第十三章 握手言和

IP与FC融合的结果

- FC
- IP
- 协议之间的相互作用
- 协议融合

话说FC和IP各占一方，谁也不让谁，互相竞争了数年，两者各立门派，势不两立。但是“夫天下之势，分久必合，合久必分”。

数年来，两者在市场上竞争的可谓你死我活。FC仅仅拿着FC SAN的速度和稳定性来炮轰IP SAN，而IP也不甘示弱，处处举着可扩展性和成本的大旗，声讨FC SAN，闹得江湖上风风雨雨。FC凭借着它的速度优势，占据了高端市场，而IP则以成本优势在低端市场占据了一席之地。然而两人谁都想一统天下，把对方彻底驱逐出市场，但是，相持数年了，谁也没能把谁干掉。两人都累了，这么多年的互相攻击，谁也没有取得丝毫胜利，FC还是稳固的占据高端市场，IP依然驰骋低端。

终于有一天，FC和IP决定握手言和，不再投入无谓的人力、物力、财力来和对方竞争。与其大肆攻击对方，不如多用点精力来提升和发展自身的技术，同时学习对方的技术，取长补短，方为正道啊！！ FC和IP彻夜长谈，终于取得了一致的见解，决定双方各取所长，共同为江湖做贡献。

首先，FC决定由IP入股自己的公司，给FC SAN提供更高的扩展性架构解决方案；同时，FC也入股IP的公司，给IP提供研发经费，用于其研发出基于以太网的、新型的、适合存储区域网络的专用协议体系。

13.1 FC的窘境

入股 FC 公司之后，IP 便开始研究如何将 FC 协议体系转向可扩展的、开放的结构。说到可扩展且开放的网络传输协议，一定非 TCP/IP 末属。可是 FC 和 TCP/IP 是完全两套毫不相干的协议体系，如果将 FC 全部转为 TCP/IP，那岂不是叛变成 IP SAN 了么？但是如果丝毫不变，那只能是 FC SAN，还是不具备开放和扩展性。

1. FC的扩展性问题

FC 为什么扩展性差？就是因为如果通信双方距离太远的话，需要自己架设光缆，或者租用电信的专线光缆，这两者成本都很高。如果租用电信部门的专线光缆，则 FC 最低速度为 1Gb/s，且租用电信部门的 1Gb/s 带宽的专线光缆，其费用不是一般机构能承担的。

目前电信提供的专线接入，其骨干网一般采用 PDH 或者 SDH 协议传输，到终端用户所能承受的速率为 2M 的 E1 线路。当然也可以直接从高速骨干直接分离出相对高速的线路，比如 OC3，OC48 等，但是费用还是过于高昂，无法承受。

E1 线路有自己的编码格式，不能将电信部门接入的光纤直接插到 FC 设备上，因为两端的编码方式不同，不能和局端的设备建立连接，所以需要增加一个协议转换设备(准确来说是协议隧道封装设备)，将 E1 协议解封装，转换成协转设备后面的协议逻辑，比如 V35 串口、以太网等其他协议。目前已经存在 FC over Sonet，FC over ATM 等协转设备了，不过这些专线的扩展性仍然不强，而且这种方案以及对应的设备也非常昂贵和稀少。

目前看来，如果要扩展 FC 网络，让相隔很远的两地之间用上 FC 协议，最好的办法就是自己架设专用光缆，可是自己架设光缆也只能在自己可控的范围内，比如一个大厂区之内，但是如果是在市内，或者两个城市、两个省之间，私自架设光缆是绝对被禁止的。

2. 解决方案

怎么办？首先，要走出去，就一定要租用电信部门的线路。电信提供了两种线路，一种是接到 Internet 的线路，也就是接入电信部门的 Internet 运营网络，通信的双方都接入，并且使用 TCP/IP 通信。另一种，就是光缆专线，也就是通信的双方都接入电信部门的专用传输骨干网络，这条专线端到端的带宽由接入提供商保证，只要两端的设备支持，其上可以运行任何上层协议。上层帧会被底层封装协议(比如 E1 等)再成帧传送到电信部门骨干传输网络中。

虽然 Internet 接入可以获得 100Mb/s 或者 1000Mb/s 的速率，但是这只是本地带宽(从本地到局端设备之间的链路带宽)，端到端的带宽，以现在的电信部门 TCP/IP 网络环境，除非购买接入商的 QOS 或者 MPLS TE 服务，否则没有人能够保证两点间的通路带宽(速率)。

如果两地之间相距很近，那么不妨考虑 Internet 链路。因为如果两地同时接入相同城市的 ISP 网络，数据包被路由的跳数就不会很高，甚至有可能只经过 1 跳或者 2 跳便可以被对方收到。更有甚者，同城的两地可能连接在局端的同一台设备上。这样可获得的带宽速率就会非常可观，就可以像在内网通信一样利用 VPN 来让两个站点之间联通。但要澄清一点，由于 Internet 链路不能时刻保障稳定的带宽，所以这种方法只适合对数据传输实时性要求不高，但同时又要求高带宽的情况。

而专用线路虽然保证了带宽，但是只能承受 E1 等低速专线，且价格相对 Internet 接入要贵很多。而且目前只有 V35－E1 封装解封设备和 E1－以太网封装解封设备，并没有 E1－FC 封装解封设备。而 V35 串口和以太网这两种二层协议，都普遍被用来承载 IP 协议，所以目前来说，E1 一般用来承载 IP 作为网络层协议。有些路由器自带 E1 封装解封模块，可以不用外接协转，直接连接从光端机分离出来的一路或者几路 G703 或者 BNC 接头，直接编码与解码 E1 协议。但是这些也都是 IP 路由器，和 FC 丝毫没有关系。

可以看出，FC 如果脱离了“后端专用”这四个字到开放领域，显然是无法生存的。而 IP SAN，则软硬通吃，只要有 IP 的地方，不管其下层是什么链路协议，就可以部署 IP SAN。这就是为何称 TCP/IP 为协议中的秦始皇的原因，秦始皇统一了货币，到哪里都通用，同样，TCP/IP 也统一了下层凌乱的各种协议。

13.2 协议融合的迫切性

说到这里，租用 Internet 线路，只能承载 IP，而租用点对点专线，也普遍用来承载 IP，可能感觉 FC 的扩展似乎就是死路一条了。但是，IP 想起了 ISCSI，当初自己不就是把 SCSI 协议给封装到了 TCP/IP 协议中来传输，才扩展了 SCSI 协议么？也就是说如果将一种协议封装到另一种协议中传输，就可以使用另一种协议带来相应的好处了。不妨就这么假设一下，FC 不可扩展，TCP/IP 扩展性很强，那么如果把 FC 协议封装到 TCP/IP 协议中来传输，是不是也可以获得 TCP/IP 的扩展性呢？这个想法比较大胆，因为 FC 本身也是作为一种可以传输其他协议的协议，FC 甚至可以承载 IP，作为 IP 的链路层，那么为什么现在却反过头来需要用 IP 来承载呢？

