA) on the client's home network is responsible for the address of the client. All packets destined to the client's (invariant) IP address are sent to the Home Agent. If the client is on the home network, the HA forwards the packets directly to the client. If it is on a foreign or visited network, the HA forwards the packets to a *Foreign Agent* (FA) that is on the visited network.

To do this, it has to set up a tunnel to the FA—which is usually a *Generic Routing Encapsulation* (GRE) or IP-in-IP tunnel.

After stripping out the original packet from the tunnel, the FA is responsible for forwarding the packet to the client. This description is a simplification—numerous other steps are involved here. The important factor in a wireless LAN scenario for Layer 3 client mobility is where the Mobile IP endpoint resides. Some client stations include a software stack for a MIP client.

This *Client MIP* (CMIP) software:

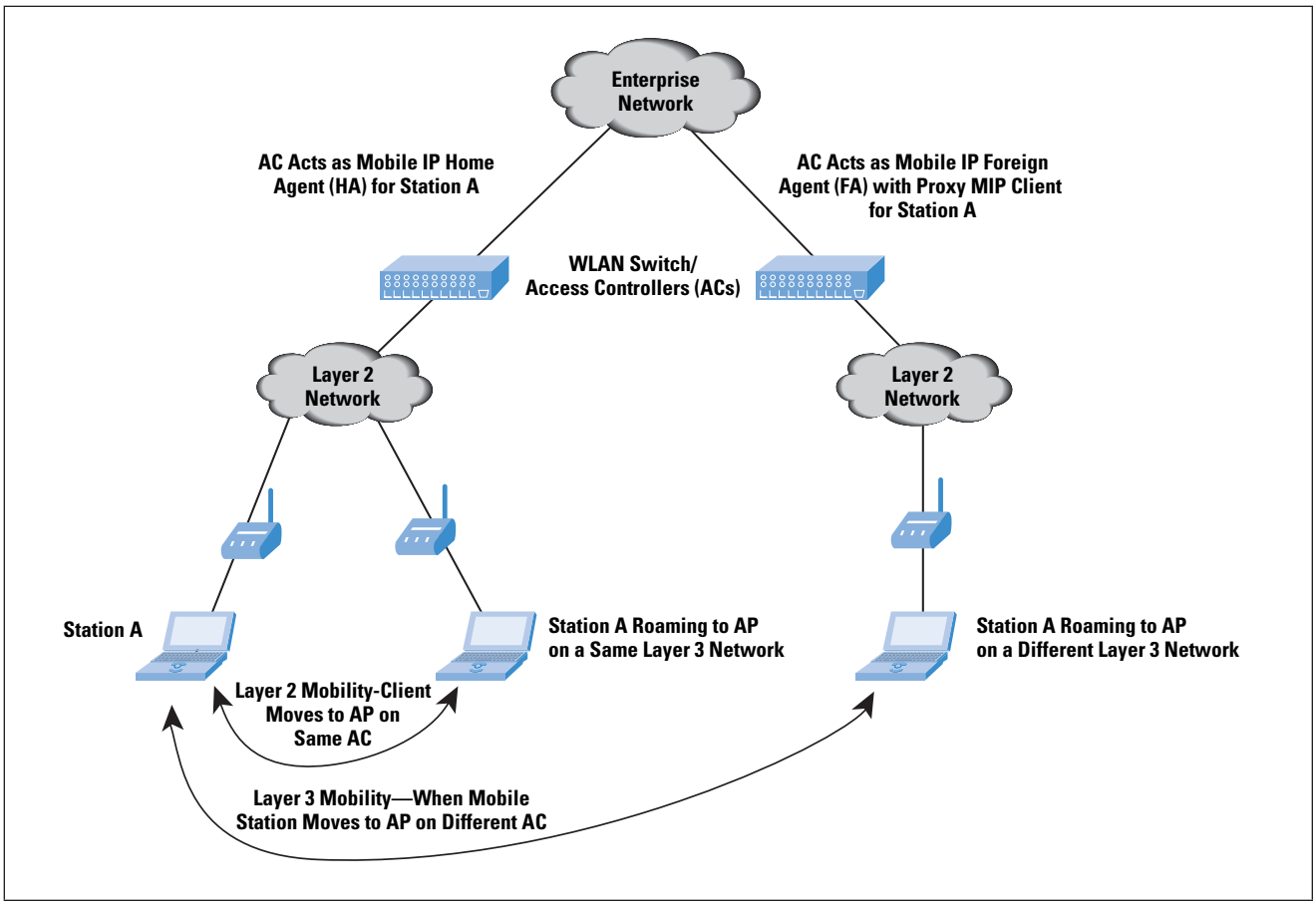
- Strips out the MIP header in the packet
- Inserts a new header to spoof the client's higher-layer applications into believing that the packets were destined for the client's IP address on the foreign network

The CMIP approach was the recommended approach for implementing MIP. However, it has the disadvantage of having to add a MIP client to every mobile station in the network—a setup that can become cumbersome when there are a large number of mobile stations.

The Centralized AC + WTP architecture offers a way of alleviating this problem. Some AC/WLAN switch vendors have implemented the MIP function on the AC so that the client never needs to be changed. Some implementations call this a *Proxy MIP* function.

The AC acts as an FA to terminate the tunnel from the HA and also performs the translation of the packets to the client's address on the visited network when forwarding packets to the client. When the client sends Layer 3 packets out, it sends them through the AC, which, in turn, modifies the headers for the source IP address and tunnels the packets to the HA. This process is called "reverse tunneling" (see Figure 4).

Figure 4: Layer 2 and Layer 3 Mobility in Centraliized WLAN Network Architecture



When you consider a large enterprise network topology with multiple ACs and APs, you can envision the MIP tunnels to be established between the various ACs. (That is, they act as Foreign Agents for one set of users and as Home Agents for another set.) From a scalability perspective, it is important that the ACs have the necessary horsepower and switching capability (switching between tunnels from the APs to the ACs to the tunnels between the ACs).

WLAN Switches and Centralized Architectures – Common Myths

Previous sections considered various aspects of the Centralized AC + WTP architecture and some of the implementation factors. This section outlines some common myths about these architectures and implementations. The intent is to examine this still-evolving area to facilitate clarity.

1. *Myth 1: ACs need to perform switching functions—hence the name WLAN switches.*

   There is no such requirement. In fact, the earliest ACs were appliances (and in some cases, PCs running Linux). The control function is the important part of the implementation—the switching is often included to accelerate the forwarding of traffic to and from the APs.

2. *Myth 2: Rogue WTP detection is a standard function of ACs.*

   This is a desired function in several implementations but is not necessarily "standard." One reason is that this is an area of differentiation among vendors (for example, the algorithms they use to classify a WTP as a rogue WTP). Another reason is that the ACs have to rely on APs or air monitors, and this reliance varies according to implementation.

3. *Myth 3: The delineation between Fat, Thin, and Fit APs is clearly defined.*

   There are several types of implementations of AP (and AC) functions, so this myth is not necessarily true. For a sample of the taxonomy (snapshot) of WTP and AC implementations, see RFC 4118[4].

