《计算机网络原理》

课程编号：40240513

讲课教师：吴建平 徐明伟 尹霞

本科生必修课

计算机科学与技术系

主要教学内容和学时分配

第一章	引言	3
第二章	计算机网络的体系结构	6
第三章	数据通信的基本原理	3
第四章	物理层	3
第五章	数据链路层	9
第六章	局域网和介质访问控制	6
第七章	网络层	6
第八章	传送层	3
第九章	计算机网络应用	6
第十章	计算机网络新技术/复习	3
共计		48

第七章 网络层

第三部分

主要内容（1）

7.1 网络层概述

7.2 路由算法

- 7.2.1 最优化原则
- 7.2.2 最短路径路由算法
- 7.2.3 洪泛算法
- 7.2.4 基于流量的路由算法
- 7.2.5 距离向量路由算法
- 7.2.6 链路状态路由算法
- 7.2.7 分层路由
- 7.2.8 移动主机的路由

主要内容（2）

7.3 拥塞控制算法

- 7.3.1 拥塞控制的基本原理
- 7.3.2 拥塞控制算法

7.4 网络互连

- 7.4.1 级联虚电路
- 7.4.2 无连接网络互连
- 7.4.3 隧道技术
- 7.4.4 互联网路由
- 7.4.5 分段
- 7.4.6 防火墙

主要内容（3）

7.5　互联网网络层协议

7.5.1　IPv4和IPv6协议

7.5.2　互联网控制协议

7.5.3　内部网关路由协议：OSPF

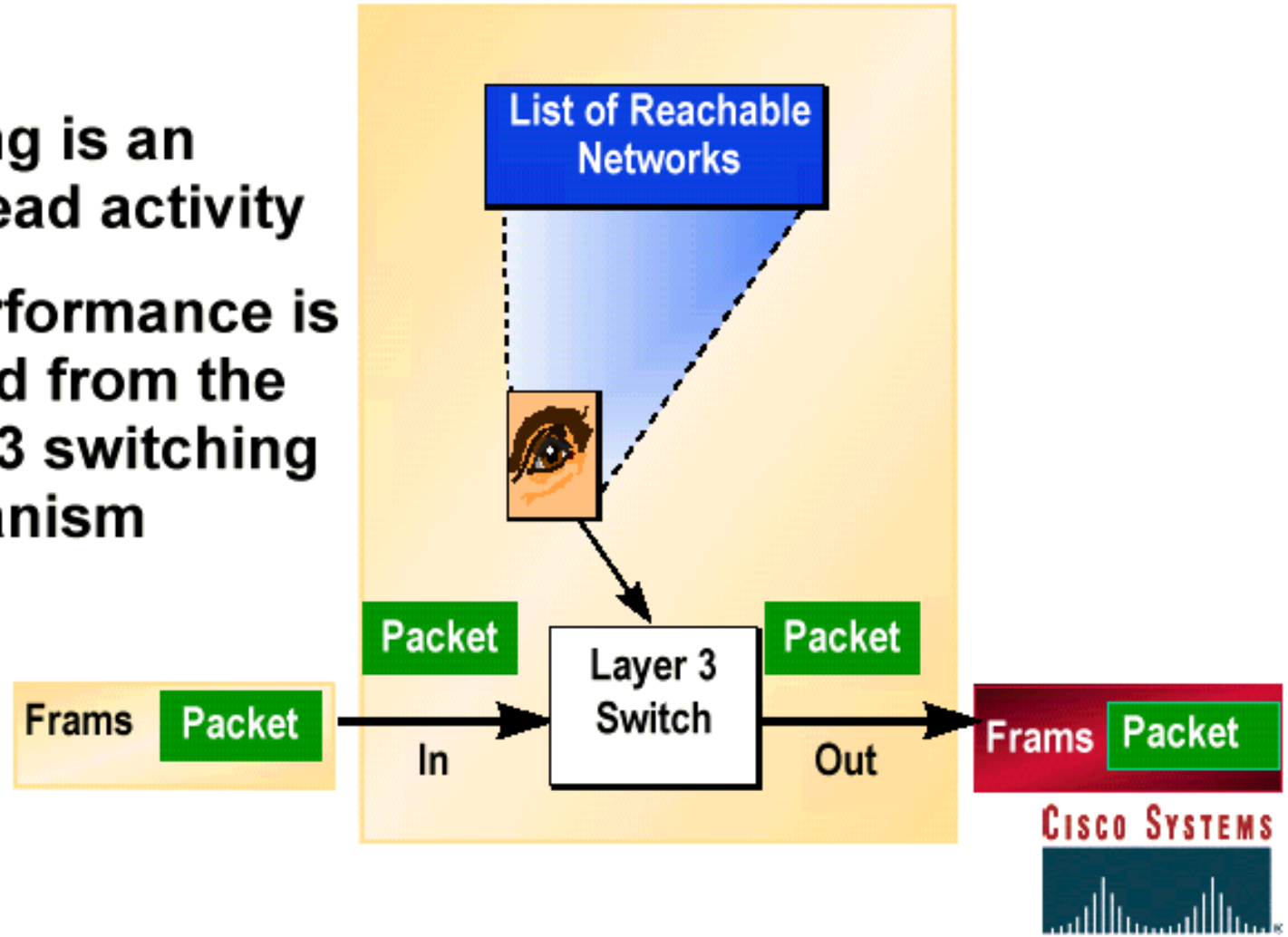
7.5.4　外部网关路由协议：BGP

7.6　路由器体系结构和关键技术

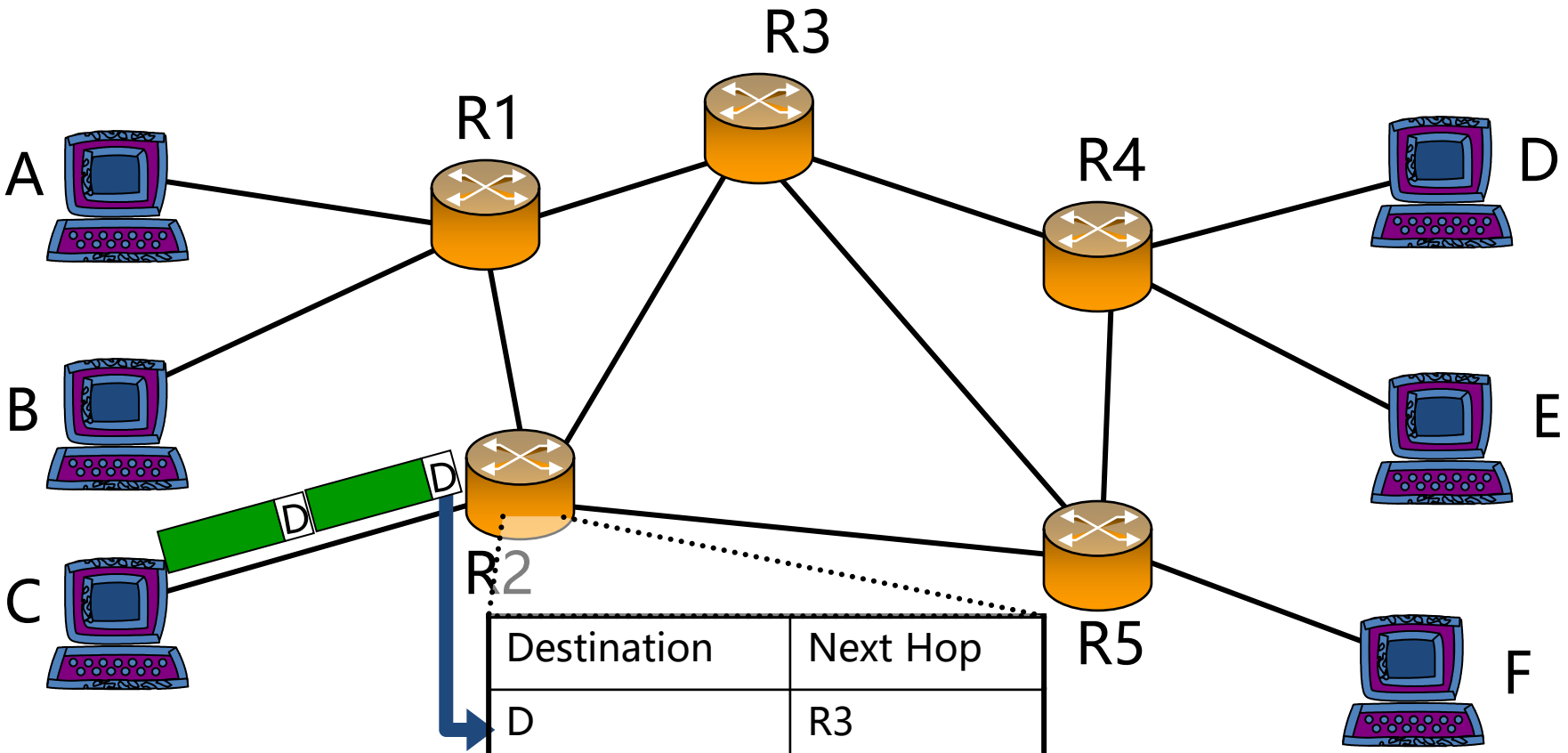
路由器体系结构和关键技术

What Is a Router?

- Routing is an overhead activity
- All performance is derived from the Layer 3 switching mechanism

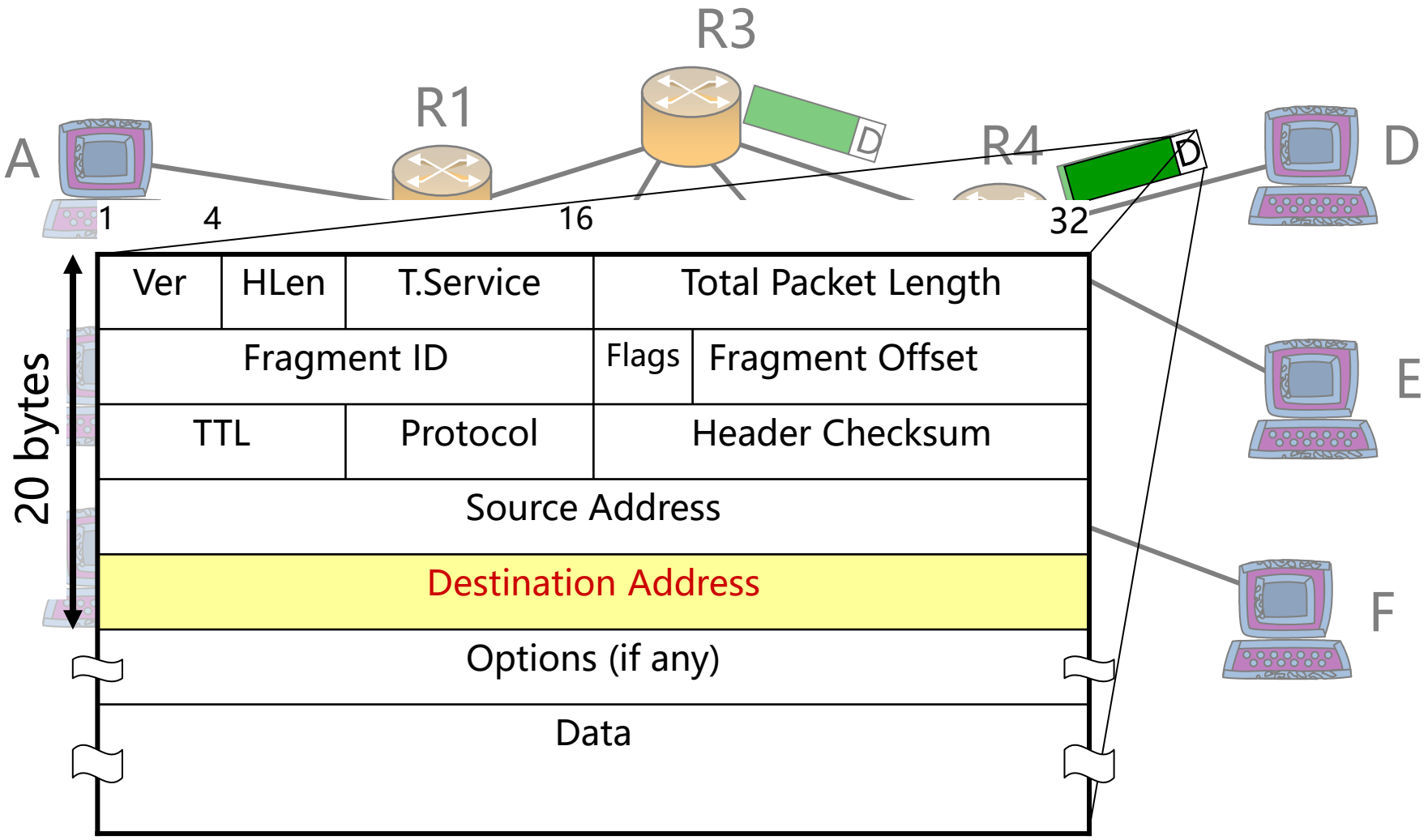


什么是路由

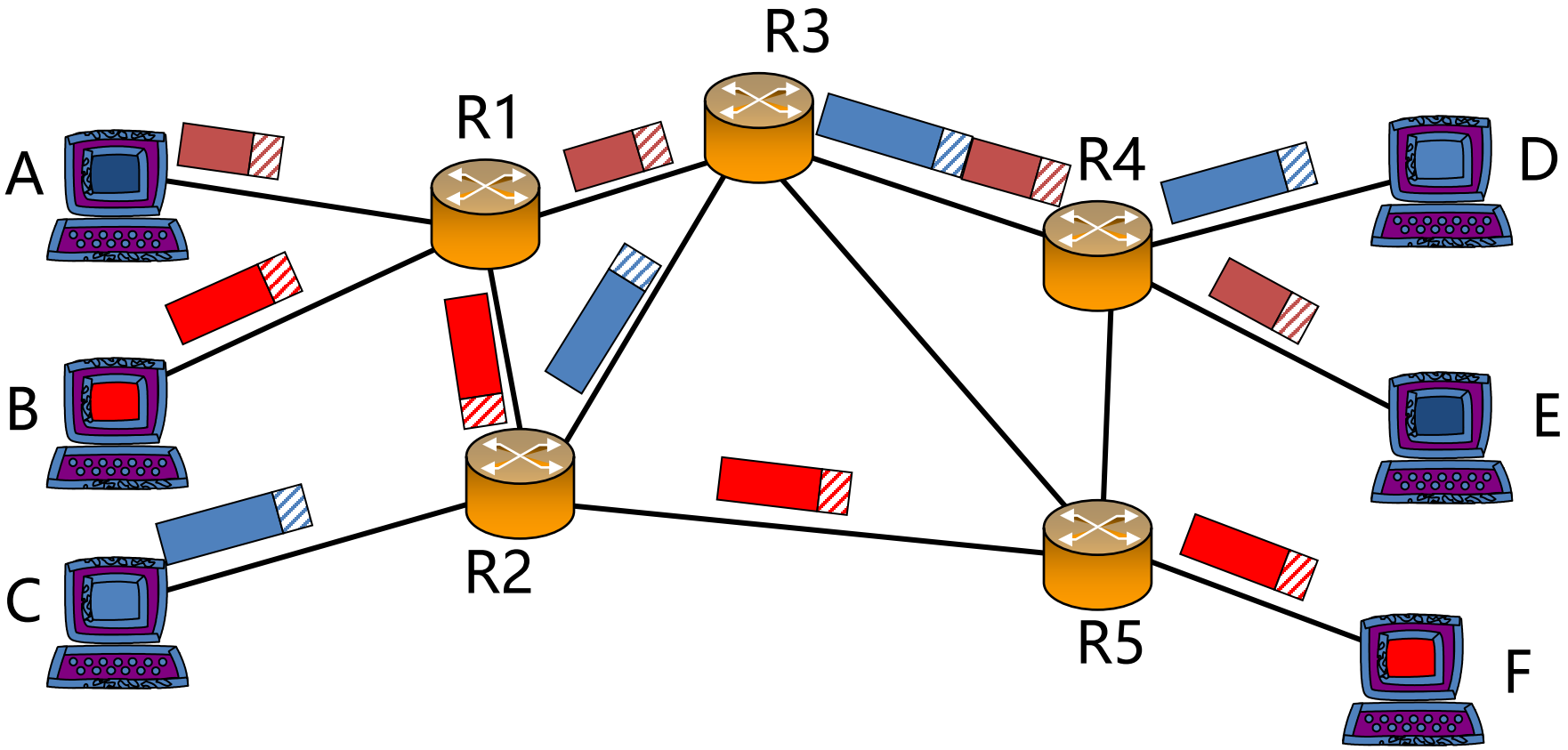


Destination	Next Hop
D	R3
E	R3
F	R5

什么是路由

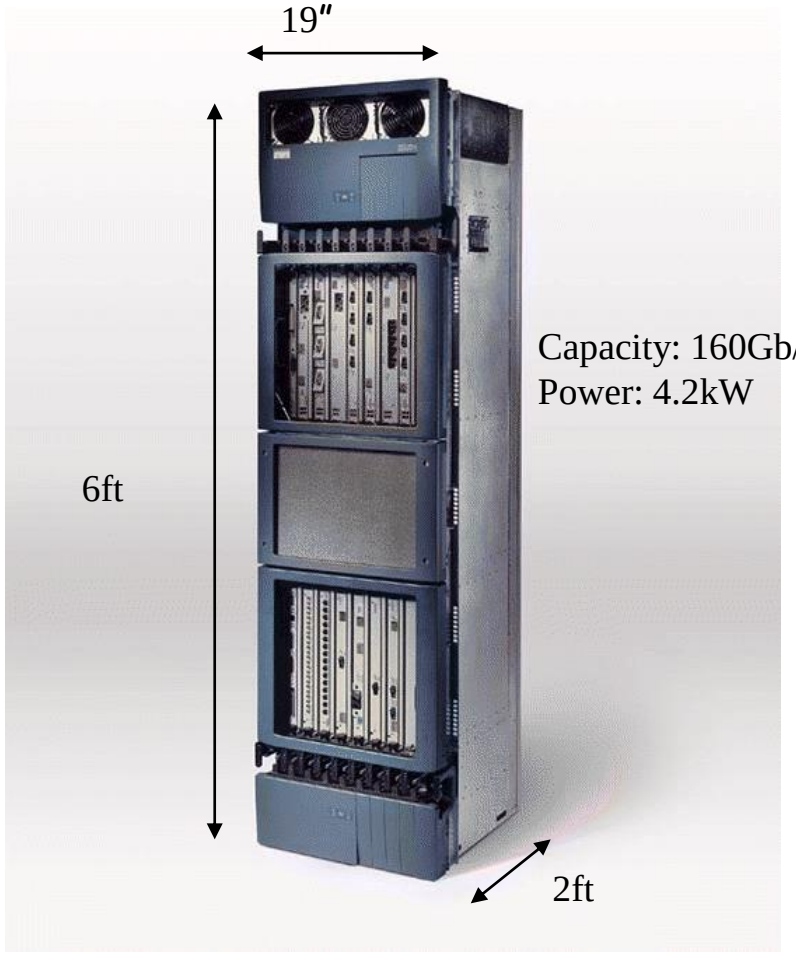

R3
R1
A
D
R4
D
D
1
4
16
32
20 bytes
Ver
HLen
T.Service
Total Packet Length
Fragment ID
Flags
Fragment Offset
TTL
Protocol
Header Checksum
Source Address
Destination Address
Options (if any)
Data
E
F