Protocol over Protocol，PoP，即一种协议被打包封装或者映射到另一种协议之上。这种思想在网络协议领域中经常使用。我估且称其为“协议融合”，认为其已经可以形成一个独立的科目。

要谈协议融合，还得从以太网和 TCP/IP 说起。

以太网和TCP/IP——不能不说的故事

前面已经详细的介绍了以太网和 TCP/IP 协议。我们知道，以太网是一个网络通信协议。

记得某人曾经说过一句话：“网络就是水晶头。”这句话比较有意思，它反映出说这句话的人对网络的不了解，但是也证明他平时所见到的网络，确实只有水晶头，且以太网普遍使用水晶头，那么“网络就是水晶头”这句话，也不是那么可笑了。它从某种角度也反映出了以太网在当今的普及程度。

前面讲到以太网是可以寻址的，也就是说它涉及了 OSI 第三层网络层的内容。大家都连接到一个以太网环境中，不需要任何其他上层协议，就可以区分开对方，进行通信。既然这样，为什么连新闻联播的主持人都知道 Internet 是利用 TCP/IP 协议而不是以太网来通信呢？为何我们总是说以太网+TCP/IP 协议二元组，而不是仅仅说以太网，或者 TCP/IP 协议？

因为以太网和 TCP/IP 协议是逻辑上分开的，它们各自是不同的协议体系，那么为什么总是把他们组合起来说呢？它们之间有什么割舍不断的恩恩怨怨呢？这其中原因，还要从 IP 讲起。

1. IP本位

前面也说过了，IP 就是一个身份标志，一个用来与其他人区别的一个 ID。以太网协议中规定的 MAC 地址，从原理上讲，就足够用来区分网络中各个节点了。但是前面也分析过，完全靠 MAC 来寻址的缺点：一是 MAC 地址太长，48bit，用于路由寻址时效率太低；二是世界上并不是每个环境中都用以太网来建立网络的，除了以太网，还有其他各种方式的网络系统，各自有各自的寻址方式，如果要让所有类型的网络之间无障碍的相互通信，就需要一个秦始皇来统一天下的货币。

IP 就是这个被选中的货币。不管以太网，或者串口协议，或者 FDDI 等类的局域网方式，我们最终都要让其之间相互通信，才能形成 Internet。

如果你是秦始皇，你会怎么来处理各国众多的货币呢？虽然秦始皇最终将其他货币回收废除了，但是 IP 却不能在短时间内将所有网络形式都废除，而用以太网统一，因为现在已经不是一个人就说了算的时代了。秦始皇可以在各个使用不同货币的地方设立一个专门的兑换机构，只要到了这个地方，就兑换成这里使用的货币。

同样，我们也给每个网络设立一个网络地址兑换设备，也就是协议，将统一的 IP 地址兑换成这个网络的自用私有地址，用这种方式实现各种类型网络的相互联通。网络中的兑换机制，是通过 ARP 协议实现的，ARP 协议可以将一种网络地址映射成另一种网络地址。每种网络要想用 IP 来统一，都必须运行各自的 ARP 协议，比如以太网中的 ARP 协议，帧中继网络中的 ARP 协议等。

对于以太网来说，IP 就是统一货币，MAC 就是以太网货币。另外，还有各种各样其他类型的货币，比如主机名(Hostname)、域名等。大家在访问网站的时候，其实就是和提供网站服务的服务器来建立通信，获取它的网页和其他服务，在 IE 浏览器中输入这个网站的域名之后，DNS 兑换程序会自动向 DNS 服务器查询，获得这个域名所对应的 IP 地址，然后

用 IP 地址与服务器通信。

数据包带着 IP 地址到了服务器所在的局域网之后，会通过局域网的路由器发出 ARP 请求，来把 IP 地址再兑换成服务器所在局域网络的地址，如果服务器所在的局域网是以太网，则对应成 MAC 地址，然后通过以太网交换设备，找到这个 MAC 地址所在的交换机端口，将数据包发向这个端口，从而被服务器收到。

为什么要经过多次兑换呢？首先把 IP 转换成域名，是为了方便记忆，不必记忆那些复杂的 IP 地址。其次把 MAC 转换为 IP，是为了天下统一，相互流通。

其实如果所有人都用以太网联网，那么就可以完全抛弃 IP 这一层寻址了，但是实际是不可能的，以太网现在还没有一统天下，而且就算一统天下了，人们也似乎不愿意抛弃 IP，就像在同一个局域网内，还是用 IP 来直接通信，而不是直接用 MAC。TCP/IP 实在是被使用的已经太普遍了，以至于就算牺牲一点性能，局域网内通信也普遍使用 IP。而实际上，以太局域网内部通信的话，NetBEUI 协议的性能比 TCP/IP 协议要高许多。

其实整个 Internet，不仅仅都是以太网，以太网适合局域网联网通信，但是不适合广域网情况，广域网的联网协议，比如 PPP，HDLC，Frame Relay，x25，ATM 等，也像以太网一样各有各的寻址体系。在一个 Internet 上有这么多种不同地址的网络，它们之间若要相互融合、寻址，就必须在各种地址之间，相互翻译、转换、映射，数据包每经过一种网络，就转换一次，这样非常麻烦。IP 地址的出现使得所有联网的节点，不管用的是以太网，还是 Frame Relay，统统都分配一个 IP 地址给每个节点，对外最终以 IP 地址作为寻址地址，而将 IP 地址再映射到自己所在网络的所使用的地址上，比如 IP 映射到以太网的 MAC，或者 IP 映射到 Frame Relay 的 DLCI，映射到 ATM 的地址等。

用来进行地址映射的程序，称为 Address Resolution Protocol，即 ARP。很多人听到 ARP，就认为是以太网，其实这也是错误的，ARP 不仅仅代表以太网中的 IP 地址和 MAC 地址的映射，它代表任何种类地址之间的映射对应关系，从这一点来说，DNS 协议也应该归入广义的 ARP 协议中。

IP 统治了 OSI 的第三层，将原来占据第三层的凌乱地址种类统一了。映射到(承载于)以太网的 IP，称为 IPoE(IPoE 也就是“基于以太网的 TCP/IP”)；映射到帧中继的 IP，称为 IpoFR；映射到 ATM 的 IP，称为 IPoA 等。从此一种新的概念诞生了：PoP，即 Protocol over Protocol。

2. IP缺乏传输保障功能

IP 统一了天下还不够，因为 IP 最大的作用就是寻址和路由以及适配链路层 MTU，它并不提供其他功能，而作为一个健全的网络传输协议，必须具有传输保障功能。而以太网是一个面向无连接的网络，它不保障数据一定会传送的对方，是一个不负责任的网络，不管目的端口有没有收到，源端口只管向外发送。而 Frame Relay 协议，其前身 x25 协议，是一个有着很好传输保障功能的协议，在 TCP/IP 没有出现之前，x25 的传输保障机制做得非常到位，因为 x25 的设计初衷，就是为了运行到及其不稳定的链路上。而随着链路质量的不断提高，x25 的做法显得越来越因噎废食了，所以其改良版本 Frame Relay，就逐渐替代了 x25，FR 抛弃了 x25 中很多无谓的传输保障机制，而仅仅留下一些流控机制。相对于以太网的不负责任，FR 起码在链路层面，实现了比较好的流控措施。