4. *Myth 4: Layer 2 and Layer 3 mobility are standard in AC + WTP architectures.*

   This is not really true. The Proxy MIP implementation for Layer 3 mobility is a step in this direction, but most AC vendors rely on proprietary mechanisms for AC-AC communication and Layer 3 mobility.

5. *Myth 5: Security functions such as firewall, intrusion detection, and so on are not a function of ACs.*

   Some vendors have debunked this argument and implemented such functions in their AC. This is an area for vendor differentiation.

Summary

This article has provided the functions and deployment of WLAN switches by detailing the architectures that rely on a centralized controller managing a set of wireless termination points. It outlined some major aspects of the CAPWAP control functions and the concerns related to Layer 2 and Layer 3 mobility while implementing an AC + WTP architecture. Although protocol standardization is being done in the IETF for this emerging area, there is still sufficient scope for vendor differentiation.

References

[1] "IEEE 802.11i and Wireless Security," David Halasz, **www.embedded.com**, August 25, 2004.

[2] Rich Seifert, *The Switch Book: The Complete Guide to LAN Switching Technology,* ISBN 0471345865, Wiley, 2000.

[3] B. O'Hara, et al., "Configuration and Provisioning for Wireless Access Points (CAPWAP): Problem Statement," RFC 3990, February 2005.

[4] L. Yang, et al., "Architecture Taxonomy for Control and Provisioning of Wireless Access Points (CAPWAP)," RFC 4118, June 2005.

[5] P. Calhoun, Editor, "CAPWAP Protocol Specification," (work in progress), Internet Draft, **draft-ietf-capwap-protocol-specification-00**, February 24, 2006.

[6] C. Perkins, Editor,"IP Mobility Support for IPv4," RFC 3344, August 2002.

[7] Edgar Danielyan, "IEEE 802.11," *The Internet Protocol Journal,* Volume 5, No. 1, March 2005.

[8] Gregory R. Scholz, "Securing Wireless Networks," *The Internet Protocol Journal,* Volume 5, No. 3, September 2002.

T. SRIDHAR is Vice President of Technology at Flextronics in San Jose, California. He received his BE in Electronics and Communications Engineering from the College of Engineering, Guindy, Anna University, Madras, India, and his Master of Science in Electrical and Computer Engineering from the University of Texas at Austin. He can be reached at **T.Sridhar@flextronics.com**

IPv6 Internals

by Iljitsch van Beijnum

This article discusses some of the protocol details you should be aware of when planning a transition from IPv4 to IPv6. Although it is not intended as a complete step-by-step guide, this article explains the differences between IPv4 and IPv6 as they relate to actually operating a network. Vendor- and operating system-specific details can be found in the book from which this text was adapted, and further information is available in the references.

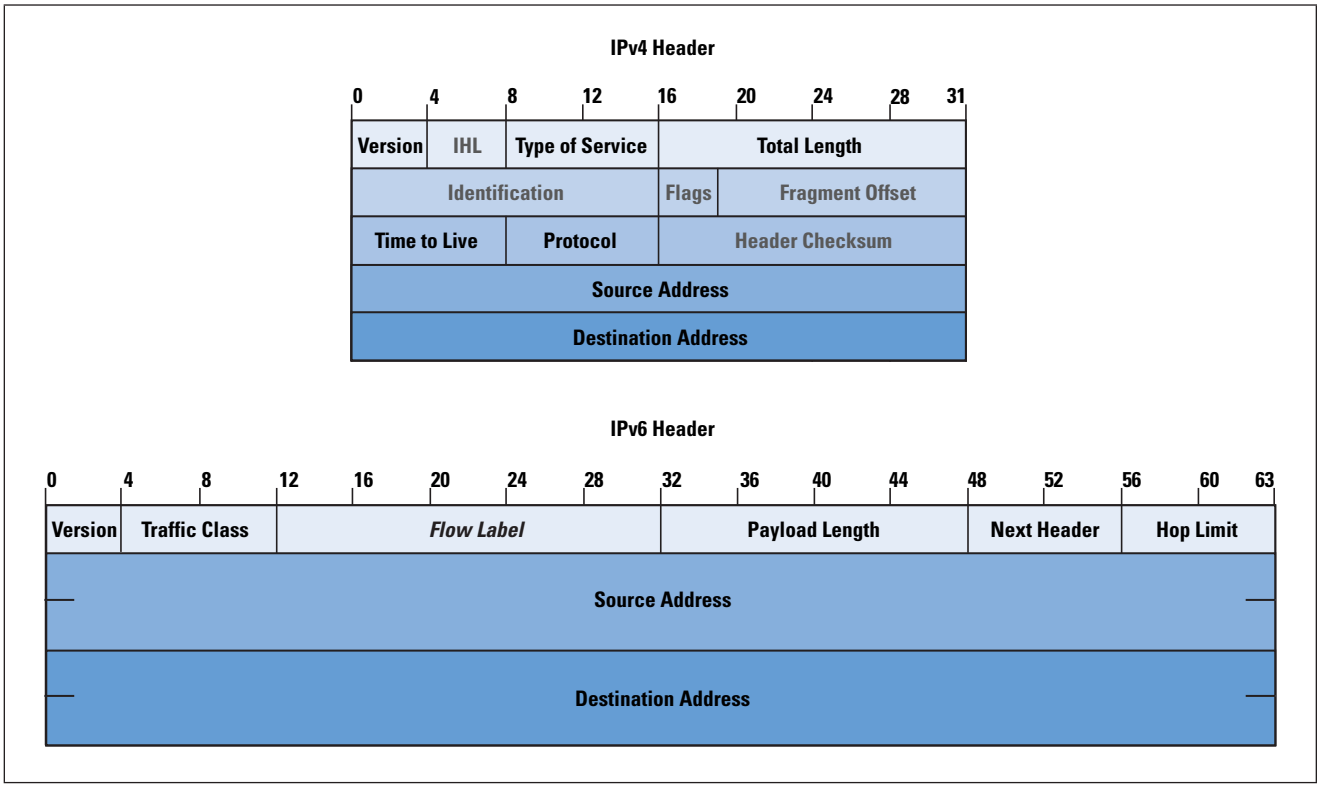
The easiest way to observe—in action—the mechanisms discussed in this article is to set up an IPv6 router on the local subnet and enable IPv6 on the operating system of your choice, if it is not enabled by default. If "native" IPv6 connectivity is not possible, you can set up automatic IPv6 tunneling or use a manually configured IPv6-in-IPv4 tunnel. Getting portable IPv6 address space from a *Regional Internet Registry* (RIR)[1] is a topic worthy of its own article, but *6to4*[2] creates 65,536 IPv6 subnets from a single IPv4 address, and service providers that provide IPv6 connectivity—either natively or over manually configured tunnels—are usually quite generous with IPv6 address space. However, you need to renumber when changing *Internet Service Providers* (ISPs), or when changing IPv4 addresses with 6to4. Most router vendors currently support IPv6 routing, but all widely used general-purpose operating systems can also route IPv6.

When you have IPv6 connectivity, the browser that comes with your system should be able to work over IPv6 (visit **`http://www.kame.net/`**), and there are v6 versions of *ping* and *traceroute* (called *ping6* and *traceroute6*) to determine IPv6 connectivity. More and more applications work over IPv6, but many still do not.

