

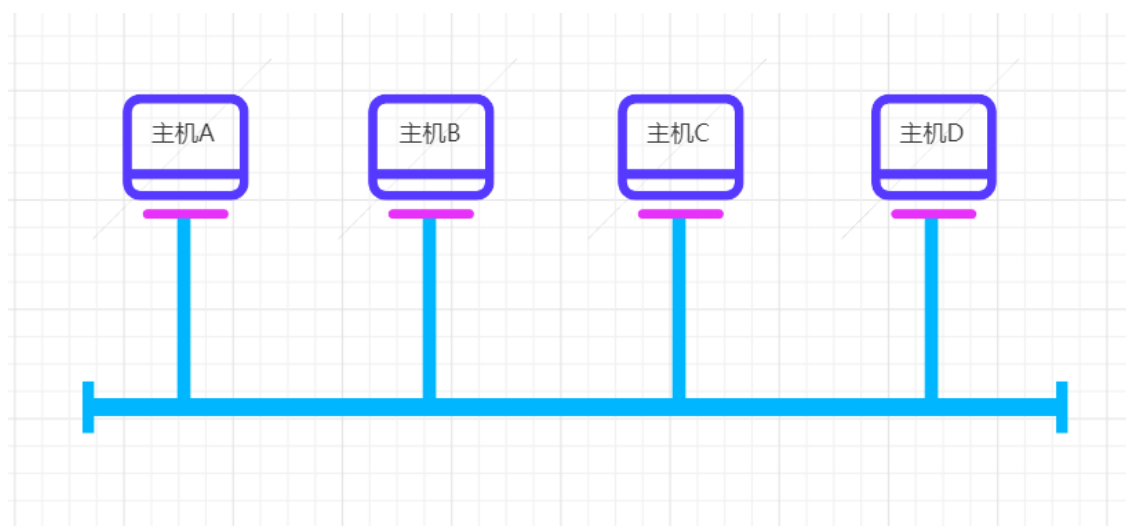
如果你要互连多台机器，比如多台电脑、打印机、路由器，等等，那你就需要一个设备来连接。

大家好，我名字叫以太网交换机，我还有其他名字叫做：交换式集线器、二层交换机等，下面为了简单，我就简称为交换机，毛遂自荐下，以上任务可以交给我来做，今天我们互相认识下，以及跟我的老前辈集线器（HUB）比较下，看看我两有啥不同、有啥相同，我为什么取代了前辈的地位，在二层协议下独步天下、称霸武林，下面请让我慢慢道来。

二层交换机工作在数据链路层，二层交换机工作于OSI模型的二层（数据链路层），故而称为二层交换机，主要功能包括物理编址、错误校验、帧序列以及流控。而三层交换机位于三层（网络层），是一个具有三层交换功能的设备，即带有三层路由功能的二层交换机，二层交换机用于小型的局域网络，它的快速交换功能、多个接入端口和相对便宜的价格为小型网络用户提供了很完善的解决方案。而三层交换机的最重要的功能是加快大型局域网络内部的数据的快速转发，加入路由功能也是为这个目的服务的。

首先，老前辈集线器是工作在物理层的，而我工作在数据链路层，所谓站的越高、看的越远，我的诞生就是为了解决集线器前辈自身存在的缺陷，至于我解决了什么缺陷，下面会详细说明。

在介绍老前辈集线器之前，我们有必要说一说更早的时候，有线领域局域网是用什么设备来互联的，我们仍然以早期最常见的总线型以太网拓扑结构为例。



一、早期使用同轴电缆的连接方式

最初是用很粗的同轴电缆作为传输媒体，要把它弯曲是十分困难的事情，并且，如果弯曲太过，咔嚓，内部的数据线就被折断了，网络就断了，线缆也只能被扔进垃圾桶。

后来演进到使用价格相对便宜的细同轴电缆，主机和总线之间通过机械连接头连接，这种方式十分流行，因为那个时代普遍认为有源器件不可靠，而无源的电缆线才是最可靠的。

细同轴电缆的连接方式这里简单说下，它要用到下面三样东西：

器件说明	图片
BNC 缆线连接器： 焊接或拧接在电缆的端部	
BNC T型头： 用于连接机器的网卡和网络中的电缆	
BNC 终端器： 用于防止信号到达电缆断口后反射回来产生干扰	

BNC 表示 BNC 连接器（BNC 是 Bayonet Neill-Concelman 的缩写，表示“尼尔-康塞曼卡口”），是一种很常见的 RF 端子同轴电缆终结器。BNC 电缆连接器由一根中心针、一个外套和卡座组成。它包括 BNC 连接器基座、外套和探针三部分。BNC 连接器得名于其锁定方式和其发明者贝尔实验室的保罗·尼尔（发明了 N 端子）和安费诺公司的工程师卡尔·康塞曼（发明了 C 端子）。BNC 电缆连接器必须连接到每段电缆的两端。