什么是路由



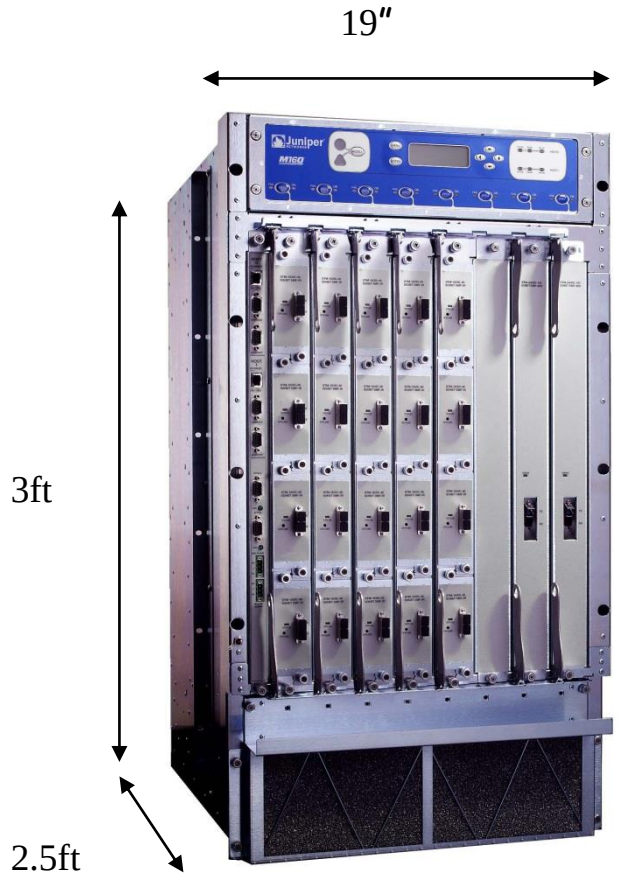
路由器是什么样子

Cisco GSR 12416



Capacity: 160Gb/s
Power: 4.2kW

Juniper M160



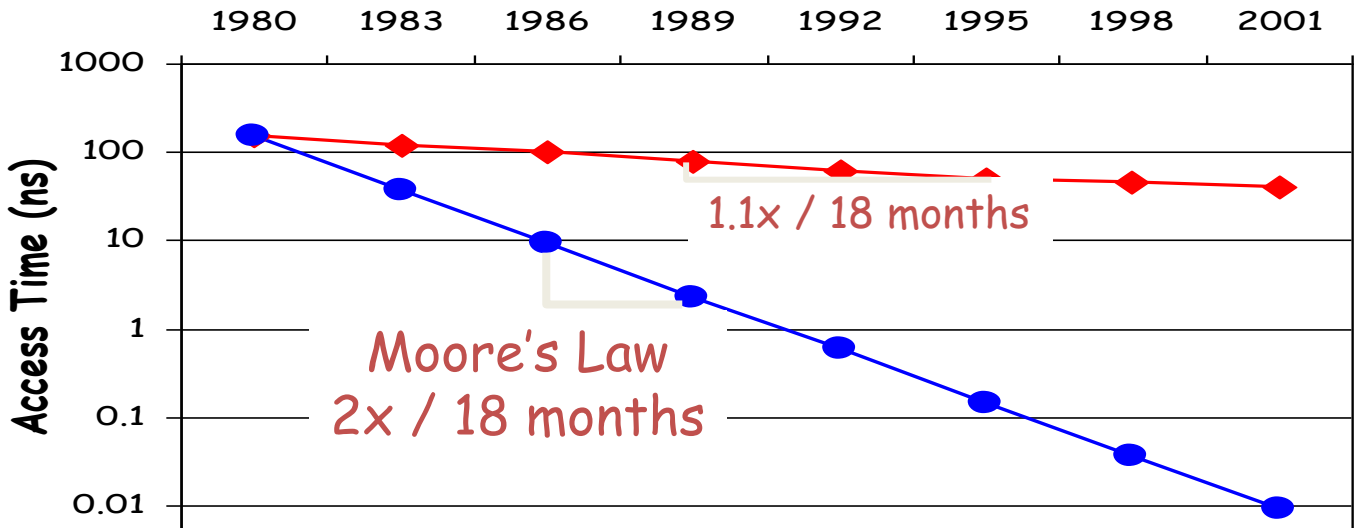
Capacity: 80Gb/s
Power: 2.6kW

为什么设计制造高速路由器有很大的难度?

1. 内存速度是瓶颈，没法跟上摩尔定律。：
2. 对路由器性能要求的提升速度超过了摩尔定律，带宽增加速度超过了处理能力增加的速度。

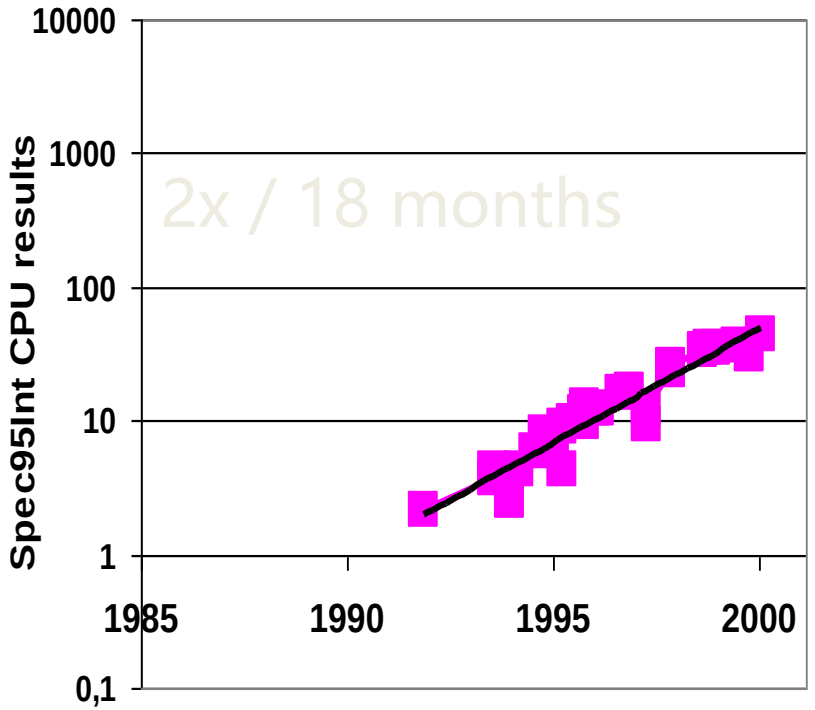
为什么设计制造高速路由器有很大的难度?

1. **It's hard to keep up with Moore's Law:**
   - The bottleneck is memory speed.
   - Memory speed is not keeping up with Moore's Law.

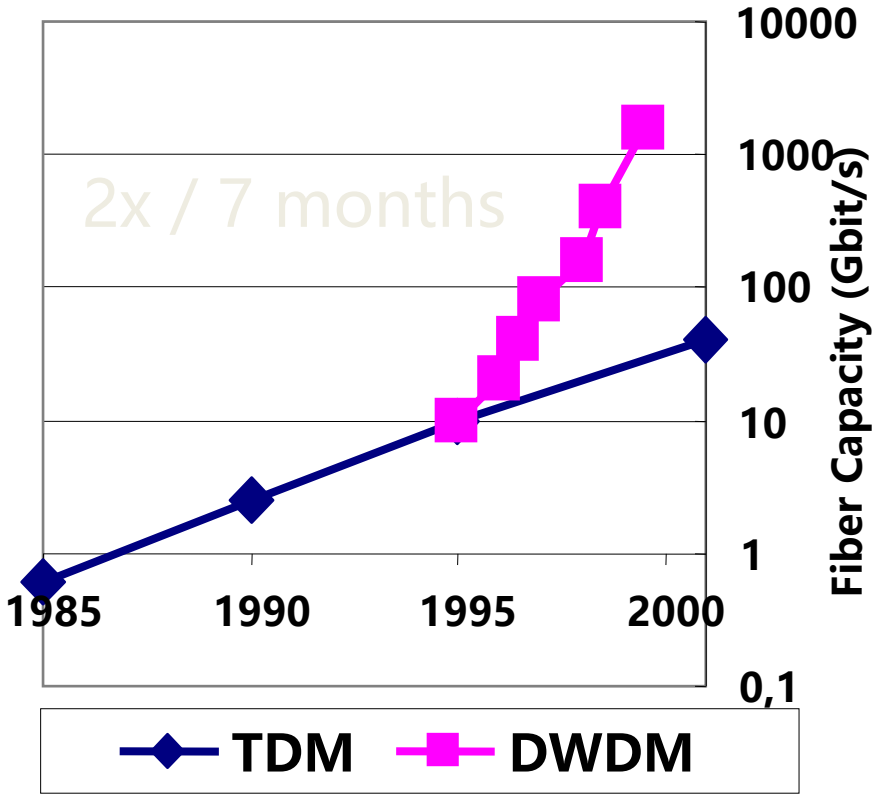


摩尔定律和光纤定律

报文处理能力



链路速度



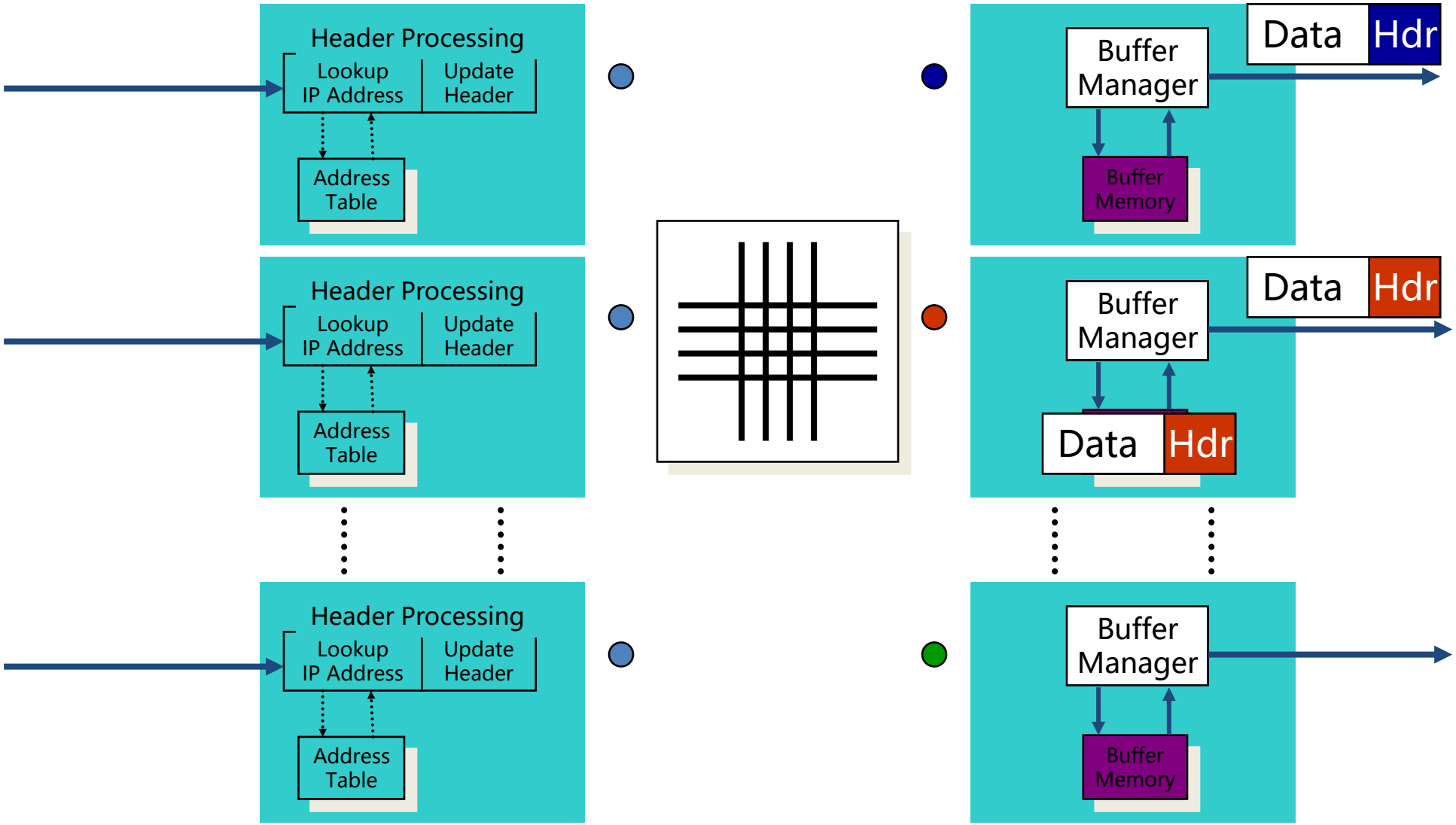
Source: SPEC95Int & David Miller, Stanford.

路由器的性能增长超过了摩尔定律

- 商业路由器的容量增长
  - Capacity 1992 ~ 2Gb/s
  - Capacity 1995 ~ 10Gb/s
  - Capacity 1998 ~ 40Gb/s
  - Capacity 2001 ~ 160Gb/s
  - Capacity 2003 ~ 640Gb/s

- 平均增长率: 2.2x / 18 months.

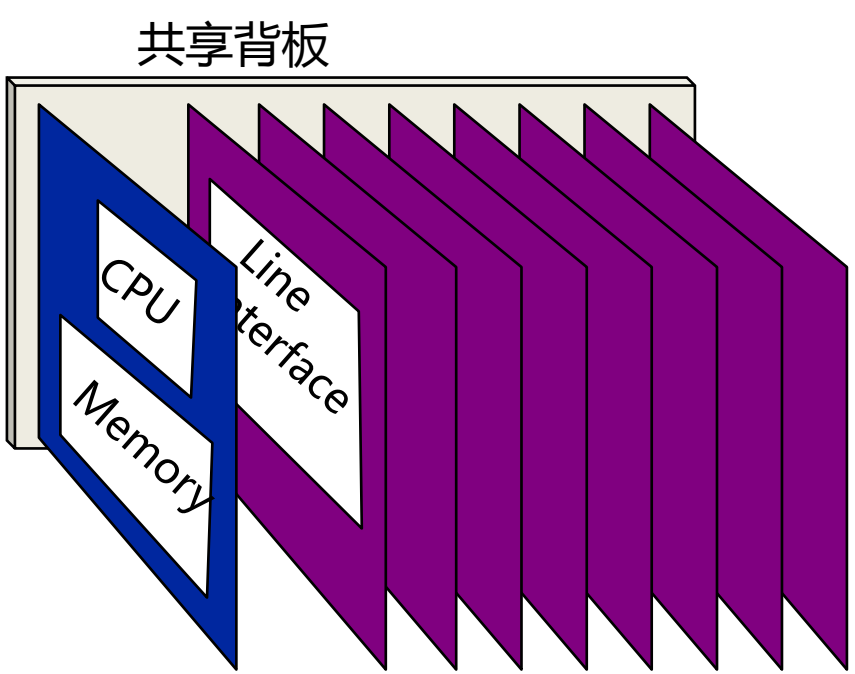
通用路由器框架

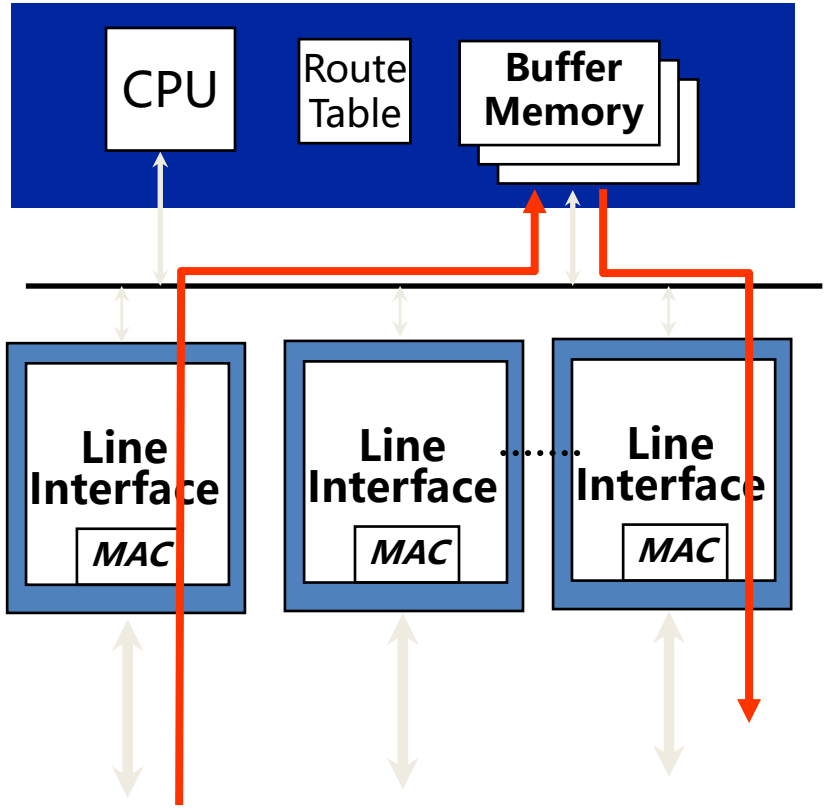

Header Processing
Lookup IP Address
Update Header
Address Table
Header Processing
Lookup IP Address
Update Header
Address Table
Header Processing
Lookup IP Address
Update Header
Address Table
Buffer Manager
Buffer Memory
Data
Hdr
Buffer Manager
Buffer Memory
Data
Hdr
Data
Hdr
Buffer Manager
Buffer Memory

路由器体系结构的发展

- 单总线，单处理器
  - 传统的计算机结构，处理器成为处理瓶颈
- 单总线，多处理器，每个接口卡上有一个处理器，主处理器负责协调。
  - 包转发由本地处理器完成，减少总线负担
- 交换结构 + 专用硬件（ASIC，FPGA，NP）
- 交换结构实现无阻塞转发
  - 硬件实现对IP包的处理和路由查找
- Internet Routing Matrix
  - 多结点级连，类似并行计算机

