

# 网络编程懒人入门(六)：史上最通俗的集线器、交换机、路由器功能原理入门-网络编程/专项技术区 - 即时通讯开发者社区！



关注我的公众号

即时通讯技术之路，你并不孤单！

IM开发 / 实时通信 / 网络编程

本文引用了知乎网友“薛定谔不在家”的部分文字内容，感谢原作者的分享。

## 1、前言

即时通讯网整理了大量的网络编程类基础文章和资料，包括《[TCP/IP协议 卷1](#)》、《[\[通俗易懂\]深入理解TCP协议](#)》系列、《[网络编程懒人入门](#)》系列、《[不为人知的网络编程](#)》系列、《[P2P技术详解](#)》系列、《[高性能网络编程](#)》系列、甚至还有图文并貌+实战代码的《[NIO框架入门](#)》等，目的是帮助即时通讯类应用的开发者，至少要掌握网络编程最基本的原理，所谓知其然更要知其所以然。尤其现在移动网络大行其道的时代，在网络环境如此复杂的情

况下，能写好一套技术精湛、用户体验等俱佳的IM或消息推送系统，显然不是随使用Netty、MINA、AFNetworking、okhttp等服务端和客户端框架就能搞定的事。总之，即时通讯技术归根结底还是网络编程技术的应用，只有更深入地了解了网络编程及其相关知识，才能更好地写出优质的应用。

实际上计算机网络编程或者网络通信技术最基本的物理载体，就是集线器、交换机、路由器这些基本设备，了解这些基本设备的工作原理，对于程序员来说是基本素养，总不能什么事都甩锅给网管，何况中小公司根本就没有条件配备专职网管，还是得程序员亲自动手。但技多不压身，何况这些设备和技术总比那些毫无技术含量的插删改查代码撸起来有意思。话不多说，回归正题吧。

本文旨在简单地说明集线器、交换机与路由器的区别，因而忽略了很多细节，三者实际的发展过程和工作原理并非文中所写的这么简单。如果你看完本文能大概了解到三者的异同，本文的目的就达到了。至于更具体的技术问题，欢迎在留言中探讨。

另外，如果您正打算从零开发移动端IM，则建议您从《[新手入门一篇就够：从零开发移动端IM](#)》一文开始，此文按照IM开发所需的知识和技能要求，拟定了详尽的学习提纲和建议等。

## 2、系列文章

本文是系列文章中的第6篇，本系列文章的大纲如下：

- [《网络编程懒人入门\(一\)：快速理解网络通信协议\\_（上篇）\\_》](#)
- [《网络编程懒人入门\(二\)：快速理解网络通信协议\\_（下篇）\\_》](#)
- [《网络编程懒人入门\(三\)：快速理解TCP协议一篇就够》](#)
- [《网络编程懒人入门\(四\)：快速理解TCP和UDP的差异》](#)
- [《网络编程懒人入门\(五\)：快速理解为什么说UDP有时比TCP更有优势》](#)
- [《网络编程懒人入门\(六\)：史上最通俗的集线器、交换机、路由器功能原理入门》](#)（本文）
- [《网络编程懒人入门\(七\)：深入浅出，全面理解HTTP协议》](#)
- [《网络编程懒人入门\(八\)：手把手教你写基于TCP的Socket长连接》](#)
- [《网络编程懒人入门\(九\)：通俗讲解，有了IP地址，为何还要用MAC地址？\\_》](#)

本站的《脑残式网络编程入门》也适合入门学习，本系列大纲如下：

- [《脑残式网络编程入门\(一\)：跟着动画来学TCP三次握手和四次挥手》](#)
- [《脑残式网络编程入门\(二\)：我们在读写Socket时，\\_](#)

[究竟在阅读什么? 》](#)

- [《脑残式网络编程入门\(三\): HTTP协议必知必会的一些知识》](#)
- [《脑残式网络编程入门\(四\): 快速理解HTTP/2的服务器推送\(Server Push\)》](#)

如果您觉得本系列文章过于基础，您可直接阅读《不为人知的网络编程》系列文章，该系列目录如下：

- [《不为人知的网络编程\(一\): 浅析TCP协议中的疑难杂症\(上篇\)》](#)
- [《不为人知的网络编程\(二\): 浅析TCP协议中的疑难杂症\(下篇\)》](#)
- [《不为人知的网络编程\(三\): 关闭TCP连接时为什么会TIME\\_WAIT、CLOSE\\_WAIT》](#)
- [《不为人知的网络编程\(四\): 深入研究分析TCP的异常关闭》](#)
- [《不为人知的网络编程\(五\): UDP的连接性和负载均衡》](#)
- [《不为人知的网络编程\(六\): 深入地理解UDP协议并用好它》](#)
- [《不为人知的网络编程\(七\): 如何让不可靠的UDP变的可靠? 》](#)
- [《不为人知的网络编程\(八\): 从数据传输层深度解密HTTP》](#)
- [《不为人知的网络编程\(九\): 理论联系实际，全方位深入理解DNS》](#)

