
Chapter 5 The Network Layer: Control Plane

In this chapter, we'll complete our journey through the network layer by covering the **control-plane** component of the network layer—the *network-wide* logic that controls not only how a datagram is forwarded among routers along an end-to-end path from the source host to the destination host, but also how network-layer components and services are configured and managed. In **Section 5.2**, we'll cover traditional routing algorithms for computing least cost paths in a graph; these algorithms are the basis for two widely deployed Internet routing protocols: OSPF and BGP, that we'll cover in **Sections 5.3** and **5.4**, respectively. As we'll see, OSPF is a routing protocol that operates within a single ISP's network. BGP is a routing protocol that serves to interconnect all of the networks in the Internet; BGP is thus often referred to as the “glue” that holds the Internet together. Traditionally, control-plane routing protocols have been implemented together with data-plane forwarding functions, monolithically, within a router. As we learned in the introduction to **Chapter 4**, software-defined networking (SDN) makes a clear separation between the data and control planes, implementing control-plane functions in a separate “controller” service that is distinct, and remote, from the forwarding components of the routers it controls. We'll cover SDN controllers in **Section 5.5**.

In **Sections 5.6** and **5.7** we'll cover some of the nuts and bolts of managing an IP network: ICMP (the Internet Control Message Protocol) and SNMP (the Simple Network Management Protocol).

5.1 Introduction

Let's quickly set the context for our study of the network control plane by recalling [Figures 4.2 and 4.3](#). There, we saw that the forwarding table (in the case of destination-based forwarding) and the flow table (in the case of generalized forwarding) were the principal elements that linked the network layer's data and control planes. We learned that these tables specify the local data-plane forwarding behavior of a router. We saw that in the case of generalized forwarding, the actions taken ([Section 4.4.2](#)) could include not only forwarding a packet to a router's output port, but also dropping a packet, replicating a packet, and/or rewriting layer 2, 3 or 4 packet-header fields.

In this chapter, we'll study how those forwarding and flow tables are computed, maintained and installed. In our introduction to the network layer in [Section 4.1](#), we learned that there are two possible approaches for doing so.

- **Per-router control.** [Figure 5.1](#) illustrates the case where a routing algorithm runs in each and every router; both a forwarding and a routing function are contained