Differences Between IPv4 and IPv6

All knowledge about IPv6 begins with studying the IPv6 header format and the ways in which it is different from the IPv4 header format. Even though at the time the IPv6 specifications were written 64-bit CPUs were rare, the IPv6 designers elected to optimize the IPv6 header for 64-bit processing. For this reason, I have drawn the IPv6 header 64 bits wide in Figure 1, a little different from the way it is usually depicted. Because 64-bit CPUs can read one 64-bit-wide memory word at a time, it is helpful that fields that are 64 bits (or a multiple of 64 bits) wide start at an even 64-bit boundary. Because every 64-bit boundary is also a 32-bit boundary, 32-bit CPUs aren't affected negatively by 64-bit optimization. The IPv4 header is presented in the usual form that highlights its 32-bit background.

Figure 1: The IPv4 and IPv6 headers



The fields in the IPv4 header that are not present in the IPv6 header have gray text; the field that is present in IPv6 but not in IPv4 is shown in italic. The changes from IPv4 to IPv6 follow:

- *Version* now always contains 6 rather than 4.
- The *Internet Header Length* (IHL) field that indicates the length of the IPv4 header is no longer needed because the IPv6 header is always 40 bytes long.
- *Type of Service* is now *Traffic Class*. The original semantics of the IPv4 Type of Service field have been superseded by the *diffserv* semantics per RFC 2474[3]. However, in IPv4, both interpretations of the field are in use (although most routers either cannot or are not configured to look at the field anyway). The IPv6 RFCs do not mandate a specific way to use the Traffic Class field, but generally the RFC 2474 diffserv interpretation is assumed.
- The *Flow Label* is new in IPv6. The idea is that packets belonging to the same stream, session, or flow share a common flow label value, making the session easily recognizable without having to look "deep" into the packet. Recognizing a stream or session is often useful in *Quality of Service* mechanisms. Although few implementations actually look at the flow label, most systems do set different flow labels for packets belonging to different TCP sessions. A zero value in this field means that setting a flow label per session is either not supported or not desired.

- The *Total Length* is the length of the IPv4 packet including the header, but in IPv6, the *Payload Length* does not include the 40-byte IPv6 header, thereby saving the host or router receiving a packet from having to check whether the packet is large enough to hold the IP header in the first place—making for a small efficiency gain. Despite the name, the Payload Length field includes the length of any additional headers, not just the length of the user data.
- The *Identification, Flags,* and *Fragment Offset* fields are used when IPv4 packets must be fragmented. Fragmentation in IPv6 works very differently (explained later), so these fields are relegated to a header of their own.
- *Time to Live* (TTL) is now called *Hop Limit*. This field is initialized with a suitable value at the origin of a packet and decremented by each router along the way. When the field reaches zero, the packet is destroyed. This way, packets cannot circle the network forever when there are loops. Per RFC 791[4], the IPv4 TTL field should be decremented by the number of seconds that a packet is buffered in a router, but keeping track of how long packets are buffered is too difficult to implement, regardless of buffering time. The new name is a better description of what actually happens.
- The *Protocol* field in IPv4 is replaced by *Next Header* in IPv6. In both cases, the field indicates the type of header that follows the IPv4 or IPv6 header. In most cases, the value of this field would be 6 for TCP or 17 for the *User Datagram Protocol* (UDP). Because the IPv6 header has a fixed length, any options such as source routing or fragmentation must be implemented as additional headers that sit between the IPv6 header and the higher-layer protocol such as TCP, forming a "protocol chain."
- The IPv4 *Header Checksum* was removed in IPv6.
- The *Source Address* and *Destination Address* serve the same function in IPv6 as in IPv4, except that they are now four times as long at 128 bits.

All IPv6 hosts and routers are required to support a maximum packet size of at least 1280 bytes. For lower-layer protocols that cannot support a *Maximum Transmission Unit* (MTU) of 1280 bytes, the relevant "IPv6 over …" standard must have a mechanism to break up and reassemble IPv6 packets so that the minimum of 1280 bytes can be accommodated. In IPv4, the official minimum size is 68 bytes—too low to be workable.

Checksums

In IPv4, the IP header is protected by a header checksum, and higher-layer protocols generally also have a checksum. The checksum algorithm for the IPv4 header, *Internet Control Message Protocol* (ICMP), ICMPv6, TCP, and UDP is the same one's complement addition, except that in IPv4, UDP packets may forego checksumming and simply set the checksum field to zero. In IPv6, this practice is no longer allowed: UDP packets must have a valid checksum.

The TCP, UDP, and ICMPv6 checksums are computed over a "pseudoheader" and the TCP, UDP, or ICMPv6 header, and user data, respectively. The pseudoheader consists of the source and destination addresses, the upper-layer packet length, and the protocol number. Including this information in the checksum calculation ensures that TCP, UDP, or ICMPv6 do not process packets that were delivered incorrectly, for instance, because of a bit error in the IP header.

IPv6 no longer has a header checksum to protect the IP header, meaning that when a packet header is corrupted by transmission errors, the packet is very likely to be delivered incorrectly. However, higher-layer protocols should be able to detect these problems, so they are not fatal. Also, lower layers almost always employ a *Cyclic Redundancy Check* (CRC) to detect errors.

Extension Headers

To allow special processing along the way, IPv4 allows extension of the IP header with one or more options. These options are rarely used today, both because they do not really solve common problems and because packets with options cannot be processed in the "fast path," and many routers and firewalls block some or all options. Not unlike the checkout counters at a grocery store, many routers have several "paths" that packets may follow: a fast one, implemented in hardware or highly optimized software, that supports only the most common operations (no checks), and one or more slower paths that use more advanced but slower software code that supports less common operations such as looking at IP options. However, many modern routers have only a fast path, so using additional features does not lead to a performance penalty.

Because the header is of fixed length in IPv6, options cannot be tagged onto the IP header as in IPv4. Instead, they are put in a header of their own that sits between the IPv6 header and the TCP or UDP (or other higher-level protocol) header. The most common extension headers follow:

- *Hop-by-Hop Options:* See the section that follows.
- *Routing:* This header is similar to the *Source Route* option in IPv4.
- *Fragment:* This header is used for fragmentation; see later in this article.
- *Authentication:* This header authenticates the user data and most header fields.
- *Encapsulating Security Payload* (ESP): This header encrypts or authenticates user data.
- *Destination Options:* See the section that follows.

The Hop-by-Hop Options and Destination Options headers are container headers: they have room for multiple suboptions. The Hop-by-Hop Options are processed by all routers along the way. All other options are normally ignored by routers and processed only by the destination. Obviously firewalls, or routers configured to perform filtering, may also look at these options. The Hop-by-Hop Options, Routing, Fragment, and Destination Options extension headers are defined in RFC 2460[5]. The Authentication and ESP extension headers are part of *IP Security* (IPsec).

