1

INTRODUCTION TO SWITCH/ ROUTER ARCHITECTURES

1.1 INTRODUCING THE MULTILAYER SWITCH

The term multilayer switch (or equivalently switch/router) in this book refers to a networking device that performs both Open Systems Interconnection (OSI) network reference model Layer 2 and Layer 3 forwarding of packets (Figure 1.1). The Layer 3 forwarding functions are typically based on the Internet Protocol (IP), while the Layer 2 functions are based on Ethernet. The Layer 2 forwarding function is responsible for forwarding packets (Ethernet frames) within a Layer 2 broadcast domain or Virtual Local Area Network (VLAN). The Layer 3 forwarding function is responsible for forwarding an IP packet from one subnetwork, network or VLAN to an another subnetwork, network, or VLAN.

The IP subnetwork could be created based on well-known IP subnetworking rules and guidelines or as a VLAN. A VLAN is a logical group of devices that can span one or more physically separate network segments that are configured to intercommunicate as if they were connected to one physical Layer 2 broadcast domain. Even though the devices may be located on a number of different physical or geographically separate network segments, the devices can intercommunicate as if they are all connected to one physical broadcast domain.

For the Layer 3 forwarding functions to work, the routing functions in the multilayer switch learn about other networks, paths to destination networks and destinations, through dynamic IP routing protocols or via static/manual configuration information

Switch/Router Architectures: Shared-Bus and Shared-Memory Based Systems, First Edition. James Aweya.

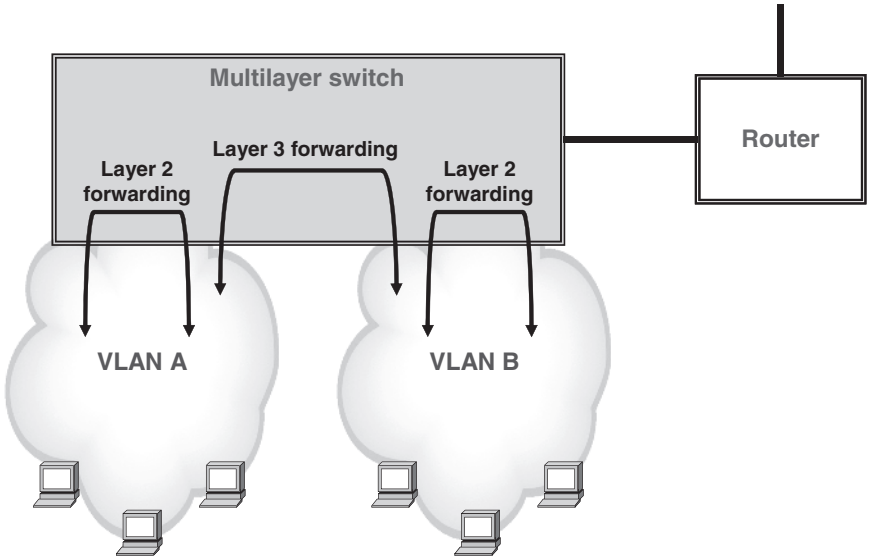

FIGURE 1.1 Layer 2 forwarding versus Layer 3 forwarding.

provided by a network administrator. The dynamic IP routing protocols – RIP (Routing Information Protocol), OSPF (Open Shortest Path First) Protocol, IS–IS (Intermediate System-to-Intermediate System) Protocol, BGP (Border Gateway Protocol) – allow routers and switch/routers to communicate and distribute network topology information between themselves and provide updates when the network topology changes occur. The routers and switch/routers via the routing protocols learn about the network topology to try to select the best loop-free path on which to forward a packet from its source to its destination IP address.

1.1.1 Control and Data Planes in the Multilayer Switch

The Layer 3 and Layer 2 forwarding functions can each be split into subfunctions – the control plane and data (or forwarding) plane functions (Figure 1.2). Comprehensive discussion of the basic architectures of routers is given in [AWEYA2000] and [AWEYA2001]. The Layer 2 functions in an Ethernet switch and switch/router involve relatively very simple control and data plane operations.

The data plane operations in Layer 2 switches involve MAC address learning (to discover the ports on which new addresses are located), frame flooding (for frames with unknown addresses), frame filtering, and frame forwarding (using a MAC address table showing MAC address to port mappings). The corresponding control plane operations in the Layer 2 devices involve running network loop prevention protocols such as the various variants of the Spanning Tree Protocol (STP), link aggregation-related protocols, device management and configuration tools, and so on.