关于移动端网络特性及优化手段的总结性文章请见：

- [《现代移动端网络短连接的优化手段总结：请求速度、弱网适应、安全保障》](#)
- [《移动端IM开发者必读\(一\)：通俗易懂，理解移动网络的“弱”和“慢”》](#)
- [《移动端IM开发者必读\(二\)：史上最全移动弱网络优化方法总结》](#)

### 3、帝国时代

我相信我们都玩过一款特别火的游戏：帝国时代。小时候想要玩帝国时代，需要到软件城购买盗版光盘安装，大概3块钱一张左右的样子，当时已经觉得很便宜了，谁想到现在有了网络之后是免费。

小A是一个帝国时代大神，他打通了游戏的所有关卡，可以一个人单挑8个疯狂的电脑玩家。渐渐地他觉得无聊了，想要找小伙伴一起PK。

但是两个电脑需要互联才行，如何实现两台设备的互联呢？

小A很聪明，他发明了一个类似于USB口一样的可以传输数据的端口，他将其命名为网口。小A通过一根网线将自

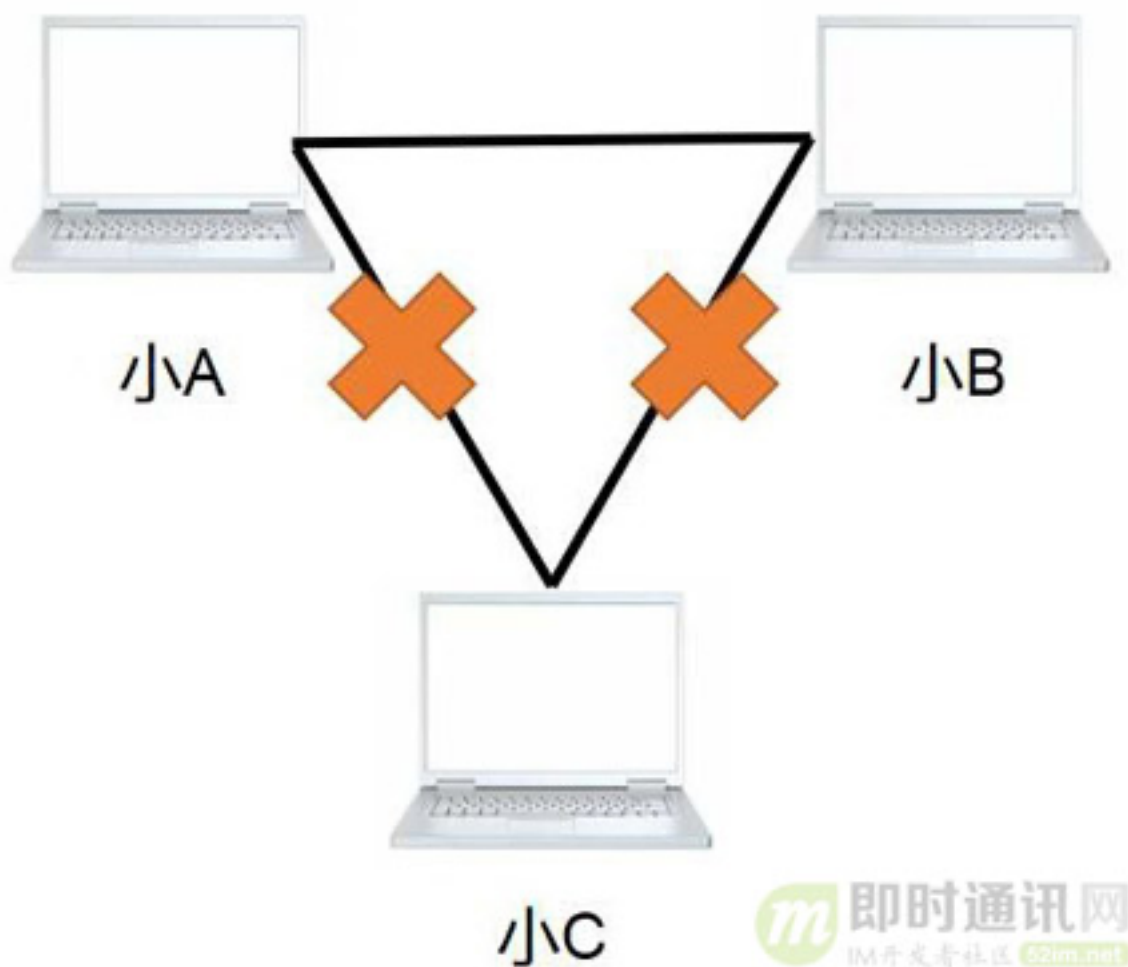
己的电脑与小B的网口相连，实现了两台电脑间的互连（如下图）。