Note that there is no standard extension header format, meaning that when a host encounters a header that it does not recognize in the protocol chain, the only thing it can do is discard the packet. Worse, firewalls and routers configured to filter IPv6 have the same problem: as soon as they encounter an unknown extension header, they must decide to allow or disallow the packet, even though another header deeper inside the packet would possibly trigger the opposite behavior. In other words, an IPv6 packet with a TCP payload that would normally be allowed through could be blocked if there is an unknown extension header between the IPv6 and TCP headers.

ICMPv6

The IPv6 version of the ICMP generally serves the same purposes as its IPv4 counterpart, but there are some changes. In IPv4, when a router or the destination host cannot process the packet properly, it sends back an ICMP error message along with the original IP header and the first 8 bytes of the higher-layer header. For UDP and TCP, this is enough for the source of the original host to see which TCP session or UDP association generated the offending packet. Because IPv6 supports an arbitrary number of extension headers between the IPv6 header and the higher-layer header, ICMPv6 returns as much of the original packet as will fit in the minimum MTU size of 1280 bytes. In addition to error messages, which are recognizable by an ICMP type of 127 or lower, there are also informational messages, with a type of 128 or higher. Because informational messages are not the result of an error, they do not include an original packet or part thereof. The most common ICMPv6 message types follow:

1: Destination unreachable
2: Packet too big
3: Time exceeded
4: Parameter problem
128: Echo request
129: Echo reply
130: Multicast listener query
131: Multicast listener report
132: Multicast listener done
133: Router solicitation
134: Router advertisement
135: Neighbor solicitation
136: Neighbor advertisement
137: Redirect message

ICMP and ICMPv6 messages also include a "code" that indicates the exact nature of the ICMP message within a certain type. As with ICMP, ICMPv6 calculates a checksum over the control message, but unlike ICMP, the ICMPv6 checksum calculation also includes a pseudoheader. Another departure from IPv4 is the fact that hosts and routers are required to limit the number of ICMPv6 messages they send. So if a router receives 100 packets per second toward an unreachable destination, it is not supposed to send back ICMPv6 packets at the same rate of 100 per second. The ICMPv6 redirect message works slightly different from the ICMP redirect message in IPv4. Like its IPv4 counterpart, the ICMPv6 redirect can be used by a router to inform a host that it should use a different router to reach the destination in question. But routers can also use the IPv6 Redirect to tell a host that the destination is reachable on the local subnet. Thus two hosts that have addresses in different prefixes can communicate directly after receiving redirects from a router.

Neighbor Discovery

When a system wants to send an IPv6 packet to another system connected to the same subnet or link, it needs to know what MAC address (or "link address" in the new IPv6 terminology) it should address the packet to, unless the interface in question is a point-to-point interface. Neighbor discovery allows systems to discover each other's MAC addresses, similar to *Address Resolution Protocol* (ARP) on Ethernet with IPv4.

Each IPv6 system joins the "solicited node" multicast group that corresponds to each of its addresses. Because the solicited node group address consists of the prefix `ff02:0:0:0:0:1:ff00::/104` followed by the bottom 24 bits of the address in question, addresses in different prefixes based on the same interface identifier (including the link-local address) all map to the same solicited node address.

Whenever a system needs to find out the link address for another system residing on the same link, it sends a neighbor solicitation to the solicited node address to which the IPv6 address of the remote system maps. The source host includes its own MAC address in the neighbor solicitation, so the neighbor knows where to send the reply.

Neighbor Unreachability Detection

RFC 2461[6] specifies a procedure for neighbor unreachability detection. IPv6 hosts and routers actively track whether their neighbors are reachable by periodically sending neighbor discovery messages directly to the neighbor. If the neighbor answers, it is reachable; if it does not, there must be some kind of problem, and the system discards the neighbor's MAC address and tries a regular multicast neighbor discovery procedure, allowing IPv6 systems to detect dead neighbors and neighbors that change their MAC address. But it is most useful to detect dead routers. On a subnet with more than one router, a host can simply install a default route toward another router when the router that it has been using becomes unreachable.

If a router loses its IPv6 address and no longer runs IPv6, Windows XP, Linux, MacOS, and FreeBSD all switch over to another router without incident. However, turning off the active router has much more severe effects: at the very least, ongoing downloads stall for a while, and in some cases, the session breaks. I have no explanation for this difference in behavior.

Stateless Address Autoconfiguration

Hosts and routers always configure link-local addresses on every interface on which IPv6 is enabled. The link-local address is nearly always derived from the interface MAC address, but to guarantee uniqueness, it is necessary to perform *Duplicate Address Detection,* which is discussed later.

When a host has a link-local address, it can obtain one or more global IPv6 addresses by using RFC 2462[7, 12], *Stateless Address Autoconfiguration*. IPv6 routers send out *Router Advertisement* (RA) packets (ICMPv6 type 134) periodically and in response to router solicitations. The information in RAs includes:

- An 8-bit *cur hop limit* field that tells hosts what value to use in the Hop Limit field of outgoing IPv6 packets
- The *managed address configuration* (M) flag—This flag is not well-defined, but the basic idea is that when it is set, hosts use a stateful mechanism (presumably *Dynamic Host Configuration Protocol Version 6* [DHCPv6]) to configure their addresses, and when the flag is not set, they use stateless address autoconfiguration.
- The *other stateful configuration* (O) flag—This flag is similar to the M flag, but indicates that the host should use a stateful mechanism to discover nonaddress configuration information.
- A 16-bit *router lifetime* value in seconds—This value tells hosts how long the default route that was created as the result of this RA should remain valid.
- The 32-bit *reachable time* value in milliseconds—This value indicates how long a neighbor should be considered reachable after receiving a "reachability confirmation," which is generally a neighbor advertisement but could be any packet.
- The 32-bit *retrans timer* value in milliseconds—The retrans timer tells hosts how long they should wait before retransmitting neighbor solicitation messages when there is no answer.

When fields that determine a value are set to zero, this means the value is not specified in the RA, so hosts must discover that value through other means. In addition to the preceding, router advertisements may also contain one or more options, such as:

- *Source link-layer address,* the router MAC address
- *MTU,* the maximum packet size that should be used on this subnet
- *Prefix information,* which specifies the prefixes used on the subnet and their properties

The prefix information option, in turn, has its own list of attributes:

- The *address prefix* itself and its length—For stateless address autoconfiguration to work, the prefix must be 64 bits long.
- The *on-link* flag—This flag tells hosts that the prefix is "on-link," so systems with addresses within this prefix are reachable on the subnet in question without help from a router.
- The *autonomous address configuration* flag—This flag tells hosts that they can create an address for themselves by combining this prefix with an interface identifier.
- A 32-bit *valid lifetime* in seconds—This value indicates how long the prefix should be considered on-link and how long autoconfigured addresses using the prefix can be used.
- A 32-bit *preferred lifetime* in seconds—This flag tells hosts how long autoconfigured addresses using this prefix are preferred.

Duplicate Address Detection

To avoid the situation where two IPv6 systems use the same address, systems perform *Duplicate Address Detection* for (nearly) all new IPv6 addresses before they are used. Duplicate address detection is done for global unicast addresses—and not just for those created using stateless address autoconfiguration, but also for link-local addresses. For obvious reasons, there is no duplicate detection for anycast addresses, because the whole point of anycast is that multiple systems have the same address.