但是，不管是以太网，还是 FR，都没有实现端到端的传输保障。端到端，是相对于“过路”来说的。过路是指在两个终端之间通信路径上的网络设备之间的路径。链路层的传输保障就是一种过路保障，因为链路层只保证相连的两个设备之间传送数据正常无误，但是不能保障通信最终端接收和发送的数据正常无误。因为在一个典型的包交换网络中，数据包一般都是一跳一跳的被传送的，每一跳两端的设备用链路层协议进行传输保障。

但是最终目的是要让通信的最终两端无误的收到数据，才能算作真正的传输保障，即端到端的保障。而 FR 协议所做的，只是在过路的时候保障链路正确传输。如果链路正确传输给了终端，而终端到最终上层的某个环节出错了，那么数据同样也是错误的，所以，要实现端到端的传输保障，一定要在最终传输终端上运行一个侦错和纠错逻辑，用来发现链路层所发现不了的错误。图 13.1 为端到端保障与过路保障的示意图。

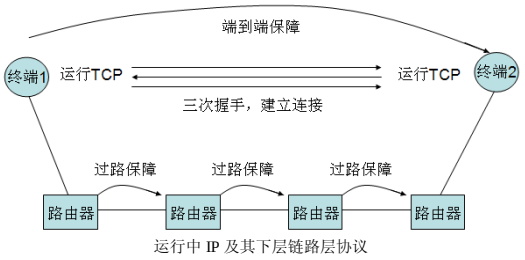

图13.1　过路保障与端到端的保障

3. TCP保驾护航

为了实现这个目的，TCP 出现了。TCP 作为一个程序运行在通信的两个终点，不管两点之间用什么样的链路连接，经过了多少网络设备，TCP 程序始终运行在通信终端上，监控终端最终发送和接收到的数据包的顺序、缓存区、校验等信息，检查是否出现丢包、阻塞等事件，一旦发现错误，立刻纠正重发数据包。

TCP 不是运行在通信路径上的，而是运行在通信终点的两端设备上。即使过路链路保障机制再健全，TCP 也是有必要的，因为数据包只有被终端正确接收到，才能算真正的传输保障。

所以，在 IP 之上，又凌驾了一层 TCP 逻辑，用来保障端到端的无误传输。而 FR 等链路层协议的保障机制，只能保障本段链路传输无误，不能保障端到端的正确收发，所以只能沦为数据链路层协议的角色了，用来承载 IP 和 TCP。

我们可以体会到，协议之间也是在互相利用、互相排挤、吞并，融合，以适应不同的应用环境，因为不可能为每一种应用环境都设计一种协议，协议之间互相利用、融合，才是最好的解决办法。

4. 最佳拍档——TCP/IP和以太网

现在可以回答上面没有找到答案的那个问题了，为什么以太网偏要和 TCP/IP 组合成一对呢？因为以太网使用得太广泛了，而 OSI 的第三层、第四层，也几乎被 IP、TCP 给统一了，所以以太网+TCP/IP，当然就成了一对好搭档了。

虽然一个协议可能实现 OSI 的所有 7 个层次，但是如果它要和其他协议合作，那么就要有个分工，而不能越权，比如 IPoA，ATM 只要传输 IP 包到目的就可以，而不管数据是

否出错、乱序等，虽然 ATM 可能有这个功能。以太网虽然自己可以寻址，但是它还是配合 IP，进行 IP 到 MAC 的映射，统一使用 IP 寻址，它默默无闻，所有光辉都被 TCP/IP 所披挂。

13.3 网络通信协议的 4 级结构

网络通信协议，一般可以分成 Payload 层、信息表示层、交互逻辑层和寻址层。其中最重要的是交互逻辑层，它是一个协议的灵魂。

1. Payload层

Payload 是协议所承载的与本协议逻辑无关的最终数据，是通信终端通过本协议最终需要传送给对方的数据。Payload 也就是协议所运输的货物。Payload 层中的数据，既可以是最终应用产生的数据，也可以是另一种协议的信息表示层+Payload 数据。如果 Payload 封装的是最终应用产生的数据，则表示这个协议是直接被上层应用程序来调用，从而完成程序之间的远程网络通信的。

如果 Payload 封装的是另一种协议的信息表示层+Payload 数据，那么就证明这个协议此时正在承载那个协议。比如协议 A 封装了协议 B 的信息表示层+Payload，则就可以说协议 A 封装了协议 B，或者协议 A 承载了协议 B，或者说协议 B is over 协议 A(BoA)。我们后面会描述一种协议被 Map(映射)到另一种协议，而不是被封装，这种融合方式称为 AmB，是彻底的协议转换，而不是仅仅做隧道封装。

2. 信息表示层

信息表示层就是附加在 Payload 数据之外的一段数据，也称作协议开销，因为这段数据和最终应用程序无关，是运行在通信双方的通信协议用来交互各自的状态，从而使双方作出正确动作的一段重要数据。这段数据可以想象成提货单或者信封。信封封装了信纸，信封上的地址、姓名等信息，就是信息表示层，它可以让对方检测到当前通信所处的状态。

3. 交互逻辑层

这一层其实就是运行在通信双方协议系统上的动作程序代码逻辑，它根据对方传送过来的信息表示层数据来作出相应的动作逻辑，再生成自己的信息表示层发送给对方，然后对方再做相同的处理判断动作，就这样完成通信双方之间的正确动作。交互逻辑层其实就是协议的设计思想。交互逻辑层对于每种协议都不相同，但是很多都类似，可以说网络通信协议基本思想是类似的，因为它们所实现的目的都是一样的，就是将数据通过网络传输到目的地。

正因为如此，各种协议的交互逻辑层才可以相互融会贯通，将一种协议的逻辑，映射翻译到另一种协议的逻辑，从而将各种协议的优点结合起来，完成目标。协议逻辑层一般都是运行在通信双方两端的，但是像 IP 路由协议等，通信双方经过的路径上的所有设备，也都需要运行，因为 IP 包是一跳一跳被接收并且转发的。

4. 寻址层

它是帮助协议来找到需要通信的目标的一套编址和寻址机制。比如 IP 地址、MAC 地址、

DLCI 地址、电话号码等。如果是点对点传输协议，则可以忽略此层，因为不需要寻址。而且不同协议之间的寻址层，可以互相映射翻译。

以上的这四层，是任何一个网络通信协议所必须具备的，不管多么简单或者多么复杂的协议。

5. 通信协议的相似性

相似性是通信协议之间相互融合的一个条件。而协议之间相互融合的另一个促成因素，就是协议使用广泛程度不同，有时如果要完成一个目标，不得不借用某种协议。

就像 TCP/IP 协议，TCP/IP 协议占领了全球 Internet 的领地。如果有一种协议想跨越地域或国家来进行通信，但是自己又无能为力，因为它没有专门为它准备的物理线路，其次它的设计，也就不适合大范围、长距离的广域网环境，那么它只能来租用 TCP/IP 协议，将自己封装到 IP 包中传送。能适合 Internet 规模的网络通信协议，唯 TCP/IP 莫属！而其他协议想要完成 Internet 范围的通信，就不得不借助 TCP/IP，搭 TCP/IP 的车，让 TCP/IP 来承载它们。它们是怎么搭上 TCP/IP 的快车呢？