为了构建一个网络，我们在 BNC T型头的一端接上一个 BNC 终端器，把连接机器的网卡的那根线连接到T型头中央的那个头上，然后其它的线接在 T型头的另一端。

因此，T型头的另一端连接的就是另一个T型头，这样一个接一个地连接，直到网络被一个 BNC 终端器给封闭为止。

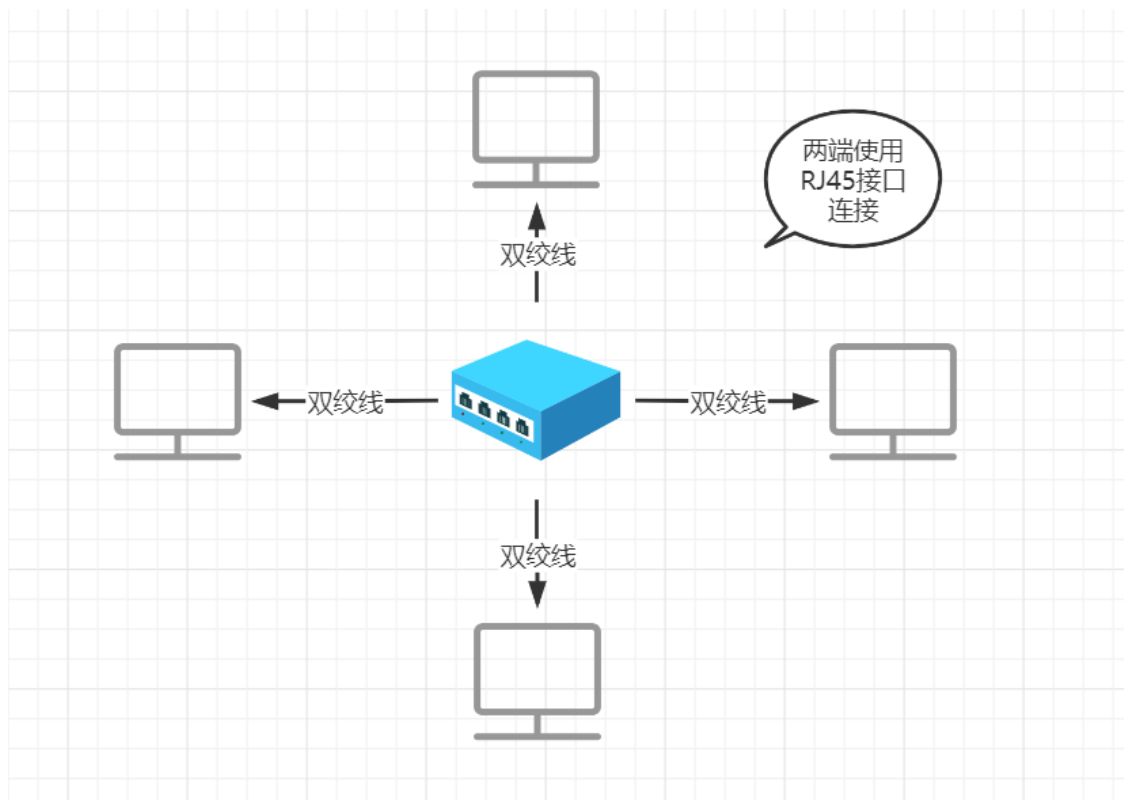
完整的 BNC 网络如下图所示：



不过这种由无源同轴电缆结合大量机械接头的总线型以太网，并不像人们想象的那么可靠，我们可以发现在这种组网方式下，一旦某个人断开自己的那段网络的话，他就会把整个网络也断开了...

二、集线器HUB+双绞线电缆

后来，我的前辈集线器HUB和双绞线电缆横空出世，如下图所示，通过集线器和双绞线互连四台主机形成星型拓扑的以太网：



看起来是星型结构，但是前辈偷偷告诉我，实际上在使用他们的以太网在逻辑上仍然是一个总线网，各站共享总线资源，使用的还是CSMA/CD协议，我们可以认为集线器就是担任了总线的角色，仍然会产生冲突。

并且前辈是只工作在物理层，它的每个接口仅仅简单地转发比特，不进行碰撞检测，碰撞检测的任务是交给各站点的网卡的。

此外，前辈说他们一般都有少量的容错能力和网络管理功能，例如若网络中某个网卡出现了故障，不停地发送帧，此时，集线器可以检测到这个问题，在内部断开与此故障网卡的连线，使整个以太网仍然能正常工作。

