

路由器

To: 计算机C
From: 计算机A
Asdasdasd

计算机A
台式机

计算机B
笔记本

计算机C
手机

计算机D
平板

冯诺依曼式计算机——冯诺依曼（计算机之父）

运算器:

CPU, GPU（显卡）

存储器:

内存（断电数据清空, 读写速度快）,

硬盘（辅存）：（数据可以持久化, 读写速度, 相对较慢）

控制器:

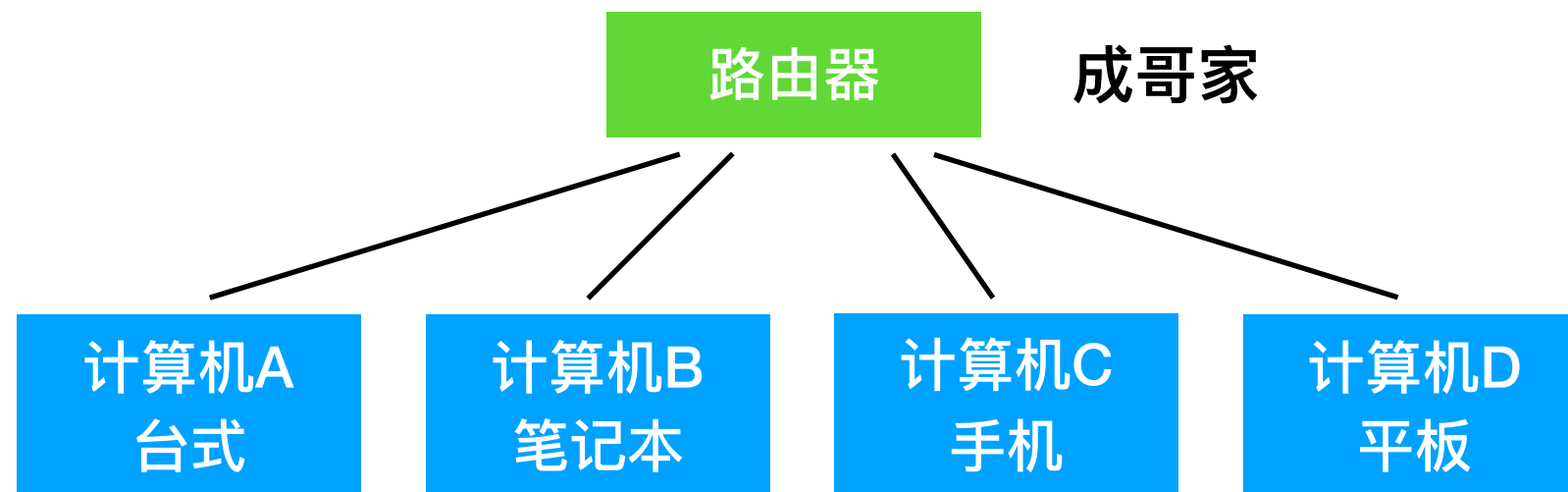
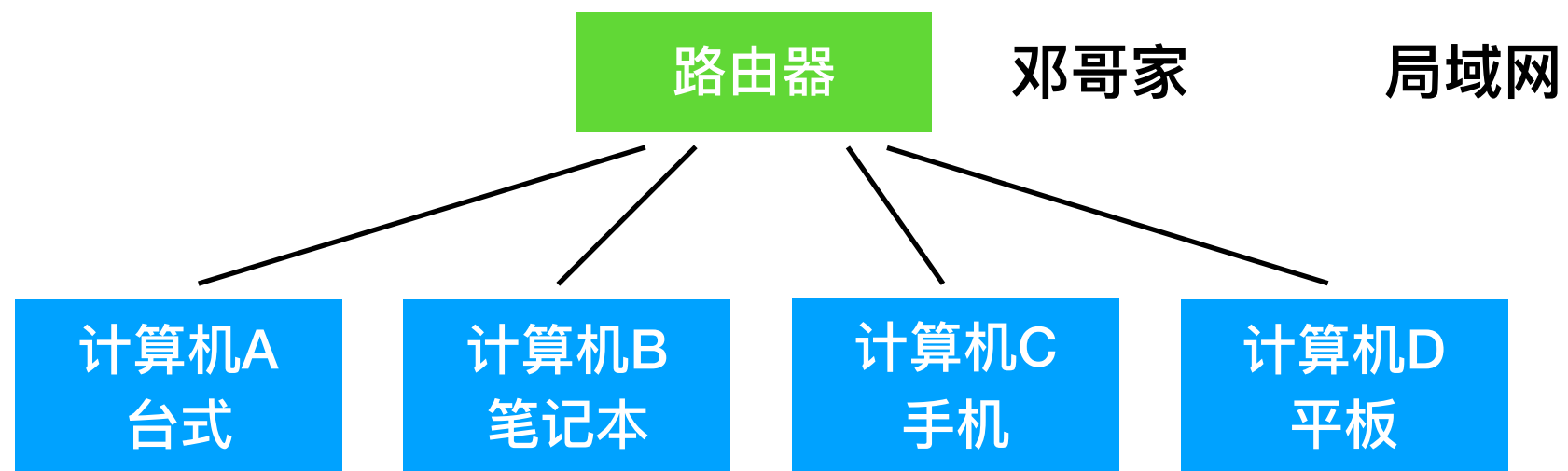
主板上的一些器件