我们不妨类比一下。在整理本章的时候，恰逢大连刚刚开通了一艘新的火车箱滚装船。我想用这个例子来比喻协议融合，再适合不过了。从山东烟台到大连，最近的路径就是走渤海湾水路，如果搭乘陆路火车，则需要绕一大圈，所以很多货运汽车，甚至火车，都选择乘船到大连，下船后，车厢用火车头拉走，这样，在增加很少成本的条件下，节约了大量时间。协议融合同样遵循这个原则，只要能使总体拥有成本降低，性价比提高，任何协议都可以融合。

13.4 协议融合的 3 种方式

协议和协议之间的相互作用，有三种基本的思想。

- 第一种是调用(Use)，也就是一种协议完全利用另一种协议。
- 第二种是隧道封装(Tunnel)，一种协议将另一种协议的完整数据包全打包遂道封装到新协议数据包中。
- 第三种是映射(Map)，也就是一种协议对另一种协议进行映射翻译，只将原来协议的 Payload 层数据提取出来，重新打包到新协议数据包中。

1. 调用关系

所谓调用，也就是一种协议自身没有某些功能，需要使用另一种协议提供的功能。比如 TCP 调用 IP，因为 TCP 没有寻址功能，所以它利用 IP 来寻址。而 IP 又可以调用以太网，因为 IP 只是一个寻址功能，它没有链路传输的功能，所以它利用以太网提供的链路传输(交换机、Hub 等)。IP 调用 PPP 来传输等，也就是上层协议为了达到通信目的，使用另一种协议为其服务。这种关系严格来说，不算是融合。

2. 隧道关系

隧道封装，顾名思义，就是将一种协议的完整数据包(包括 Payload 和协议开销)作为另一种协议的 Payload 来进行封装，打包传输到目的地，然后解开外层协议的封装信息，露出

内部被封装承载的协议完整数据包，再提交给内层协议处理逻辑模块进行处理。也就是说，进行协议转换的设备根本就不需要去理解内层协议到底是什么东西，到底想要干什么，只要将数据包统统打包发出去。Tunnel 的出现，往往是由于被 Tunnel 的协议虽然和外层协议都在某一方面具有相似甚至相同的功能，但是在某些特定的条件下，被 Tunnel 协议不比外层协议表现得优秀，不适合某种特定的环境，而这种环境，恰恰被外层协议所适合。这就像用船来装火车箱一样。Tunnel 的另一个目的是伪装内层协议。

3. 映射关系

Map 是比 Tunnel 更复杂、更彻底的协议融合方式。所谓 Map，也就是映射，就是将内层协议的部分或者全部逻辑，映射翻译到外层协议对应的功能相似的逻辑上，而不是仅仅做简单的封装。Map 相对于 Tunnel，是内外层协议的一种最彻底的融合，它将两种协议的优点，融合得天衣无缝。内层协议的 Payload 层在 Map 动作中是不会改动的，因为 Payload 层的数据只有两端通信的应用程序才能理解。

13.5 Tunnel和Map融合方式各论

例如火车、汽车是两种运输工具，它们看似有太多的不同，但是它们的目的都是相同的，都是为了将货物运送的目的地。而火车需要跑在铁道上，但汽车需要跑在公路上(物理层不同，链路层不同)；火车因为铁轨很平滑，需要用钢铁轮子，而汽车因为公路很颠簸，需要用充气轮胎；火车不需要红绿灯来制约，而汽车跑在公路上，会有很多红绿灯来制约它；火车由于跑在专用的铁轨上，所以它能达到很高的时速，而汽车由于跑在共享的公路上，它能，但是不敢达到太高的时速，火车只能按照它的轨道来运行，而汽车几乎随处可去……

以上列举出了火车和汽车的种种特点，相应的飞机、轮船、火箭等都可以拿来对比，这些特点就像各种通信协议自身的特点一样。同样都是运输货物，但是它们都适应了不同的需要。只不过网络通信协议运输的不是货物，而是一串 0 和 1，是高低变化的电平，是数据，是信息。不同的通信协议同样也是为了满足不同的情况、不同的需求。TCP/IP 协议满足了 Internet 范围的网络通信；FC 协议满足了后端存储的专用高速公路这个环境，二者都各自占有自己的领地，谁也取代不了谁。就像铁路不可能为了和民航竞争，而把轨道往天上修，航空公司也不可能为了和陆运公司竞争，而让飞机跑在公路上。

TCP/IP 适合整个 Internet 范围的通信，而 SCSI 协议不适合，所以如果 SCSI 协议需要跨越大范围通信，就要将其承载到 TCP/IP 上，也就形成了 ISCSI 协议，然而 TCP/IP 根本就不关心什么是 SCSI，更不知道 SCSI 是怎样一种作用逻辑，它只是负责封装并传输。同样，因为以太网是个面向无连接的网络，没有握手过程，也没有必要有终端认证机制、没有 NCP 机制(PPP 协议中用来协商上层协议参数的机制)，而 PPP 却有这些机制，它非常适合 ISP 用来对接入终端进行认证和管理，但是 PPP 使用程度远远不如以太网广泛，怎么办？融合吧！于是形成了 PPPoE 协议。

13.5.1 Tunnel方式

ISCSI 和 PPPoE 这两个协议，是典型的 Tunnel 模式。前面已经给 Tunnel 下过定义了。首先一种 PoP 的模式被定义为 Tunnel 的前提，就是这两种协议对某一特定的功能，均有自己的实现。如果一种协议在某方面的功能，另一种协议没有实现，那么另一种协议就是“调用”那种协议，而不是被 Tunnel 到那种协议。比如，IPoE 就是典型的调用，而不是 Tunnel 或者 Map，因为 IP 没有链路层功能。

IP 与 Ethernet 之间的编址逻辑是映射关系而不是使用关系，即 IP 地址与 MAC 地址的相互映射。

用 ISCSI 来分析，TCP/IP 可以实现寻址和传输保障，SCSI 协议也可以实现寻址和传输保障，所以它们具备了这个前提；同样 PPPoE 也是一种 Tunnel 方式的融合协议，因为 PPP 和 Ethernet 都是链路层协议。

1. VPN的引入

Tunnel 的另一个作用，就是伪装。有时候虽然两种协议实现的功能、适用环境都相同，但还是将其中一种 Tunnel 到另一种之上，这是为什么呢？有些情况确实需要这种实现方式。比如 IP 协议中的 GRE，通用路由封装，就是这样一种协议。它将 IP 协议承载到 IP 协议本身之上，自己承载自己，再封装一层，这样就可以使得一些不能在公网路由的 IP 包，封装到可以在公网路由的 IP 包之中，到达目的后再解开封装，露出原来的 IP 包，再次路由。这就是伪装。