## 4、集线器（Hub）

两个小伙伴很开心，联机玩了起来，这时被路过的小C看见了，小C也要加入进来。

但是我们知道，每台电脑只有一个网口，无法实现三台电脑的相互连接，那要怎么办呢？

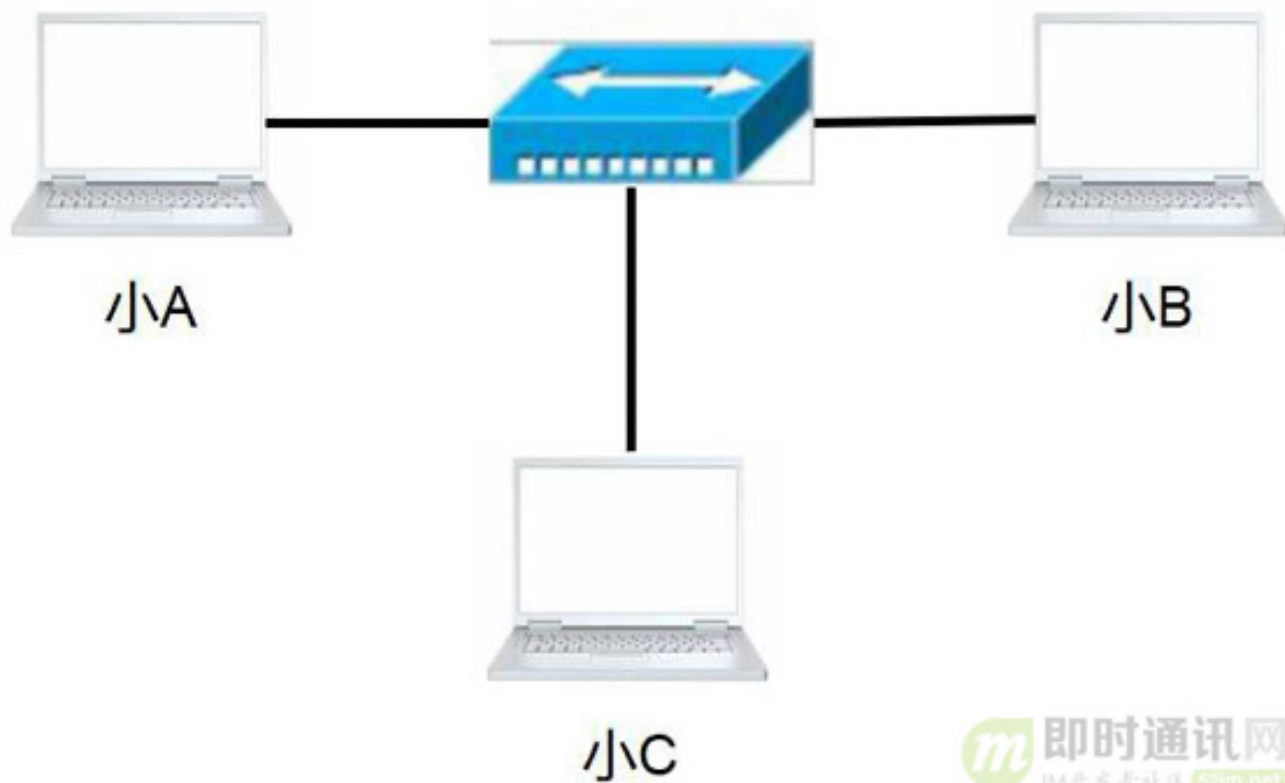


这时候小B出了一个主意：咱们再找一台计算机，给他多设计几个网口，我们每个人都连到这台计算机的网口上，不也实现咱们哥几个之间的互连了吗。

说干就干，于是他们设计出了一款微型计算机，他本身具备多个网口，专门实现多台计算机的互联作用，这个微型计算机就是集线器（HUB）。