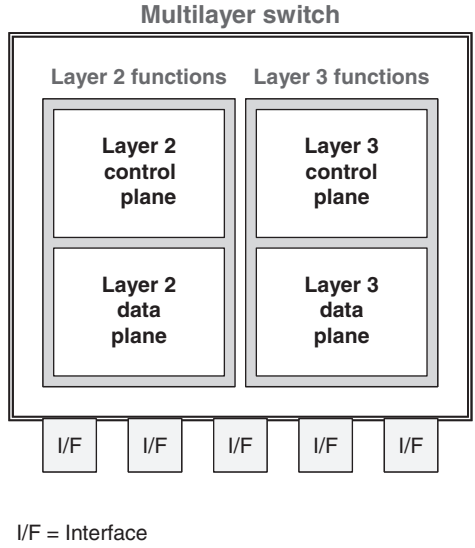

FIGURE 1.2 Control and data planes in a multilayer switch.

Even though the Layer 2 functions can be split into two planes of control and data operations, this separation (of control plane and data plane) is usually applied to the Layer 3 functions performed by routers and switch/routers. In a router or switch/router, the entity that performs the control plane operations is referred to as the routing engine, route processor, or control engine (Figure 1.3).

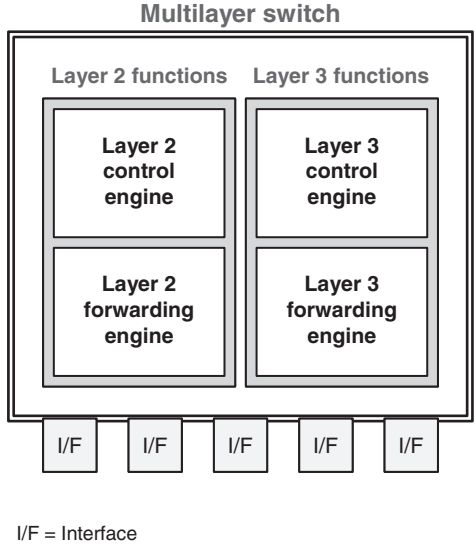

FIGURE 1.3 Control and forwarding engines in multilayer switches.

The entity that performs the data (or forwarding) plane operations is referred to as the forwarding engine or forwarding processor. By separating the control plane operations from the packet forwarding operations, a designer can effectively identify processing bottlenecks in the device. This knowledge allows the designer to develop and/or use specialized software or hardware components and processors to eliminate these bottlenecks.

1.1.2 Control Engine

Control plane operations in the router or switch/router are performed by the routing engine or route processor, which runs the operating system software that has modules that include the routing protocols, system monitoring functions, system configuration and management tools and interfaces, network traffic engineering functions, traffic management policy tools, and so on.

The control engine runs the routing protocols that maintain the routing tables from which the Layer 3 forwarding table is generated to be used by the Layer 3 forwarding engine in the router or switch/router (Figure 1.4). In addition to running other protocols such as PIM (Protocol Independent Multicast), IGMP (Internet Group Management Protocol), ICMP (Internet Control Messaging Protocol), ARP (Address Resolution Protocol), BFD (Bidirectional Forwarding Detection), and LACP (Link Aggregation Control Protocol), the control engine is responsible for maintaining sessions and exchanging protocol information with other router or network devices.

The control engine typically is the module that provides the control and monitoring functions for the entire router or switch/router, including controlling system power supplies, monitoring and controlling system temperature (via cooling fans), and monitoring system status (power supplies, cooling fans, line cards, ports

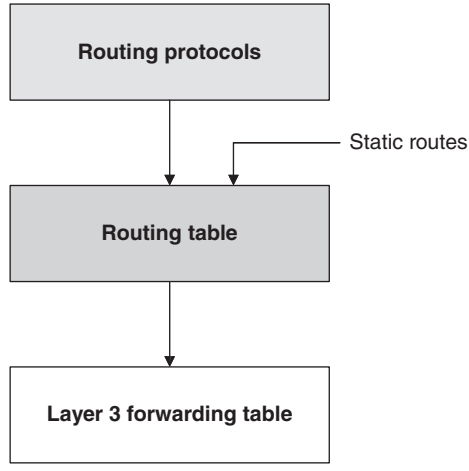

FIGURE 1.4 Routing protocols and routing table in the control engine.

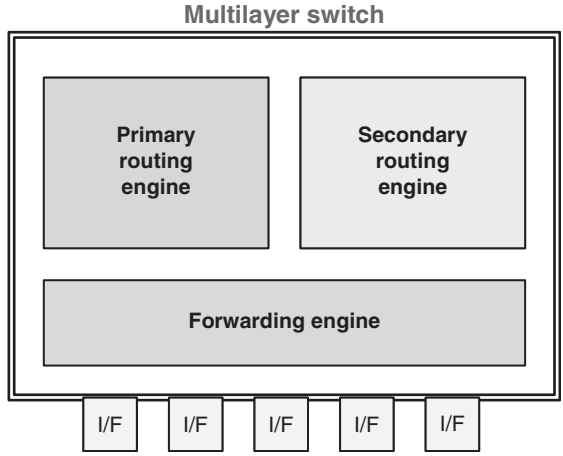

FIGURE 1.5 Multilayer switch with primary and secondary routing engines.

and interfaces, primary/secondary router processors, primary/secondary forwarding engines, etc.). The routing engine also controls the router or switch/router network management interfaces, controls some chassis components (e.g., hot-swap or OIR (online insertion and removal) status of components on the backplane), and provides the interfaces for system management and user access to the device.