利用这种思想，人们设计出了 VPN，即 Virtual Private Network，用来将相隔千里的两个内部网络，通过 Internet 连接起来，两端就像在一个内网一样，经过 Internet 的时候，使用公网地址封装内网的 IP 包。这是最简单的 VPN。在这基础上，又可以对 IP 包进行加密、反修改等，形成 IPSEC 体系，将其和原始的 VPN 结合，形成了带加密和反修改的 IPSEC VPN，真正使得这种 PoP，穿越外层协议的时候，能够保障数据安全。

2. 例解Tunnel

下面再举个例子来说明，到底什么是 Tunnel。

邮政系统，目前已经是举步维艰。21 世纪之前，网络还没有很普及，除了电话、电报，写信似乎是大家长距离通信的唯一选择。寄信人将自己的信件(数据，Payload)装入信封(协议信息表示层数据段)，填好收信人地址、邮编、名称(通信协议的信息表示层、寻址层)等，交给邮局(网络交换路由设备)，由邮局进行层层路由转发，最终到达目的地。

IP 网络和邮政系统极其相似。而为什么邮政系统目前已经陷入了困境呢？原因就是竞争。

进入 21 世纪之后，物流业快速兴起，它们借助陆路、水路、航路、铁路等 “链路层”，加上自己的一套流程体系(协议交互逻辑)，充分利用这些资源达到物流目的。以前只有邮政一种方式，而现在物流公司多如牛毛，每个公司都有自己不同的物流体系，但是基本思想

大同小异，都是要将用户的货物运送到目的地。

21 世纪，虽然网络已经很发达，但是网络只能走信息流，走不了实物流。所以物流公司还是能占据一定市场。

我们来看看 21 世纪，用户是怎么来寄出一封信件或者包裹的。同样寄出一封信，如果还是用古老的协议，比如信封 + 80 分邮票的形式，还是可以的，大街上现在还有邮筒。但是很多快递公司，也提供信件包裹服务，只不过他们用的信封，比普通信封大，结实，而且他们信封上的标签，所包含的信息更加具体和丰富，比如增加了收件人电话、发件日期、受理人签字、委托人签字等。邮政信封具有的，快递信封都具有。

这样就可以看出这两种协议的不同之处了。用户可以把信件封装到邮政普通信封直接发送，也可以封装到快递公司信封中发送，也就是选用其中一种协议。

那么如果用户先把信件(最终数据)封装到普通信封中，填好信封头信息(协议信息表示层和寻址层)，然后将封装好的普通信封，再封装到快递公司的信封中，并再次填一份快递公司的信封头信息。快递公司按照这些信息，将信件送到目的地，目的收到之后，解开外层信封，然后解读内层信封的信息头，再次转发，或者直接打开。刚才描述的这种情况，就是一个典型的协议 Tunnel 方式的相互作用，把邮政协议，Tunnel 到快递公司的协议，这种 Tunnel 的目的，就是为了获得快速、优质的服务，因为普通邮政协议提供不了快速高效的服务。

我们再来看这种情况，比如快递公司 A，在北京没有自己的送货机构，但是青岛有人需要向北京送货，怎么办？

此时当然要考虑借助在北京有送货机构的快递公司 B，让他们代送，将信件封装到快递公司 A 的信封，然后再将 A 的信封装入快递公司 B 的信封，让快递公司 B 做转发，到目的地之后，B 的送货员剥开外层信封，最终用户会收到一个快递公司 A 的信封，客户就认为是快递公司 A 全程护送过来的，其实不是。这样就很好地伪装了信件。这是 Tunnel 的另一个目的。

13.5.2 Map 方式

说完了 Tunnel，我们再来说说 Map。Map 就是将一种协议的逻辑，翻译映射成另一种协议的逻辑，Payload 数据完全不变，达到两种协议部分或者完全融合。

还是快递公司的例子。两个快递公司(两种协议)，快递公司 A 在青岛没有自己的送货机构，但是 B 有。所以 A 和 B 达成协议，A 将青岛地区的送货外包给 B，凡是 A 公司在青岛的业务，都由 B 来运送，但是表面上必须保持 A 的原样，这种方式目前实际已经广泛使用。起初的做法是：先将客户信件装入 A 信封，然后再封装一层 B 信封，带着 A 信封来转发，也就是 Tunnel。后来，B 公司嫌这种方法浪费了成本，因为额外携带了一个 A 信封，这增

加了信件的重量和信封成本。所以 B 公司琢磨出一套方法。

1】 先让 B 公司的取件人了解寄件人所要提供的信息，此时取件人担当 A 公司的角色，用户认为取件人是 A 公司的，用户按照 A 公司的协议，将信封头信息告诉取件人；

2】 然后取件人此时并没有将信件装入 A 公司信封，而是直接装入了 B 公司信封，但是在填写 B 公司信封头的时候，取件人将用户提供的针对 A 公司特有的信封头信息，转换翻译成 B 公司特有的信封头信息；

3】 经过 B 公司转发后，到达目的地之后，送货员再次将 B 公司的信封头信息，转换翻译成 A 公司所特有的信封头信息。

这样，两端的用户，同样也丝毫感觉不出中间环节其实是 B 公司完成的。但是这种方式相对于 Tunnel 方式的确节约了 B 公司的成本，使得开销变小了，提高了转发效率。这种方式的协议之间的相互作用，就是 Map。

1. IP和以太网之间的寻址关系Map

最简单的 Map 就是 IP 和以太网之间的寻址关系 Map。IP 地址必须映射到 MAC 地址，才能享受以太网的服务。正如 IP 和以太网之间的 Use+Map 关系一样，实际上，各种协议之间的相互作用，不可能只是其中一种作用方式，寻址体系之间一定需要 Map(同种协议自身 Tunnel 的情况除外)，交互逻辑层可以 Tunnel，也可以 Map，Payload 一定需要 Tunnel。所以针对协议不同的层次，都有相对应的相互作用方式。

2. 协议交互逻辑的Map

协议交互逻辑的 Map，比寻址层的 Map 要复杂的多。寻址层的 Map 只要维护一张映射表就可以，交互逻辑的 Map 则需要维护一个代码转换逻辑模块。

两种协议的状态机的互相融合作用是很复杂的。比如 TCP 的流控机制和 FC 协议的流控机制之间的 Map，TCP 是靠窗口机制实现端到端的流控，FC 靠 Buffer to Buffer(过路流控)和 End to End(端到端流控)两种机制实现流控。如果把 FC 协议承载到 TCP/IP 协议之上，那么就会出现 Tunnel 模式和 Map 模式，当然 Tunnel 中也可能需要 Map，Map 中也同样需要一定的 Tunnel 成分。