前辈的实际模样想必大家都很有趣，我贴个它的照片：



他告诉我他身上长满了RJ45接口的母头，我照了照镜子，发现我跟他长的竟然很像。。。

江湖上大家都说他是大嘴巴，我不小心了解了下他的工作原理，原来他真的是大嘴巴，怪不得被我干掉了，我们来详细看看老大哥集线器是如何工作的：

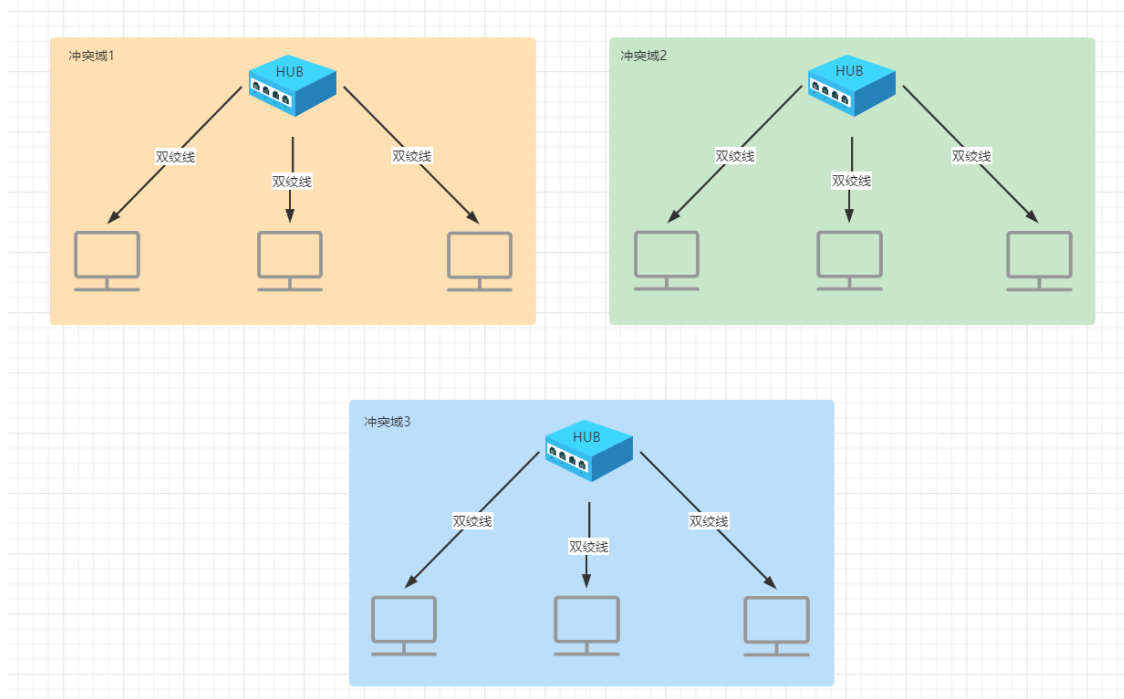
假设现在有 5 台机器连接在集线器上，分别是机器 A、机器 B、机器 C、机器 D 和 机器 E。如果机器 A 要和机器 C 通信，它就把信息发送给集线器。

但是集线器很“笨”，不会读，只知道把信息“广而告之”。它只能把机器 A 发给它的信息“广播”给所有连接在集线器上的机器。机器 B，机器 D 和机器 E 也会收到机器 A 发的信息，但是看到信息不是给它们的，就可以丢弃信息。机器 C 看到信息是发给它的，就会读取信息。