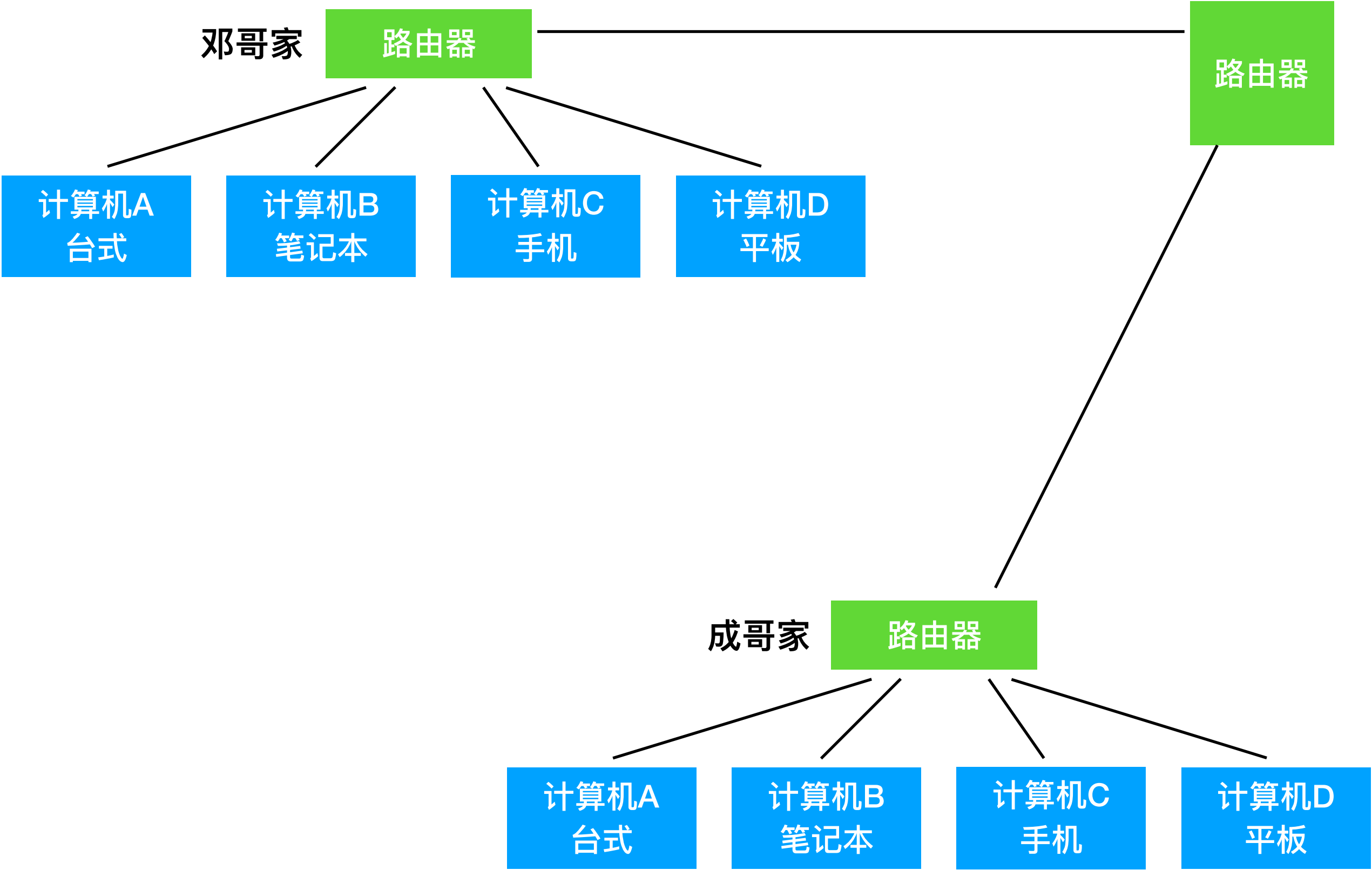
输入设备:

键盘, 鼠标, 麦克风, 网口

输出设备:

显示器, 耳机, 网口





IP地址

IP地址的格式：

IP地址分为四个段：xxx.xxx.xxx.xxx，每个段0~255，每个段，都是由8个0、1组成的。

IP地址的分类：

一个IP地址分为两个部分：网络ID，主机ID

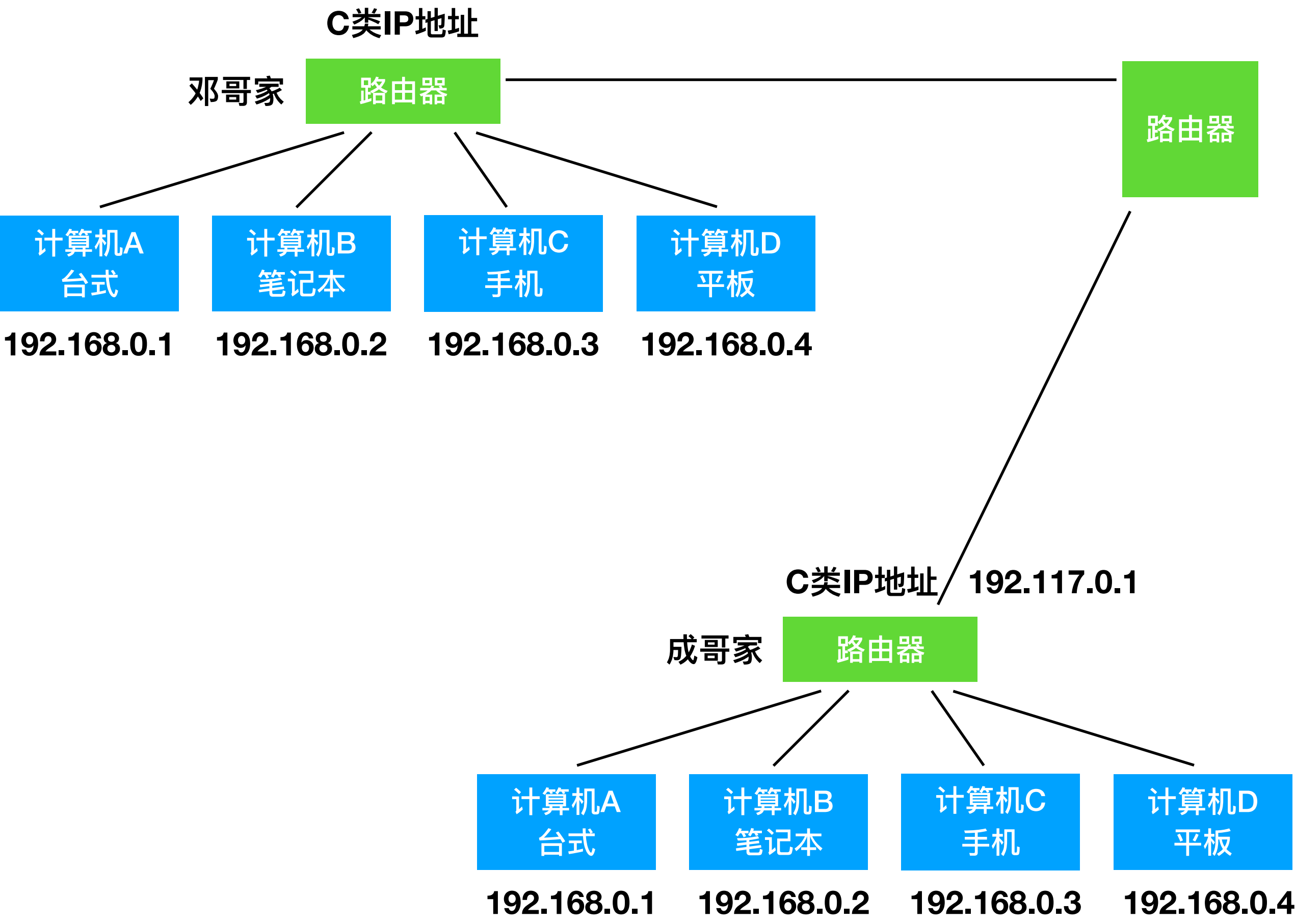
A类：0.0.0.0 ~ 127.255.255.255（一个网络能有1600+万台）