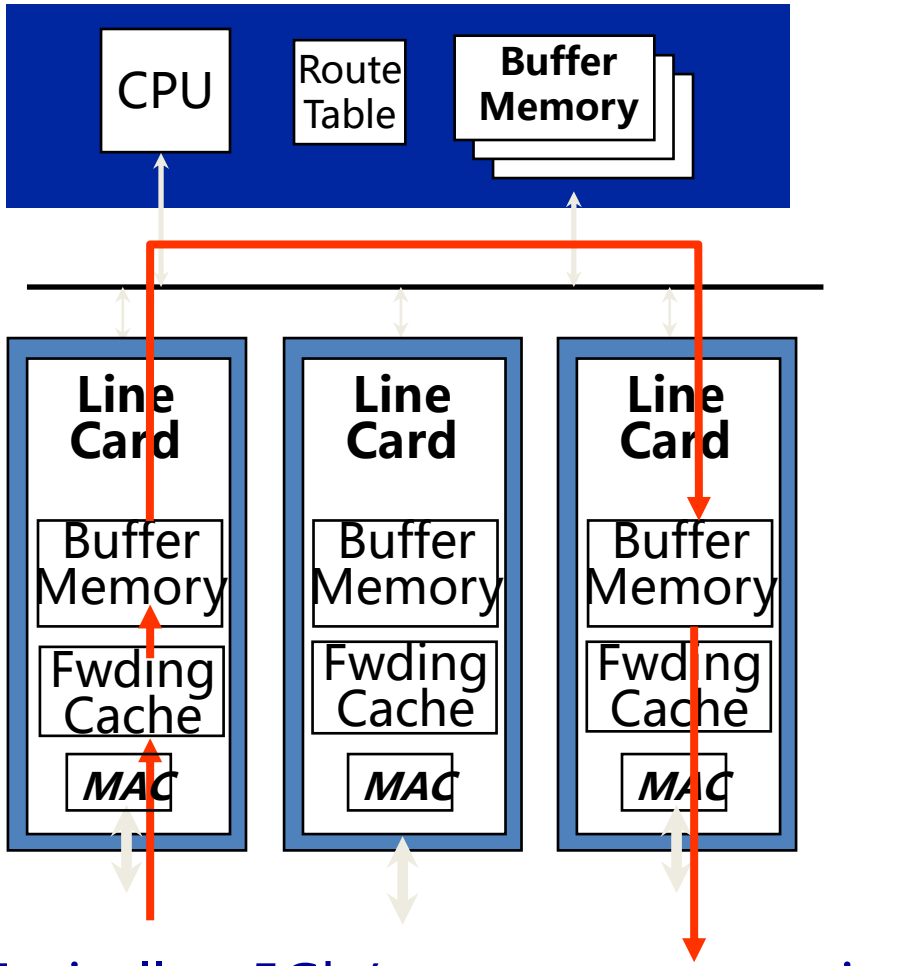
第一代路由器





Typically <0.5Gb/s aggregate capacity

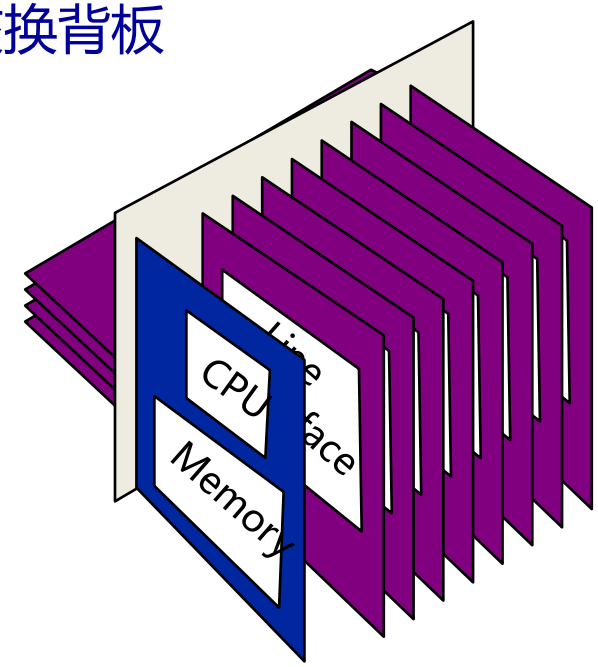
第二代路由器

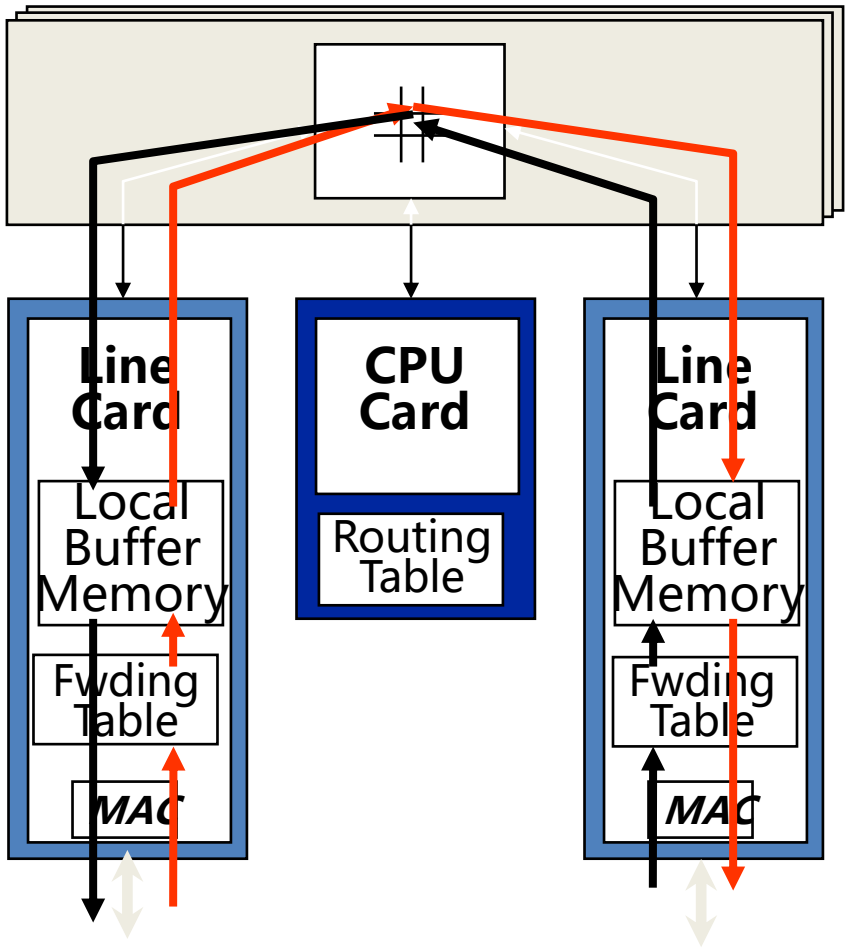


Typically <5Gb/s aggregate capacity

第三代路由器

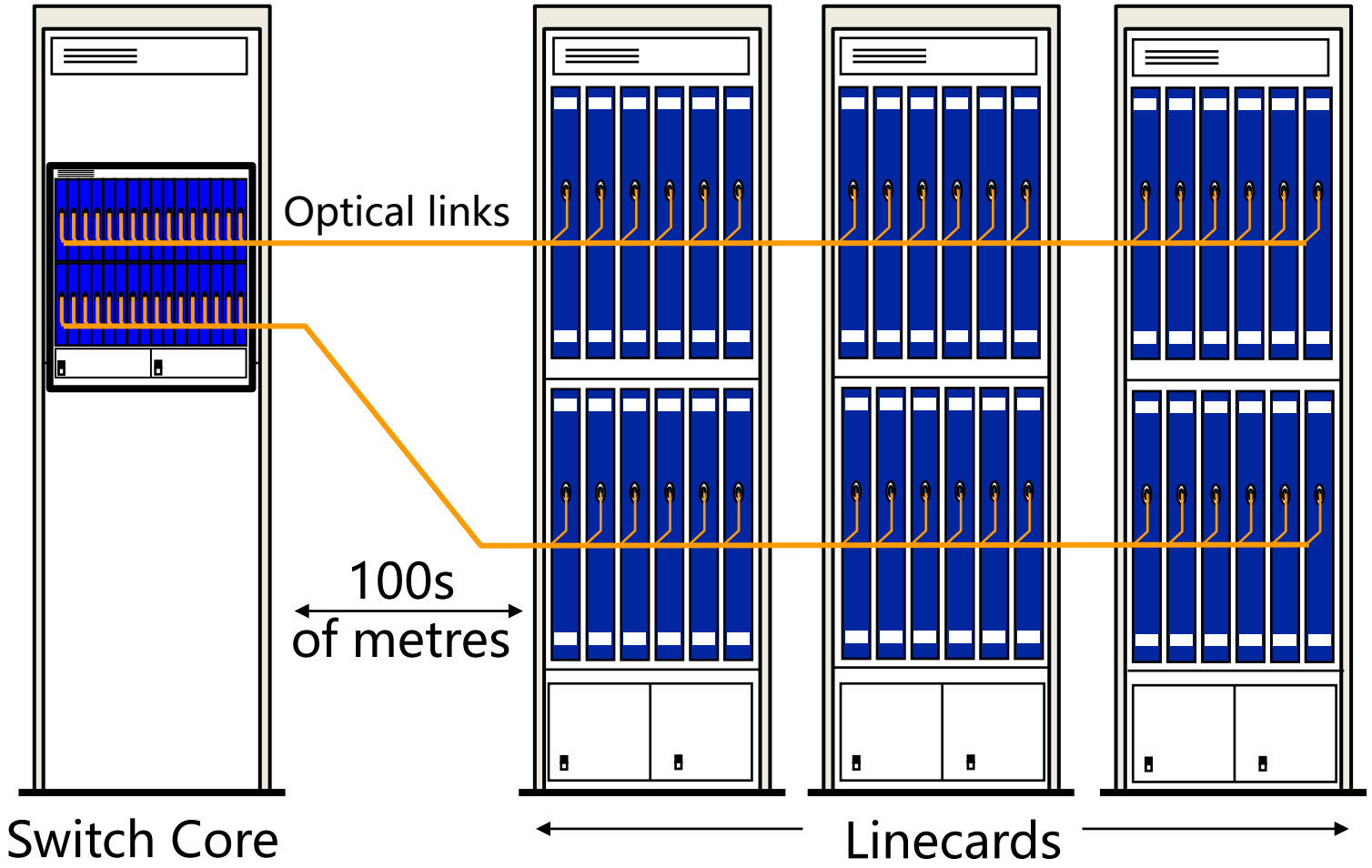
交换背板





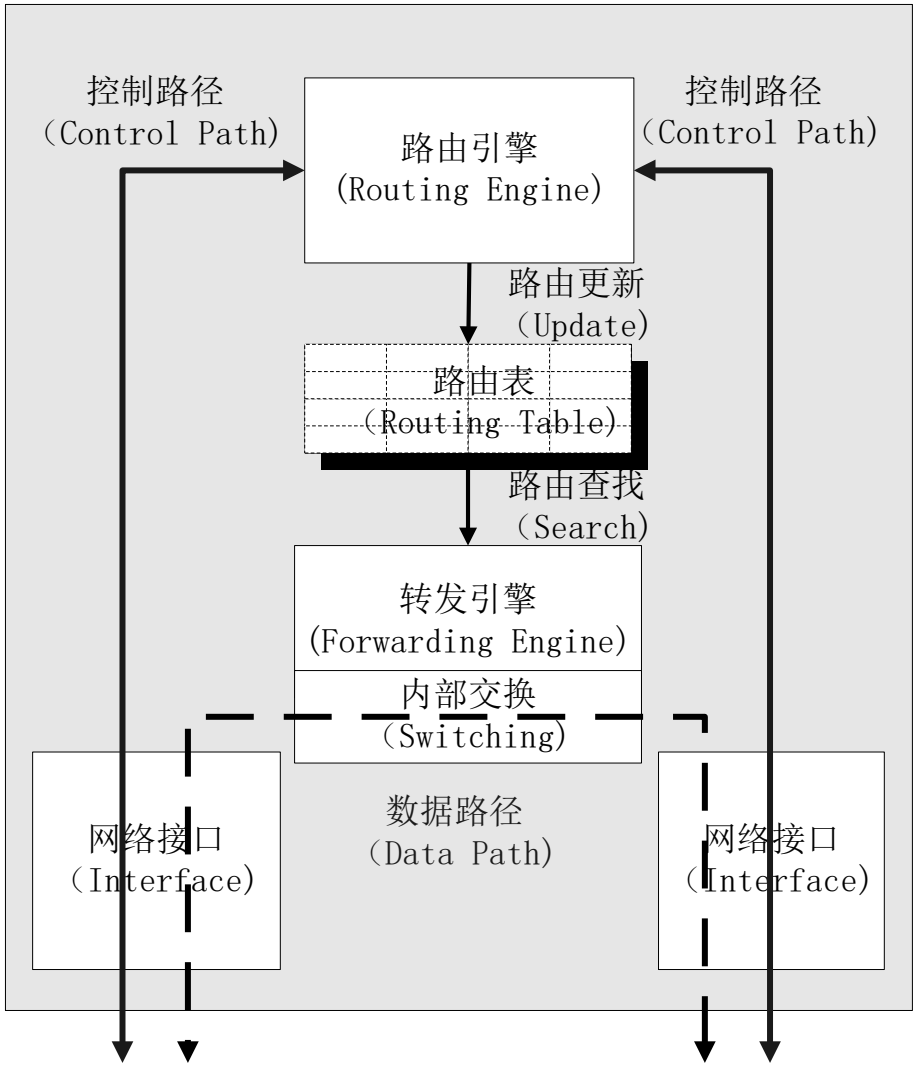
Typically <50Gb/s aggregate capacity

第四代路由器/交换机



0.3 - 10Tb/s routers in development

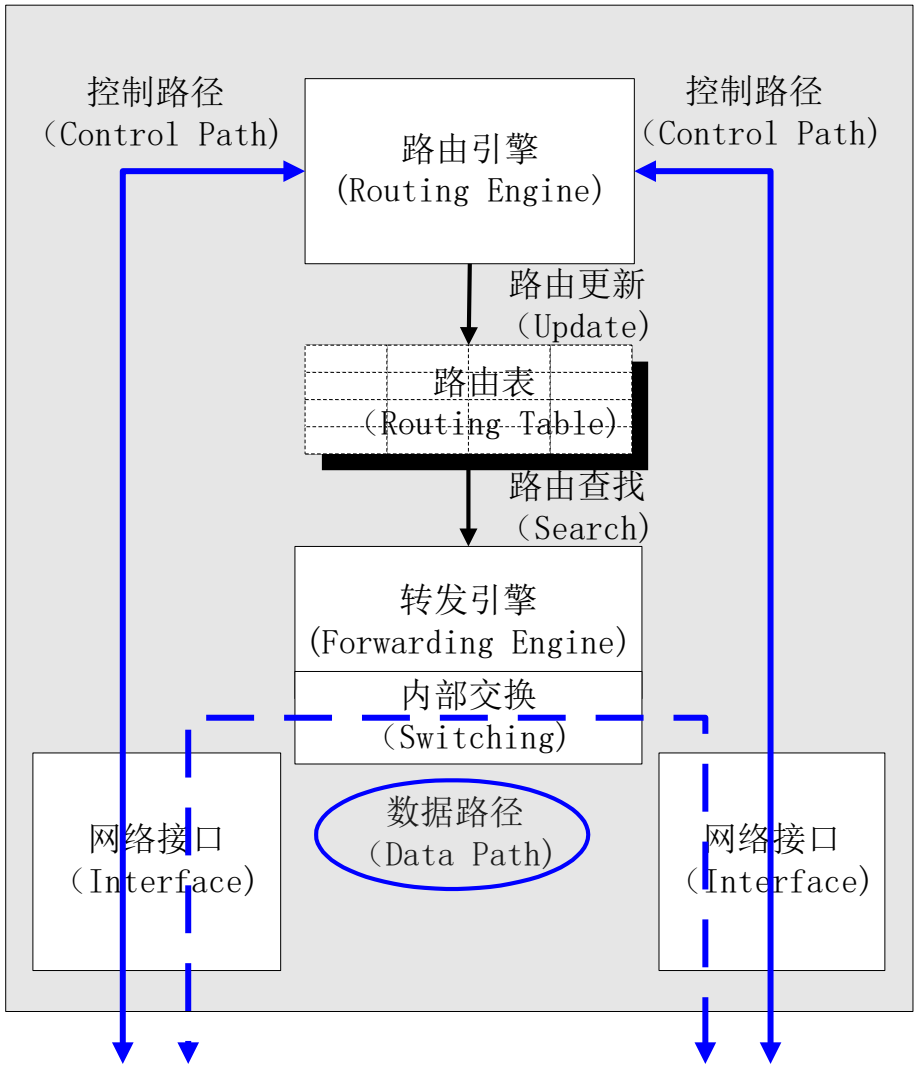
路由器基本结构


控制路径
（Control Path）
路由引擎
(Routing Engine)
控制路径
（Control Path）
路由更新
（Update）
路由表
（Routing Table）
路由查找
（Search）
转发引擎
(Forwarding Engine)
内部交换
（Switching）
网络接口
（Interface）
数据路径
（Data Path）
网络接口
（Interface）

路由器基本结构

- 网络接口
  - 完成网络报文的接收和发送。
- 转发引擎
  - 负责决定报文的转发路径。
- 内部交换
  - 为多个网络接口以及路由引擎模块之间的报文数据传送提供高速的数据通路。
- 路由引擎
  - 由运行高层协议（特别是路由协议）的内部处理模块组成。
- 路由表
  - 路由表包含了能够完成网络报文正确转发的所有路由信息，它在整个路由器系统中起着承上启下的作用。