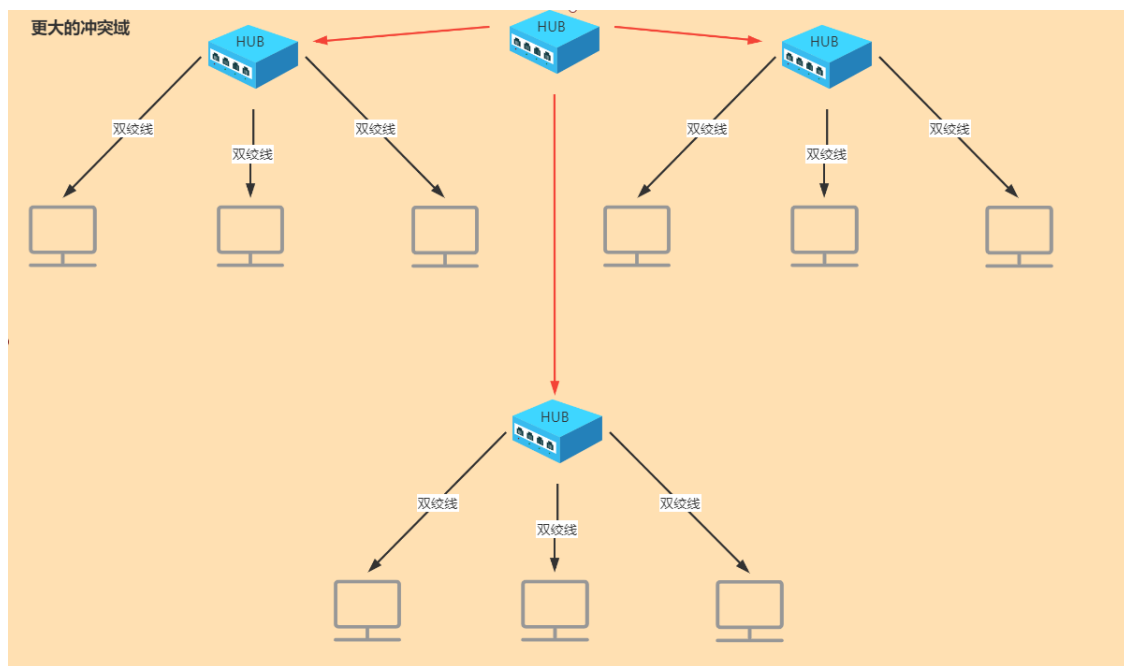
集线器比较“呆”，有点“大嘴巴”，不过也还算好使。

所以说集线器不利于信息的保密，它发送数据时是没有针对性的，而是采用广播方式发送。

集线器可以在物理层拓展以太网，假设现在有三个独立的以太网：



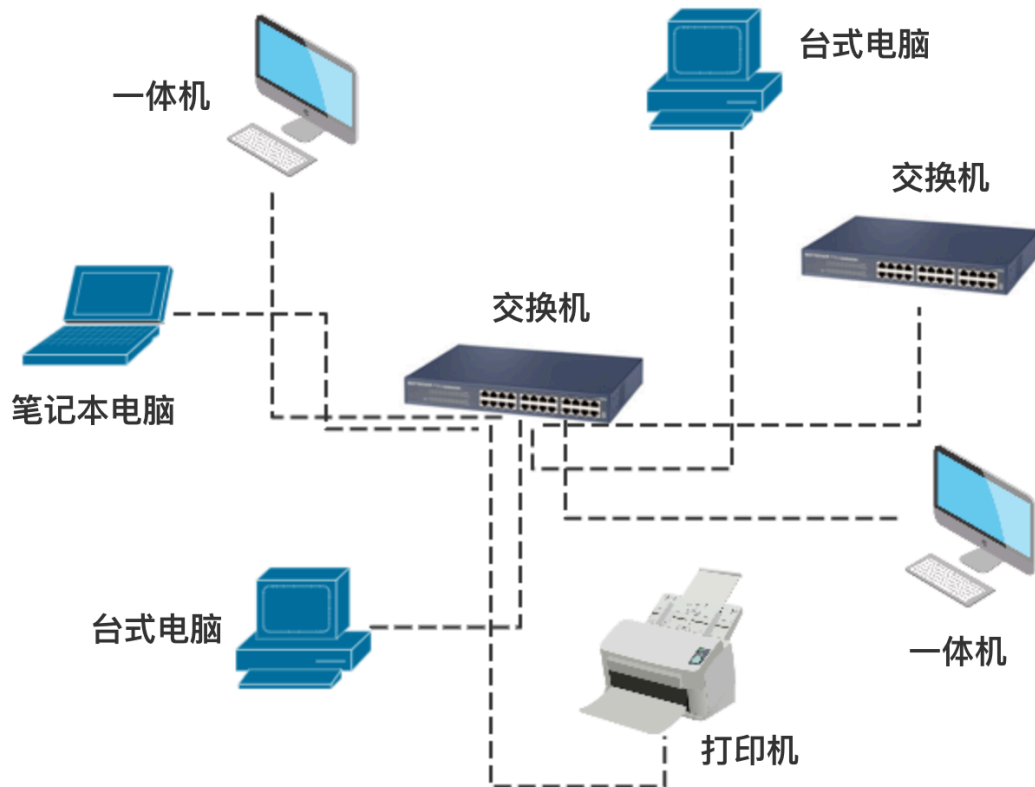
此时再使用一个集线器将他们互连起来，这样，原来三个独立的以太网互连成为一个更大的以太网，并且原来三个独立的碰撞域（冲突域）就合并成了一个更大的碰撞域，换句话说形成了一个更大的总线型以太网。



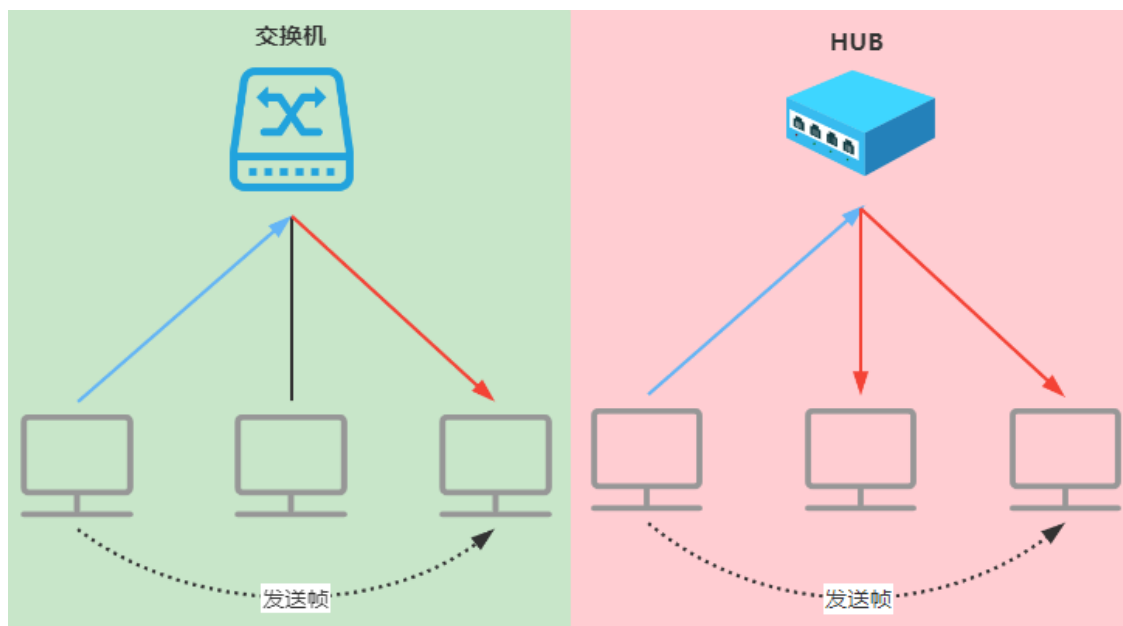
哈！前辈啊前辈，你说你跟总线有啥区别？还不是通过大喇叭到处喊，还不是到处充满了冲突？还是休息休息吧，我该登场了！

