网络层：数据平面

概述

转发和路由选择

每台路由器的数据平面的主要作用是从其输入链路向其输出链路转发数据报（**转发**）；控制平面的主要作用是协调这些本地的每路由器转发动作，使得数据报沿着源和目的地主机之间的路由器路径最终进行端到端传送（**路由**）。

转发(forwarding)是指将分组从一个输入链路接口转移到适当的输出链路接口的路由器本地动作。转发发生的时间尺度很短(通常为几纳秒)，因此通常用硬件来实现。路由选择(routing)是指确定分组从源到目的地所采取的端到端路径的网络范围处理过程。路由选择发生的时间尺度长得多(通常为几秒)，因此通常用软件来实现。

每台网络路由器中有一个关键元素是它的转发表(forwarding table)。路由器检查到达分组首部的一个或多个字段值，进而使用这些**首部值**在其转发表中**索引**，通过这种方法来转发分组。这些值对应存储在转发表项中的值，指出了该分组将被转发的路由器的输出链路接口。

网络层控制平面的传统方法

网络层控制平面的传统方法是指**每个路由器都运行一个路由选择算法**，根据网络拓扑和链路代价，计算出到达目的网络的最优路径，并将这些路径存储在路由表中。路由表中的每一项包含了一个目的网络的地址、一个下一跳路由器的地址和一个输出接口。当一个路由器收到一个数据报时，它会根据数据报的目的地址，查找路由表，找到匹配的路由表项，然后将数据报转发到相应的输出接口。这种方法是基于数据报的目的地址进行转发的，也称为目的地址转发。

网络层控制平面的传统方法有两种主要的路由选择算法：链路状态算法和距离矢量算法。链路状态算法要求每个路由器获取整个网络的拓扑信息，然后运行迪杰斯特拉算法，计算出到达每个目的网络的最短路径。距离矢量算法要求每个路由器只维护到达每个目的网络的最小代价和下一跳路由器，然后定期与邻居路由器交换这些信息，根据贝尔曼-福特方程，更新自己的路由表。这两种算法各有优缺点，例如，链路状态算法可以快速收敛，但需要较多的消息传输和计算，而距离矢量算法可以节省资源，但可能出现无穷计数和环路的问题。

网络层控制平面的SDN方法

网络层控制平面的 SDN 方法是指使用软件来管理和配置网络的一种方法，它将控制平面和数据平面分离，使网络更加灵活和可编程。控制平面是负责决定数据包如何在网络中转发的部分，数据平面是负责实际转发数据包的部分。SDN 方法可以提高网络的效率、安全性和可扩展性，以适应不断变化的业务需求和流量模式。SDN **方法的核心组件是** SDN **控制器**，它是一个软件应用程

序，可以通过开放式 API 与网络设备进行通信，下发控制指令和获取网络状态信息。SDN 控制器可以实现网络的集中管理、编排、分析和自动化，从而降低网络的复杂性和成本。**远程控制器计算和分发转发表以供每台路由器所使用。**