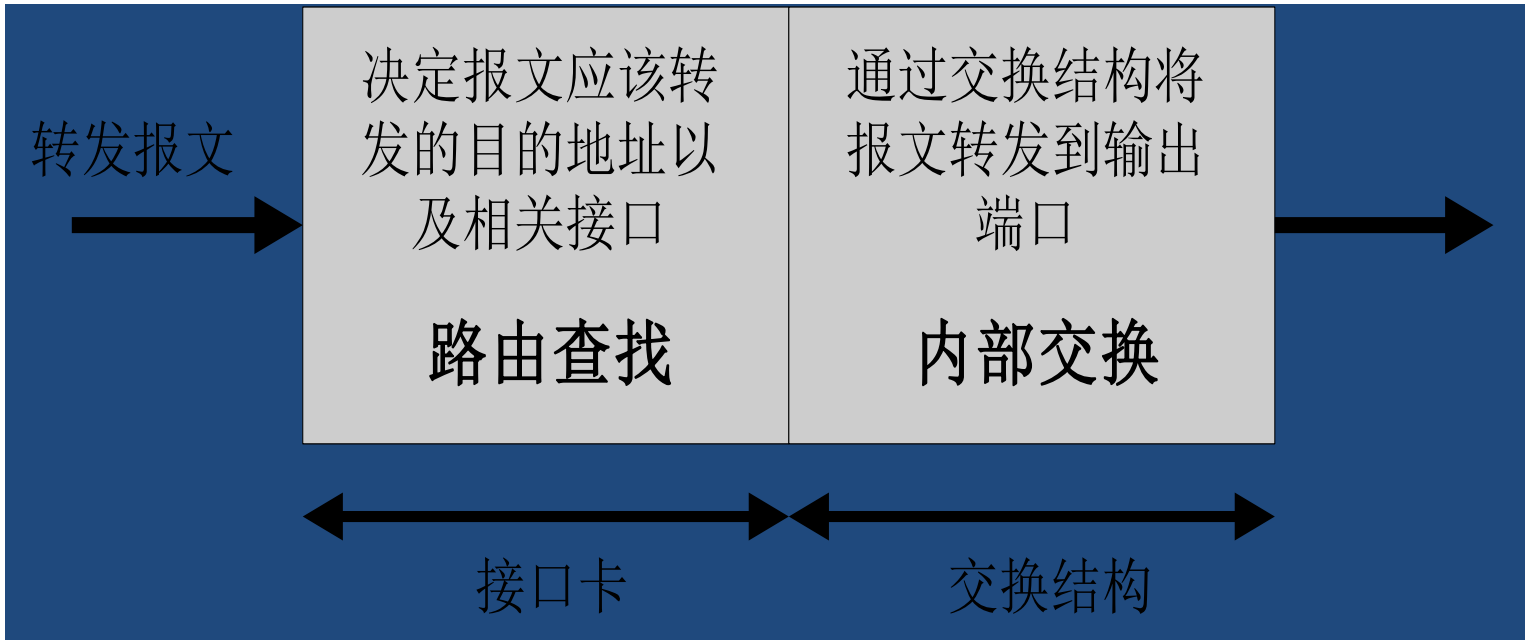
报文处理路径


控制路径
(Control Path)
路由引擎
(Routing Engine)
控制路径
(Control Path)
路由更新
(Update)
路由表
(Routing Table)
路由查找
(Search)
转发引擎
(Forwarding Engine)
内部交换
(Switching)
网络接口
(Interface)
数据路径
(Data Path)
网络接口
(Interface)

报文处理路径

- 路由器提供了两种不同的报文处理路径：
  - **数据路径**：处理目的地址不是本路由器而需要转发的报文，因此数据路径是整个路由器的关键路径，它的实现好坏直接影响着路由器的整体性能
  - **控制路径**：处理目的地址是本路由器的高层协议报文，特别是各种路由协议报文。虽然控制路径不是路由器的关键路径，但是它负责完成路由信息的交互，从而保证了数据路径上的报文沿着最优的路径转发。

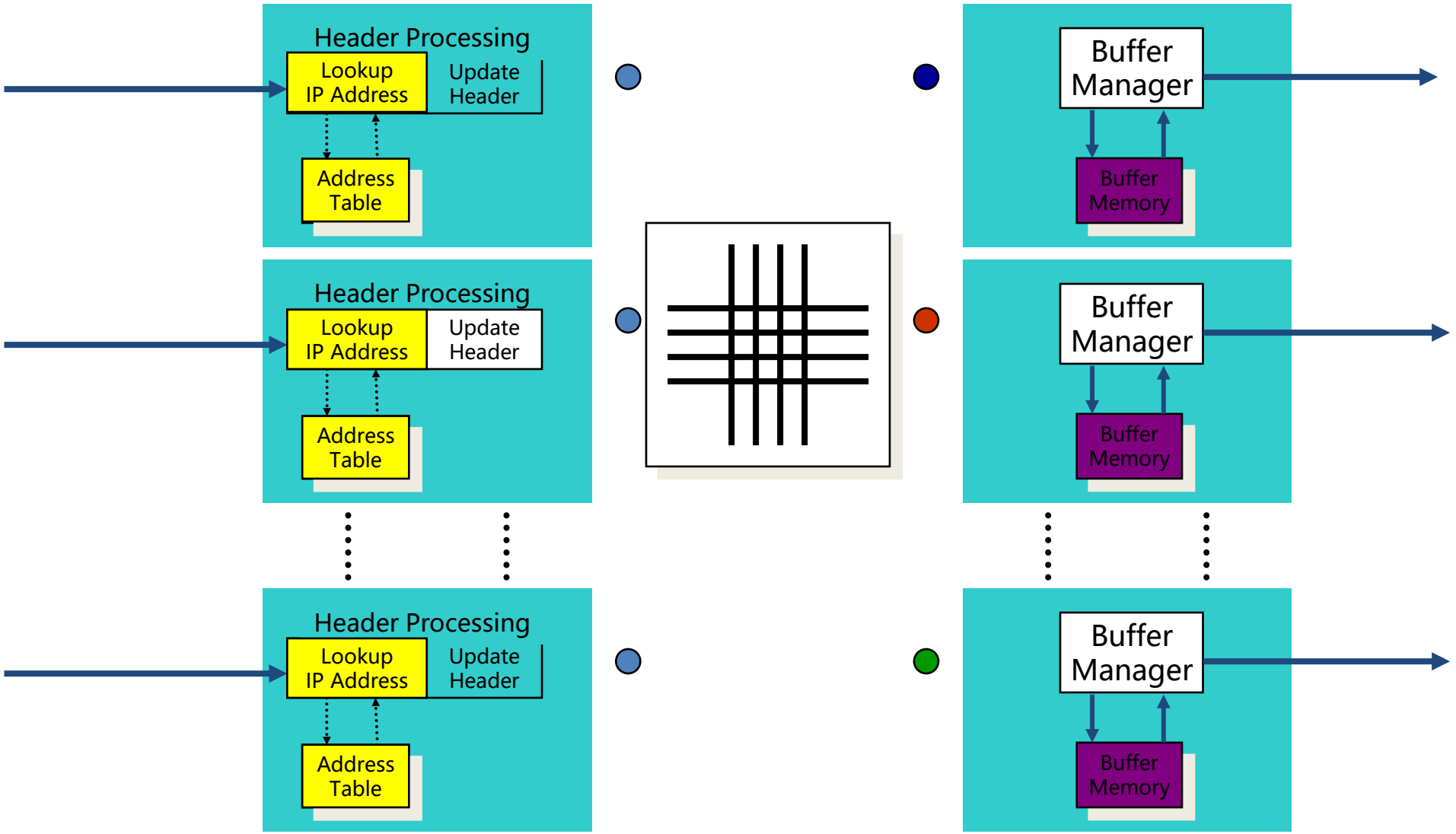
数据路径的工作流程


转发报文
决定报文应该转发的目的地址以及相关接口
路由查找
通过交换结构将报文转发到输出端口
内部交换
接口卡
交换结构

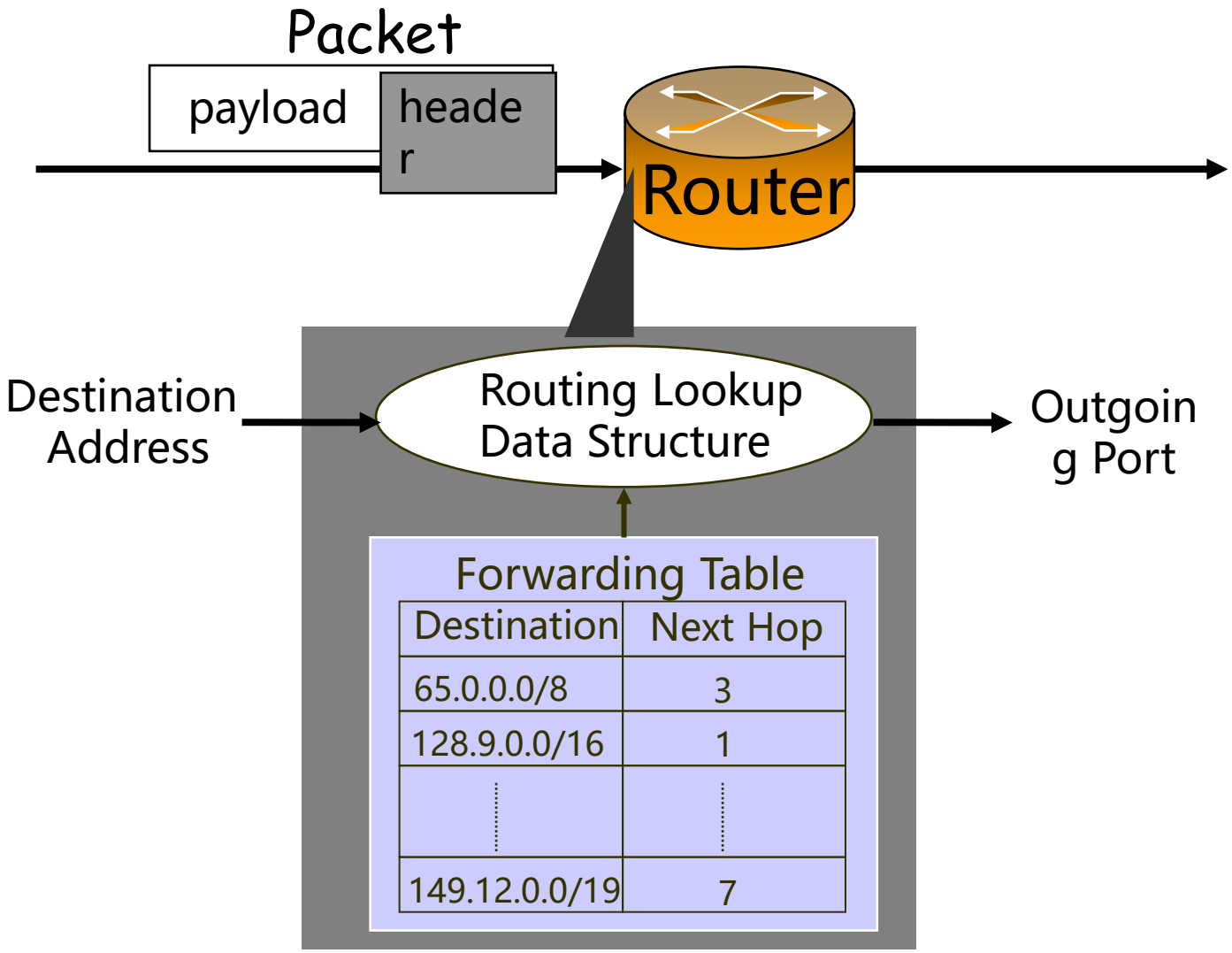
数据路径的工作流程

- RFC1812规定IP路由器必须完成两个基本功能：
  - 首先路由器必须能够对每个到达本路由器的报文做出正确的转发决策，决定报文向哪一个下一跳路由器转发。为了进行正确的转发决策，路由器需要在转发表中查找能够与转发报文目的地址最佳匹配的表项，这个查找过程被称为路由查找（Route Lookup）。
  - 其次路由器在得到了正确的转发决策之后必须能够将报文从输入接口向相应的输出接口传送，这个过程被称为内部交换过程（Switching）。

IP地址查找


Header Processing
Lookup IP Address
Update Header
Address Table
Header Processing
Lookup IP Address
Update Header
Address Table
Header Processing
Lookup IP Address
Update Header
Address Table
Buffer Manager
Buffer Memory
Buffer Manager
Buffer Memory
Buffer Manager
Buffer Memory

路由查找过程示意图


Packet
payload
heade
r
Router
Destination
Address
Routing Lookup
Data Structure
Outgoin
g Port
Forwarding Table
Destination
Next Hop
65.0.0.0/8
3
128.9.0.0/16
1
149.12.0.0/19
7

IP Address Lookup

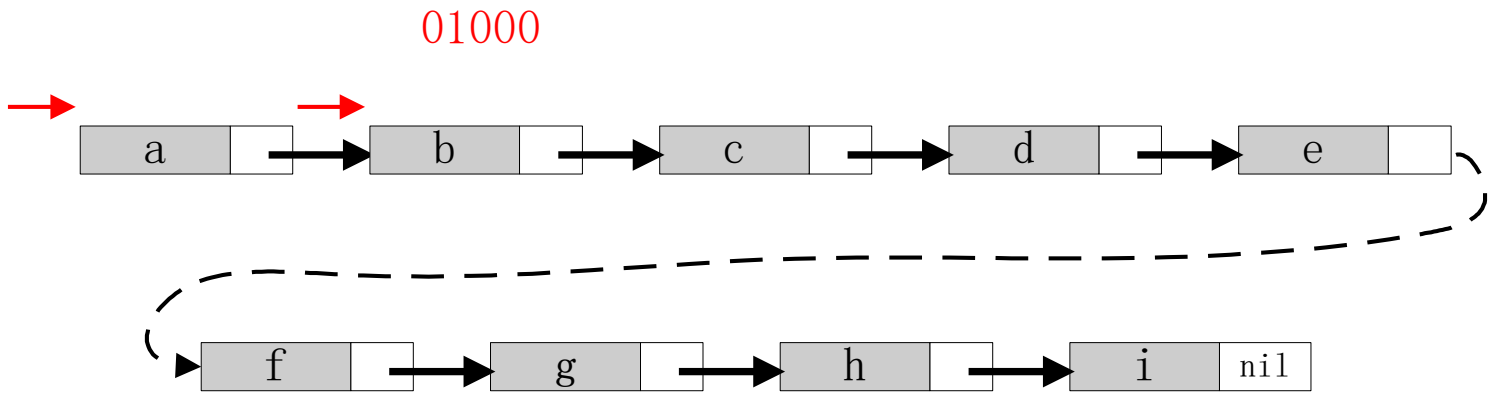
- 路由查找的难点在于:
  1. 不是精确匹配，是最长匹配 .
  2. 路由表很大，目前大约是80万条表项，并且还在不断增长 .
  3. 路由查找必须很快：对应10Gb/S的链路，速度要求是30ns .

路由查找算法的分类

- 基于地址前缀值的路由查找算法
  - 通过对整个地址前缀空间进行地址关键字穷举法来避免对地址前缀长度进行考虑。
  - 线性查找法、地址区间的二分查找法、TCAM硬件查找法
- 基于地址前缀长度的路由查找算法
  - 从前缀长度的角度入手进行路由查找
  - trie树(包括二分支、多分支)、前缀长度空间的二分查找法

线性查找法

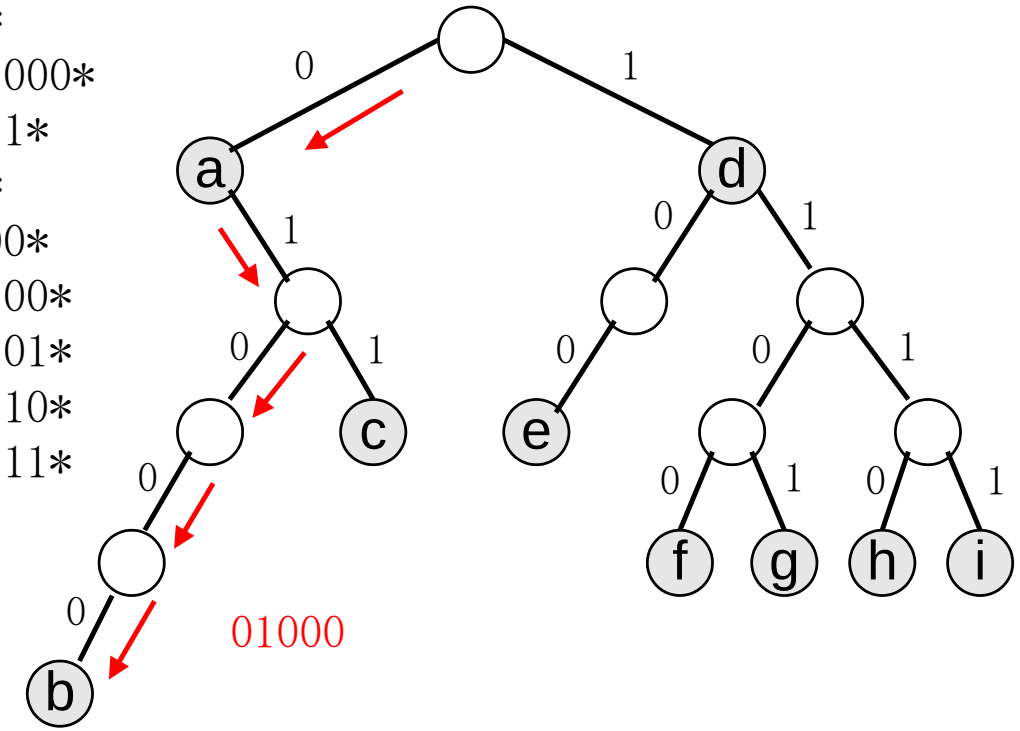
Prefixes
a 0*
b 01000*
c 011*
d 1*
e 100*
f 1100*
g 1101*
h 1110*
i 1111*



- 实现简单
- 查找效率低，查找过程的算法复杂度为O（N)
- 存储空间复杂度为O(N)，插入删除复杂度为O(1)
- 一种改进：将前缀长度大的路由表项放在链表的前列，平均查找性能会有改进，但是路由更新复杂度变为O(N)
- 只适用于路由表项比较少的早期低端路由器

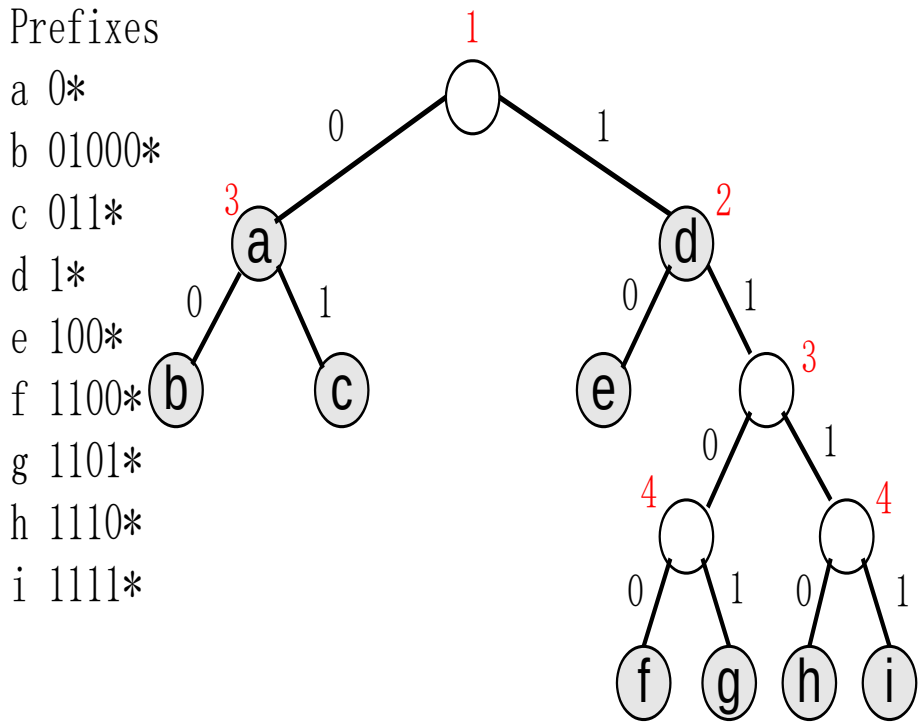
二分支树查找法（A Binary Trie）

Prefixes
a 0*
b 01000*
c 011*
d 1*
e 100*
f 1100*
g 1101*
h 1110*
i 1111*



- 数字查找树的每个结点不是关键字，而是组成关键字的符号
- 实现简单
- 适用性好，可以用于任何长度关键字的查找
- 查找效率低，最差情况下O(W)，W = logN
- 存储效率较低
- Trie树查找过程实际上就是在长度空间内的顺序查找操作