In high-performance platforms, more than one routing engine can be supported in the router switch/router (Figure 1.5). If two routing engines are installed, one typically functions as the primary (or master) and the other as the secondary (or backup). In this redundant routing engine configuration, if the primary routing engine fails or is removed (for maintenance/repairs) and the secondary routing engine is configured appropriately, the latter takes over as the master routing engine.

Typically, a router or switch/router supports a set of management ports (e.g., serial port, 10/100 Mb/s Ethernet ports). These ports, generally located on the routing engine module, connect the routing engine to one or more external devices (e.g., terminal, computer) on which a network administrator can issue commands from a command-line interface (CLI) to configure and manage the device. The routing engine could support one or more USB ports that can accept a USB memory device that allows for the loading of the operating system and other system software.

In our discussion in this book, we consider the management plane as part of the control plane – not a separate plane in its own right (Figure 1.6). The management plane is considered a subplane that supports the functions used to manage the router or switch/router via some connections to external management devices (a terminal or computer). Examples of protocols supported in the management plane include Simple Network Management Protocol (SNMP), Telnet, File Transfer Protocol (FTP), Secure FTP, and Secure Shell (SSH). These management protocols allow configuring, managing, and monitoring the device as well as CLI access to the device.

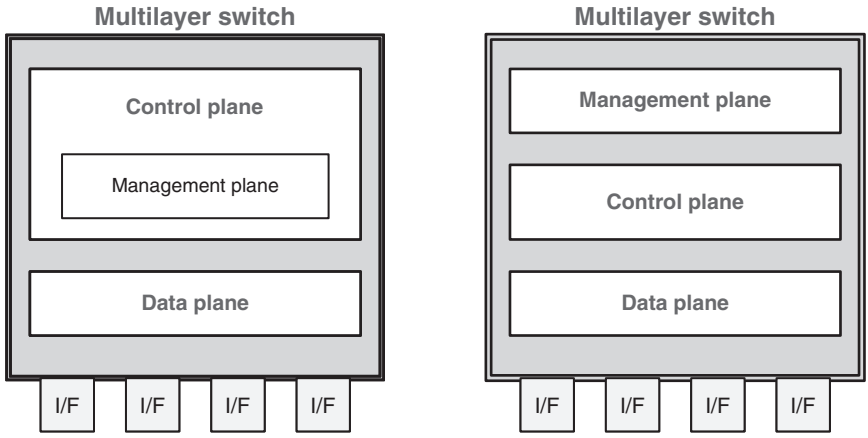

FIGURE 1.6 Control plane versus management plane.

A console port (which is an EIA/TIA-232 asynchronous serial port) could allow the connection of the routing engine to a device with a serial interface (terminal, modem, computer, etc.) through a serial cable with an RJ-45 connector (Figure 1.7). An AUX (or auxiliary) port could allow the connection of the routing engine

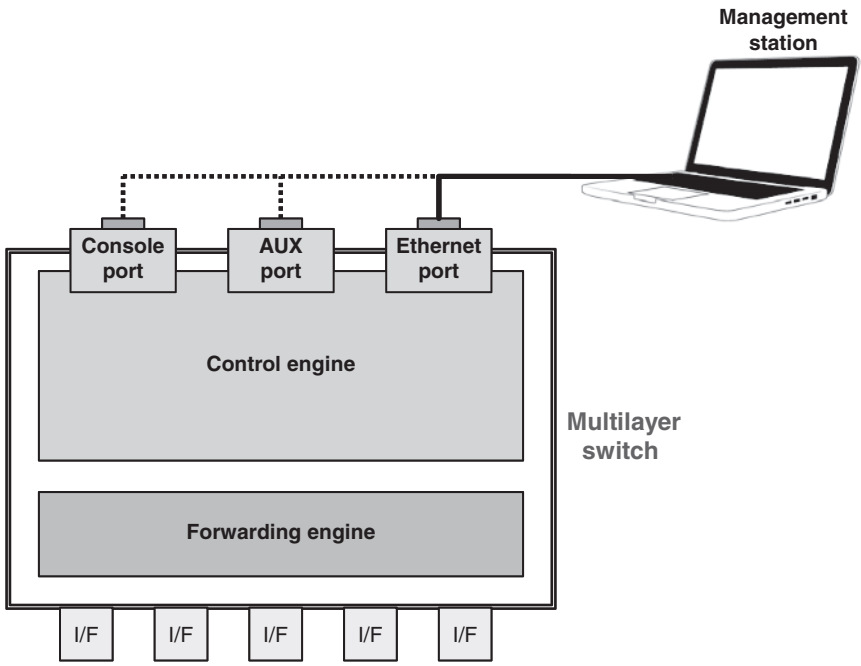

FIGURE 1.7 Management ports.