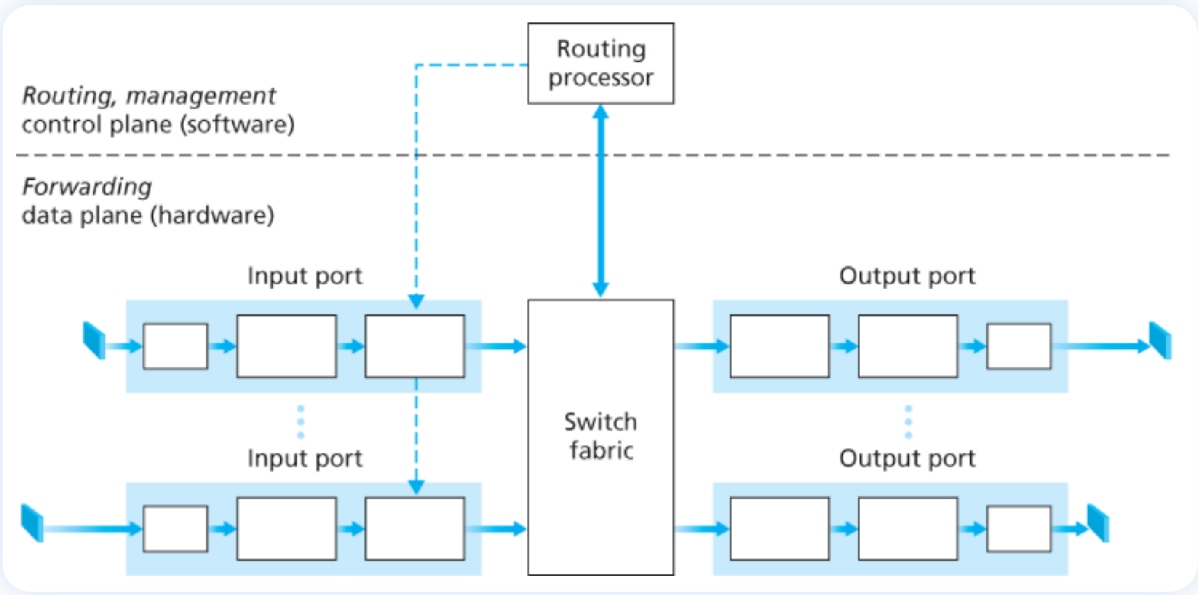
分组交换机概念

分组交换机是指一台通用分组交换设备，它根据分组首部字段中的值，从输入链路接口到输出链路接口转移分组。某些分组交换机称为链路层交换机（link layer switch），基于链路层帧中的字段值做出转发决定，这些交换机因此被称为链路层（第2层）设备。其他分组交换机称为路由器（router），基于网络层数据报中的首部字段值做出转发决定

路由器工作原理

路由器结构

- 输入端口（Input port）
- 交换结构（Switch fabric）
- 输出端口（Output port）
- 路由选择处理器（Routing processor）



- 上半部分为控制平面：软件、毫秒级
- 下半部分为数据平面：硬件、纳秒级

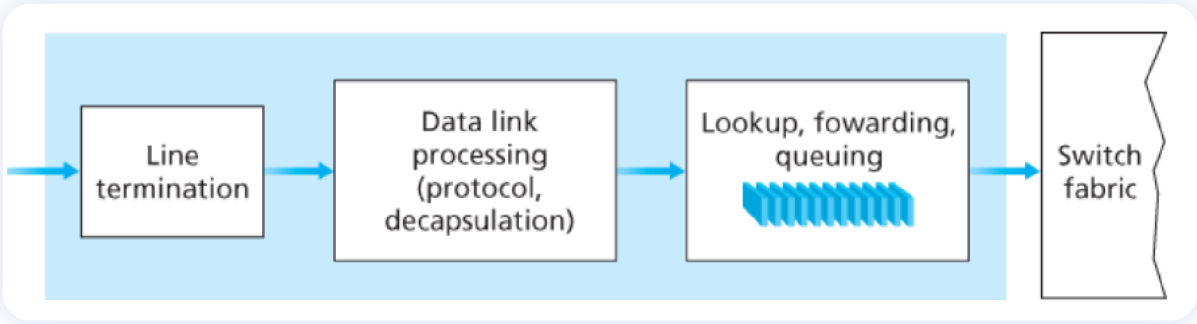
输入端口（Input port）

路由器输入端口是路由器的一个重要组成部分，它负责接收来自输入链路的数据包，并进行物理层和数据链路层的处理，以及转发表的查找和转发功能。路由器输入端口的内部结构通常包括以下几个模块：

- **线路终端模块**，实现了路由器与输入链路的物理层连接，负责数据包的接收和发送。
- **数据链路处理模块**，实现了路由器与输入链路的数据链路层协议，负责数据包的封装和解封装，以及差错检测和恢复。
- **查找/转发模块**，实现了路由器的转发功能，负责根据数据包的目的地址，在转发表中查找匹配的输出端口，并将数据包转发到交换结构中。