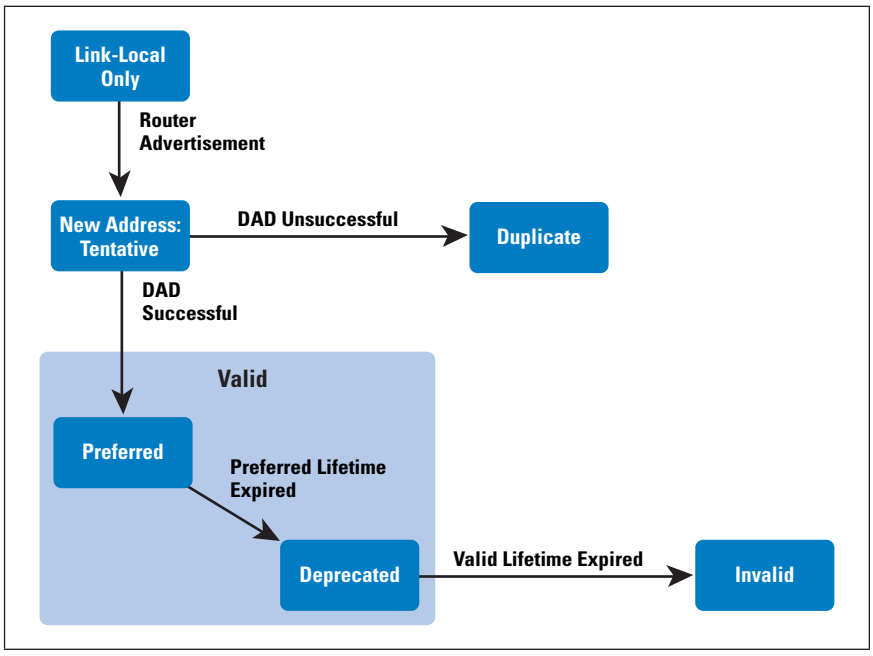


Figure 2: The Lifecycle of an IPv6 Address

As depicted in Figure 2, a host starts with only a link-local address. Duplicate address detection is also done for the link-local address, but this is not shown in the figure.

When a host receives a router advertisement that contains one or more prefixes with the autonomous address configuration flag set, the host creates addresses with interface identifiers derived from the IEEE *64-bit Extended Unique Identifier* (EUI-64) and possibly also a randomly generated one, if the host uses RFC 3041[8] address privacy. The host marks the resulting addresses as "tentative" and proceeds to execute the duplicate address detection procedure by joining the solicited node multicast group for the address in question and sending out one or more neighbor solicitation messages for the address. (If the number of duplicate address detection retries is configured to be zero, no duplicate detection is performed.) Only when there is no answer is the address used. If there is a conflict, the system is supposed to log the error and wait for manual intervention.

Address Lifetime

After successfully maneuvering past the duplicate address detection hurdle, addresses configured through stateless address autoconfiguration can be used until the "preferred lifetime" from the router advertisement message expires. In most cases, the lifetime does not expire because new RAs refresh the timers. But if there are no more RAs, eventually the preferred lifetime elapses and the address becomes "deprecated." New sessions should not use deprecated addresses but should choose "preferred" (nondeprecated) addresses, if available. However, existing sessions will continue to use the deprecated address. Eventually, the "valid lifetime" also runs out, and the deprecated address is removed from the interface, breaking any sessions that are still using the address.

Renumbering

Having different preferred and valid timers for the router advertisement itself and also for any prefixes contained in it makes it possible to do two things: renumber easily and cause more problems. It is even possible to do both at the same time. With stateless autoconfiguration, renumbering is easy: you simply give the router an address in the new prefix and set the preferred lifetime for the old prefix to zero, making hosts create one or more new addresses and deprecate any existing ones in the old prefix as soon as they receive the resulting router advertisement. After that, all new communication should start using the new address immediately. Existing TCP sessions and UDP associations continue to use the same address as before. After some time, all communication that started before the change should have stopped so that the old addresses can be removed safely.

This process is slightly more complex than it seems at first glance: as a precaution against attackers, hosts are not supposed to trust a valid lifetime of less than 7200. So make sure that the hosts have received at least one RA after setting the valid lifetime to 7200, and then set both the lifetimes to zero and remove the autonomous address configuration flag for the prefix. Two hours later, all hosts should have removed the addresses in this prefix, so you can remove the prefix from the router.

Beware that when you renumber because you are switching from one ISP to another, it is unavoidable that at some point, packets with source addresses in address space from ISP A end up at ISP B, or the other way around. If ISP B employs antispoofing or ingress filtering, it will not allow these packets through, so reduced connectivity will result. You can ask one ISP to remove the filters temporarily and then send out all your outgoing traffic over that ISP (or one that did not filter in the first place). However, do not expect too much cooperation from your ISP unless you are a valued customer.

Address Prefix and Router Lifetime Mismatch

Earlier, I mentioned the potential for causing more problems because router advertisements and the prefixes they contain have independent lifetimes. This scenario allows for four permutations:

- The RA lifetime is valid, and the prefix lifetime is valid: IPv6 works.
- The RA lifetime is invalid, and the prefix lifetime is invalid: IPv6 is disabled.
- The RA lifetime is valid, but the prefix lifetime is invalid: The system has an IPv6 default route but no global IPv6 address.
- The RA lifetime is invalid, but the prefix lifetime is valid: The system has a global IPv6 address but no IPv6 default route.

When a host has no global addresses but does have an IPv6 default route (case 3), it cannot reach the rest of the IPv6 Internet. Unfortunately, FreeBSD and MacOS hosts do not know that: they try anyway, with long delays as a result. Only after trying all the remote destination IPv6 addresses and timing out, the system falls back on IPv4 (for applications that try more than one address). Linux, on the other hand, does not install the IPv6 default route or ignores it when no global IPv6 addresses are present, so the timeout is immediate.

Windows XP does install the default route but magically manages to avoid lengthy timeouts anyway. On the other hand, Windows XP suffers timeouts when it has an IPv6 address but no default route (case 4) because Windows implements the on-link assumption: it first performs neighbor discovery on the local subnet for any IPv6 addresses. Only after neighbor discovery times out does Windows revert to IPv4. FreeBSD and MacOS, however, do not implement the on-link assumption, so they immediately notice that the IPv6 destination address is unreachable and revert to IPv4—if an IPv4 address is available and the application cycles through all addresses. With Linux, the default route does not seem to expire even though the timers eventually reach zero and lower. But addresses do expire and are removed when the lifetime for the associated prefix times out.

Address Selection

Choice is good, but it comes with problems of its own. The explicit support for multiple addresses in IPv6 requires the system or applications to choose which address to use for a given communication session. The coexistence of IPv4 and IPv6 in the same host makes this situation even more pressing. RFC 3484[9] provides guidelines in this area—it lists no fewer than 10 rules for choosing a destination address and 8 rules for selecting a source address. Most of these rules are fairly obvious, such as preferring a nondeprecated address over a deprecated one and not using a link-local source address to communicate with a destination that has a global address. It gets more interesting with the "policy table." On systems that support this mechanism, such as Windows XP and FreeBSD 5.4, the administrator can instruct the system to prefer certain address ranges over others.