(through a serial cable with an RJ-45 connector) to a computer, modem, or other auxiliary device. Furthermore, a 10/100 Mb/s Ethernet interface could connect the routing engine to a management LAN (or a device that has an Ethernet connection) for out-of-band management of the router or switch/router.

The routing table (also called the Routing Information Base (RIB)) maintains information about the network topology around the router or switch/router and is constructed and maintained from information obtained from the dynamic routing protocols, and static routes configured by the network administrator. The routing table contains a list of routes to particular IP network destinations (or IP address prefixes). Each route is associated with a metric that is a "distance" measure used by a routing protocol in performing the best path computation to a destination.

The best path to a destination is determined by a routing protocol based on metric (quantitative value) it uses to "measure" the distance it takes to reach a destination. Different routing protocols use different metrics to measure the distance to a given destination. Then the best path to a destination selected by a routing protocol is the path with the lowest metric. Usually the routing protocol selects the best path by evaluating all the possible multiple paths available to the same destination and selects the shortest or optimum path to reach that network. Whenever multiple paths from the router to the same destination exist, each path uses a different output or egress interface on the router to reach that destination.

Typically, routing protocols have their own metrics and rules that they use to construct and update routing tables. The routing protocol generates a metric for each path through the network where the metrics may be based on a single characteristic of a path (e.g., RIP uses a hop count) or several characteristics of a path (e.g., EIGRP uses bandwidth, traffic load, delay, reliability). Some routing protocols may base route selection on multiple metrics, where they combine them into a single representative metric.

If multiple routing protocols (e.g., RIP, EIGRP, OSPF) provide a router or switch/router with different routes to the same destination, the administrative distance (AD) is used to select the preferred (or more trusted) route to that destination (Figure 1.8). The preference is given to the route that has the lowest administrative distance. The administrative distance assigned to a route generated by a particular routing protocol is a numerical value used to rank the multiple paths leading to the same destination. It is a mechanism for a router to rate the trustworthiness of a routing information source (including static routes). The administrative distance represents the trustworthiness the router places on the route. The lower the administrative distance, the more trustworthy the routing information source.

For example, considering OSPF and RIP, routes supplied by OSPF have a lower administrative distance than routes supplied by the RIP. It is not unusual for a router or switch/router to be configured with multiple routing protocols in addition to static routes. In this case, the routing table will have more than one routing information source for the same destination. For example, if the router runs both RIP and EIGRP, both routing protocols may compute different best paths to the same destination. However, RIP determines its path based on hop count, while EIGRP's

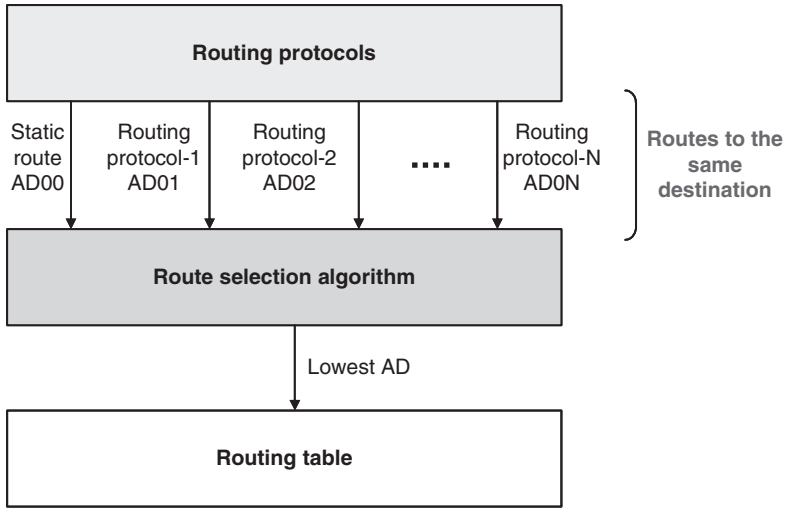

FIGURE 1.8 Use of administrative distance in route selection for the routing table.

best path is based on its composite metric. The administrative distance is used by the router to determine the route to install into its routing table. A static route takes precedence over an EIGRP discovered route, which in turn takes precedence over a RIP discovered route.