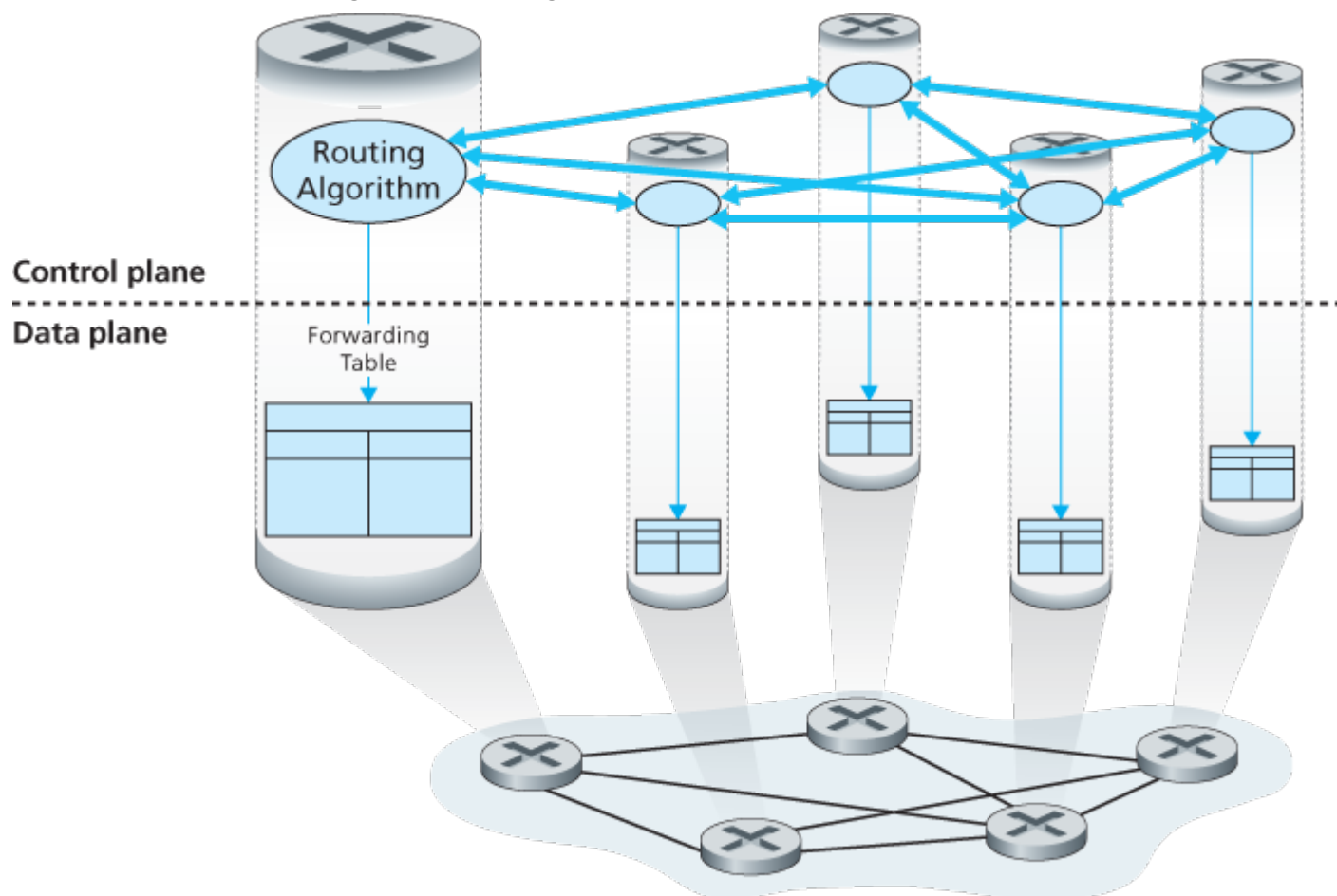


Figure 5.1 Per-router control: Individual routing algorithm components interact in the control plane

within each router. Each router has a routing component that communicates with the routing components in other routers to compute the values for its forwarding table. This per-router control approach has been used in the Internet for decades. The OSPF and BGP protocols that we'll study in [Sections 5.3](#) and [5.4](#) are based on this per-router approach to control.

- **Logically centralized control.** [Figure 5.2](#) illustrates the case in which a logically centralized controller computes and distributes the forwarding tables to be used by each and every router. As we saw in [Section 4.4](#), the generalized match-plus-action abstraction allows the router to perform traditional IP forwarding as well as a rich set of other functions (load sharing, firewalling, and NAT) that had been previously implemented in separate middleboxes.