Path MTU Discovery and Fragmentation

Because routers cannot fragment IPv6 packets, *Path MTU Discovery* (PMTUD) is mandatory in all cases where links with MTUs larger than 1280 bytes are used for IPv6, so it is imperative that routers generate ICMPv6 packet-too-big messages and that these messages make it back to the source of the offending packet. Filtering out these ICMPv6 messages makes it impossible to communicate reliably.

If you decide that you must filter ICMPv6 packet-too-big messages, you *must* use an MTU equal to the IPv6 mandatory minimum of 1280 bytes across your network so there is no need for PMTUD.

Upon reception of a packet-too-big message, TCP reduces its packet size to accommodate the smaller MTU on the path in question. However, protocols that run over UDP often cannot arbitrarily reduce their packet size. In IPv4, UDP packets are generally sent without the "don't fragment" bit set, so routers fragment them if necessary. In IPv6, this setup is not possible; if the packet is too large, the source host has to fragment it. The source host does this by first splitting the packet into unfragmentable and fragmentable parts. The IPv6 header and any headers that must be processed by routers along the way make up the unfragmentable part; the payload data and any headers that have to be processed only on the destination host are the fragmentable part. The fragmentable part is then split into as many parts as required to fit in the path MTU, and each part is transmitted as a packet containing the unfragmentable part, a fragment header, and one of the fragments of the fragmentable part.

The fragment header is 8 bytes, and except for a "next header" field and two reserved fields, it contains the same fragment offset, more fragments, and identification fields as the IPv4 header. The identification field is now 32 bits long and is used to indicate which fragments belong to the same original packet. All fragments except the last one have the "more-fragments" bit set and are multiples of 8 bytes.

After receiving the first fragment (which is not necessarily the first fragment of the original packet), a host waits up to 60 seconds for all other fragments to come in and, if they do, reassembles the original packet by combining all the fragments with the same source and destination addresses and identification field into a single packet. If one or more fragments is lost, the packet cannot be reassembled, so the entire packet is lost.

Note that IPv6 fragmentation has the same problem as IPv4 fragmentation: the TCP or UDP port numbers are available only in the first fragment, making it hard for firewalls and the like to filter fragmented packets. Common solutions are to reassemble the packet prior to filtering or to discard all fragments.

DHCPv6

DHCPv6 (RFC 3315[10]) is the IPv6 version of the DHCP. Because IPv6 has stateless address autoconfiguration, DHCP occupies a very different part of the landscape in IPv6 compared to IPv4. Although the details are different in the by-now-expected places (address length, use of multicasts, some streamlining), the DHCPv6 protocol itself is quite similar to the IPv4 version of DHCP. The more important differences are the way in which the protocol is used. DHCPv6 has three purposes:

- *Address configuration:* Giving out addresses to individual hosts
- *Nonaddress configuration:* Giving out other configuration information, such as DNS resolver addresses and domain search lists
- *Prefix delegation:* Giving out entire prefixes to routers (RFC 3633[11])

A DHCPv6 client interested in an address or other configuration information sends out a *solicit* message indicating its needs to the link-local scope multicast address **`ff02::1:2`**, port 547. (Server-to-client messages are addressed to port 546.) DHCPv6 servers that receive the *solicit* message either directly or forwarded by a relay and can accommodate the request respond with an *advertise* message. The client considers the offers in the various *advertise* messages and directs a *request* message to the server of its choice. The server then replies with a *reply* message, confirming the address or configuration information. Alternatively, if the client wants to receive only configuration information and no addresses or prefixes, it can send a *request-information* message, and the server immediately sends back a reply message, so only half the messages are exchanged and the whole process completes much faster. The client can also use the "rapid commit" option to indicate that it wants to use the expedited procedure for address or prefix assignment if it is fairly certain that it will take up the offer from the first DHCPv6 server that responds.

As expected, IPv6 addresses assigned with DHCPv6 come with a preferred and a valid lifetime. Sometime before this timer expires, the client sends a *renew* message, asking the server if it can continue to use the address. When it has no more use for the address, the client sends a *release* message. Less common situations have other messages.

To allow servers to recognize clients, each device that implements DHCPv6 has *DHCP Unique Identifier* (DUID). In IPv4, DHCP clients use a MAC address or user-supplied string as a Client Identifier. In DHCPv6 this client identifier is always the DUID. Devices can create their DUID in various ways, as long as the DUID is unique and not subject to change, if at all possible.

DHCPv6 supports an authentication mechanism that allows clients and servers to interact in a secure way, so third parties cannot inject false DHCP messages or modify legitimate ones. However, this mechanism must be preconfigured manually on all servers and clients, partially negating the advantages of DHCP over manual configuration.

An interesting use of DHCPv6 is *Prefix Delegation* (PD). With DHCPv6 PD, routers request a prefix that they then use to number one or more of their interfaces, supporting stateless address autoconfiguration for hosts connected to that interface. By creatively borrowing the DHCP timers and reusing them in router advertisements, a whole site can be renumbered by changing a single setting in a DHCPv6 configuration on a DHCPv6 server or a router functioning as a DHCPv6 PD server.

Ed.: This article is adapted from chapter 8 of *Running IPv6* by Iljitsch van Beijnum, published by Apress in 2005, ISBN 1590595270. The article differs from the chapter in that it has been edited for size and the vendor-specific examples have been removed. Used with permission. For information about the book, see:
`http://www.apress.com/book/bookDisplay.html?bID=10026`

References

[1] Karrenberg D., Ross G., Wilson P., and Nobile L., "Development of the Regional Internet Registry System," *The Internet Protocol Journal,* Volume 4, No. 4, December 2001.

[2] Carpenter, B., Fink, B., and Moore, K., "Connecting IPv6 Routing Domains Over the IPv4 Internet," *The Internet Protocol Journal,* Volume 3, No. 1, March 2000.

[3] Nichols, K., Blake, S., Baker, F., and Black, D., "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers," RFC 2474, December 1998.

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

[5] Deering, S. and Hinden, R., "Internet Protocol, Version 6 (IPv6) Specification," RFC 2460, December 1998.

[6] Narten, T., Nordmark, E., and Simpson, W., "Neighbor Discovery for IP Version 6 (IPv6)," RFC 2461, December 1998.

[7] Narten, T. and Thomson, S., "IPv6 Stateless Address Autoconfiguration," RFC 2462, December 1998.

[8] Narten, T. and Draves, R., "Privacy Extensions for Stateless Address Autoconfiguration in IPv6," RFC 3041, January 2001.

[9] Draves, R., "Default Address Selection for Internet Protocol Version 6 (IPv6)," RFC 3484, February 2003.

[10] Droms, R., Ed., Bound, J., Volz, B., Lemon, T., Perkins, C., and Carney, M., "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)," RFC 3315, July 2003.

[11] Troan, O. and Droms, R., "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) Version 6," RFC 3633, December 2003.

[12] François Donzé, "IPv6 Address Autoconfiguration," *The Internet Protocol Journal,* Volume 7, No. 2, June 2004.