As another example, if OSPF computes a best path to a specific destination, the router first checks if an entry for that destination exists in the routing table. If no entry exists, the OSPF discovered route is installed into the routing table. If a route already exists, the router decides whether to install the OSPF discovered route based on the administrative distance of the OSPF generated route and the administrative distance of the existing route in the routing table. If the OSPF discovered route has the lowest administrative distance to the destination (compared to the route in the routing table), it is installed in the routing table. If the OSPF discovered route is not the route with the best administrative distance, it is rejected.

A routing protocol may also identify/discover multiple paths (a bundle of routes not just one) to a particular destination as the best path (Figure 1.9). This happens when the routing table has two or more paths with identical metrics to the same destination address. When the router has discovered two or more paths to a particular destination with equal cost metrics, the router can take advantage of this to forward packets to that destination over the multiple paths equally.

In the above situation, the routing table may support multiple entries where the router installs the maximum number of multiple paths allowed per destination address. The routing table will contain the single destination address, but will associate it with multiple exit router interfaces, one interface entry for each equal cost path. The router will then forward packets to that destination address across the multiple exit interfaces listed in the routing table. This feature is known as

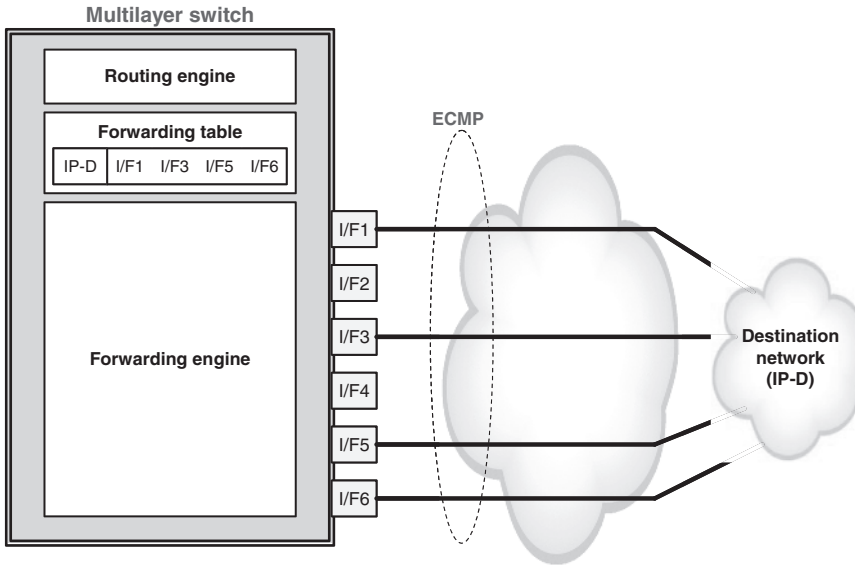

FIGURE 1.9 Equal cost multipath routing.

equal-cost multipath (ECMP) and can be employed in a router to provide load balancing or sharing across the multiple paths.

A router or switch/router may also support an important feature called virtual routing and forwarding (VRF), which is a technology that allows the router or switch/router to support concurrently multiple independent virtual routing and forwarding table instances (Figure 1.10). VRF is a feature that can be used to create logical segmentation between different networks on the same routing platform. The routing instances are independent, thereby allowing VRF to use overlapping IP addresses even on a single interface (i.e., using subinterfaces) without conflicting with each other.

VRF allows, for example, a network path between two devices to be segmented into several independent logical paths without having to use multiple devices for each path. With VRF, the traffic paths through the routing platform are isolated, leading to increased network security, which can even eliminate the need for encryption and authentication for network traffic.

A service provider may use VRF to create separate virtual private networks (VPNs) on a single platform for its customers. For this reason, VRF is sometimes referred to as VPN routing and forwarding. Similar to VLAN-based networking where IEEE 802.1Q trunks can be used to extend a VLAN between switching domains, VRF-based networking can use IEEE 802.1Q trunks, Multiprotocol Label Switching (MPLS) tags, or Generic Routing Encapsulation (GRE) tunnels to extend and connect a path of VRFs together.

While VRF has some similarities to a logical router, which may support many routing tables, a (single) VRF instance supports only one routing table. Furthermore,