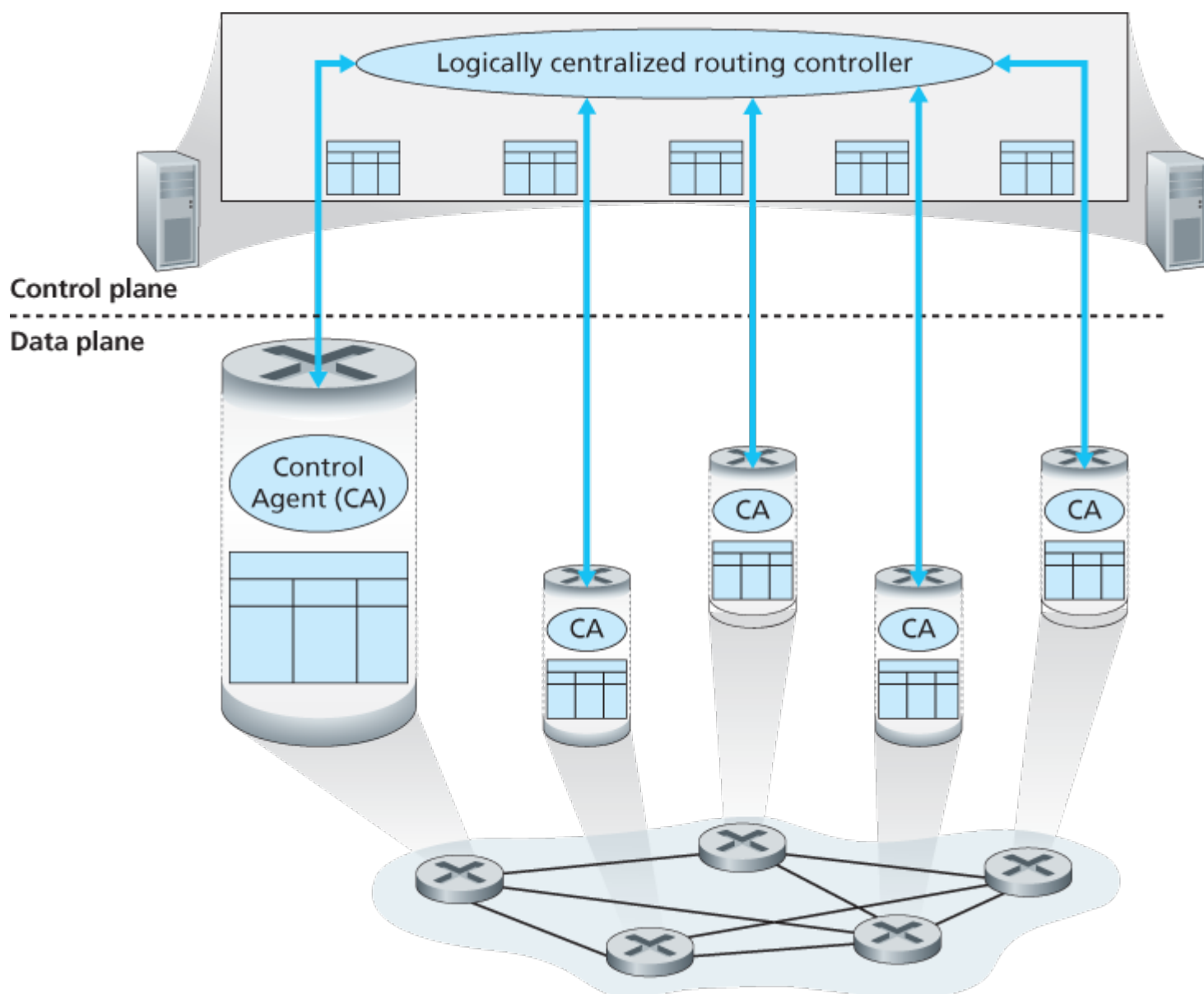


Figure 5.2 Logically centralized control: A distinct, typically remote, controller interacts with local control agents (CAs)

The controller interacts with a control agent (CA) in each of the routers via a well-defined protocol to configure and manage that router's flow table. Typically, the CA has minimum functionality; its job is to communicate with the controller, and to do as the controller commands. Unlike the routing algorithms in [Figure 5.1](#), the CAs do not directly interact with each other nor do they actively take part in computing

the forwarding table. This is a key distinction between per-router control and logically centralized control.

By “logically centralized” control [\[Levin 2012\]](#) we mean that the routing control service is accessed as if it were a single central service point, even though the service is likely to be implemented via multiple servers for fault-tolerance, and performance scalability reasons. As we will see in [Section 5.5](#), SDN adopts this notion of a logically centralized controller—an approach that is finding increased use in production deployments. Google uses SDN to control the routers in its internal B4 global wide-area network that interconnects its data centers [\[Jain 2013\]](#). SWAN [\[Hong 2013\]](#), from Microsoft Research, uses a logically centralized controller to manage routing and forwarding between a wide area network and a data center network. China Telecom and China Unicom are using SDN both within data centers and between data centers [\[Li 2015\]](#). AT&T has noted [\[AT&T 2013\]](#) that it “supports many SDN capabilities and independently defined, proprietary mechanisms that fall under the SDN architectural framework.”