路由器输入端口的工作流程如下：

- 当一个数据包到达输入端口时，线路终端模块会将数据包从物理层信号转换为数字比特流，并传送给数据链路处理模块。
- 数据链路处理模块会对数据包进行数据链路层的处理，例如检查数据包的完整性和正确性，如果有差错，可以进行重传或丢弃。如果数据包无误，数据链路处理模块会将数据包的数据链路层首部去掉，只保留网络层的数据包，也就是IP数据报，并传送给查找/转发模块。
- 查找/转发模块会对IP数据报进行转发表的查找，根据最长前缀匹配原则，找出与数据报的目的地址最匹配的转发表项，确定数据报的输出端口。如果没有匹配的转发表项，数据报会被转发到默认的输出端口。**查找/转发模块会在数据报上添加一个标签，表示数据报的输出端口，并将数据报转发到交换结构中。**



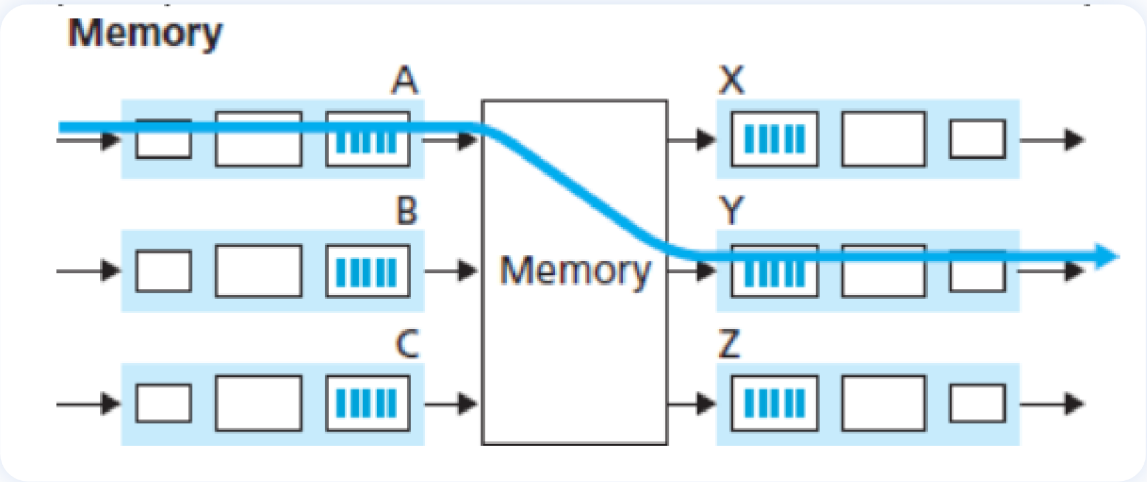
交换结构（Switch fabric）

路由器交换结构是路由器的核心功能，它负责将分组从输入端口转发到合适的输出端口。根据不同的实现方式，路由器交换结构可以分为以下几种类型：

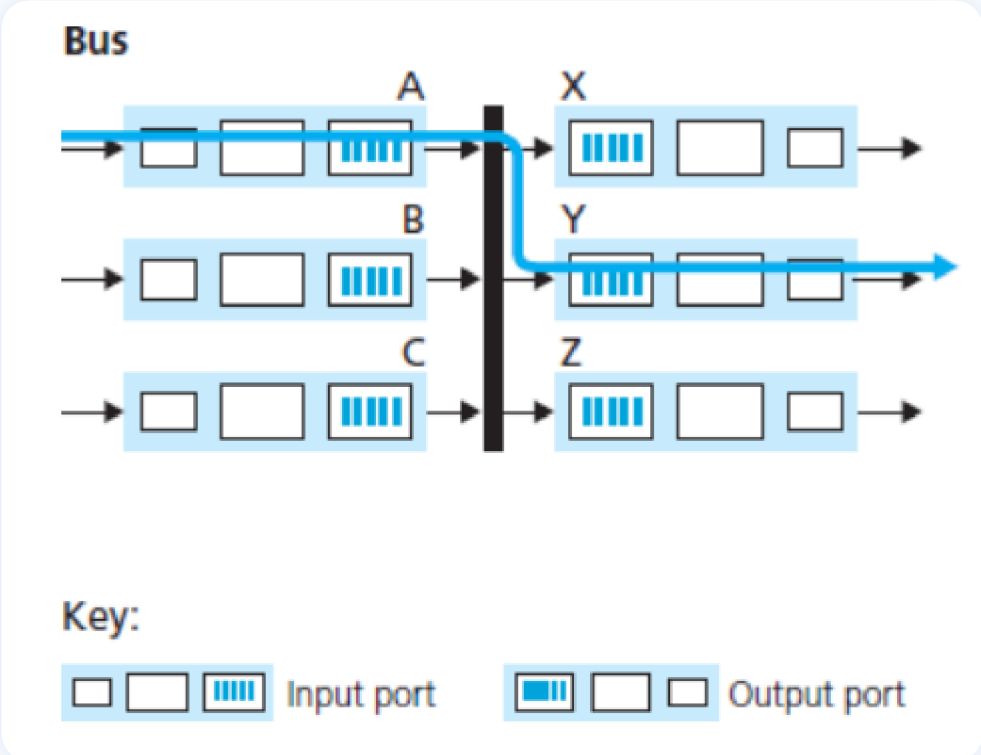