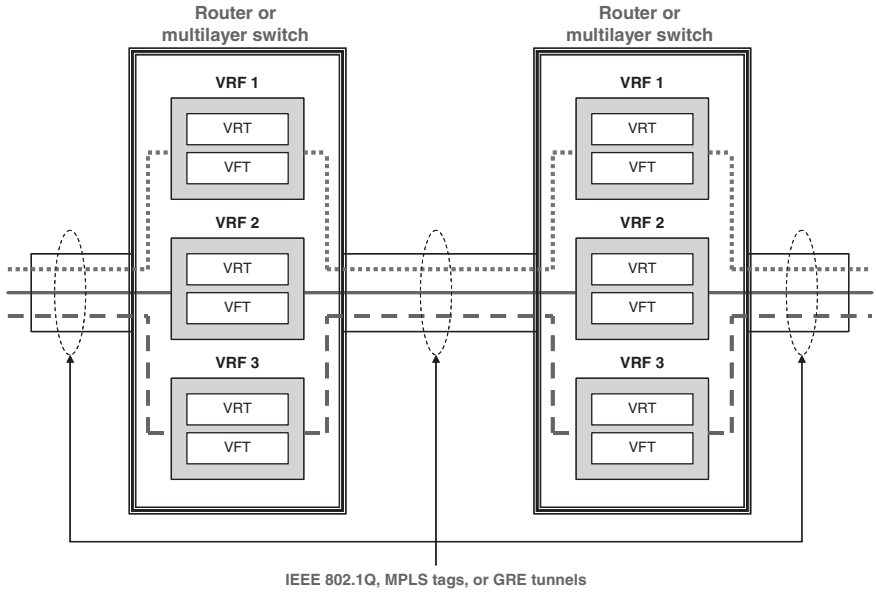

FIGURE 1.10 Virtual routing and forwarding (VRF).

VRF uses a forwarding table that specifies the next hop node for each packet forwarded, a list of nodes along a path that may forward the packet, and routing protocols and a set of forwarding rules that specify how to forward the packet. These requirements isolate traffic and prevent packets in a specific VRF from being forwarded outside its VRF path, and also prevent traffic from outside (the VRF path) from entering the specific VRF path.

The routing engine also maintains an adjacency table, which typically in its simplest form may be an ARP cache. The adjacency table (also known as Adjacency Information Base (AIB)) contains the MAC addresses and egress interfaces of all directly connected next hops and directly connected destinations (Figure 1.11). This table is populated with MAC address discoveries obtained from ARP, statically configured MAC addresses, and other Layer 2 protocol tables (e.g., Frame Relay and ATM map tables). The network administrator can explicitly configure MAC address information in the adjacency table, for example, for directly attached data servers.

The adjacency table is built from information obtained from ARP that is used by IP hosts to dynamically learn the MAC address of other IP hosts on the same Layer 2 broadcast domain (VLAN or subnet) when the target host's IP address is known. For example, an IP host that needs to know the MAC address of another IP host connected to same VLAN can send an ARP request using a broadcast address. The sending host then waits for an ARP reply from the target IP host. When received, the ARP reply includes the required MAC address and the associated IP address. This MAC address can then be used to address Ethernet frames (destination MAC address) originating from the sending IP host to the target IP host on the same VLAN.

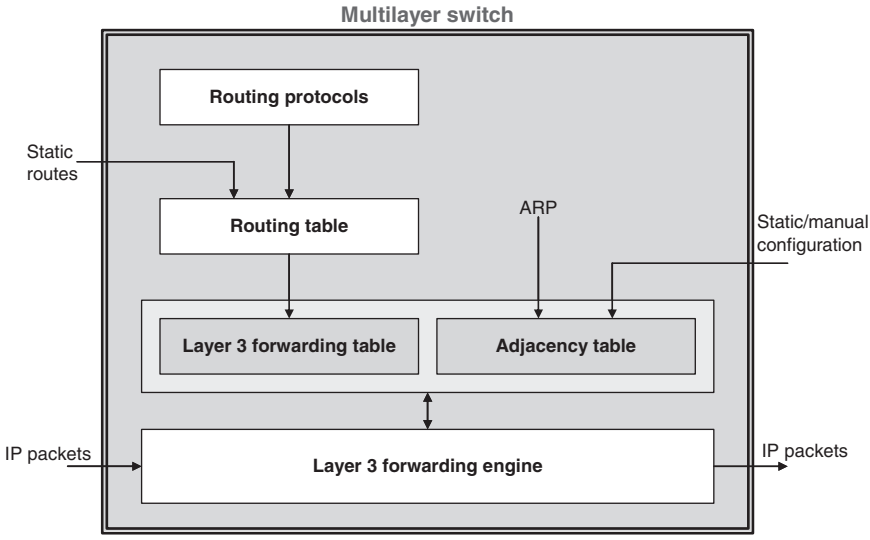

FIGURE 1.11 Layer 3 forwarding table and adjacency table.

1.1.3 Forwarding Engine

The data or forwarding plane operations (i.e., the actual forwarding of data) in the router or switch/router are performed by the forwarding engine, which can consist of software and/or hardware (ASICs) processing elements. The Layer 3 forwarding engine performs route lookup for each arriving IP packet using a Layer 3 forwarding table. In some implementations, the adjacency table is not a separate module but integrated in the Layer 3 forwarding table to allow for one lookup for all next hop forwarding information. The forwarding engine performs filtering and forwarding of incoming packets, directing outbound packets to the appropriate egress interface or interfaces (for multicast traffic) for transmission to the external network.