顾名思义，集线器起到了一个将网线集结起来的作用，实现最初级的网络互通。

集线器是通过网线直接传送数据的，我们说他工作在物理层（如下图所示）。



## 5、交换机

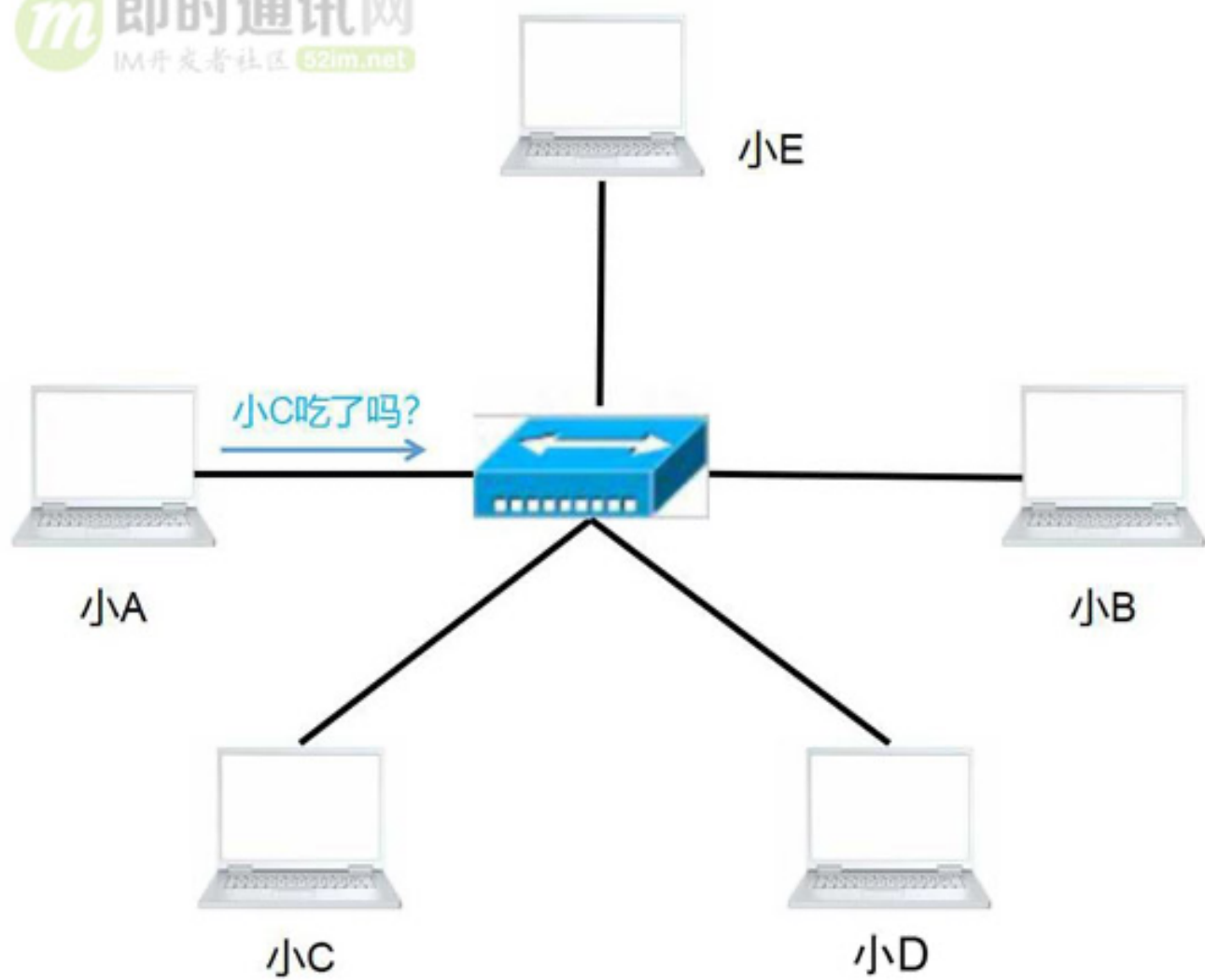
有了集线器后，越来越多的小伙伴加入到游戏中，小D、小E等人都慕名而来。

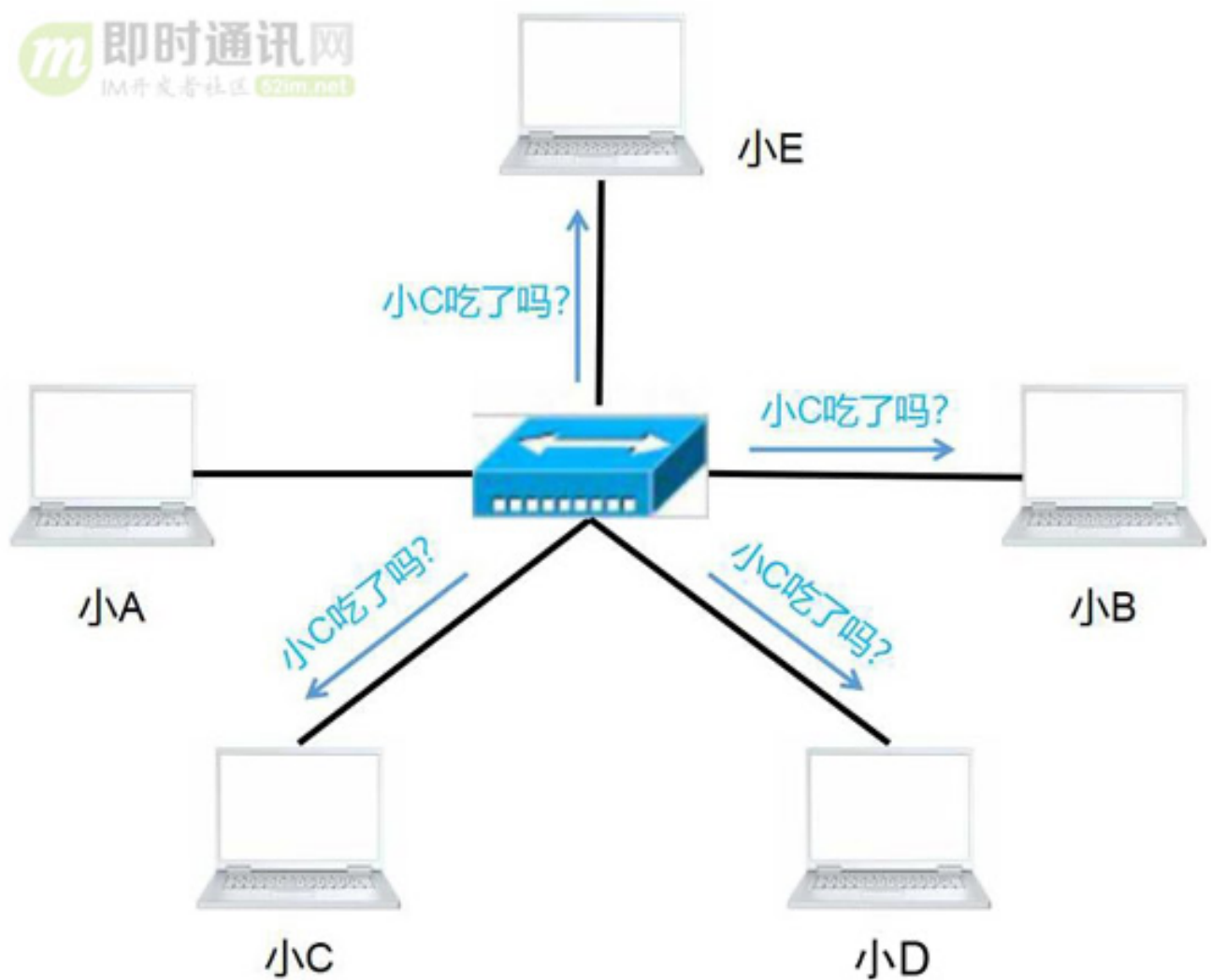
然而集线器有一个问题，由于和每台设备相连，他不能分辨出具体信息是发送给谁的，只能广泛地广播出去。