三、交换机SWITCH+双绞线电缆、集线器HUB+双绞线电缆

实际上我也是长满了RJ45接口（母头），我也可以把其它机器（电脑、打印机、传真机、扫描仪，等等）连接起来，或者把别的交换机连接进来。如下图所示：



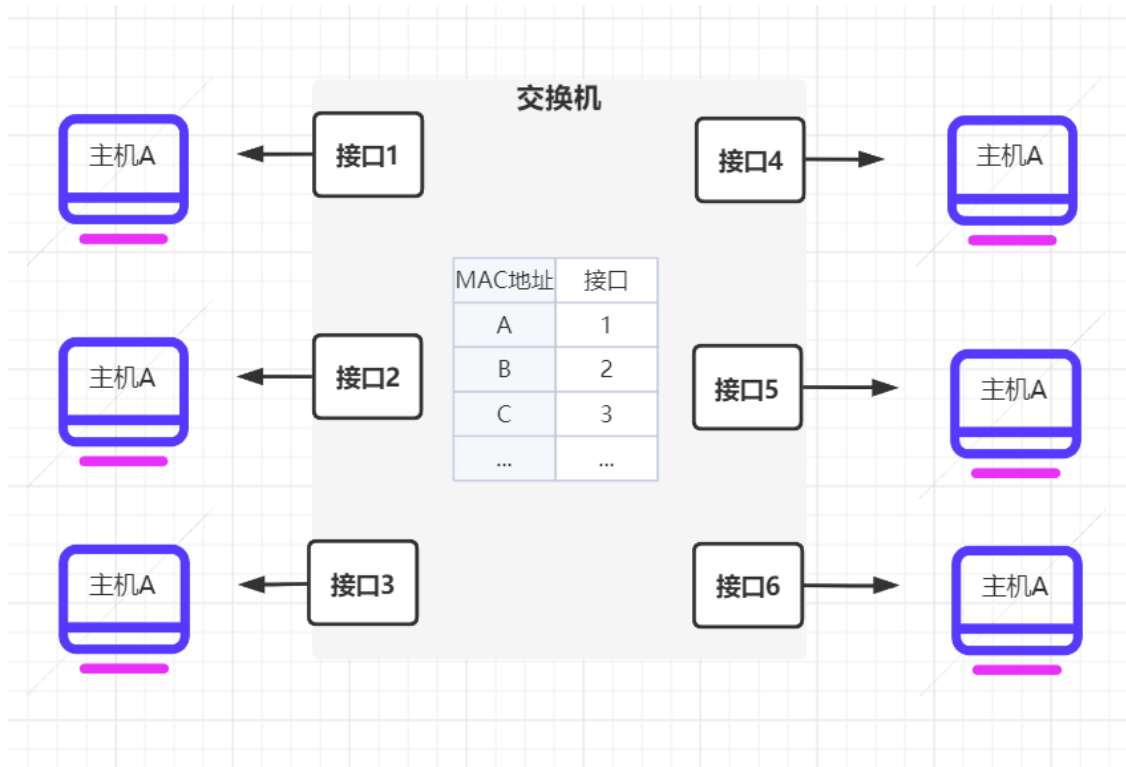
先来直观看下我和集线器的区别，假设一个主机要给另一个主机发送单播帧：



注：红线表示帧经过此处，黑线表示帧没有经过。

可以看到，单播帧进入到达我这边后，会将单播帧转发给目的主机，而不是网络中的其他主机；而集线器不同，单播帧发给集线器后，集线器广而告之所有其他主机。

那么我是如何知道谁发给谁的呢？实际上我也要学习，通过学习才能知道哪个主机在哪个接口下面，我会在学习的过程中会记录一张MAC地址和接口映射表，我通过读这张表就知道应该往哪里发啦：