我们不妨称作：以 Tunnel 为主的模式和以 Map 为主的模式。

如果是 Tunnel 为主的模式，那么 TCP/IP 根本不管 FC 协议的交互逻辑是怎么样的，TCP 仅仅把 FC 当成 Payload 来封装并传送。

而 Map 模式中，进行 Map 操作的设备或者软件，就需要既了解 TCP/IP 协议的交互逻辑，又了解 FC 协议的交互逻辑，因为只有了解了双方的逻辑，才有可能进行 Map。比如，FC 协议发出了一个信号，说本方缓存将满，请降低发送速度。Map 设备收到这个信号之后，就会 Map 成 TCP/IP 可识别的信号，即本方处理受阻，窗口减小至某某数值，这就是 FC 协议到 TCP/IP 协议关于流控机制 Map 的一个方法。

如果在 Tunnel 模式中，FC 协议发出的这个流控信号，则会被 TCP/IP 给 Tunnel 传送到对方，然后再由对方的 FC 协议模块来根据这个信号来判断流控机制应该做出的动作，动态调整发送速率。

这个信号是直接原封不动的被传送到 FC 协议的对端处理机上处理，而不是像 Map 模式中在本地就终结了 FC 逻辑。Tunnel 模式中，TCP/IP 不参与任何 FC 协议内部的逻辑。

除了 FC 流控逻辑的映射，其他 Flogin 登录机制、连接机制等映射，也都有自己的实现。比如，FC 发起一个 Plogin 过程，那么 Map 设备可以 Map 到 TCP/IP 的一个握手过程等。

Tunnel 和 Map 这两种模式，在第 8 章还有一个将 FC al 的环接入 FC Fabric 中的例子。

13.6 FC与IP协议之间的融合

哗啦……，早晨的微风把 IP 从美梦中吹醒。原来 IP 做了一场美梦。根据梦中的指示，IP 鬼使神差的将 FC 协议映射到了 IP 上。并做了两种模式，一种是以 Tunnel 为主的模式，称作 FCIP；另一种是以 Map 为主的模式，称作 IFCP。

在 FCIP 模式中，通信的双方各增加一个 FCIP 网关，任何 FC 协议的逻辑，哪怕是一个小小的 Ack 帧，都需要封装到 TCP/IP 协议中传输。两端的 FC 协议处理机不会感知到中间 TCP/IP 的存在，它们认为对方就是一个纯粹的 FC 设备。

在 IFCP 模式中，通信的双方各增加一个 IFCP 网关，作为协议转换设备使用。IFCP GW 将 FC 协议终止在本地，提取 Payload 数据，对外以 TCP/IP 设备的形式出现并传输数据，到达对方之后，对方的 IFCP GW 再从 IP 包中提取出 Payload，然后将其封装到 FC 帧中，对其内部以 FC 设备的形式出现。通信双方中间的 TCP/IP 协议，将大部分或者全部 FC 的逻辑都映射成 TCP/IP 的逻辑。

比如每当一个 FC 设备需要和远端的 FC 设备通信，发起 Plogin，那么 IFCP GW 就向对方建立一条 TCP 连接，用多条 TCP 连接和不同的 IP 地址来区分不同的 FC 设备。此外，还需要保存一个 TCP 端口或者 IP 地址对 FC 设备 24bit 的 Fabric 地址的映射表。如果两端的 FC 设备的 ID 有冲突，这个映射表还需要考虑 NAT，将地址翻译成其他 ID。相对于 IFCP，FCIP 协议则不能识别 FC 的逻辑，因为它只是 Tunnel，如果两端 Fabric 中有 ID 冲突的，那么也只能冲突着了。

至此，FC 协议终于可以享受 TCP/IP 带来的扩展性了，FC 搭上了 TCP/IP 的车，远隔千里都可以跑上 FC 协议了。IP 大获成功！IP 和 FC 从此握手言和！

伟大的 SCSI 协议

可以说整个网络存储系统，都起源于一个协议体系，这个协议体系就是 SCSI 协议。网络存储的任何内容，最终都是为了将这个协议体系发扬光大。人们将这个协议强行划分解体成了多个层次，然后把它的最上面的几层，与另一个协议体系——Fabre Channel 协议的下几层进行融合，形成了 FCP 协议，这种协议目前运行在各个厂家的高端磁盘阵列上。还有曾经一度时间，以太网甚至也看好了 SCSI 协议，想与其融合成所谓的“ESCSI”协议，但结果没有成功。以太网失败之后，它的好兄弟 IP 接着上，最终成功的与 SCSI 协议进行

了融合，生成了 ISCSI 协议，目前也被广泛应用于一些低端盘阵。

为何不是 IATA 或者 FATA 呢？原因就是因为 SCSI 协议体系本身就比 ATA 协议体系高效并且功能强大，此外，SCSI 的硬盘性能也普遍比 ATA 硬盘转速快性能高，用于服务器系统，所以 SCSI 当然是首选了。另外，一个巴掌拍不响，SCSI 协议本身就想把自己给“嫁”出去，因为它很早就已经迫不及待地将自己分成了很多层次，来吸引其他协议。

协议融合的结果，就形成了目前形形色色的网络存储世界，各种融合协议，各种产品，各种解决方案，好不热闹！而原本的 SCSI 协议，除了一些磁带机以及主机本机硬盘外，已经不再使用。SCSI 融合入了各种协议中，它无处不在，虽然它的躯体已经是七零八落，但是它的精深思想，以及为技术而献身的精神，将在形形色色的技术中永放光芒！！

13.7 无处不在的协议融合

之所以提出“协议融合”这个名词，而不是“协议映射”或者“协议隧道”，是因为“融合”这个词更加通俗易懂；另外，也更加具有生物学色彩。计算机就是人类所创造的另一种形式的“生物”，人类就是计算机的上帝。

1. 协议融合和基因融合

分子生物学家们将不同功能的基因段整合到一起，然后用核糖体蛋白机器读取其代码，表达成肽链，然后折叠成三维结构的新功能蛋白质分子，比如抗冻小麦、发光的白鼠等。这就是基因融合。这个过程与协议融合类似。

协议融合是无处不在的，正如不同快递公司之间的合作一样。甚至连劳动合同方面都出现了融合，劳务派遣公司与劳动者签订合同，然后将劳动者输送到用工单位工作，用工单位不必维护人事系统，将人事系统外包给劳务派遣公司。

2. 航空公司的协议融合

目前，国际上的大多数的大型航空公司都利用 IBM 或者 Unisys 的大型机系统作为订票和离港系统的处理机。世界各地的售票和离港终端都通过某种网络系统与大型机连接并且通信。航空业的大机与终端通信协议也经历了纯种和融合阶段。

IBM 利用 ALC 协议与其终端通信，Unisys 主机则通过 UTS 协议与其终端通信。但是随着 IP 网络的成本不断降低，质量不断提高，UTS 和 ALC 这两种古老的纯种协议，不得不考虑将自己嫁给 IP 网络，从而出现了 MATIP 协议，也就是将这些协议承载于 IP 之上。Cisco 公司也为航空业专门开发了这种融合协议，称为 ALPS 协议。然而 ALPS 最终没有成为 RFC 标准，而 MATIP 协议，却最终登上了 RFC 宝座。MATIP 协议的文本可以查看 RFC2153。