例如小A本来想问小C：你吃了吗？结果小B，小D和小E等所有连接在集线器上的用户都收到了这一信息。

由于处于同一网络，小A说话时其他人不能发言，否则信息间会产生碰撞，引发错误，对这种情况，我们称为各设备处于同一冲突域内。







这样的设备用户体验极差，于是小伙伴们一起讨论改进措施。

这时聪明的小D发话了：我们给这台设备加入一个指令，让他可以根据网口名称自动寻址传输数据。

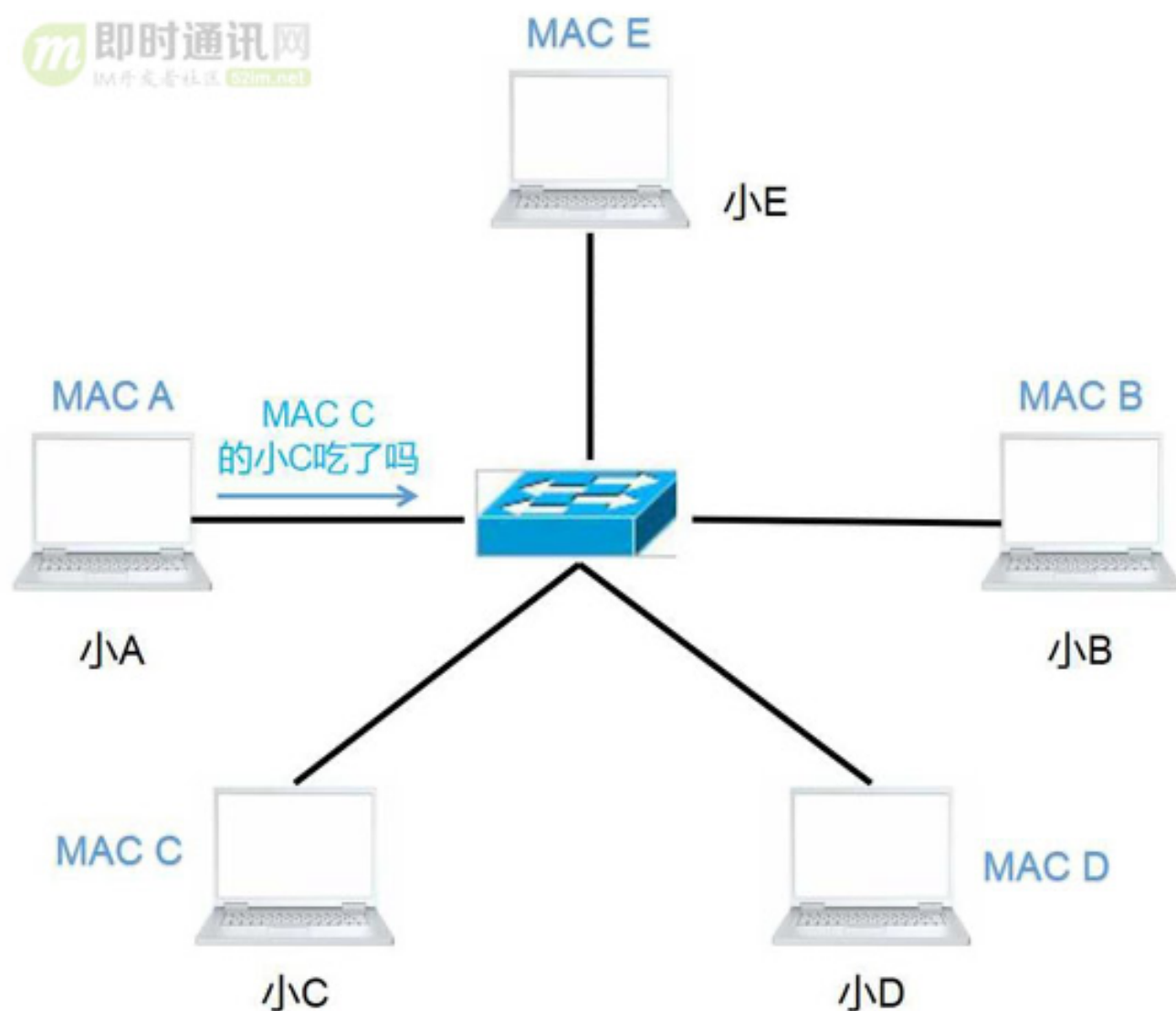
比如我把小A的网口命名为macA，将小C的命名为macC，这时如果小A想要将数据传给小C，则设备会根据网口名称macA和macC自动将资料从A的电脑传送到C的电脑中，而不让小B、小D和小E收到。

(补充说明： 这里的macA, macB指的就是MAC地址，相

当于一个人的身份证，独一无二。)