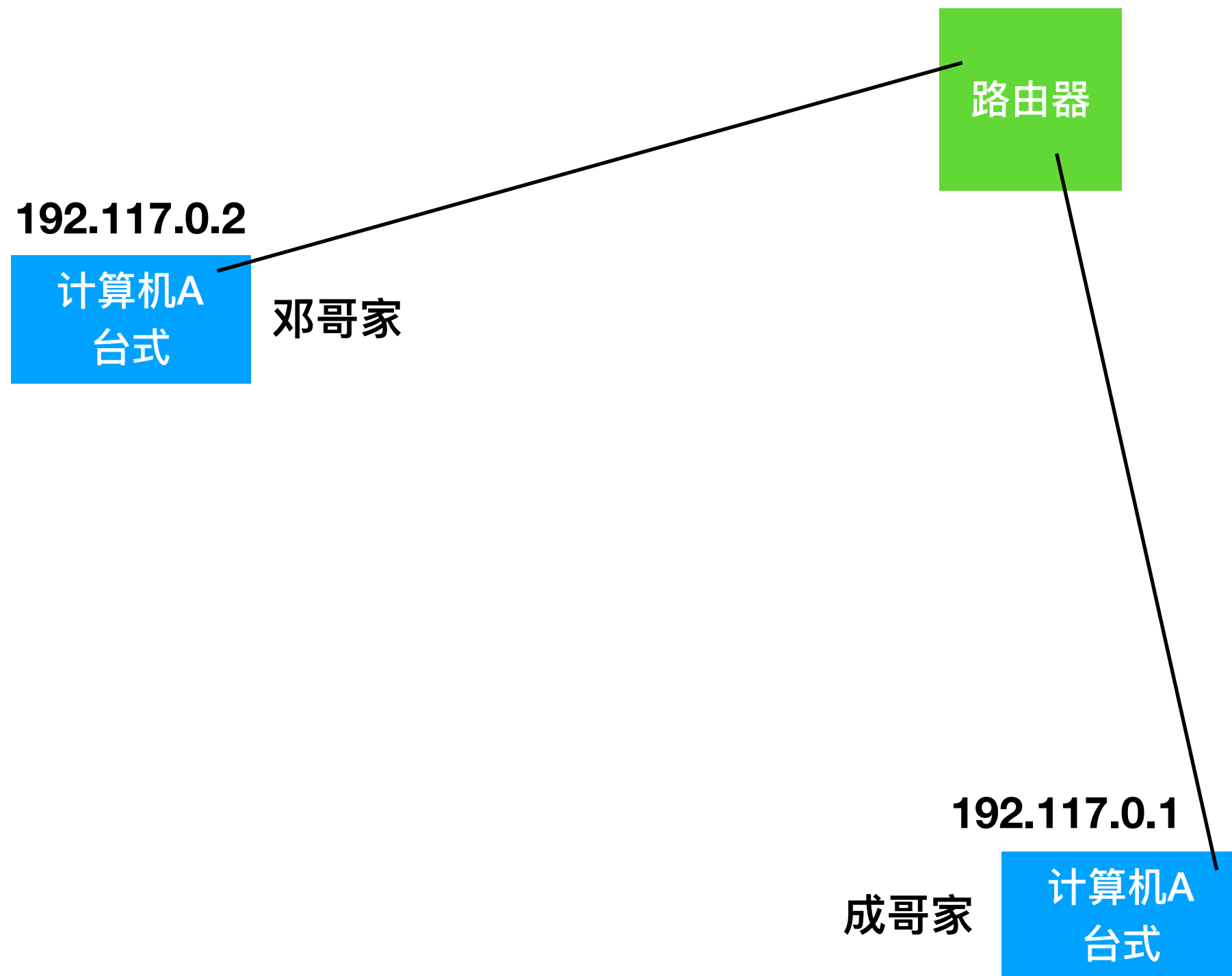
B类：128.0.0.0 ~ 191.255.255.255（172.16.0.0 ~ 172.31.255.255）