路径压缩树（Path-Compressed Trie）

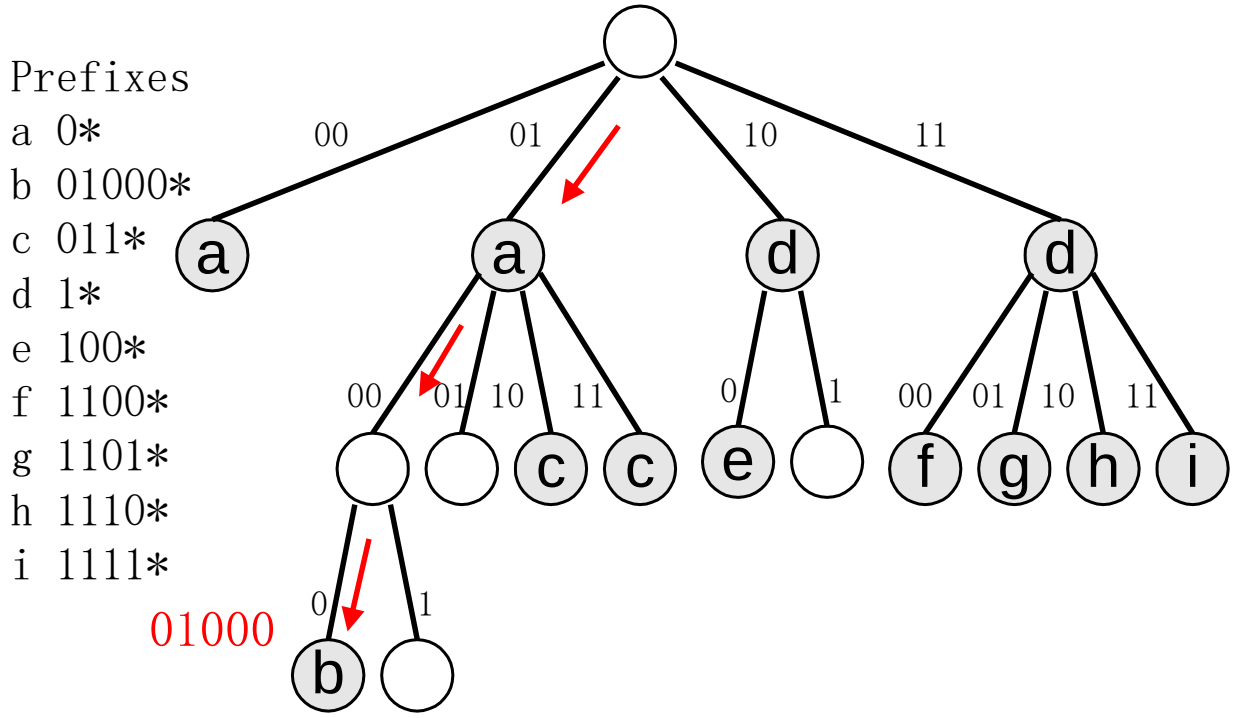


- Trie树中单分支结点的存在既增加了搜索的深度，又增加了存储空间
- 由于忽略了地址中某些位的匹配操作，结点处需要有一个变量指示下一个要检查的位
- 路径压缩trie树前缀结点保存的是地址前缀
- 当到达叶子结点或前缀匹配失败时，查找结束
- 最早在PATRICIA算法中提出，并在BSD UNIX中实现。

查找算法使用的辅助策略

- 前缀扩展（Prefix Expansion）
  - 将一条长度较短的前缀展开成多条长度较长的前缀集
  - 前缀扩展技术可以把包含各种前缀长度的前缀集转化为只包含较少前缀长度的前缀集
- 独立前缀转化（Disjoint Prefix Transformation）
  - 为了解决前缀地址空间的重叠和最长匹配问题，可以将地址前缀集转化为完全独立的前缀集
  - 根据独立前缀集构造的trie树中所有前缀结点都出现在叶子上，也称为叶子扩展（leaf pushing）技术
- 压缩技术（Compression Techniques）
- 优化技术（Optimization Techniques）
- 存储层次（Memory Hierarchy）

多分支树查找法



- 优点
  - 提高了查找效率，复杂度为O(W/k)，其中k为trie树步宽
- 缺点
  - 存储空间的需求大，利用率不高
  - 更新效率低
- 前缀扩展
  - 图中a(0*)扩展成a(00*)以及a(01*)

基于大容量RAM的快速路由查找算法

Table16表项结构

0	路由转发信息
1	TableNext表指针

TableNext表项结构

路由转发信息

表Table16

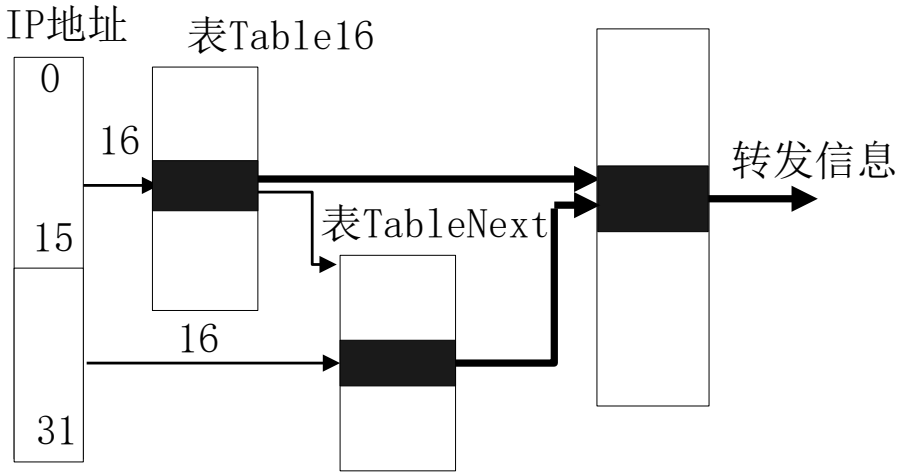
...		...
165.255	-	- -
166.0	0	A
166.1	0	A
...		...
166.110	0	A
166.111	1	
166.112	0	A
...		...
166.255	0	A
167.0	-	- -
...		...

表TableNext

0*256+0	B
...	..
0*256+255	B
...	..
68*256+0	C
...	..
68*256+255	C
69*256+0	B
...	..
69*256+255	B
...	..

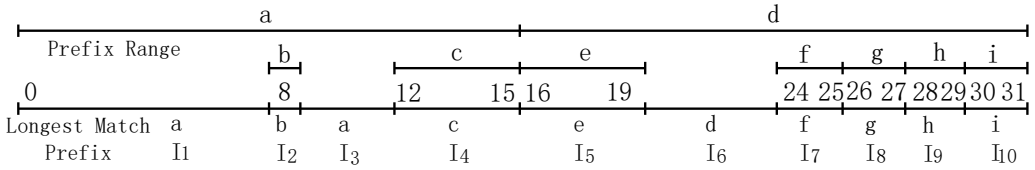
- 采用多分支trie树查找算法思想，trie树的深度为2，步宽分别为16/16（或24/8）。
- 算法采用两种查找表结构，分别为Table16和TableNext。Table16表相当于多分支trie树的第一层结点，它主要保存那些路由地址前缀小于等于16的表项；TableNext表相当于多分支trie树的第二层结点，主要保存那些路由地址前缀大于16的表项。
- 例如加入路由项166/8(A)，166.111/16(B)，166.111.68/24(C)，查找表如左图所示。
- 在查找表中查找地址166.1.2.3和166.111.1.2

基于大容量RAM的快速路由查找算法

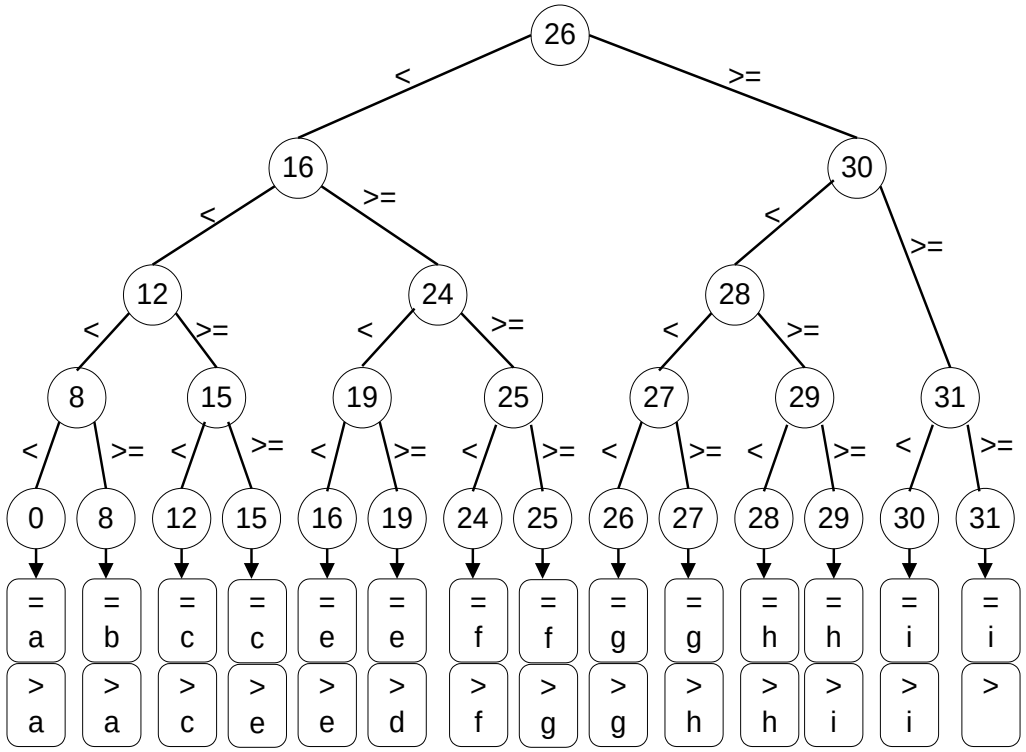


- 对于任何一个IP地址，查找过程最多只需要访问两次查找表（Table16和TableNext）就可以得到转发信息，大大加快了路由查找的速度。
- 如果我们在实现中将Table16和TableNext表从物理上分开（比如说采用两个不同的RAM），那么访问不同的表就可以同时进行，从而可以使用流水线查找。在查找过程完全满足流水线操作的条件下，查找只需要一次存储器访问，达到了查找算法的最高性能。

地址区间的二分查找法

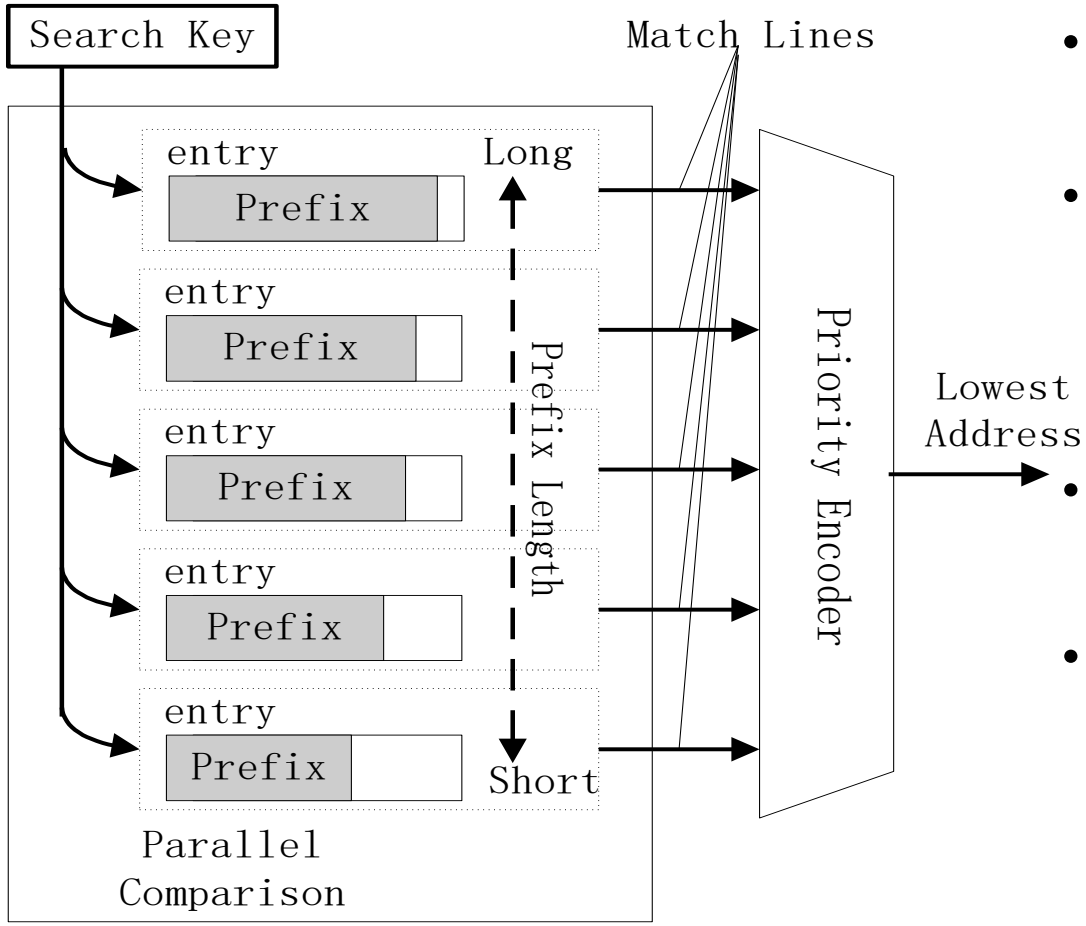


前缀对应的地址区间图



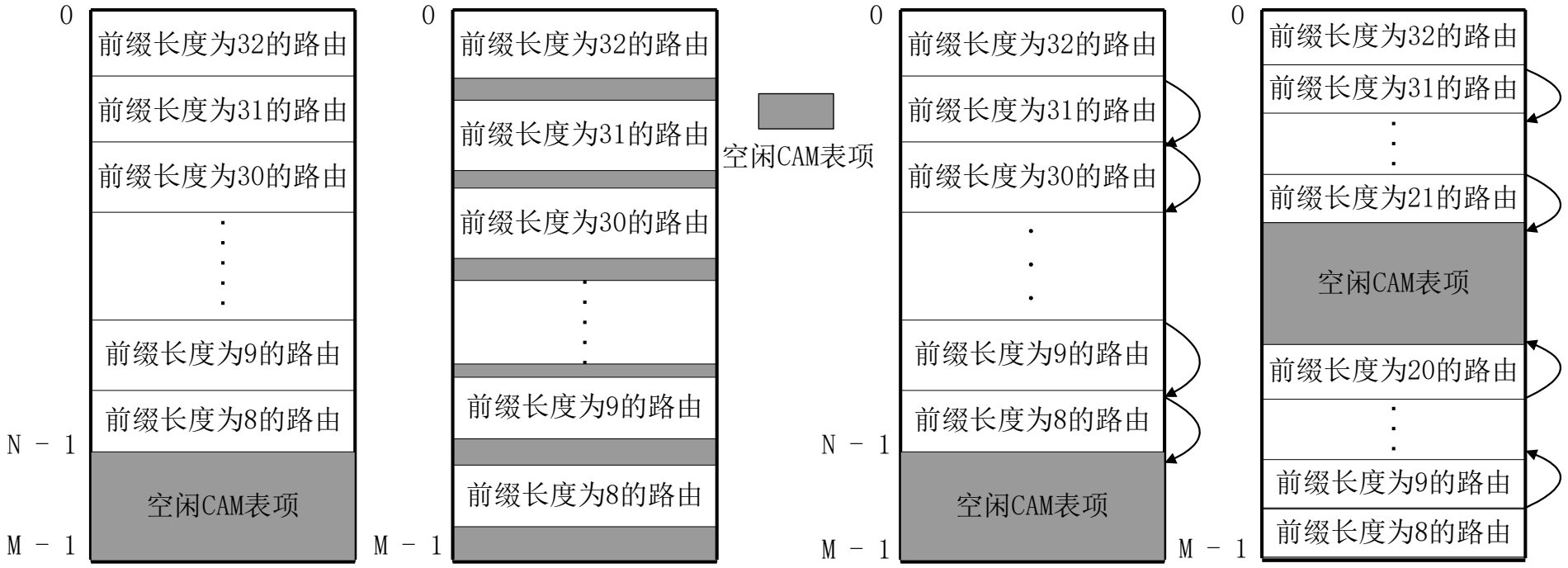
- 地址前缀在整个地址空间内代表了一段连续的地址区间
- 任何地址区间对应的最长前缀应该是包含此区间的前缀中地址范围最窄的一项
- 例：10110（22）
- N个地址前缀，查找算法复杂度为O（$\log_2 2N$）
- 改进：多路查找算法，复杂度降为O（$\log_k 2N$）
- 区间二分查找法的最大问题是地址前缀的更新

TCAM (Ternary Content Addressable Memory)



- 在一个硬件时钟周期内完成关键字的精确匹配查找
- TCAM规定在所有匹配的表项中选取地址最低的表项作为最后结果，因此需要保证在低地址存储前缀较长的前缀
- 优点
  - 查找速度快(15-20ns)
- 缺点
  - 单位容量的芯片价格高
  - 功耗较大
  - 更新效率低