也就是说，这台设备解决了冲突的问题，实现了任意两台电脑间的互联，大大地提升了网络间的传输速度，我们把它叫做交换机。

由于交换机是根据网口地址传送信息，比网线直接传送多了一个步骤，我们也说交换机工作在数据链路层（如下图）。





这回小伙伴们高兴了，他们愉快地玩耍起来。

## 6、路由器

渐渐地，他们在当地有了名气，吸引了越来越多的小伙伴加入到他们的队伍中。有一天，一个外村的小伙突然找上门来，希望能和他们一起互联，实现跨村间的网络对战。

小A说可以呀，于是他们找了一根超长的网线将两个村落的交换机连在了一起。结果发现一件奇怪的事：两个村落间竟然不能相互通信。

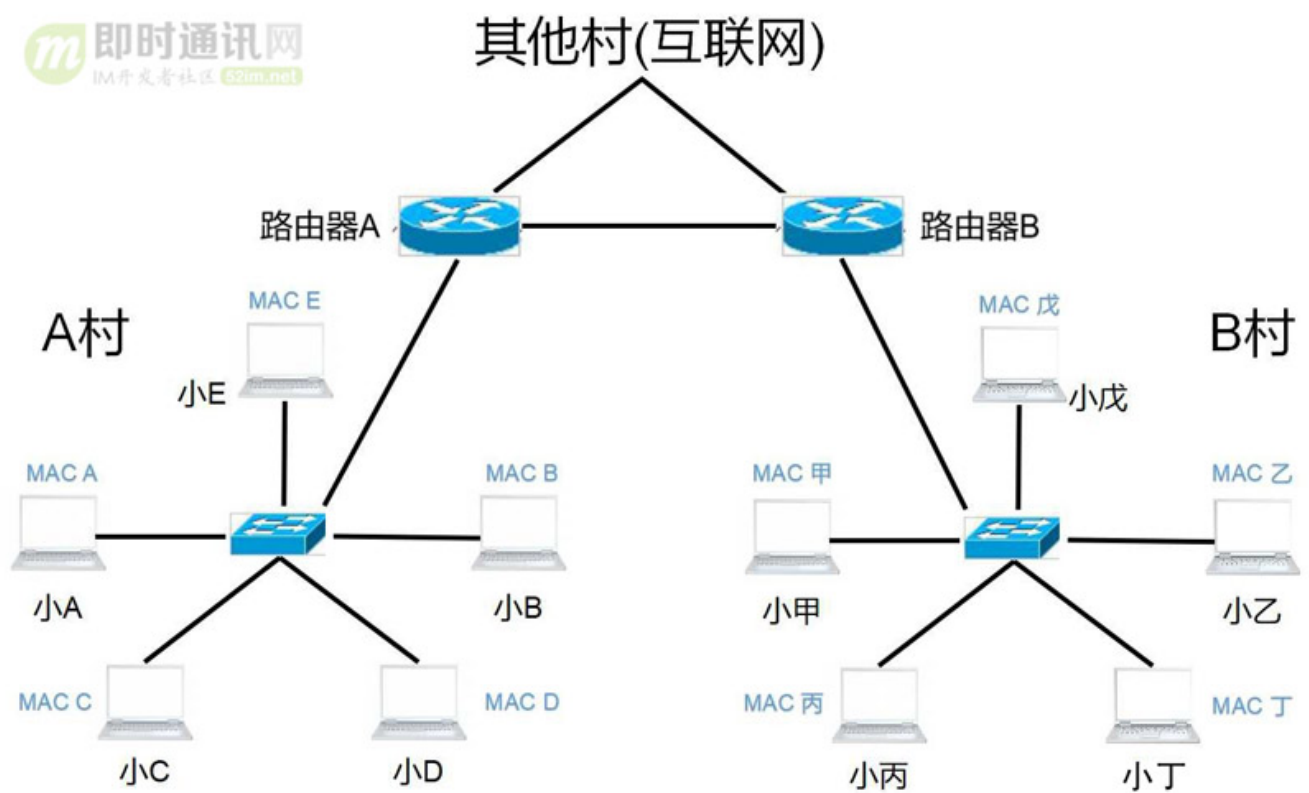
怎么回事？原来那边的电脑和他们用的不是一套操作系统，这导致信息间的传送形式的不匹配。在这期间，还有其他村落的人也来找过小A，可是小A发现，每个村子之间用的操作系统都不一样。

这可咋办呐？难道以后只能各自村子玩各自的了吗？为了解决这一问题，各村的小伙伴们坐在一起组织了一场会议，最终得出了一套解决方案：采用同样的信息传送形式（像不像秦始皇统一度量衡？）。

那如何实现呢？小伙伴们规定，不同的村子间先在各自的操作系统上加上一套相同的协议。不同村落通信时，信息经协议加工成统一形式，再经由一个特殊的设备传送出去。这个设备就叫做路由器。