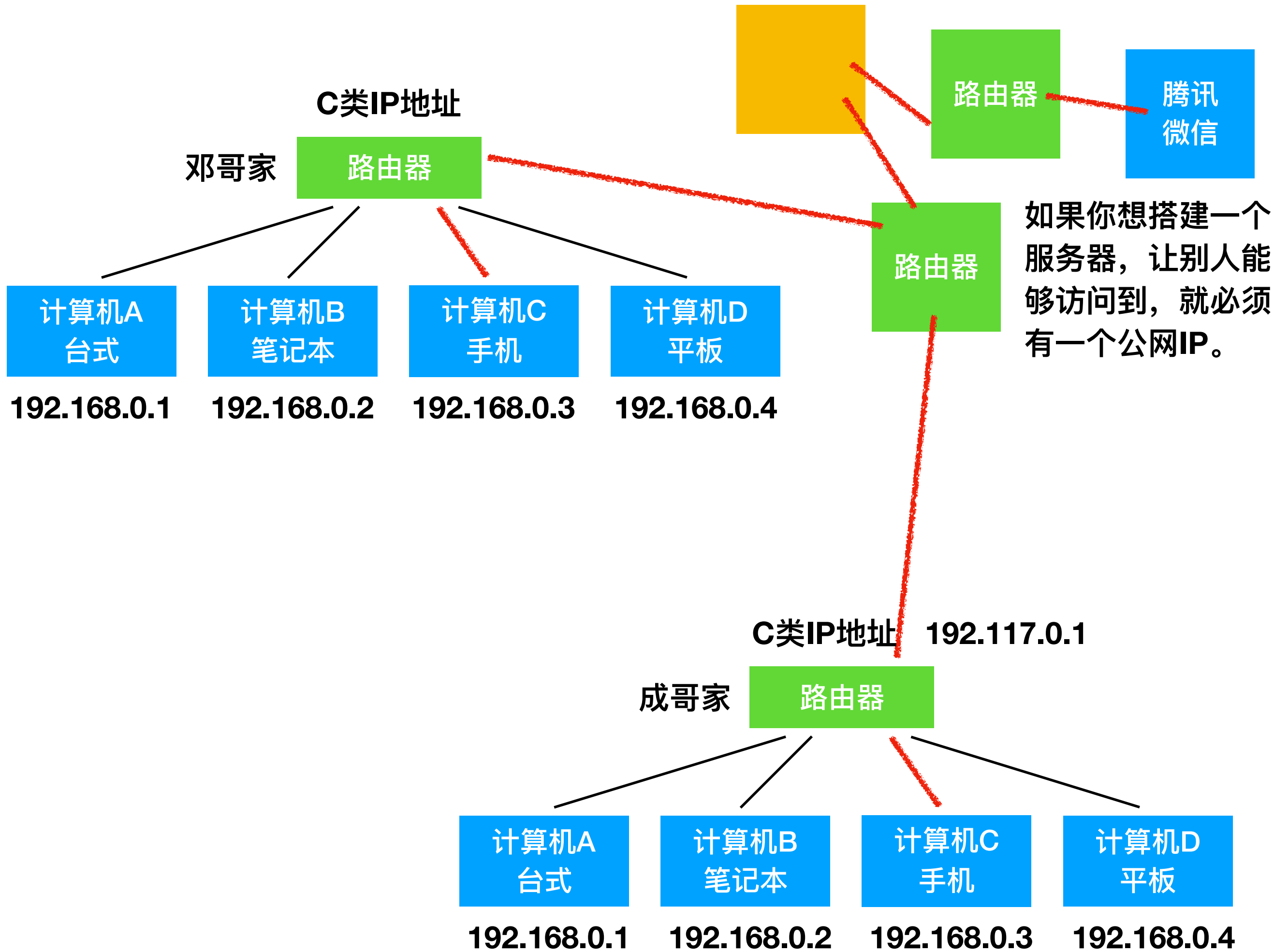
C类：192.0.0.0 ~ 223.255.255.255（192.168.xxx.xxx）