ILJITSCH VAN BEIJNUM holds a Bachelor of Information and Communication Technology degree from the Haagse Hogeschool in The Hague, Netherlands. In 1995, he found himself in the emerging Internet Service Provider business. There he learned about system administration, IP networking, and especially routing. After first starting a small ISP with four others and working as a senior network engineer for UUNET Netherlands, he became a freelance consultant in 2000. Not long after that, he started contributing to the IETF Multihoming in IPv6 working group. He wrote the book *BGP: Building Reliable Networks with the Border Gateway Protocol,* ISBN 0-596-00254-8, published by O'Reilly in 2002, and *Running IPv6,* ISBN 1590595270, published by Apress in 2005. E-mail: **iljitsch@muada.com**

Book Reviews

Electronic Brains

Electronic Brains, Stories from the Dawn of the Computer Age, by Mike Hally, ISBN 0-309-09630-8, Joseph Henry Press, 2005.

Electronic Brains is a personal account from the early days of computing that describes the childhood of a technology that is little more than 50 years old. The book originated as a BBC radio programme, still accessible at **`http://www.bbc.co.uk/radio4/science/electronicbrains.shtml`**. Mike Hally traveled over the globe looking for the first "computors" and the stories from the dawn of a new age. This book contains the results of the investigation, giving a first-hand testimony of hard work, passion, and amazing developments that shaped the second half of the last century.

Organization

Chapter 1, "From ABC to ENIAC," presents the development of what is commonly accepted as the first computer, the ENIAC, a computer that replaced calculating machines and people making the operations in ballistic trajectories analysis by hand. Credit is given to John Atanassof and Clifford Berry, the developers of ABC, possibly the first operational computer in the world.

Development of the UNIVAC, the computer famed by predicting the result of the 1952 U.S. presidential election, is presented in Chapter 2. Designed by Eckert and Mauchly, the developers of ENIAC, UNIVACs were commercial computers used for processing census data and so well marketed that the term "UNIVAC" was used as a synonym for "computer."

Chapter 3 looks at the development of the Rand 409, maybe the first mass-produced computer. The 409 was a medium-sized computer, with a price tag of US$100,000 that compared favorably against UNIVAC's $1 million, achieving a sell rate of one per week.

"Computing in Great Britain" is the focus of Chapter 4, where credit is given to Maurice Wilkes and Alan Turing. A worthy detail that gives a glimpse of the technical difficulties overcome is the description of memory based on mercury delay-lines, where binary data was stored using sound pulses on tubes filled with mercury engineered in such way that the delay from transmitter to receiver allowed the electronics to do the calculation before the data in memory was needed at the receiver side.

Perhaps the strangest computer development is set forth in Chapter 5. The *Lyons Electronics Office* (LEO) was a computer developed by a large catering company to expedite its clerical operations. LEO was possibly the first commercial computer in the world, so successful that the catering company began to produce and sell it to other corporations.

Chapter 6 describes the efforts by USSR scientists to develop computing technology. More than one development was made; it is not clear which was the first soviet computer, and the developments were secret—in some cases very specialized, such as a computer with ternary logic instead of the currently used binary logic (ENIAC used decimal logic).

Chapter 7 focuses on computing developments in Australia, work that did not last because the funds were scarce and sometimes the budget was assigned to other sciences, such as radiophysics. Here we can see that computers were used for purposes totally different than their uses in cold-war countries; for example, they were used to answer crossword puzzles—strange if we consider that the disk had a capacity of 3 KB.

A strange computer, formally known as *Hydraulic Economics Computer,* is described in Chapter 8. It was not a typical computer—it was a system developed to show the interrelation between macroeconomic variables using colored water, pumps, and valves. Universities, central banks, and Ford bought the computer, and four of them survive in different parts of the world. The emergence of IBM is the subject of Chapter 9, which presents IBM as a late adopter of computing technology that eventually became the leader of the computer age. We learn that the first computer produced by IBM was the IBM 701; after that came the IBM 1401 and then the IBM 360—the system that consolidated IBM as the ruler in the computing world.

Summary

From the ABC to the well-known ENIAC and UNIVAC, *Electronic Brains* is a testimony to the people who worked day and night to accomplish something that few others understood. Motivated mainly by passion and with little to no economic support, team spirit is a common factor in all the computer developments: "...it was like a brotherhood! We would help each other in case someone got stuck on a particular activity. I would have gone anywhere with those guys. I've never had such unified job environment. We knew we were pushing back the frontiers."

Electronic Brains is an enjoyable book that I recommend to any person with interest in computers and technology. Computer historians could scoff at the rather simple analysis of technical details, but this is not a technical book. The value of *Electronic Brains* is the first-hand account of early undertakings and the multiple-country investigation that is presented. With many anecdotes, this book will serve as a witness to the pioneers of a new era, the computing era.

—Claudio Gutiérrez
claudio.gutierrez.m@gmail.com

Business 2010

Business 2010—Mapping the Commercial Landscape, by Ian Pearson and Michael Lyons, ISBN 1-84439-105-1, Published by Spiro Press, **http://www.spiropress.com/**

This interesting book explores how trends in technology, economic factors, social changes, and evolving attitudes to technology will reshape the business landscape by the year 2010. The book describes its subject matter in terms that are understandable and interesting to both technical and nontechnical audiences. It is valuable to technologists because it expands their perception of the future beyond that which is available through traditional sources such as vendor roadmap sessions by linking closely commercial, technical, and social trends.

Organisation

The book is divided into three main sections. The first looks at the major influences on future business: technological progress, changing attitudes, social forces, and economics. The implications of these factors are then examined, and finally the application of the analysis to business strategy is examined. These ideas are then pulled together in a succinct and easily understood conclusion.

Pearson and Lyons focus on the effect of particular techniques. Some of these, such as self-organising systems and the mimicking of natural phenomena ("biomimetrics"), are fairly unconventional, but others, such as increased miniaturisation, wireless devices, low-cost computing and networking, the semantic Web, and artificial intelligence will be more familiar. The Internet and its potential effect on financial transactions and taxation features heavily. The authors note that attitudes to technology are changing and adoption cycles are reducing, describing the impact that technology has had on the physical labour market and the likely future impact on knowledge workers. The authors consider the economic implications of the exploitation of information, looking at the relative cost of creation and reproduction when compared with more traditional goods and services.

The next three chapters look at the implications of this analysis, starting by looking at numerous trade-offs and counter-balancing forces, such as the effect of the "browser wars" and the relationships between customers and producers. The importance of customer and worker information to a commercial organisation and the problems arising from its exploitation are described. The discussion then considers how the knowledge economy changes the importance of physical assets and commercial relationships, followed by an examination of the political and organisational implications of technology.

Finally the authors look at the business effects, starting with the ease of transferring information between systems. They note that corporate intranets make both the devolving of authority through outsourcing and the imposition of increased command and control through micromanagement easier.

Pearson and Lyons suggest that new technology alters the value chains that influence businesses, leading to more temporary business relationships, their replacement by "value-nets," and the rise of the virtual company. This section concludes by looking at globalisation—how goods and services are paid for and some of the implications for taxation.