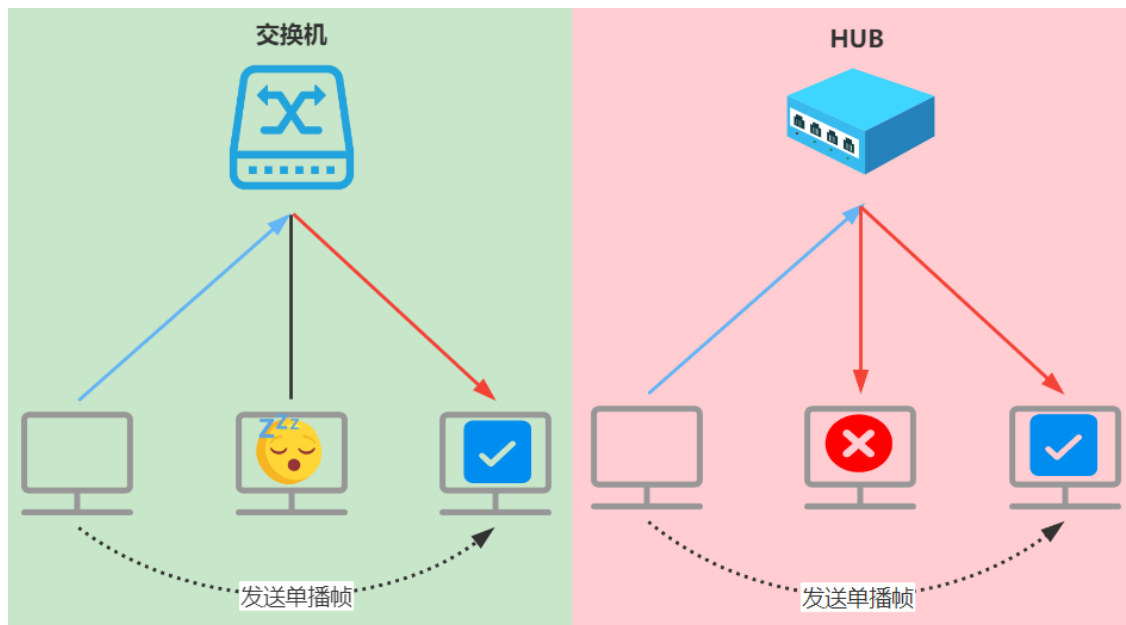
至于我如何学习的，这篇文章就不提前剧透了，现在您只要知道，我是通过这张表来明确知道往哪个接口发送消息，因此不用像集线器那样只能靠广播来传递。

此外，通过上图可以看到：

- 我通常有多个接口，每个接口都可以直接与一台主机或另一个以太网交换机相连，一般工作在全双工模式。
- 我具有并行性，能同时连通多对接口，使多对主机能同时通信，得益于已经摒弃CSMA/CD这个协议，因此在我这里不会产生碰撞。
- 对了，我一般具有多种速率的接口，例如10Mb/s、100Mb/s、1Gb/s、10Gb/s接口的多种组合。
- 我工作在数据链路层，当然我由于跟MAC地址打交道较多，因此实际上我的工作范围也包含了物理层。
- 我是一个学习能力很强的小能手，刚才图上面的MAC地址和接口的对应关系是我自己通过学习算法自动逐渐建立起来的，因此我即插即用，插上自动学习！
- 此外，我既支持把帧存储下来转发，也支持直通交换：采用基于硬件的交叉矩阵将帧不存储而直接发送出去，这样做时延会很小。

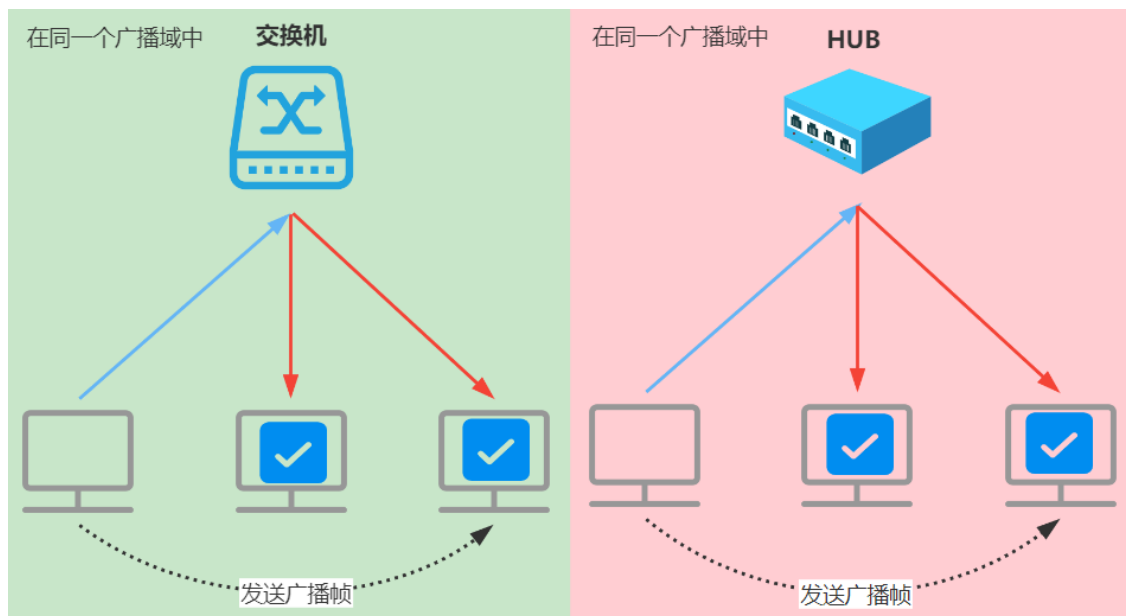
四、进一步对比交换机和集线器、交换机SWITCH+双绞线电缆、集线器HUB+双绞线电缆