- 通过内存交换：这种方式是利用路由器的内存和CPU来实现分组的转发，输入端口和输出端口相当于I/O设备，路由选择处理器相当于CPU，转发表相当于内存中的数据结构。这种方式的优点是简单，缺点是速度慢，容易造成瓶颈。
- 通过总线交换：这种方式是利用一条共享的总线来连接输入端口和输出端口，输入端口给分组分配一个标签，然后通过总线发送给所有的输出端口，输出端口根据标签判断是否接收分组。这种方式的优点是避免了CPU的参与，缺点是总线的带宽有限，不能同时处理多个分组。
- 通过互联网络交换：这种方式是利用一个**复杂的互联网络来连接输入端口和输出端口**，互联网络由多条垂直和水平的总线组成，每个交叉点由一个交换结构控制器控制，可以开启或关闭。输入端口根据转发表确定输出端口，然后通过互联网络发送分组。这种方式的优点是高速，非阻塞，缺点是复杂，成本高。

三种交换结构：共享内存、共享总线、交叉开关矩阵

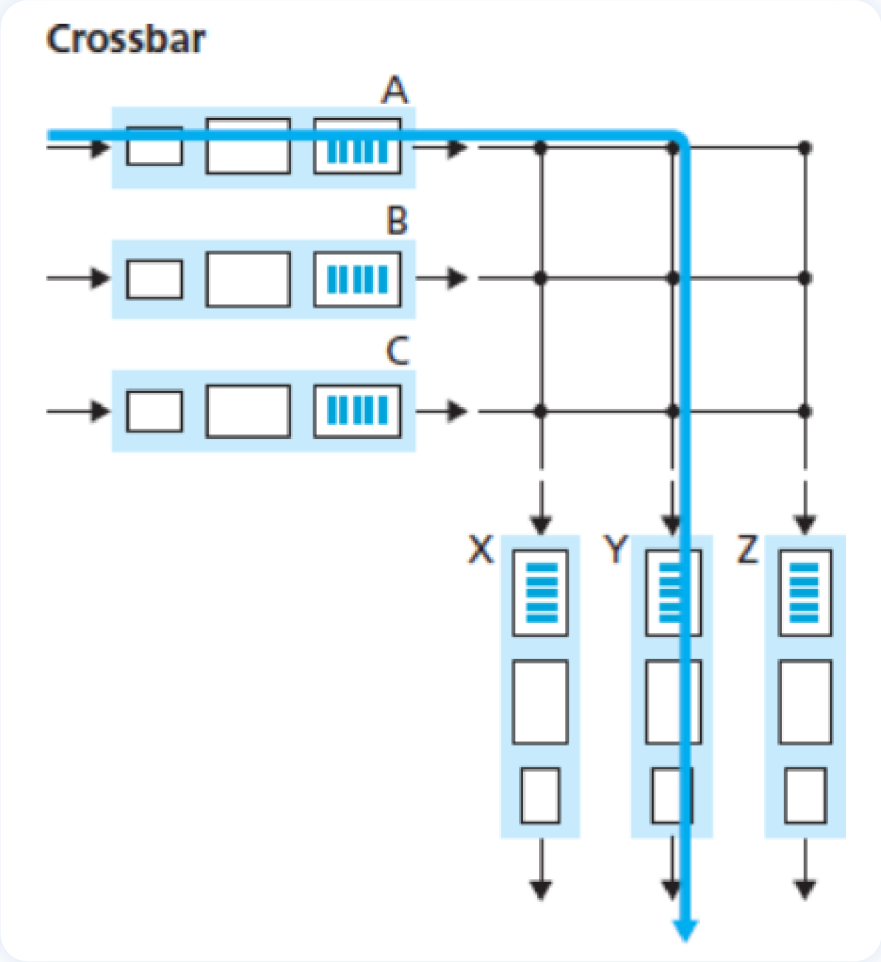
- 共享内存（Switching via memory）
  - 初代路由常采用共享内存
  - pkt在memory复制
  - 速度被内存带宽限制，每个datagram都要经过2个bus
  - Cisco Catalyst 8500