TCAM路由更新

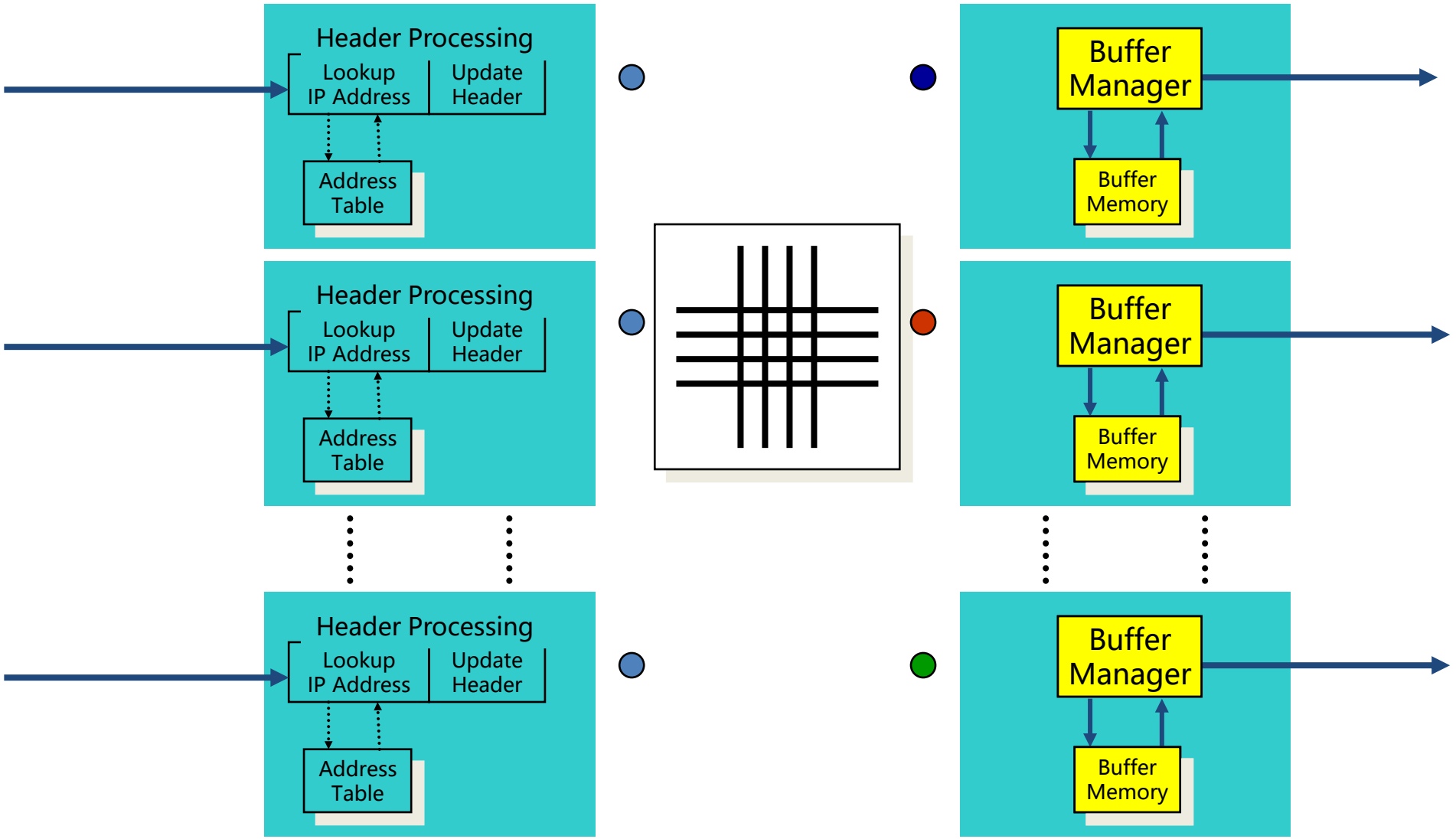


顺序移动法 -- O(N)　预留表项空间的顺序移动法 -- O(N)　选择移动法 -- O(W)　改进的选择移动法 -- O(W/2)

路由查找算法的评价标准

- 查找速度（Speed)
- 存储容量（Storage）
- 预处理和更新速度(Preprocessing and Update Speed）
- 算法实现的灵活性（Flexibility in Implementation）
- 算法的可扩展性（Scalability）

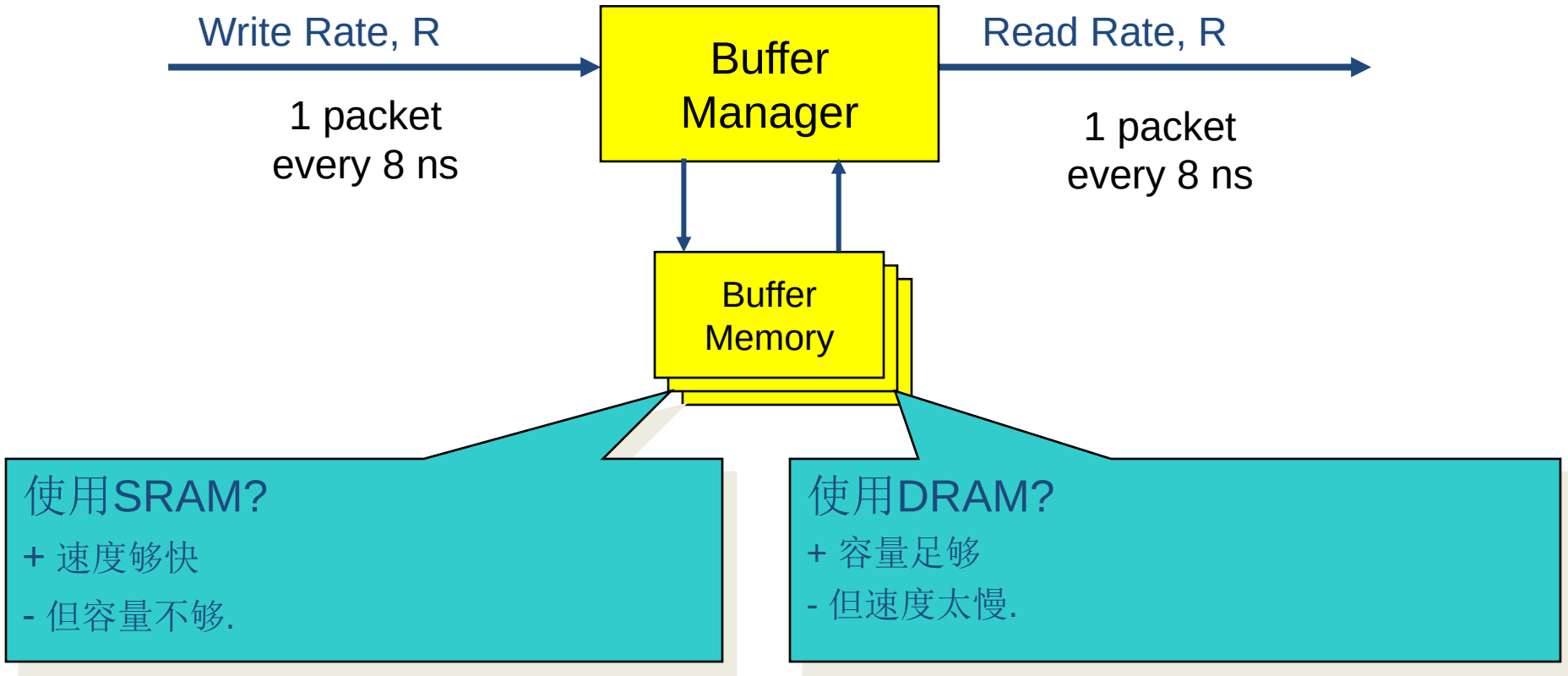
报文缓存


Header Processing
Lookup IP Address
Update Header
Address Table
Header Processing
Lookup IP Address
Update Header
Address Table
Header Processing
Lookup IP Address
Update Header
Address Table
Buffer Manager
Buffer Memory
Buffer Manager
Buffer Memory
Buffer Manager
Buffer Memory

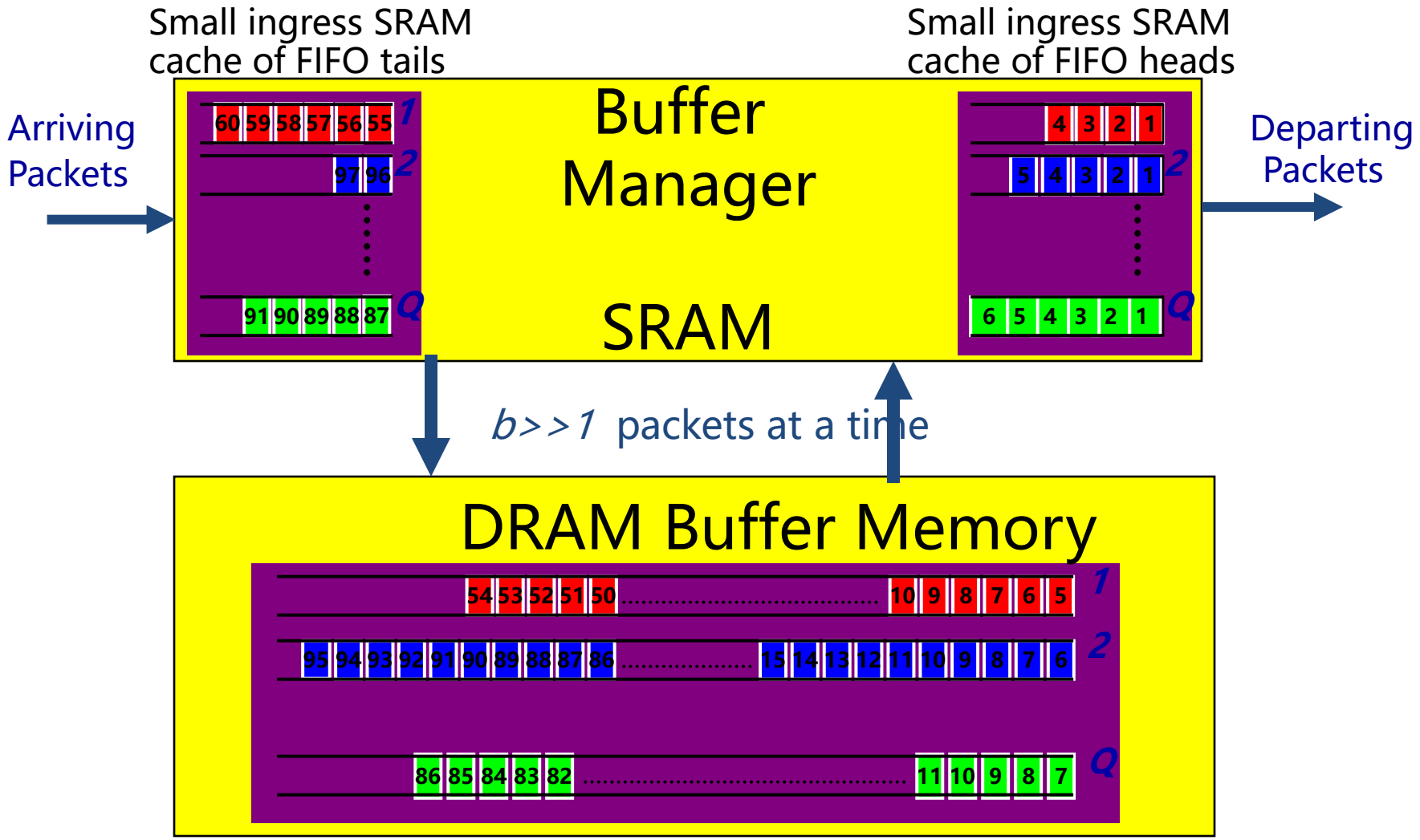
快速报文缓存

Example: 40Gb/s packet buffer

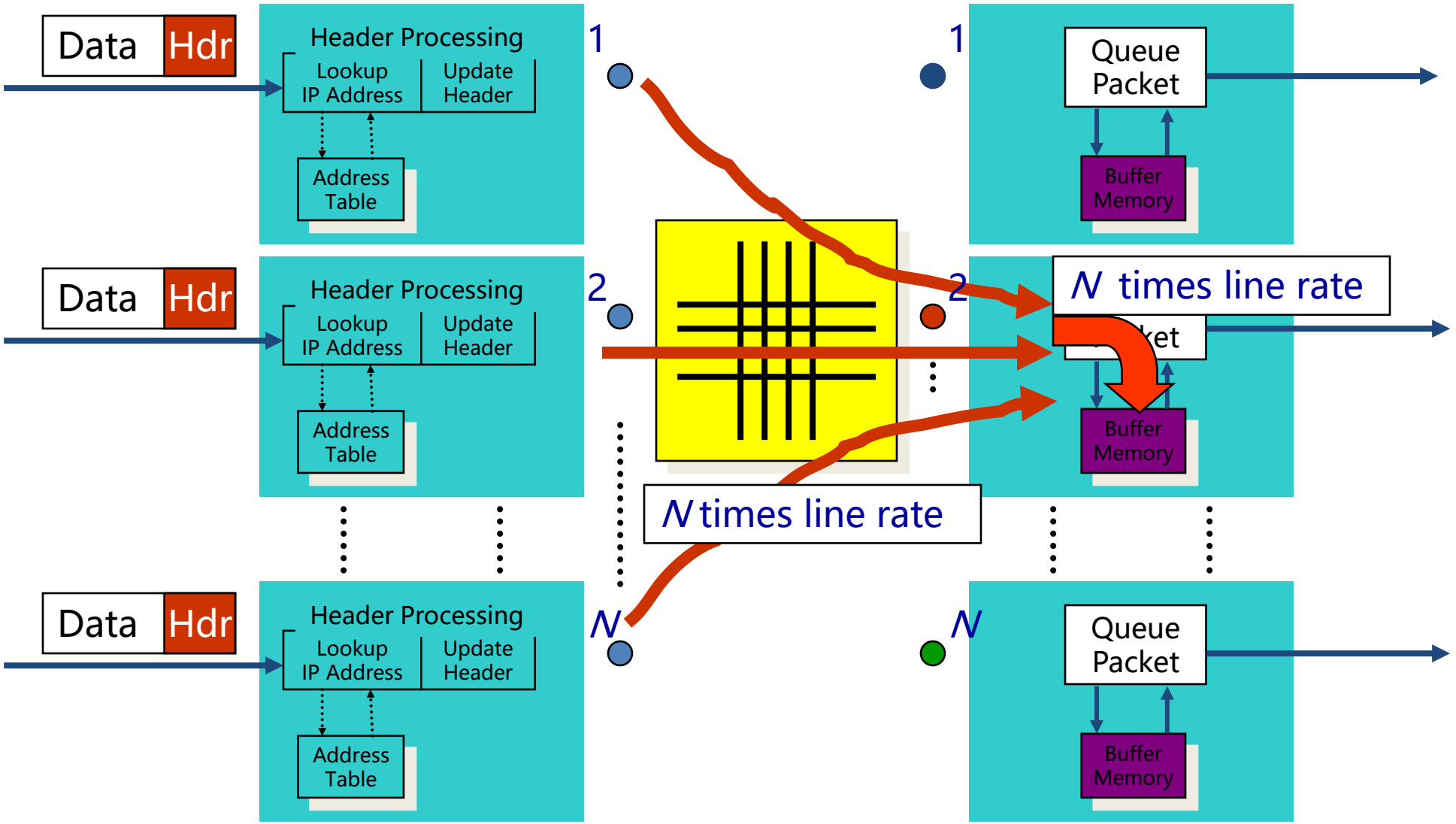
40 byte packets



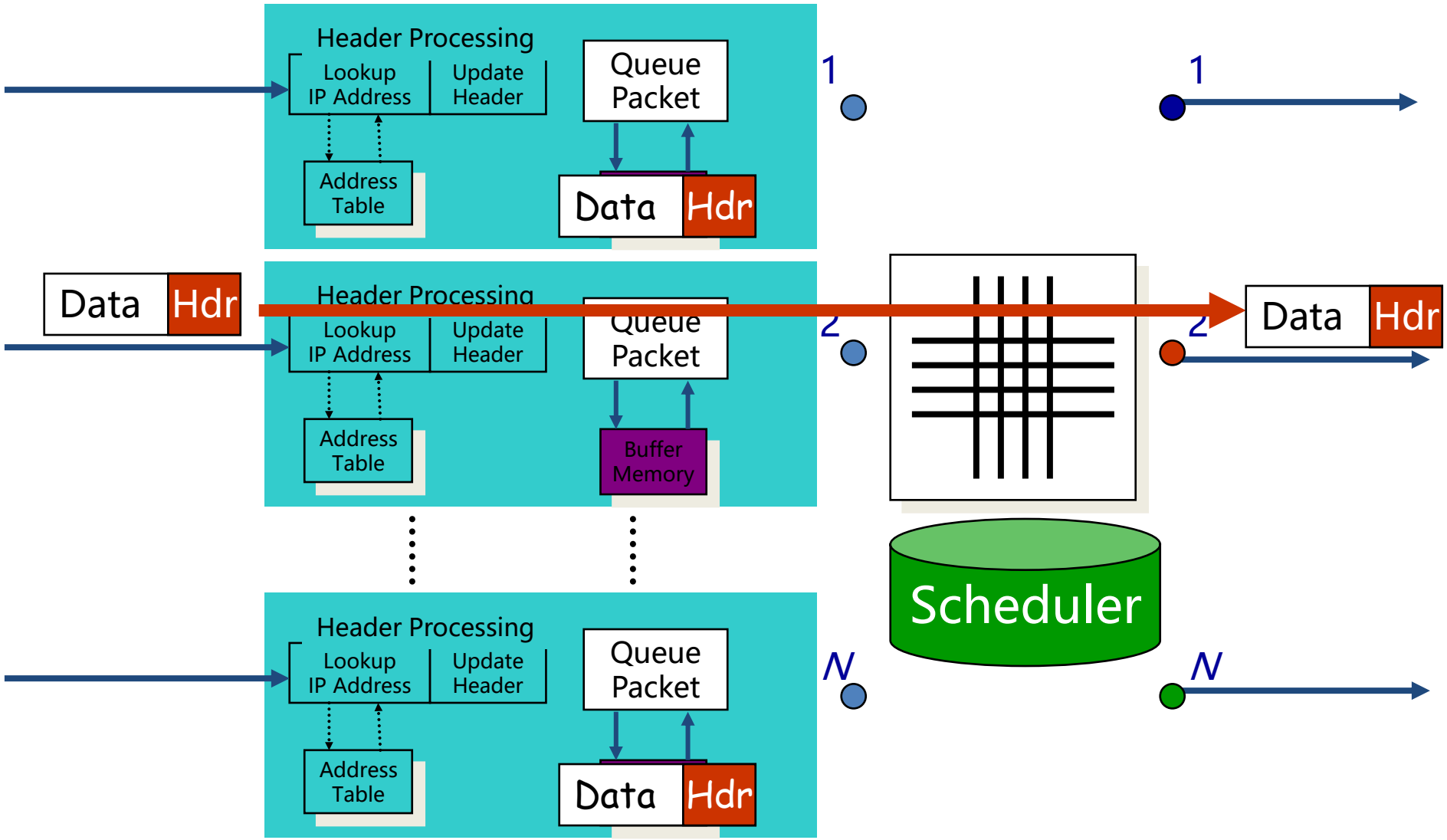
报文caches


Small ingress SRAM
cache of FIFO tails
Small ingress SRAM
cache of FIFO heads
Arriving
Packets
Departing
Packets
Buffer
Manager
SRAM
60 59 58 57 56 55
97 96
91 90 89 88 87
4 3 2 1
5 4 3 2 1
6 5 4 3 2 1
1
2
Q
b>>1 packets at a time
DRAM Buffer Memory
54 53 52 51 50
10 9 8 7 6 5
95 94 93 92 91 90 89 88 87 86
15 14 13 12 11 10 9 8 7 6
86 85 84 83 82
11 10 9 8 7