D类：（多播地址）

E类：







域名与DNS解析

www.baidu.com->域名

问：能通过域名直接访问到一台机器吗？

答：不可以的。

用域名和IP形成对应关系。

首先，计算机是不知道域名对应的IP的。

问路由器，如果路由器认识这个域名，就返回一个IP，然后计算机访问这个IP。

如果路由器不认识，他就问上一层路由器。

如果问到了城市这个级别的路由器的时候，DNS服务器。

如果DNS服务器不认识这个域名，继续向上级DNS服务器查找。

互联网建立的时候，13台总的DNS服务器。

当向浏览器的地址栏中输入一个url按回车之后，网络中都会发生什么？

比如输入的是123.xyz

看浏览器的缓存。

本机host。

C:windows/system32/drivers/etc/host

127.0.0.1 localhost

0.0.0.0

家里路由器

上级路由、城市的LDNS服务器

继续向上级的DNS服务器找。

gDNS服务器。

五层网络模型

应用层

HTTP协议，DNS协议

运输层

TCP协议、UDP协议

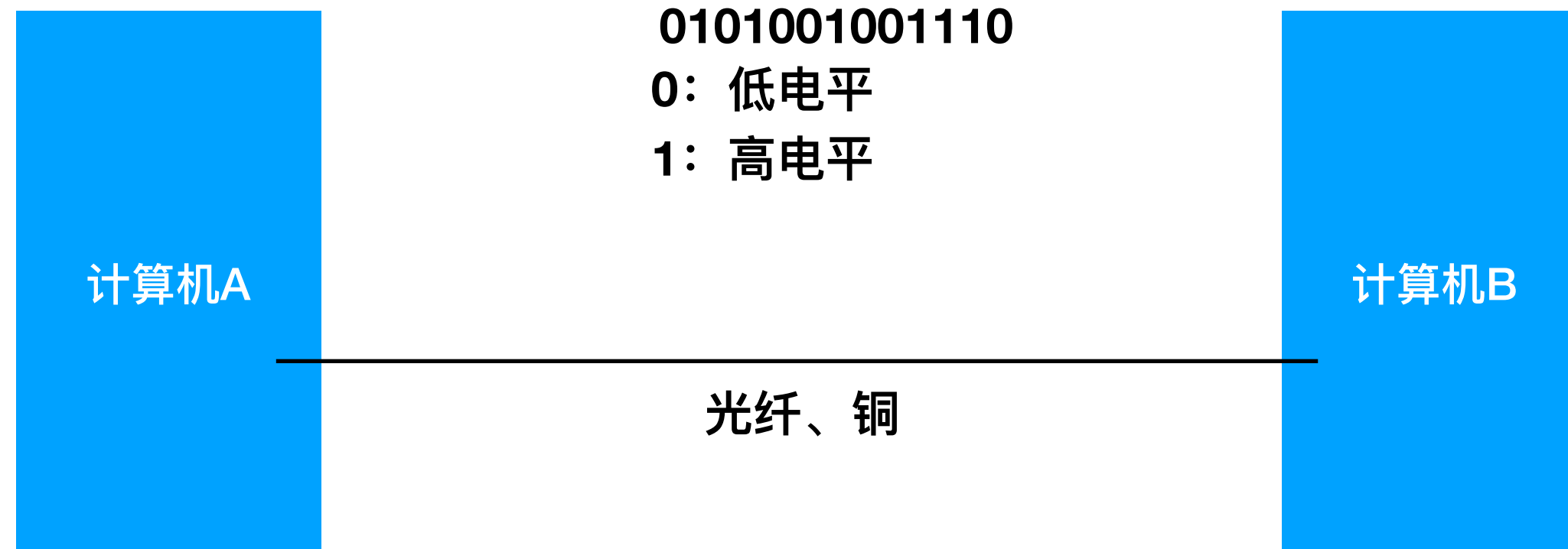
网络层

IP地址——IP协议

数据链路层

mac地址

物理层



0101001001110

0: 低电平

1: 高电平

计算机A

计算机B

光纤、铜

TCP/IP协议（对方的IP，自己的IP，对方的端口） HTTP协议（请求头） hello

HTTP协议分为两个部分：

请求： Request

请求方式 路径 协议版本

请求方式： GET, POST

响应： Response

五层网络模型

Hello

应用层(浏览器)

HTTP协议， DNS协议

运输层

TCP协议、 UDP协议

网络层

IP地址——IP协议

数据链路层

mac地址

物理层

应用层
(Web服务器)

运输层

网络层

数据链路层

物理层

HTTP协议

请求： Request

请求头

数据体

请求方式 url 协议版本

POST /path?a=1&b=2 HTTP1.1

Host: www.baidu.com

Connection: keep-alive

User-Agent: asdasdasd

Cookie

传文件的一些信息。

GET/POST — — HEAD, PUT, DELETE

响应： Response

响应头

数据体

协议版本 状态码 message

属性： 值

Get与Post请求方式的区别

1. 是基于什么前提的？如果什么前提都没有，不使用任何规范，只考虑语法和理论上的HTTP协议。

GET和POST几乎没有什么区别，只有名字不一样。

2. 如果是基于RFC规范的。

(1) 理论上的 (Specification) : **GET和POST具有相同语法的，但是有不同的语义。**
get是用来获取数据的，post是用来发送数据的，其他方面没有区别。