As already discussed, routers and switch/routers determine best paths to network destinations by sharing information about the network topology and conditions with neighboring routers. The router or switch/router communicates with their neighboring Layer 3 peers to build a comprehensive routing database (the routing table) that enables the forwarding engine to forward packets across optimum paths through the network. The information in the routing table (which is very extensive) is distilled into the much smaller Layer 3 forwarding table that is optimized for IP data plane operations.

Not all the information in the routing table is directly used or is relevant to data plane operations. The Layer 3 forwarding table (also called Forwarding Information Base (FIB)) maintains a mirror image of all the most relevant forwarding information contained in the routing table (next hop IP address, egress port(s), next hop MAC address (adjacency information)). When routing or topology changes occur

in the network, the IP routing table is updated, and those changes are reflected in the forwarding table.

The forwarding engine processes packets to obtain next hop information, applies quality of service (QoS) and security filters, and implements traffic management policies (policing, packet discard, scheduling, etc.), routing policies (e.g., ECMP, VRFs), and other functions required to forward packets to the next hop along the route to their final destinations. In the switch/router, separate forwarding engines could provide the Layer 2 and Layer 3 packet forwarding functions. The forwarding engine may need to implement some special data plane operations that affect packet forwarding such as QoS and access control lists (ACLs).

Each forwarding engine can consist of the following components:

- Software and/or hardware processing module or engine, which provides the route (best path) lookup function in a forwarding table.
- Switch fabric interface modules, which use the results of the forwarding table lookup to guide and manage the transfer of packet data units across the switch fabric to the outbound interface(s). The switch interface module will be responsible for prepending internal routing tags to processed packets. The internal routing tag would typically carry information about the destination port, priority queuing, packet address rewrite, packet priority rewrite, and so on.
- Layer 2/Layer 3 processing modules, which perform Layer 2 and Layer 3 packet decapsulation and encapsulation and manage the segmentation and reassembly of packets within the router or switch/router.
- Queuing and buffer memory processing modules, which manage the buffering of (possibly, segmented) data units in the memory as well as any priority queuing requirements.

As discussed above, the forwarding table is constructed from the routing table and the ARP cache maintained by the routing engine. When an IP packet is received by the forwarding engine, a lookup is performed in the forwarding table (and adjacency table) for the next hop destination address and the appropriate outbound port, and the packet is sent to the outbound port. The Layer 3 forwarding information and ARP information can be implemented logically as one table or maintained as separate tables that can be jointly consulted when forwarding packets.

The router also decrements the IP TTL field and recomputes the IP header checksum. The router rewrites the destination MAC address of the packet with the next hop router's MAC address, and also rewrites the source MAC address with the MAC address of the outgoing Layer 3 interface. The router then recomputes the Ethernet frame checksum and finally delivers the packet out the outbound on its way to the next hop.

1.2 EVOLUTION OF MULTILAYER SWITCH ARCHITECTURES

The network devices that drive service provider networks, enterprise networks, residential networks, and the Internet have evolved architecturally and considerably over the years and are still evolving to keep up with new service requirements and user traffic. The continuous demand for more network bandwidth in addition to the introduction of new generation of services and applications are placing tremendous performance demands on networks.

Streaming and real-time audio and video, videoconferencing, online gaming, real-time business transactions, telecommuting, the increasingly sophisticated home devices and appliances, and the ubiquity of bandwidth hungry mobile devices are some of the many applications and services that are driving the need for scalability, reliability, high bandwidth, and improved quality of services in networks. As a result, network operators including the residential network owners are demanding the following features and requirements from their networks:

- The ability to cost-effectively scale their networks with minimal downtime and impact on network operations as traffic grows.
- The ability to harness the higher link bandwidths provided by the latest wireless and fiber-optic technologies to allow for transporting large volumes of traffic.
- The ability to implement mechanisms for minimizing data loss, data transfer latency, and latency variations (sometimes referred to as network jitter), thus enabling the improved support of delay-sensitive applications. This includes the ability to create differentiated services by prioritizing traffic based on application and user requirements.

The pressures of these demands have created the need for sophisticated new equipment designs for the network from the access to the core. New switch, router, and switch/router designs have emerged and continue to do so to meet the corresponding technical and performance challenges being faced in today's networks. These designs also give operators of enterprise networks and service provider networks the ability to quickly improve and scale their networks to bring new services to market.

The first generation of routers and switch/routers were designed to have the control plane and packet forwarding function share centralized processing resources resulting in poor device and, consequently, network performance as network traffic grow. In these designs, all processing functions (regardless of the offered load) must contend for a centralized, single, and finite pool of processing resources.