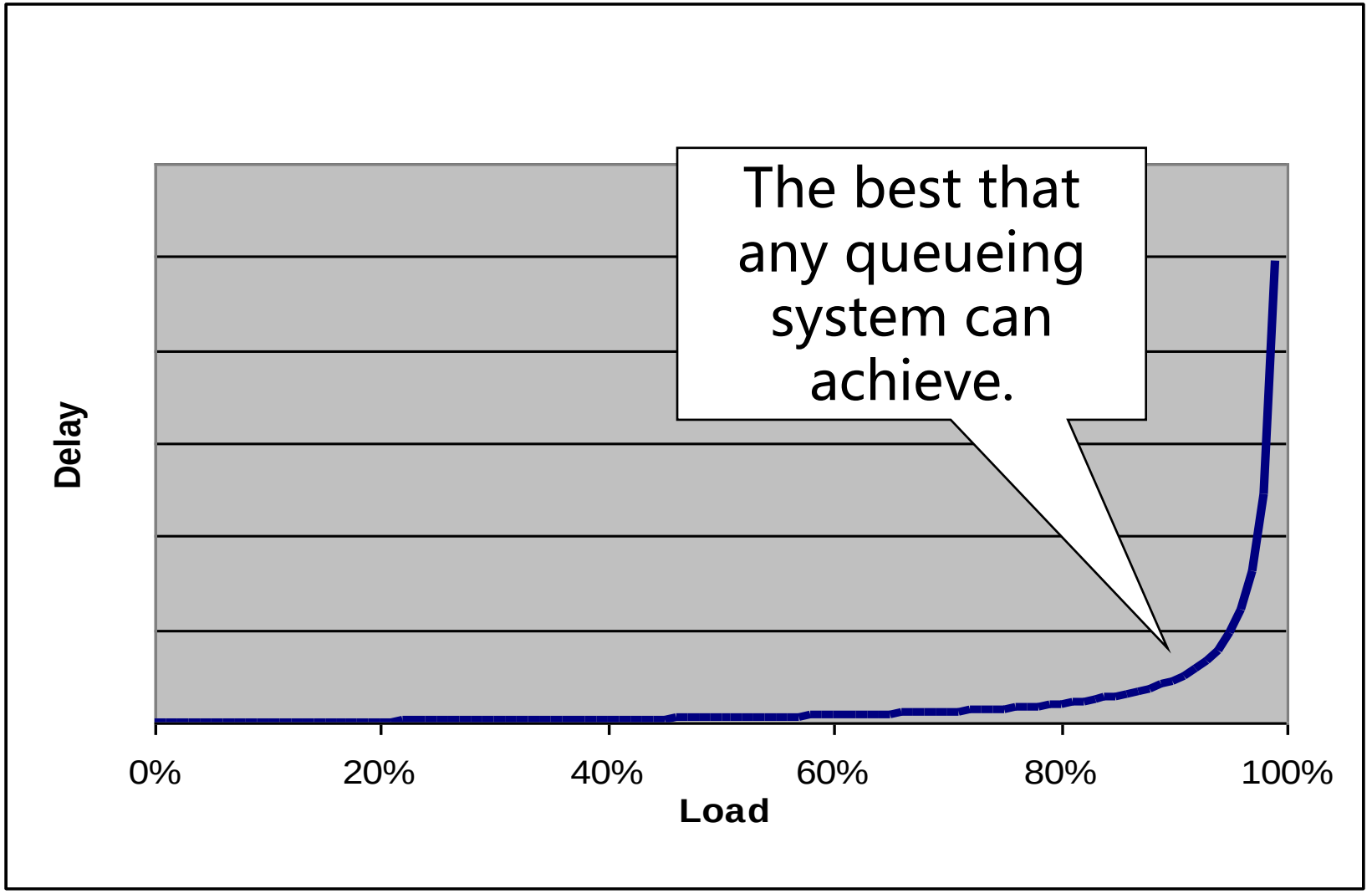
交换


Data
Hdr
Header Processing
Lookup IP Address
Update Header
Address Table
1
2
N
Queue Packet
Buffer Memory
N times line rate

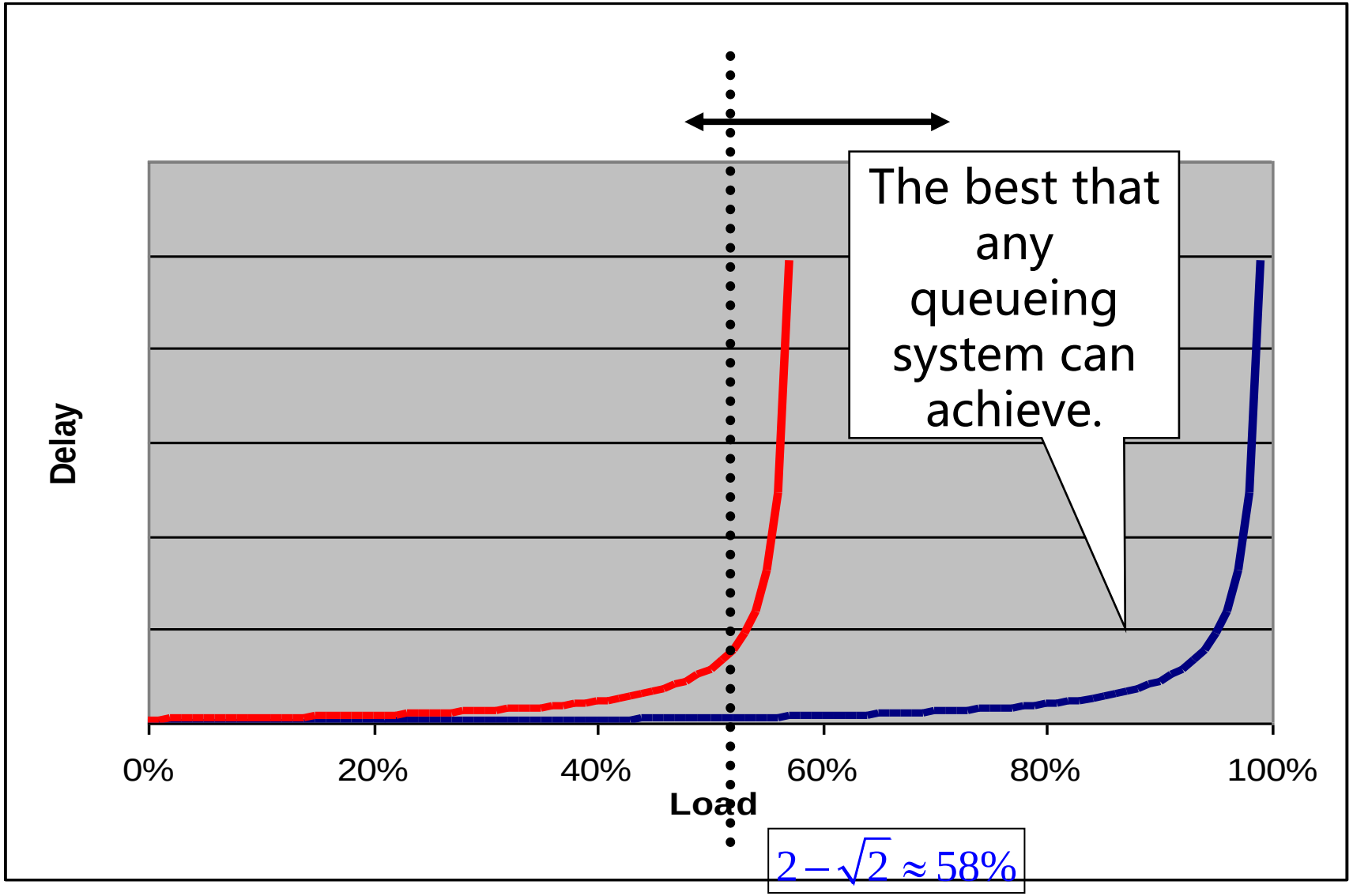
交换


Header Processing
Lookup IP Address
Update Header
Address Table
Queue Packet
Data
Hdr
Buffer Memory
1
2
N
Scheduler

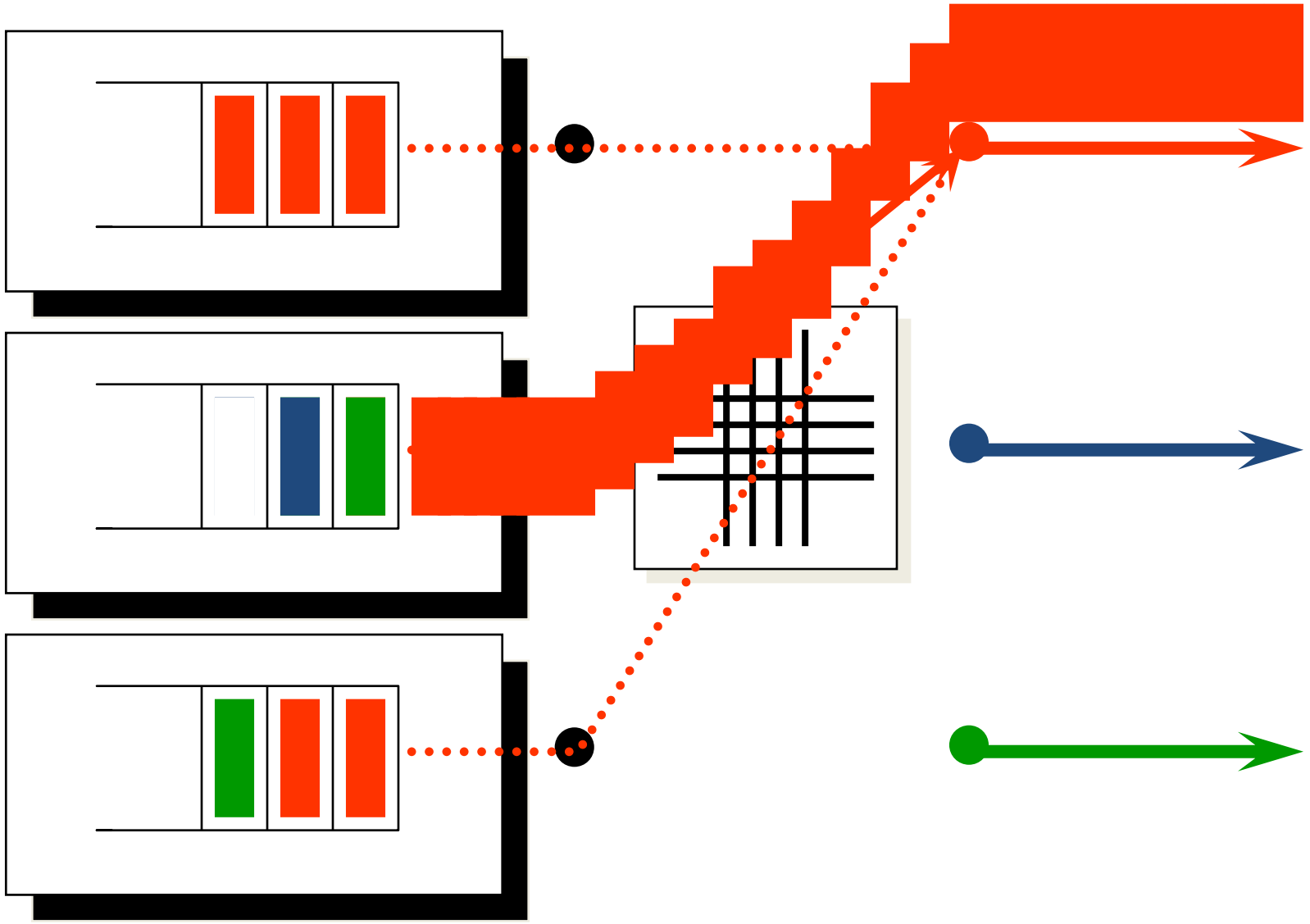
使用输入队列的路由器


The best that any queueing system can achieve.
Delay
0%
20%
40%
60%
80%
100%
Load