第一种情况：发送一个单播帧，情况应该如下：



具体上面已经解释，交换机只会发给特定的主机，而集线器永远都是广播到所有与之相连的主机上，具体上面已经解释，交换机和集线器的表现不同。

第二种情况：此时要是发送一个广播帧呢？



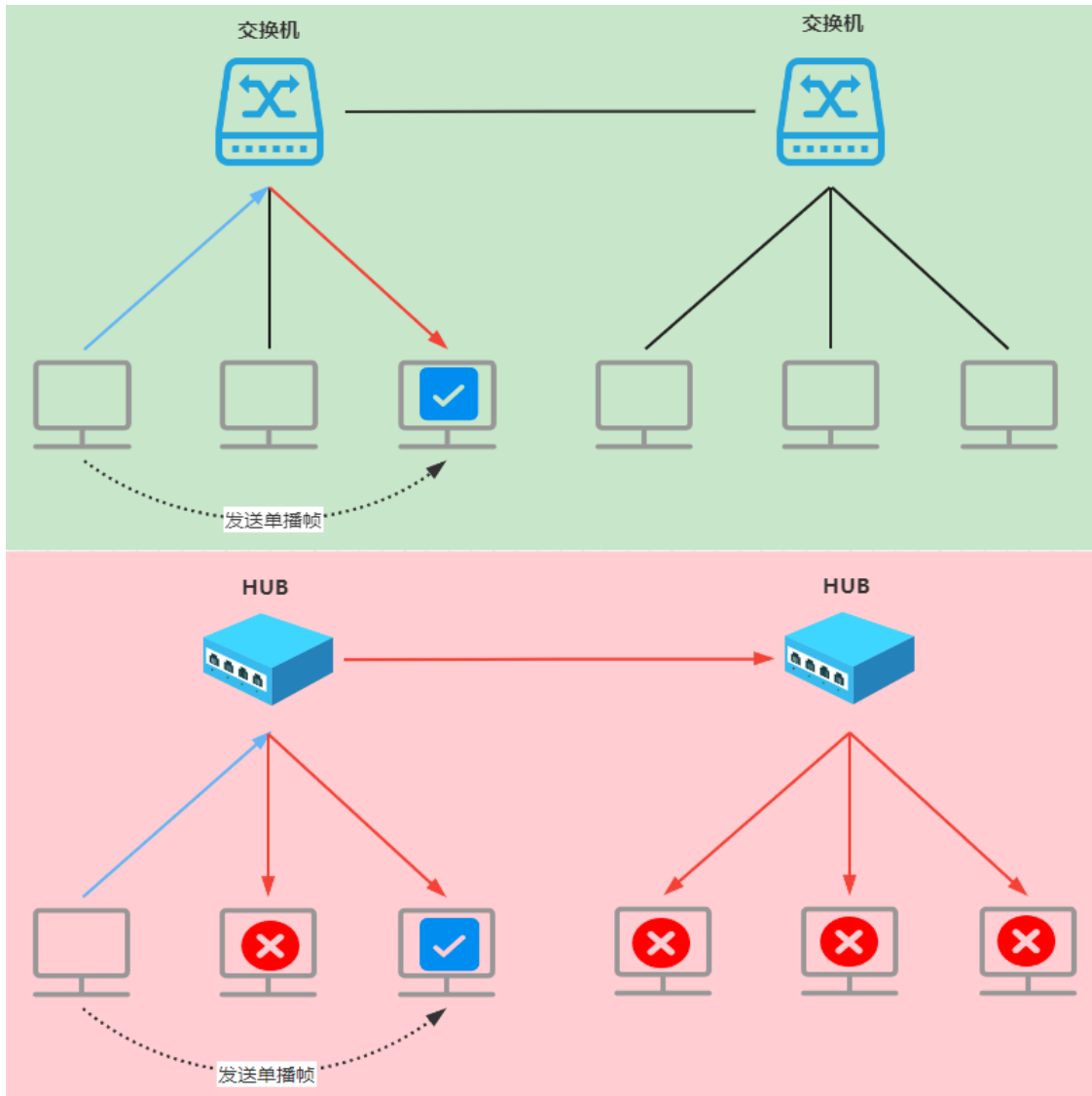
看的出来效果实际上一样，使用集线器的共享总线型以太网中的各主机属于同一个广播域；而使用交换机的交换式以太网中各主机也属于同一个广播域。

第三种情况：假设此时两台主机同时向另一台主机发送单播帧呢？

使用集线器的共享总线型以太网中必然会产生碰撞，遭遇碰撞的帧会传播到总线上的各主机；使用交换机的交换式以太网中，交换机同时收到多个帧时，会进行缓存，然后逐个转发给目的主机，不会产生碰撞。