13.8 交 叉 融 合

在本书写作之时，FCoE 这个由 FCP 与以太网结姻所产生的融合协议，正在被一些厂商炒作得沸沸扬扬。FC 协议与 SCSI 协议融合之后形成 FCP 协议，而 FCP 协议又与 Ethernet 融合形成 FCoE 协议。

如图 13.2 所示是协议融合树。

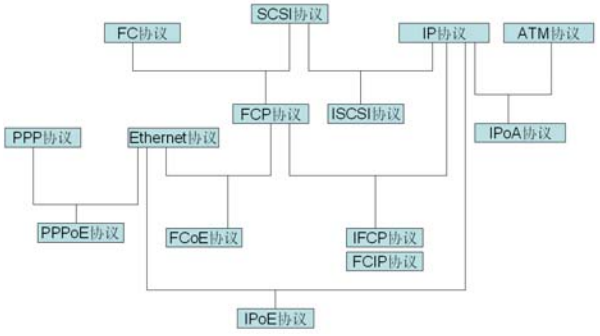

图13.2 协议融合树

从图 13.2 可以看到，FCP 协议与 IP 协议融合的后代是双胞胎。各种协议之间相互融合，甚至产生了交叉，但是一切融合都是为了更好地适合市场需求，因为在资本市场经济时代，只要能满足需求，就能获得利润。“不求最好，但求最适。”

另外，由于 IP 网络的大肆普及，众多的协议动辄就要非 IP 不嫁，而 IP 和以太网绝对是铁哥们儿，所以以太网也借 IP 的光，就凭着自己仅仅 1Gb/s 的带宽到处招摇撞骗，这不，成功地把比它多 3Gb/s 带宽的 FCP 协议给忽悠过来了。不过以太网也在潜心修炼，等练成出关之后，其 10Gb/s 的速率，将会让人望而生畏，但愿那时候以太网统一天下！

13.9 IFCP和FCIP的具体实现

上回说到 IP 根据网络通信协议之间的相互作用，成功地将 FC 协议和 TCP/IP 协议进行了融合，生成一种 FCIP 的 Tunnel 协议和一种 IFCP 的 Map 协议。

蓝图有了，那么具体怎么来将其实现呢？我们知道，不管是 FCIP 的简单 Tunnel 模式，还是 IFCP 的复杂 Map 模式，进行这种 PoP 操作的角色，一定是一端面对 FC 协议的网络，另一端面对 TCP/IP 协议的网络。

1. 协议转换器

同时面对多种协议，并在多种协议之间实现相互融合、相互转换的设备，就称作协议转换器。如果这个转换器只是起到一个桥联的作用，只在一条链路上串联，那么就称其为协议桥接器。如果这种转换器，不但要实现单条链路上的协议转换工作，而且还需要实现一些转发动作，即在多条链路、多个网络之间互相转发数据，则可以称其为协议路由转换器。如果某种协议路由器可以实现多于 2 种协议的网络互联，则称其为多协议路由转换器，因为它能在多种协议之间互相转换并做路由转发。

SAN 要想获得扩展性，即要想将相隔两地很远的两个 SAN 网络通过 IFCP 或者 FCIP 连接起来，就必须在双方的 SAN 系统前端各增加一个协议转换设备，这个设备后端连接各自的 SAN，前端连接 IP 网络，在广域网络上运行 FCIP 或者 IFCP 协议通信，达到协议转换的

目的。

两个独立的系统连接起来，就涉及了两种情况。

- 第一种：两个系统在连接之后，在逻辑上还是独立的，即一个系统不影响另一个系统，但是它们之间可以通过协议转换设备来通信。
- 第二种：两个系统融合成一个大的系统，逻辑上是一体的，只不过相处两地，之间用协议转换设备连接。就像以太网络一样，如果用光缆将两地的两个局域网直接连接起来，两地的系统同在一个广播域中，这样就相当于把两个系统融合起来了。

但是如果两地各自接一个 IP 路由器，广域网链路上承载的是基于广域链路协议之上的 IP 包，那么两地的局域网就没有被融合，可以只是相互通信而已。

有的时候，两地的系统必须融合，而有的时候，不需要融合。是否融合，需要看最终的需求。所以协议转换设备也必须能够处理这两种情况。对于需要融合的情况，协议转换设备不需对两端的 SAN 逻辑做任何附加处理，而只需要将两端的逻辑 Tunnel 或者 Map 到广域网协议上就可以了。而对于不需要融合的情况，协议转换设备就需要对两端系统的逻辑做一系列的处理、屏蔽、虚拟和欺骗了。

2. TCP/IP和以太网络实例解析

我们不妨拿 TCP/IP 和以太网络来做一个例子。

假如一个公司，在 a 地和 b 地，分别有一个办事处，每个办事处有一台以太网交换机，上面各连接了几台终端。现在为了业务资源共享，公司决定将两地的网络融合起来。公司向 ISP 申请了一条 2Mb/s 的 E1 专线(当然也可以申请 Internet 线路，两端都接入 Internet，然后做 L2VPN 或者 L3VPN)。

公司有两种选择方案。

- 一种是直接用这条专线把两地的交换机连接起来，在这条线路上直接承载以太网帧。
- 另一种选择就是两端各加一个路由器，隔离两边的局域网，但是保持它们之间的通信。

这个公司最终选择了后一种方案，原因就是为了保持双方的独立性，同时保证性能。因为毕竟是两个办事处，如果彻底进行融合，不但不安全，也不利于扩展，而且容易造成广域网流量太大，因为彻底融合之后，以太网广播就要跨广域网来互相传递，这无疑是浪费资源的。在隔离的基础上，同样能够保持双方无障碍的相互通信，只是不能像在一个局域网内那样直接利用 MAC 来点对点通信。如果 a 地某个节点需要和 b 地某个节点通信，a 地的这个节点需要先把数据发给 a 地的路由器，也就是网关设备，然后让网关来转发给 b 地。虽然多增加了一层操作，但是这样做的可扩展性、可管理性都增强了。在路由器上可以做访问控制、地址转换、QOS、策略路由等基于 IP 甚至 TCP 层次的个性化动作。如果是直接局域网融合，则这些特性都不能实现。

3. SAN系统实例解析

再来看 SAN 的情况。还是这个公司，a 地和 b 地各有一个 SAN 系统。为了实现存储资源直接共享，公司决定将这两个 SAN 联通起来。同样也存在两种情况，即彻底融合或者相对独立的连通。

如果是彻底融合的话，那么广域网链路就完全相当于一条 ISL 链路，只不过通信协议可能是 FCIP 或者 IFCP 协议。

对于 FCIP，任何 FC 帧都将被透明的传递。对于 IFCP，一部分 FC 帧会被屏蔽或者 MAP。但是这些被屏蔽或者 MAP 的帧，都是和底层通信有关的，而上层逻辑性质的帧，IFCP 也需要透传到对端。

这些业务逻辑性质的帧，比如 RSCN 帧，这种用来传递 Fabric 网络中的重要变化信息给已经注册了这项服务的节点；再比如 Plogin，Process Login 等这些都是业务逻辑性质的，和底层通信无关。