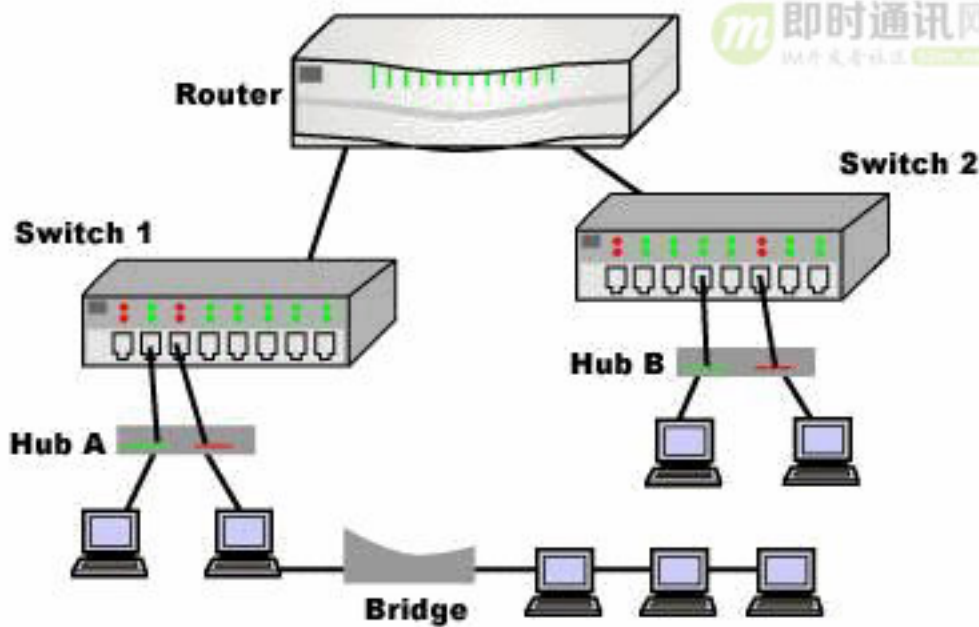
在这套协议中，每个机器都被赋予了一个IP地址，相当于一个门牌号一样。路由器通过IP地址寻址，我们说它工作在计算机的网络层。

这样，经由如此的一系列改装，小A终于带领村民们实现了整个乡镇的通信。随着越来越多的城里人也加入小A的协议，小A带领村民逐步实现了全市、全国乃至全世界的通信。这一套协议便是TCP/IP协议簇，互联网也便这样形成了（关于TCP/IP改变世界的故事，详见[《技术往事：改变世界的TCP/IP协议（珍贵多图、手机慎点）》](#)）。



然而，即便如今全网络已遍布了全世界，在小A和村里的小伙伴对战帝国时代的时候，也仍然用着交换机。只有和外面更大的世界交流的时候才用到路由器。

其实上图只是为了帮助您更好地理解路由器，一个真实的网络拓扑中，路由器、交换机、集线器是一起分工合作的，正如下图所示：



(如上图所示：Router即路由器、Switch即交换机、Hub即集线器)

## 附录：更多网络编程资料

《[TCP/IP详解 - 第11章·UDP：用户数据报协议](#)》

《[TCP/IP详解 - 第17章·TCP：传输控制协议](#)》

《[TCP/IP详解 - 第18章·TCP连接的建立与终止](#)》

《[TCP/IP详解 - 第21章·TCP的超时与重传](#)》

《[技术往事：改变世界的TCP/IP协议（珍贵多图、手机慎点）](#)》

《[通俗易懂-深入理解TCP协议（上）：理论基础](#)》

《[通俗易懂-深入理解TCP协议（下）：RTT、滑动窗口、拥塞处理](#)》

《[理论经典：TCP协议的3次握手与4次挥手过程详解](#)》

《[理论联系实际：Wireshark抓包分析TCP 3次握手、4次挥手过程](#)》

《[计算机网络通讯协议关系图（中文珍藏版）](#)》

《[UDP中一个包的大小最大能多大？](#)》

《[P2P技术详解\(一\)：NAT详解——详细原理、P2P简介](#)》

《[P2P技术详解\(二\)：P2P中的NAT穿越\(打洞\)方案详解](#)》

《[P2P技术详解\(三\)：P2P技术之STUN、TURN、ICE详解](#)》

《[通俗易懂：快速理解P2P技术中的NAT穿透原理](#)》

《[高性能网络编程\(一\)：单台服务器并发TCP连接数到底可以有多少](#)》

《[高性能网络编程\(二\)：上一个10年，著名的C10K并发连接问题](#)》

《[高性能网络编程\(三\)：下一个10年，是时候考虑C10M并发问题了](#)》

《[高性能网络编程\(四\)：从C10K到C10M高性能网络应用的理论探索](#)》

《[技术扫盲：新一代基于UDP的低延时网络传输层协议——QUIC详解](#)》

《[让互联网更快：新一代QUIC协议在腾讯的技术实践分享](#)》

《[现代移动端网络短连接的优化手段总结：请求速度、弱网适应、安全保障](#)》

《[聊聊iOS中网络编程长连接的那些事](#)》

《[移动端IM开发者必读\(一\)：通俗易懂，理解移动网络的“弱”和“慢”](#)》

《[移动端IM开发者必读\(二\)：史上最全移动弱网络优化方法总结](#)》

《[IPv6技术详解：基本概念、应用现状、技术实践（上篇）](#)》

《[IPv6技术详解：基本概念、应用现状、技术实践（下篇）](#)》



>> [更多同类文章 .....](#)