The authors ask the question—how can business adapt? They start their analysis by examining the interactions between the physical and mental worlds and cyberspace, noting that a strategic analysis works only if the forces acting on a business do not change too rapidly. As change becomes more rapid, there will be no time to develop business cases, because first-mover advantage will be the only advantage a business can have. Pearson and Lyons conclude that the critical factors in allowing cyber-economy to grow are ease of navigation and the effective use of branding. They conclude by examining who will be the winners and losers in business in the year 2010—and why.

Synopsis

This book is succinct and well-written, covering a complex but interesting field in just under 200 pages. The authors paint a convincing description of future business trends, exploring the technical, commercial, economic, and political pressures that will influence them. Their cause, effect, and potential response treatment leads the reader through the subject in a way that is both interesting and instructive. The authors are not afraid to be controversial and at times they take the reader into some very unfamiliar territory, adding extra spice to the book.

While other books are available that look at the future from a more technologically orientated perspective, this book is one of the few that manages to couple the developments in the commercial and technical worlds, thereby giving a more comprehensive viewpoint. In an age when technologists are increasingly being asked to take more of a commercial view, this can only be a good thing. The approach taken has much in common with that taken by Alvin Toffler in his books *Future Shock* and *The Third Wave*. An updated treatment like this is to be welcomed.

The Authors

Ian Pearson works for British Telecom (BT) as its chief futurologist; he is a well-known speaker on future technology trends and has published extensively in this field. Michael Lyons also works for BT and has more than 30 years of research experience in the telecoms industry. He has recently been working in the fields of decision support systems and long-term research issues, leading a research team in BT's Research and Venturing department. Pearson is described as an "unfettered thinker" and Lyons as a "pragmatic modeller," characteristics which give the book its balanced view.

—Edward Smith, BT, UK
edward.a.smith@btinternet.com

Fragments

ICANN Ratifies Global Policy for Allocation of IPv6 Address Space

On September 7, 2006, the ICANN Board ratified the *Global Policy for Allocation of IPv6 Address Space*. This policy provides for the allocation of IPv6 address space from ICANN to the *Regional Internet Registries* (RIRs).

On July 13, 2006, the Secretary of the *Address Supporting Organization* (ASO) *Address Council* (AC) forwarded to ICANN the proposed global policy for allocation of IPv6 address space. This proposed global policy had been submitted to the ASO AC by the Executive Council of the *Number Resource Organization* (NRO) on June 6, 2006, and adopted by the ASO AC on July 12, 2006. Each RIR community individually discussed the policy and approved its adoption via their own policy development processes. The IPv6 Allocation Policy document is available from the ASO Website:

http://aso.icann.org/docs/aso-global-ipv6.pdf

See also:

http://www.icann.org/announcements/announcement-11sep06.htm

http://www.nro.net

IP addressing in China and the Myth of Address Shortage

In recent years, various sources have repeated a myth that the IPv4 address pool is close to exhaustion. Many of these stories also falsely claim that there are fewer IPv4 addresses allocated to China than to some individual US universities. The *Asia Pacific Network Information Centre* (APNIC) is committed to countering this myth and has published an article in its newsletter *Apster* on this topic. The article is available here:

http://www.apnic.net/news/hot-topics/internet-gov/ip-china.html

Calendar of Internet-related Events

The *Internet Society* (ISOC) maintains an online list of meetings and conferences, see:

http://geneva.isoc.org/events/

Don't forget to tell us if you move!

We receive quite a lot of IPJ return mail marked as "undeliverable." If you change your address please let us know by either using the IPJ subscription tool or sending an e-mail with the new information to **ipj@cisco.com**. Your cooperation is much appreciated.

Call for Papers

The Internet Protocol Journal (IPJ) is published quarterly by Cisco Systems. The journal is not intended to promote any specific products or services, but rather is intended to serve as an informational and educational resource for engineering professionals involved in the design, development, and operation of public and private internets and intranets. The journal carries tutorial articles ("What is...?"), as well as implementation/operation articles ("How to..."). It provides readers with technology and standardization updates for all levels of the protocol stack and serves as a forum for discussion of all aspects of internetworking.

Topics include, but are not limited to:

- Access and infrastructure technologies such as: ISDN, Gigabit Ethernet, SONET, ATM, xDSL, cable, fiber optics, satellite, wireless, and dial systems
- Transport and interconnection functions such as: switching, routing, tunneling, protocol transition, multicast, and performance
- Network management, administration, and security issues, including: authentication, privacy, encryption, monitoring, firewalls, trouble-shooting, and mapping
- Value-added systems and services such as: Virtual Private Networks, resource location, caching, client/server systems, distributed systems, network computing, and Quality of Service
- Application and end-user issues such as: e-mail, Web authoring, server technologies and systems, electronic commerce, and application management
- Legal, policy, and regulatory topics such as: copyright, content control, content liability, settlement charges, "modem tax," and trademark disputes in the context of internetworking

In addition to feature-length articles, IPJ will contain standardization updates, overviews of leading and bleeding-edge technologies, book reviews, announcements, opinion columns, and letters to the Editor.

Cisco will pay a stipend of US$1000 for published, feature-length articles. Author guidelines are available from Ole Jacobsen, the Editor and Publisher of IPJ, reachable via e-mail at **ole@cisco.com**

This publication is distributed on an "as-is" basis, without warranty of any kind either express or implied, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. This publication could contain technical inaccuracies or typographical errors. Later issues may modify or update information provided in this issue. Neither the publisher nor any contributor shall have any liability to any person for any loss or damage caused directly or indirectly by the information contained herein.

The Internet Protocol Journal
Ole J. Jacobsen, Editor and Publisher

Editorial Advisory Board
Dr. Vint Cerf, VP and Chief Internet Evangelist
Google Inc, USA

Dr. Jon Crowcroft, Marconi Professor of Communications Systems
University of Cambridge, England

David Farber
Distinguished Career Professor of Computer Science and Public Policy
Carnegie Mellon University, USA

Peter Löthberg, Network Architect
Stupi AB, Sweden

Dr. Jun Murai, General Chair Person, WIDE Project
Vice-President, Keio University
Professor, Faculty of Environmental Information
Keio University, Japan

Dr. Deepinder Sidhu, Professor, Computer Science &
Electrical Engineering, University of Maryland, Baltimore County
Director, Maryland Center for Telecommunications Research, USA

Pindar Wong, Chairman and President
Verifi Limited, Hong Kong

The Internet Protocol Journal is published quarterly by the Chief Technology Office, Cisco Systems, Inc.
www.cisco.com
Tel: +1 408 526-4000
E-mail: ipj@cisco.com

Cisco, Cisco Systems, and the Cisco Systems logo are registered trademarks of Cisco Systems, Inc. in the USA and certain other countries. All other trademarks mentioned in this document are the property of their respective owners.

Copyright © 2006 Cisco Systems Inc. All rights reserved.

Printed in the USA on recycled paper.

The Internet Protocol Journal, Cisco Systems
170 West Tasman Drive, M/S SJ-7/3
San Jose, CA 95134-1706
USA

ADDRESS SERVICE REQUESTED

PRSRT STD
U.S. Postage
PAID
PERMIT No. 5187
SAN JOSE, CA