(2) 实现上的 (Implementation) : **各种浏览器，就是这个规范的实现者。**

常见的那些不同：

- 1) **GET的数据在URL是可见的。POST请求不显示在URL中。**
- 2) **GET对长度是有限制的，POST长度是无限的。**
- 3) **GET请求的数据可以收藏为书签，post请求到的数据不可收藏为书签。**
- 4) **GET请求后，按后退按钮、刷新按钮无影响，post数据会被重新提交。**
- 5) **GET编码类型：application/x-www-form-urlencoded
post的编码类型：有很多种。
application/x-www-form-urlencoded
multipart/form-data**
- 6) **GET历史参数会被保留在浏览器里，psot不会保存在浏览器中的。**
- 7) **GET只允许ASCII.post没有编码限制，允许发二进制的。**
- 8) **GET与POST相比，GET安全性较差，因为所发的数据是URL的一部分。**

Cookie与Session

1.如果我们用JS的变量来存数据，那么在页面关闭的时候，数据就消失了。

2.保持登录状态是怎么做到的呢？

按照正常的HTTP协议来说，是做不到的。

因为HTTP协议，上下文无关协议。

3.所以说前端页面上，有可以持久化存储数据的东西。一旦登录成功，我就记载在这个里面。

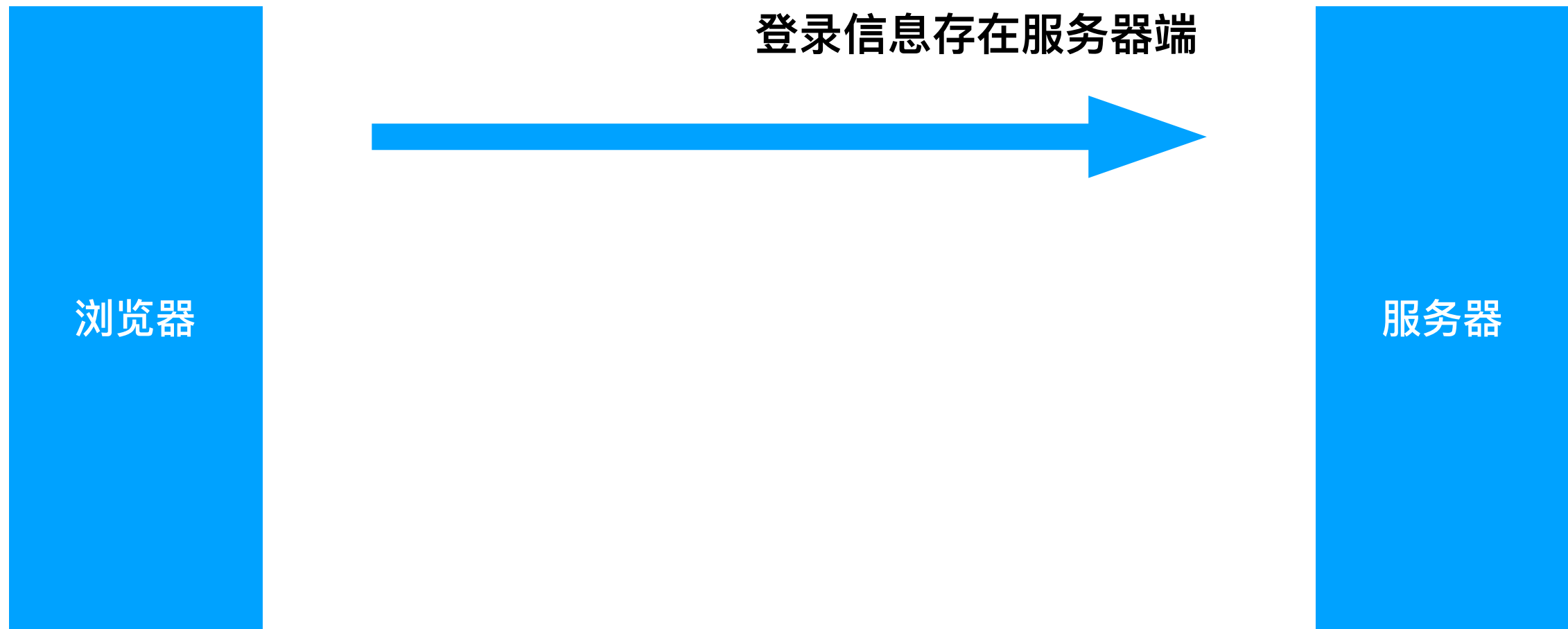
Cookie是有限制的。

Cookie是存在浏览器里的，不是存在某个页面上的。是可以长期存储的。**Cookie**即使是保存在浏览器里，也是存放在不同的域名下的。

1. 初始状态：没有登录
2. 访问百度的登录，输入用户名，密码。
3. 如果用户名和密码是正确的。百度的后端会向这个域名下，设置一个Cookie。写入用户的基本信息（加密的）。
4. 以后每一次向百度发送请求，浏览器都会自动带上这些Cookie。
5. 服务端（后端）看到了带有ID的cookie，就可以解析这个加密的ID，来获取到这个用户本身的ID。
6. 如果能获取到本身的ID，那么就证明这个用户已经登录过了。所以后端可以继续保留用户的信息。

缺点：如果某个坏人，复制了我浏览器里的cookie，他就可以在他的电脑上登录我的账号了。

XSS注入攻击。



数据存在Session上也有缺点

如果用户量非常大，上亿的用户。

在用户量很大的时候，服务器端很耗资源的。

因为后端可能不止一台服务器，用户的登录信息，一般只存在一台服务器上。

因为用户的登录操作，在哪台机器上执行的，就一般存在哪台机器上。

需要通过反向代理。（轮询，IP哈希。）

B/S结构

C/S结构

Client/Server——Client只负责内容的展示，Server负责提供内容。