- 共享总线（Switching via a bus）
  - datagram通过共享的bus从输入端口到输出端口
  - 总线冲突（bus contention）：交换速度被总线带宽限制
  - Cisco 5600

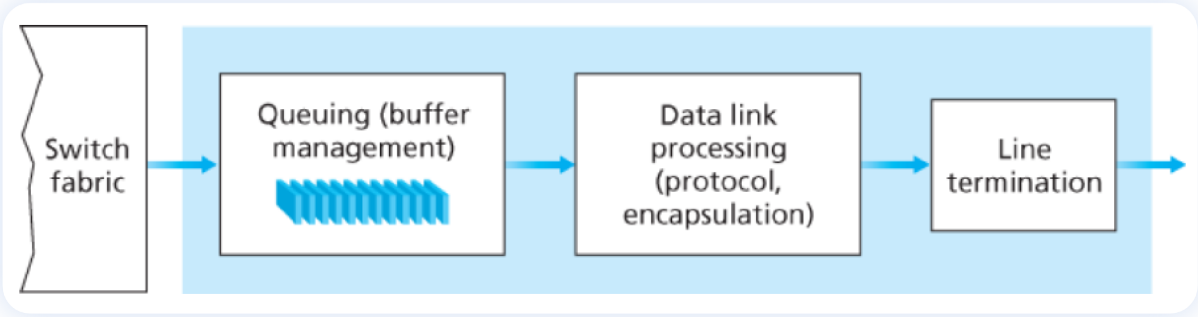


- 交叉开关矩阵（Switching via an interconnection network）
  - 客服总线带宽的限制
  - 纵横式交换机是一种由2N条总线组成的互联网络，连接N个输入端口和N个输出端口
  - 可并行转发多个分组，但如果同输入输出端口，还是必须等待前一个发送了再发下一个
  - Cisco 12000



输出端口（Output port）

1. 排队（缓存管理）（Queueing(datagram buffer)）
   - 缓存：当数据报到达快于传输速率，如果缓存不够会导致丢包
   - 调度方法：从排队中选择数据报进行传输，例如按时间顺序、按优先级顺序
2. 数据链路处理（协议，封装）（Data link processing(protocol, encapsulation)）
3. 线路端接（Line termination）



排队

输入端口排队

- 当fabric比input port慢时，发生排队
- buffer满时，出现排队延时和丢包
- 队首阻塞（Head-of-the-Line blocking，HOL）：队首的datagram会阻塞排在它后面的datagram输入

输出端口排队

- 当switch到达比output line快，发生排队
- buffer满时，出现排队延时和丢包