彻底融合之后，两个 SAN 系统就融合为了一个系统，那么这个系统就会有一个主交换机，主交换机为系统中其他交换机分配域 ID，并且两个交换机之间需要运行 FSPF 路由协议，不停的发送一些路由控制帧，再加上主交换机选举时产生的帧，主交换机失败时，整个 Fabric 的重建过程中每个交换机发出的各种帧都需要经过广域网链路进行传送。

不但这些帧要占用广域网带宽，而且一旦主交换机发生故障，那么对方的 SAN 系统会进行 Rebuild，包括重新选举主交换机、重新建立路由表等，这个过程中，IO 就会暂时中断。

由于广域网链路速度相对慢，稳定性相对差，所以一旦这条链路发生不稳定的振荡，那么就会造成主交换机重新选举。如果链路频繁闪断的话，那么两端的 SAN 系统根本无法正常工作了。

所以说，两地 SAN 系统彻底融合的话，一旦某地的系统故障，就会影响到另一个系统的正常运行，而且要占用额外多的宝贵的广域网资源。由于访问存储资源对性能和延迟要求较高，所以彻底融合两个 SAN，最好只在局域网内进行，交换机间的链路最好是裸光缆或者高速链路，否则最好采用另外一种融合方式，即逻辑独立、全局连通的融合方式。

13.10 局部隔离/全局共享的存储网络

将 SAN 系统彻底融合，扩展性差、管理性差，而且耗费广域网链路资源。所以这个公司同样也选择了相对独立的连通方式。下面来看一下，相对独立的融合，到底是个什么概念，它的作用机制是怎样的。

“a 地的 SAN 交换机(E 端口)－a 地协议转换器－广域网链路－b 地协议转换器－(E 端口)b 地 SAN 交换机”这种拓扑不管是彻底融合，还是独立融合都一样，只不过协议转换器在两种方式下所作的工作不一样。彻底融合方案中，协议转换只 Tunnel 或者 Map 通信底层的协议逻辑，而不管上层业务逻辑，也就是只要从 E 端口收到了帧，协转就将其 Tunnel 或者 Map 到 IP 协议中发送给对端。而相对独立的融合，不但要 Tunnel 或者 Map 底层协议逻辑帧，它还要理解 FC 的上层逻辑，做到“报喜不报忧”。

独立融合/全局共享

所谓独立融合，就是说两端的 SAN 系统都可以独立运作，而不依靠另一方，或者受另一方的影响。这样就不能像彻底融合那样一端为主交换机，一端为非主交换机，而要让两端独立起来。由于两端的 Fabric 中都各自只有一台 SAN 交换机，所以两端的 SAN 交换机都是主交换机，各自为政。

既然这样，怎么能和对方的 SAN 进行通信呢？协议路由器自有其招数。协议路由器与 SAN 交换机之间通过 E 端口连接，它欺骗两端 SAN 交换机，让交换机认为它正在连接着另一台交换机，而这个由协议转换器虚拟出来的交换机级别比它低，所以它自己认为自己就是主交换机。虚拟交换机和 SAN 交换机之间运行 FSPF 路由协议，所以这个虚拟交换机就获得了 SAN 交换机下面所有连接的终端节点信息。

获得这些信息之后，a 地的协转通过广域网链路将这些信息通告给 b 地的协议转换器。b 地的协议转换器同样和 b 地的 SAN 交换机之间运行着 FSPF 路由协议，同样也欺骗了 b 地交换机。b 地协议转换器收到了 a 地协议转换器发来的关于 a 地 SAN 交换机上所连接的所有节点信息之后，就利用和 b 的 SAN 交换机之间的 FSPF 路由协议，将这些节点信息通告给 b 地 SAN 交换机，所以 b 交换机就有了 a 交换机上节点的信息，同样 a 交换机也会拥有 b 交换机上节点的信息，这样，a 和 b 交换机之间就可以通信了，其实它们都不知道中途有两个中介在骗它们。

如果其中一个 SAN 系统发生故障，那么这个系统中的协转设备，会将这个重大消息屏蔽，不告诉对端的 SAN 系统。因为一旦被对方系统得知，便会发生 Fabric 的重建过程，影响本端 SAN 系统的 IO。有了 SAN 路由器，远端 SAN 访问的超时，并不会影响本地 SAN 的访问。此即所谓的“报喜不报忧”。同样，一个 SAN 系统中的诸如 RSCN 等广播类的帧，也会被协转设备根据策略而终结在本地，不会跨越广域网链路通告给对方。协转设备还应该具有访问控制功能。

这种方案被称作“SAN 路由”，因为它具有像 IP 路由类似的功能和架构。

13.11 多协议混杂的存储网络

如图 13.3 所示，其中的中枢引擎是两个互相连接的多协议路由器。这个多协议处理机，就像一台计算机的 CPU，Fabric 和以太网络就像计算机的 IO 总线，磁盘便是计算机的外设和输入设备，各种存储控制器便可以理解为计算机上的各种 IO 控制器，前端的 Fabric 和以太网便是前端的 IO 总线，主机服务器则是输出设备。即磁盘上的数据，经过输入总线输入 CPU 进行运算，然后通过输出总线，输出给主机服务器。这又是一个轮回，不折不扣的轮回，循环嵌套，永无止境。

图 13.3 所示的拓扑，可以说是一个大的统一的拓扑。存储网络不外乎就是图 13.3 中列出的元素。磁盘经过一层层的 IN 和 OUT，一层层的虚拟化或者桥接透传，最终被主机看作是一个 LUN 或者目录。不妨将其抽象，隐去复杂的部分，就形成了图 13.4 的拓扑。

再抽象一下，如图 13.5 所示。

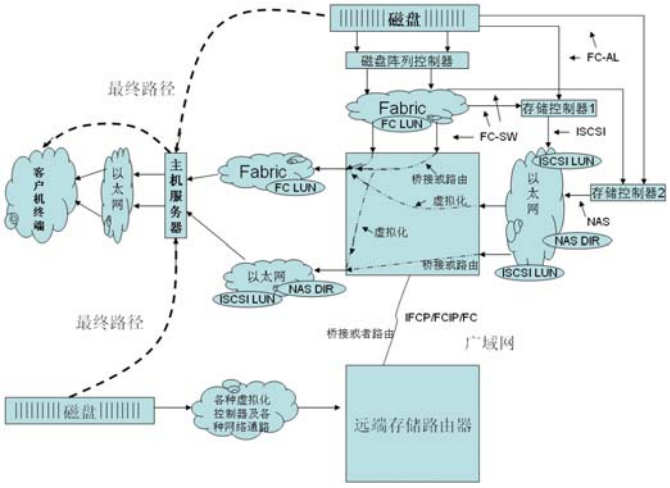

图13.3 多协议混杂的存储网络

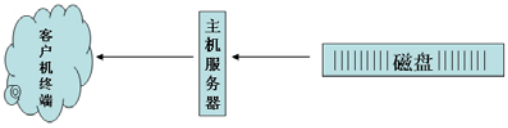

图13.4 一次抽象后的系统架构

图13.5 本质模型