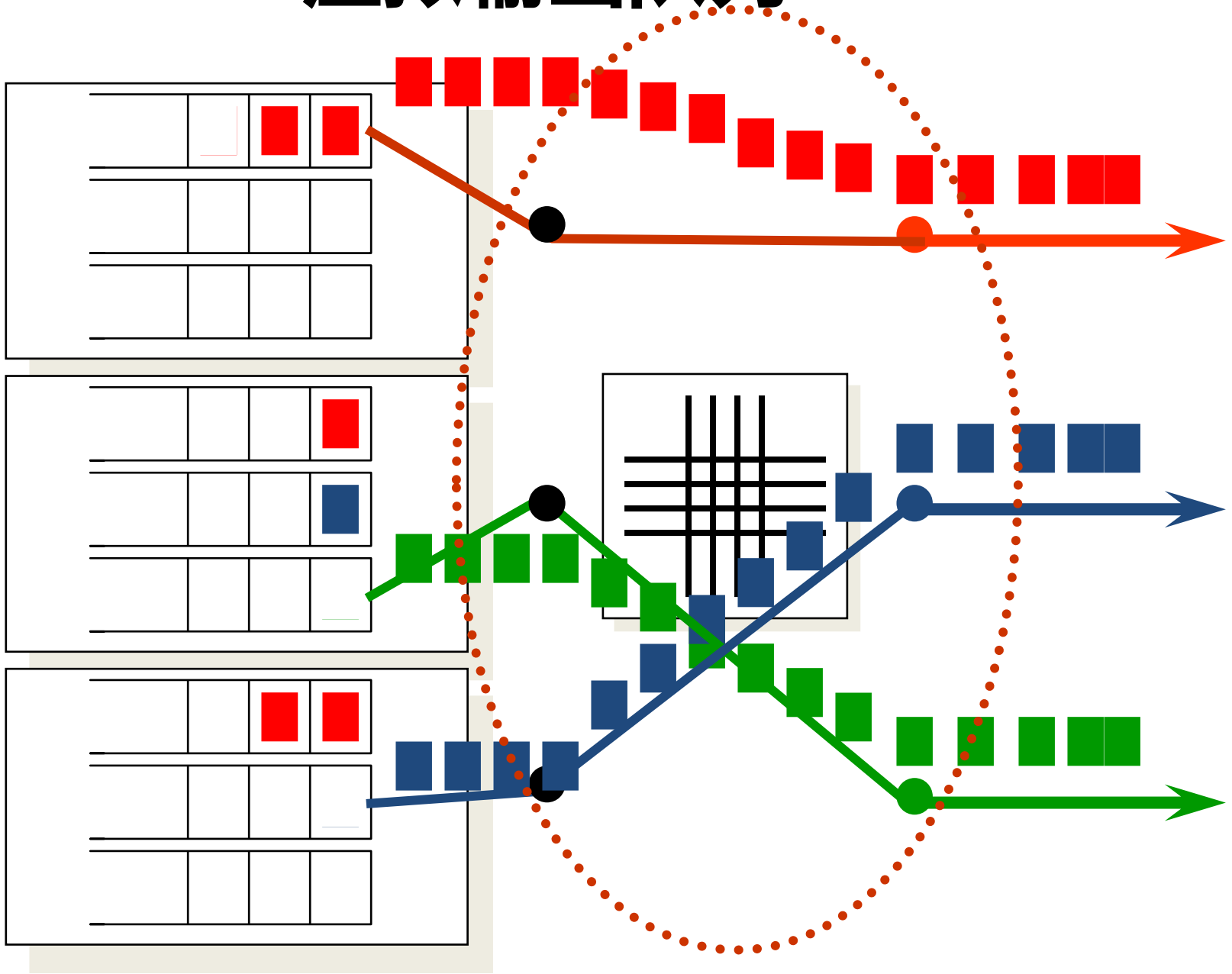
使用输入队列的路由器队头阻塞



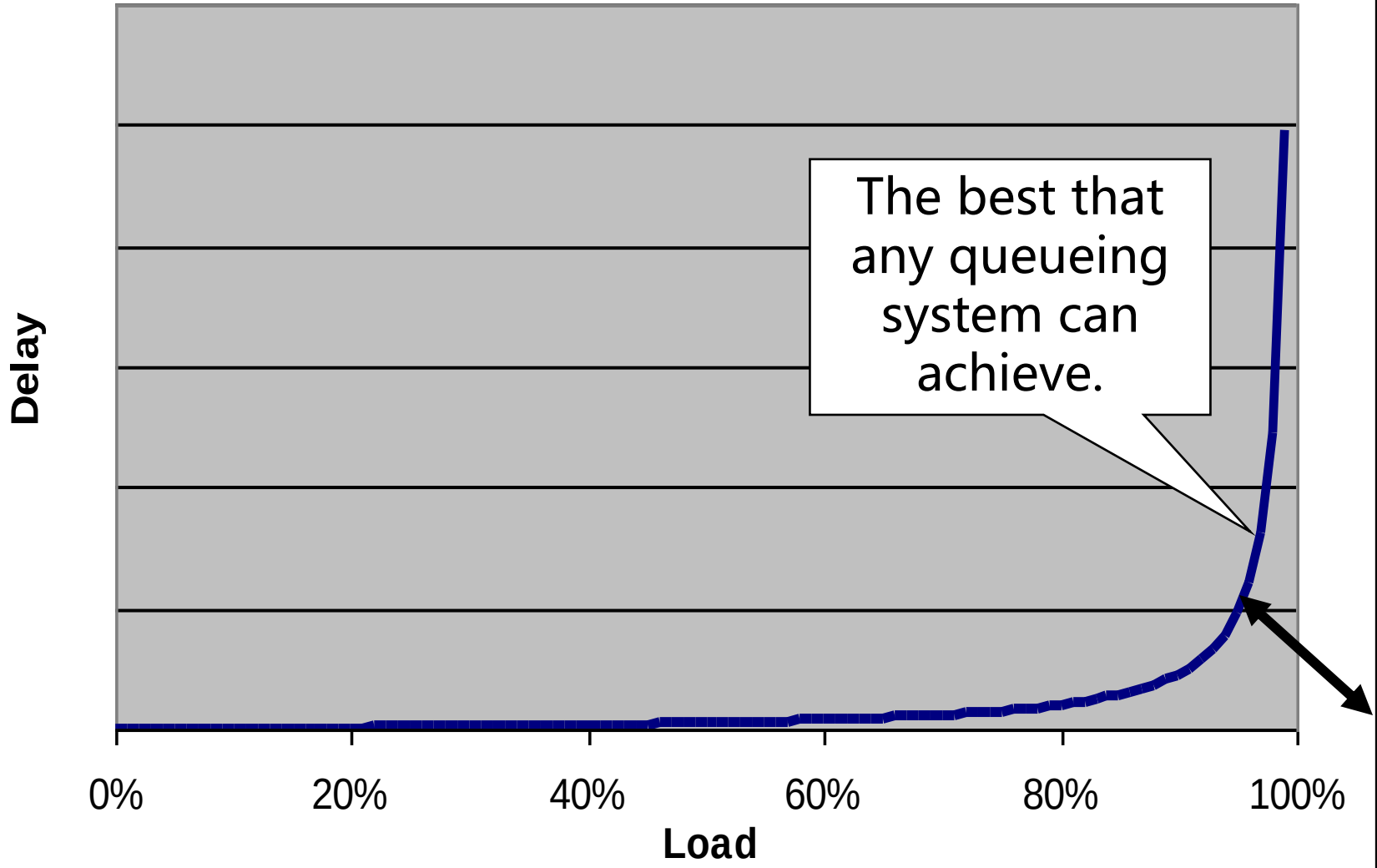
队头阻塞



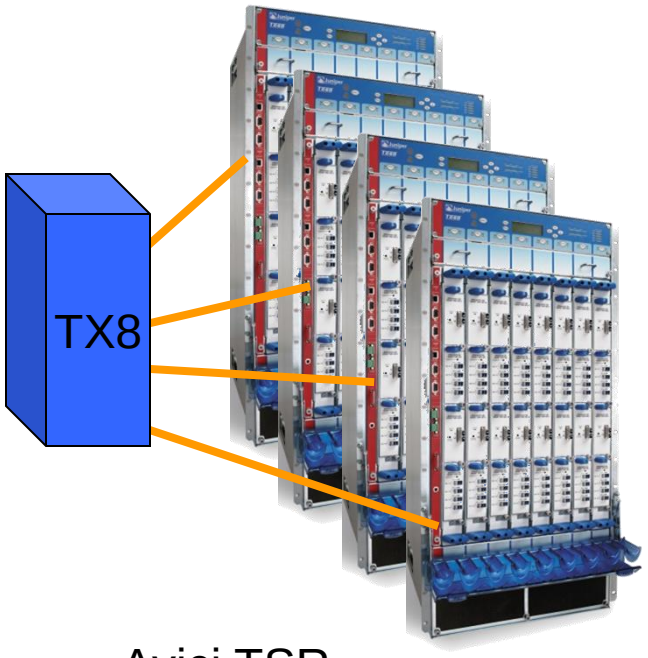
虚拟输出队列



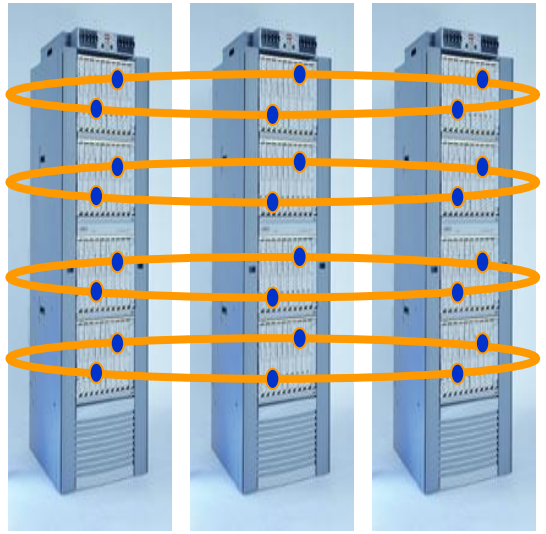
使用虚拟输出队列的路由器



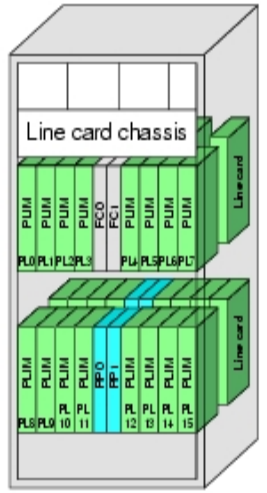
Juniper TX8/T640



Avici TSR

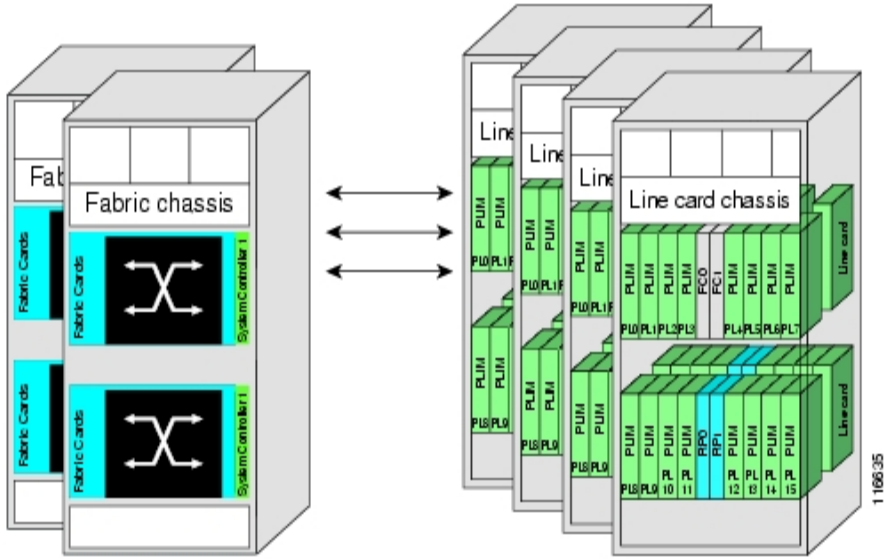


Single-Chassis System
Line Card Chassis Only

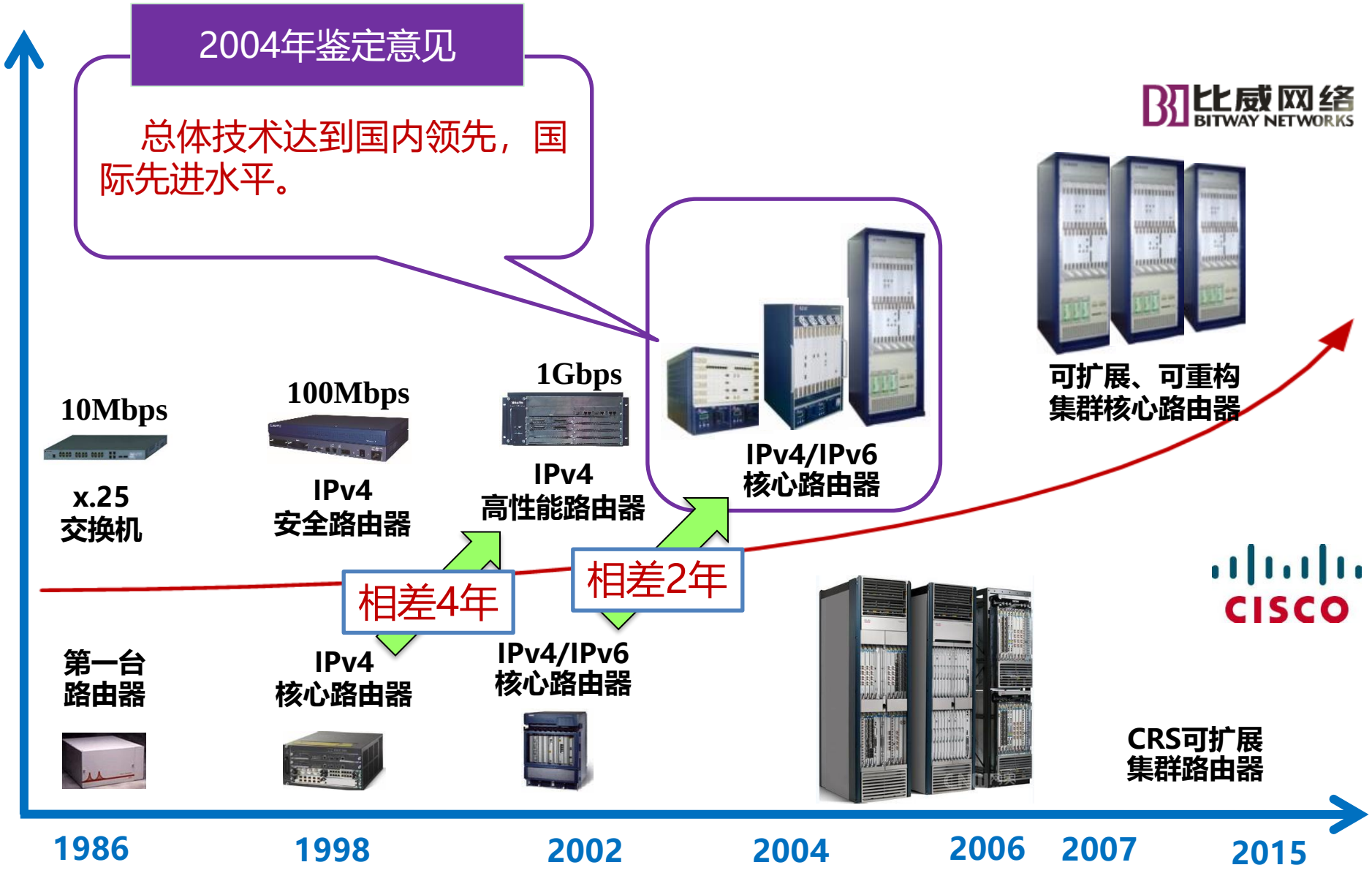


CISCO CRS-1

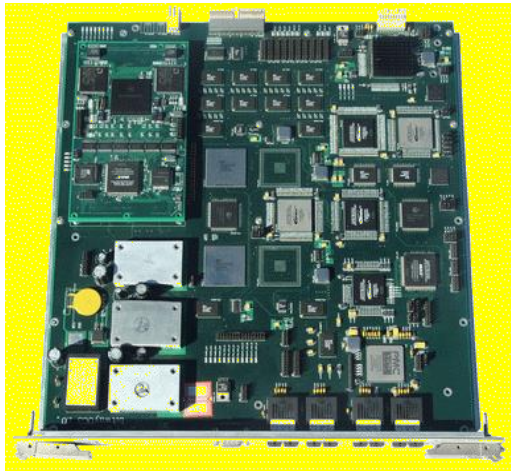
Multiple-Chassis System
Multiple Line-Card Chassis Cross Connected to One or More Fabric Chassis

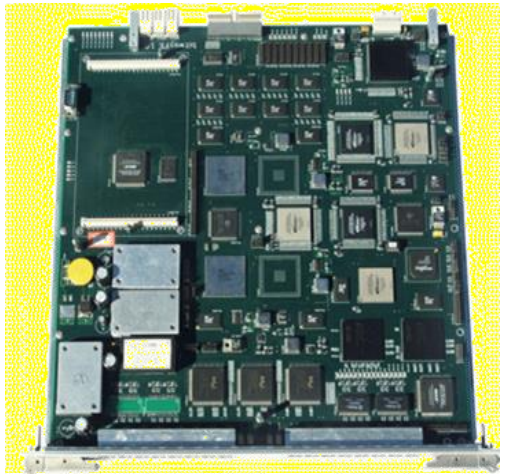
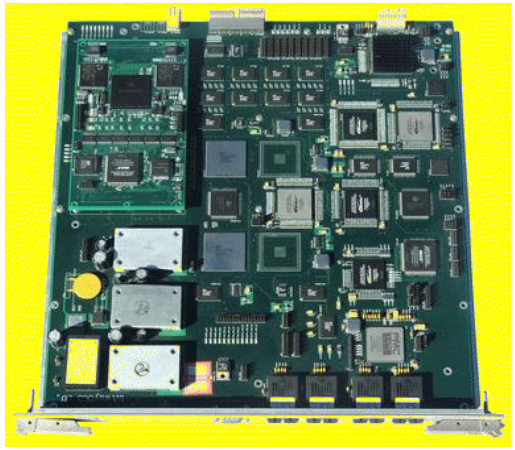
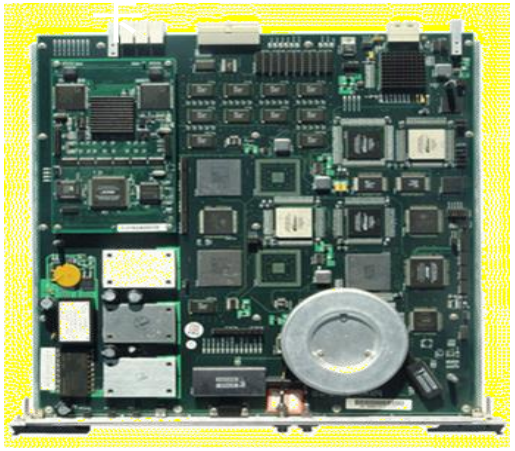
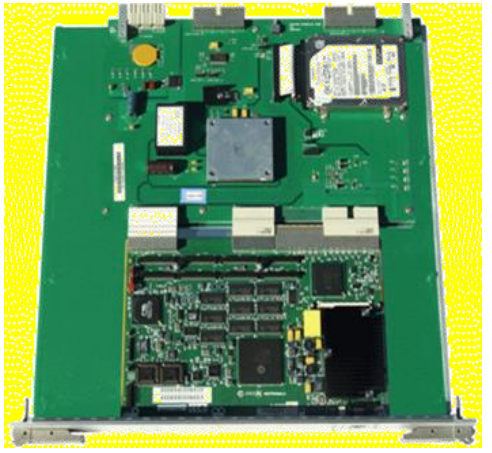


清华大学比威核心路由器研究历程



清华大学比威核心路由器核心版卡



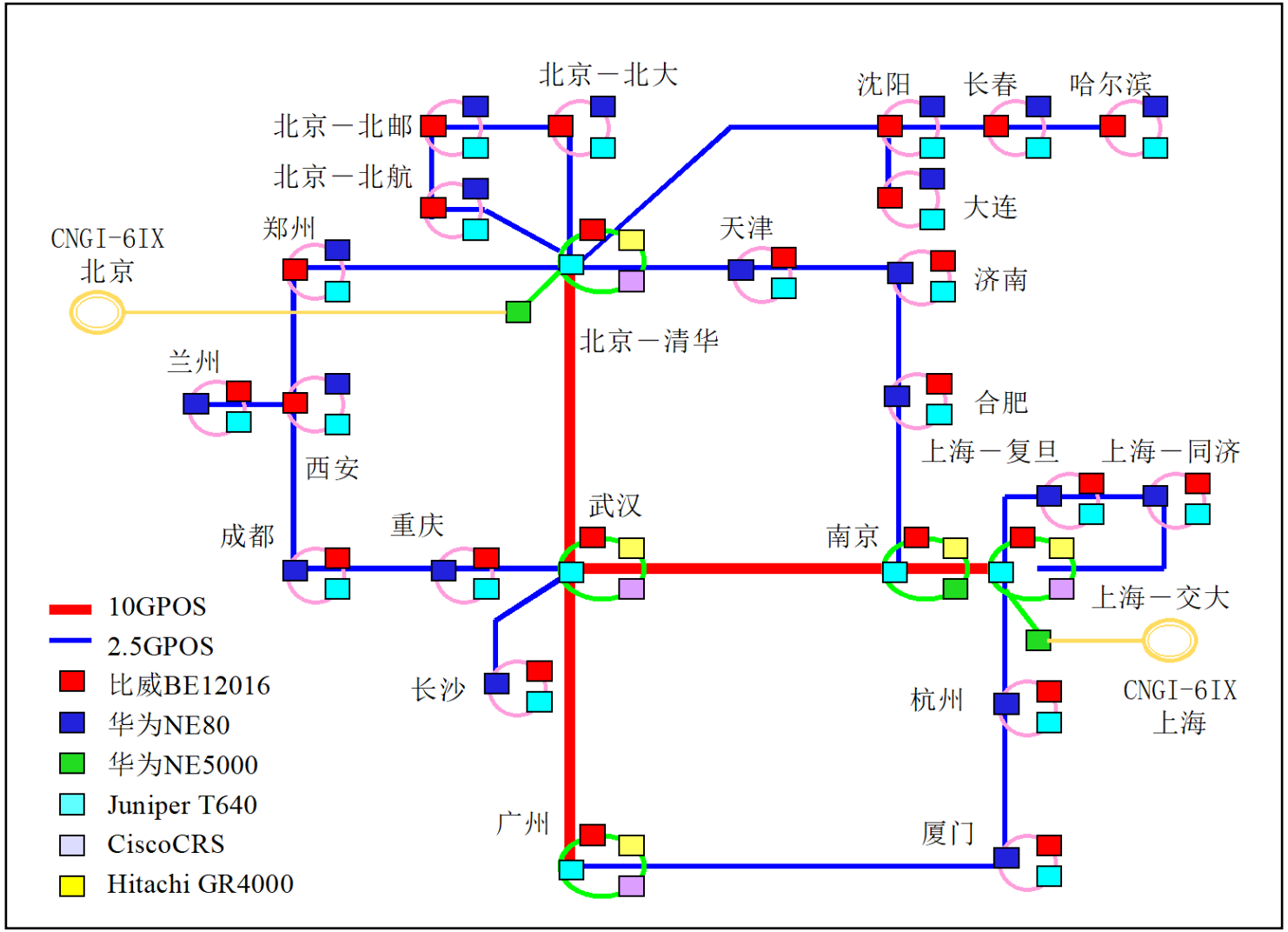
CNGI-CERNET2主干网IPv6核心路由器联调现场

华为公司IPv6核心路由器

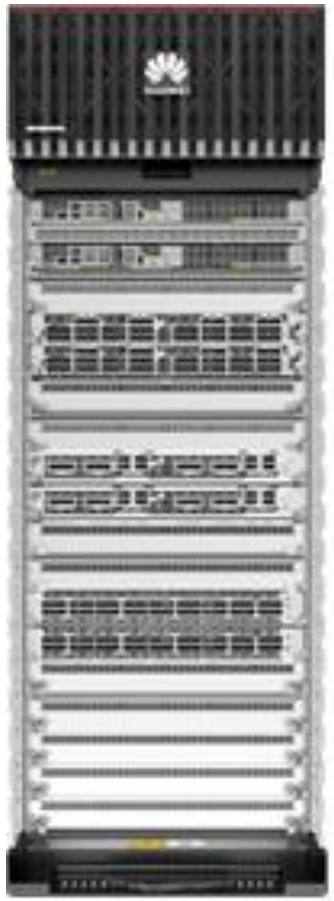
比威公司IPv6核心路由器

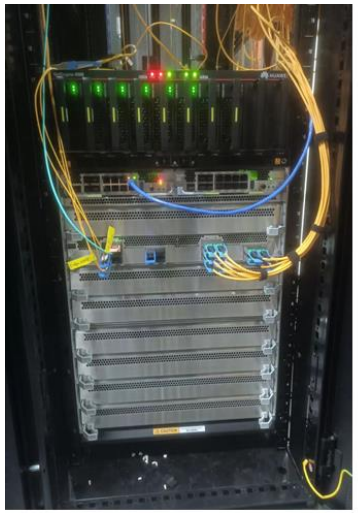
Junier 公司IPv6核心路由器

CNGI-CERNET2主干网（2005-2015）

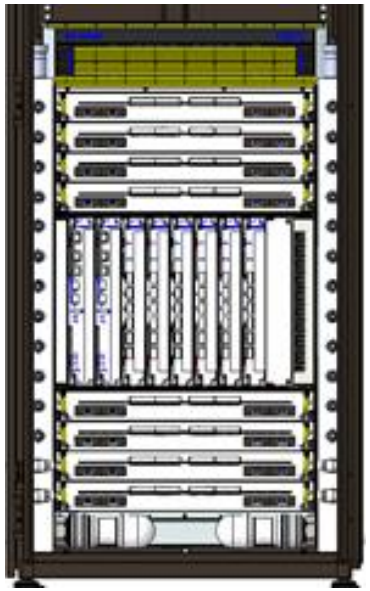


华为最新的路由器系统：NE8000系列





华三最新路由器系统



第七章 网络层

第三部分结束

本章结束