To handle the growing network traffic loads and at the same time harness the high-speed wireless and ultrafast fiber-optic interfaces (10, 40, and 100 Gb/s speeds), the typical high-end router and switch/router now support distributed forwarding architectures. These designs provide high forwarding capacities, largely

by distributing the packet forwarding functions across modular line cards on the system.

These distributed architectures enable operators to scale their networks capacity even from the platform level, that is, within a single router or switch/router chassis as network traffic loads increase without a corresponding drain on central processing resources. These distributed forwarding architectures avoid the packet forwarding throughput degradation and bottlenecks normally associated with the centralized processor forwarding architectures.

In the next chapter and also in rest of the book, we describe the various bus- and shared-memory-based forwarding architectures, starting from the architectures with centralized forwarding to those with distributed forwarding. The inner workings of these architectures are discussed in addition to their performance limitations.

1.2.1 Centralized Forwarding versus Distributed Forwarding Architectures

From a packet forwarding perspective, we can categorize broadly the switch, router, or switch/router architectures as centralized or distributed. There are architectures that fall in between these two, but focusing on these two here helps to shed light on how the designs have evolved to be what they are today.

In a centralized forwarding architecture, a processor is equipped with a forwarding function (engine) that allows it to make all the packet forwarding decisions in the system. In this architecture, the routing engine and forwarding engine both could be on one processor or the routing engine implemented on a separate centralized processor. In the simplest form of the centralized architecture, typically a single general-purpose CPU manages all control and data plane operations.

In such a centralized forwarding architecture, when a packet is received on an ingress interface or port, it is forwarded to the centralized forwarding engine. The forwarding engine examines the packet's destination address to determine the egress port on which the packet should be sent out. The forwarding engine completes all its processing and forwards the packet to the egress port to be sent to the next hop.

In the centralized architecture, there is not great distinction between a port or a line card – both have very limited packet processing capabilities from a packet forwarding perspective. A line card can have more than one port, where the line card only supports breakouts for the multiple ports on the card. Any processing and memory at a port or line card is only for receive operations from the network and data transfer to the centralized processor, and data transfer from the centralized processor and transmit to the network.

In distributed forwarding architecture, the line cards are equipped with forwarding engines that allow them to make packet forwarding decisions locally without consulting the route processor or another forwarding engine in the system. The route processor is only responsible for generating the master forwarding table that is then distributed to the line cards. It is also responsible for synchronizing the distributed forwarding tables in the line card with the master forwarding table in

the route processor whenever changes occur in the master table. The updates are triggered by route and network topology changes that are captured by the routing protocols.

In a distributed forwarding architecture, when a packet is received on an ingress line card, it is sent to the local forwarding engine on the card. The local forwarding engine performs a forwarding table lookup to determine if the outbound interface is local or is on another line card in the system. If the interface is local, it forwards the packet out that local interface. If the outbound interface is located on a different line card, the packet is sent across the switch fabric directly to the egress line card, bypassing the route processor.

As will be discussed in the next chapter, some packets needing special handling (special or exception packets) will still have to be forwarded to the route processor by the line card. By offloading the packet forwarding operations to the line cards, packet forwarding performance is greatly improved in the distributed forwarding architectures.

It is important to note that in the distributed architecture, routing protocols and most other control and management protocols always run on a routing engine that is typically in one centralized (control) processor. Some distributed architectures offload the processing of other control plane protocols such as ARP, BFD, and ICMP to a line card CPU. This allows each line to handle any ARP, BFD, and ICMP messages locally without having to rely on the centralized route processor.

Current practice in distributed router and switch/router design today is to make route processing a centralized function (which also has the added benefit of supporting route processor redundancy, if required (Figure 1.5)). The control plane requires a lot of complex operations and algorithms (routing protocols, control and management protocols, system configuration and management, etc.) and so having a single place where all route processing and routing table information maintenance are done for each platform significantly reduces system complexity. Furthermore, the control plane operations tend to have a system-wide impact and also change very slowly compared to the data plane operations.

Even though the control plane is centralized, there is no need to scale route processing resources in direct proportion to the speed of the line cards being supported or added to the system in order to maintain system throughput. This is because, unlike the forwarding engine (whether centralized or distributed), the route processor performs no functions on a packet-by-packet basis. Rather, it communicates with other Layer 3 network entities and updates routing tables, and its operations can be decoupled from the packet-by-packet forwarding process.

Packet forwarding relies only on using the best-path information precomputed by the route processor. A forwarding engine performs forwarding function by consulting the forwarding table, which is a summary of the main forwarding information in routing table created by all the routing protocols as described in Figure 1.8. Based on destination address information in the IP packet header, the forwarding engine consults the forwarding table to select the appropriate output interface and forwards the packet to that interface or interfaces (for multicast traffic).