Browser/Server——Browser只负责内容的展示，Server负责提供内容。

www.baidu.com

index.html

页面的本质是什么？ 页面的本质就是一个字符串。带有HTML格式的字符串。

浏览器 向服务器 请求一个页面的本质是什么？

(1) www.baidu.com

(2) 服务器接收到这个请求后，服务器想要把这个页面的内容（HTML格式的字符串）返回给浏览器。

(3) 页面的字符串存在哪里呢？ 存在HTML文件里。例如：index.html

(4) 服务器端要读取文件。

(5) 将读取出来的内容返回给浏览器。

最后返回的是一个字符串，这个字符串的来源可能是文件，可能是缓存，可能来自于数据库。

服务器： **严格的说**，服务器是一台计算机，这台计算机，只提供服务。（不是用户用的）

但是，我们常说的这个服务器，指的是服务容器，不是服务器。

服务容器：是一个程序。程序可以监听一个端口。读取文件，并且返回。

如果我们想通过访问**服务器（服务容器）**的方式，来访问我们自己写的页面。

我们就得装一个服务容器的程序。

跨域访问资源

哪些东西属于资源？

js文件算吗？js文件肯定是算资源的，但是js文件是允许被跨域请求的。

css文件，jpg,png等。src属性的资源都是可以被跨域请求的。href资源大部分都是可以被跨域请求的。

哪些资源算跨域请求的资源？

1. 后端接口的数据。
2. 其它域的cookie
3. 其它域的缓存

什么是其它的域？怎么样算跨域？

页面本身：有协议（http/https）,域名，端口

要请求的数据：<http://www.baidu.com:80>

协议，域名，端口这三个，有任意一个不一样就算跨域。

跨域这个行为，发生在哪里？

答案：

1. 即使跨域了（协议，域名，端口号有不一样的），请求也可以发出。
2. 服务器端也是可以接收的。
3. 服务器端也是可以正常处理的。
4. 服务器端也是可以正常返回数据。
5. 浏览器也能接收到这些数据。
6. 接收到之后，发现当前页面的域和请求的域不同，所以判定为跨域。
7. 我们的代码在这等着结果呢，但是因为浏览器判定跨域了，不会把结果传递给我们的代码。

虽然跨域了，但是我们依然需要这个数据，怎么办？

解决跨域问题：

1. 后端（别人家的）配合我们进行跨域。

pan.baidu.com ——> zhidao.baidu.com

(1) JSONP（正常的情况，返回的数据都是JSON格式。JSONP是一种特殊的格式。）

(2) 后端设置Access-Control-Allow-Origin属性以支持跨域。（聊天机器人课讲，因为需要nodejs）

2. 后端不配合我们进行跨域。

(3) iframe（只能显示，不能控制）

(4) 通过后端代理（自己的后端）（后面聊天机器人讲，因为需要nodejs）