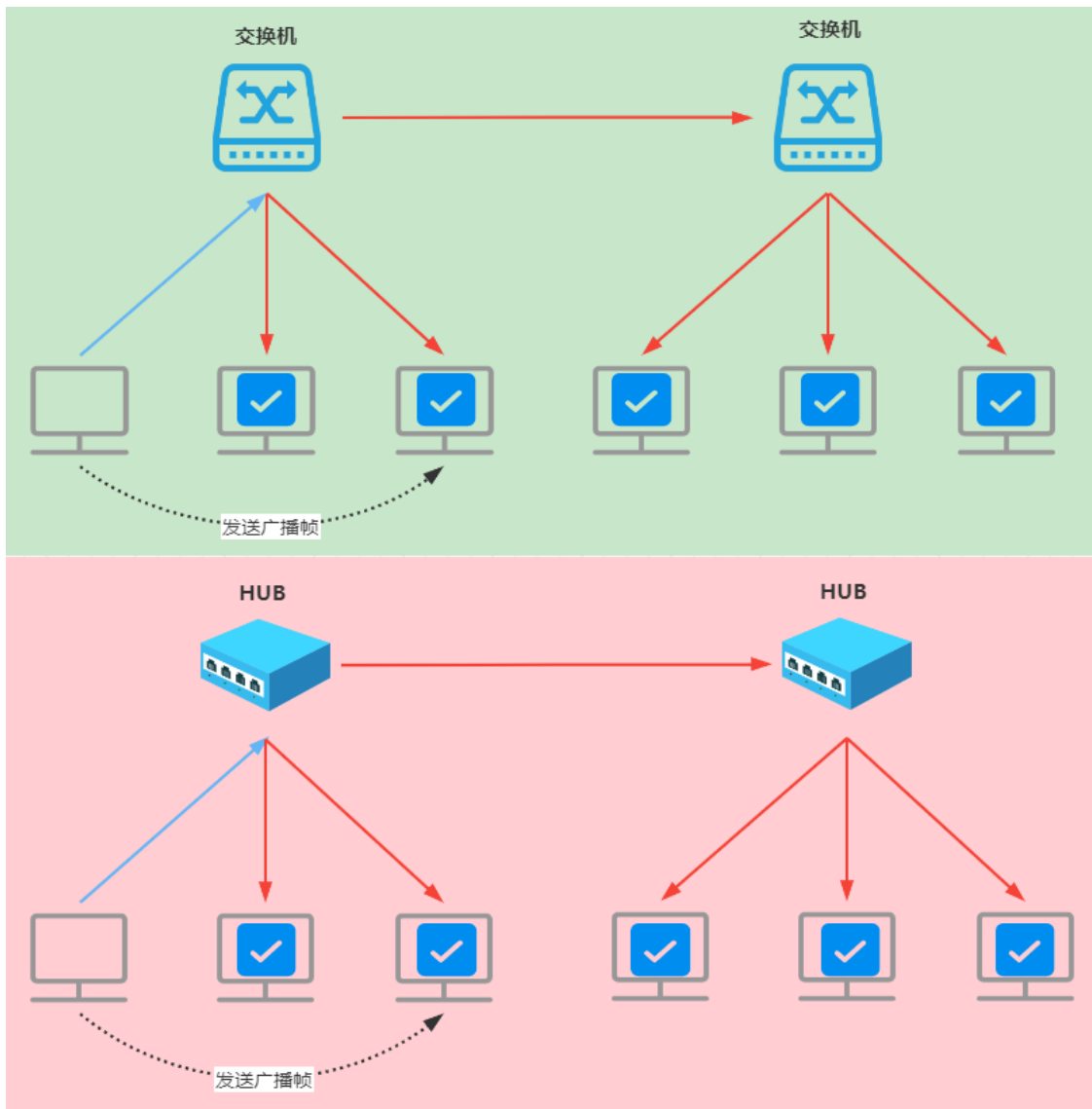
接下来对比使用集线器扩展以太网和使用交换机扩展以太网的区别。

第四种情况：发送一个单播帧：



交换机的优势在这里凸显无疑，无需赘述。

第五种情况：发送一个广播帧：



不管是使用集线器扩展以太网还是使用交换机扩展以太网，扩展后的各主机都属于同一个广播域，值得注意的是，使用集线器扩展以太网，逻辑上仍然是总线的，因此形成了一个更大的碰撞域，换句话说，参与竞争总线的主机比扩展前更多了，**因此在使用集线器扩展以太网时，不仅扩大了广播域，还扩大了碰撞域；而使用交换机扩展以太网只会扩大广播域，而不会扩大碰撞域。**

回过头来，我想已经解释清楚一开始的问题：我两有啥不同、有啥相同，我为什么取代了前辈的地位、我解决了集线器的哪些缺陷！

总而言之啊，在我这里不存在碰撞和冲突，我不是每次通过广播去发送帧，无论是效率、安全性我都完胜集线器！以使用我的交换式以太网已经取代了传统使用集线器的共享式以太网！

文中提及到了MAC地址和接口的对应表，这个表就是我的灵魂，它有个名字叫做：**CAM 表**，你一定很想知道我是如何学习和维护CAM表的吧？下一篇文章我们继续深入交流！