TCP/IP 网络协议栈的构件化研究

Research on the Component of TCP/IP Network Protocol Stack

蔡秀珍 （湄洲湾职业技术学院 福建莆田 351254）

摘 要： 软件构件化是21世纪软件工业发展的大势趋。把网络协议和构件思想结合在一起实现是目前网络协议体系结构发展的方向，特别是在基于通信设备的软件开发方面。本文首先介绍了构件技术及其优点，然后介绍了基于构件技术的嵌入式 TCP/IP 网络协议栈，最后分析了基于构件的 TCP 协议模型。此模型在通讯设备上可使用，具有实际价值。

关键词： TCP/IP 协议栈 构件技术 网络 嵌入式系统

1 引言

现在，越来越多的手机可以直接上网浏览网页、收发电子邮件、查询信息、还能进行各种商贸交易。这些复杂操作，通过嵌入式 Internet 这一概念和技术得以实现。嵌入式通讯设备融合了嵌入式通讯技术、嵌入式操作系统技术、嵌入式数据管理系统技术、无线网络技术，成为信息技术时代的典型产品。其网络应用在软件上很大程度依赖于 TCP/IP 网络协议栈的设计与实现。通常，运用在手机上的 TCP/IP 协议族，采用了层状结构来构造 Internet 通信系统。这种通信系统是基于单块式（monolithic）体系结构的，即纳入的协议是以单块方式设计并加以实现的。这使得通信设备出现稳定性差、通信方式特殊、通信速度慢和通信过程容易受其它通信的干扰等问题。

随着通讯设备的广泛普及和通讯设备对网络接入的需求，一些更加简单灵活的体系结构随之产生。将构件技术引入网络协议栈的开发，实现构件化的网络协议栈，便能很好地解决了基于单块式体系结构的通讯系统所出现各种问题。

2 基于构件的网络协议栈

2.1 构件技术

软件构件技术是建立在面向对象技术之上的，它提供了比面向对象技术更为高级的抽象。构件技术通过二进制的封装以及动态链接技术解决软件的动态升级和软件的动态替换问题。通常是对一组类的组合进行封装，表示完成一个或多个功能的特定服务，同时通过提供固定的接口来调用该构件所提供的方法，整个构件隐藏了具体的实现，采用接口提供服务。这样，在不同层次上，构件均可以将底层多个逻辑组合成高层次上的粒度更大的新构件。构件之间通过约定的接口进行数据交换和信息传递。构件的位置是相互透明的，可以在同一个用户进程空间，也可以在不同的用户进程空间，甚至在不同的机器上，而且不同的构件可以用不同的语言编写，只要它们符合事先约定的构件规范。

构件是具有强制性、封装性、透明性、互操作性和通用性的软件单元。构件的粒度可大可小，可以是一个简单的按钮实现模型，也可以是潮流计算、状态估计等应用。构件使用与实现语言无关的接口定义语言（IDL）来定义接口。IDL 文件描述了数据类型、操作和对象、客户通过它来构造一个请求，服务器则为一个指定对象的实现提供这些数据类型、操作和对象。构件技术成为了嵌入式操作系统和嵌入式应用软件的发展趋势。

2.2 构件化的 TCP/IP 网络协议栈具有的优点

①针对不同网络应用的需求，能最大限度地利用领域相关知识进行调整，提高了性能。

②底层构件可以在不影响顶层使用的情况下进行修改。

③方便用户开发调试新的协议，方便用户添加新

的协议 。

2.3 基于构件的嵌入式 TCP/IP 网络协议栈

对于运行在内核模式下的 TCP/ IP 网络协议，如数据链路层的 ARP、IP 层的路由协议、协议层的 TCP，为了实现基于构件的 TCP/ IP 网络协议栈，必须对现有的模型进行改善和扩充，其特性要对运行速度、执行效率、系统安全性有严格要求，同时在不影响这些特性下实现跨平台和分布式调度和管理。构件化网络协议栈可以包含一组协议功能，这些功能可用来构成提供适当通信服务的协议，构件化所提供的这种组合式服务，使应用程序能组装任意数目的不同功能的通信服务。因此具备了良好的可复用性、可组装性、可扩展性。

图 1 显示，协议构件化以后，各层协议构件之间的关系。使用了 TCP/ IP 网络协议的通讯构件，从各种协议构件中选择符合需求的模块，按照协议层重新组装使用，显示了极大的灵活性。

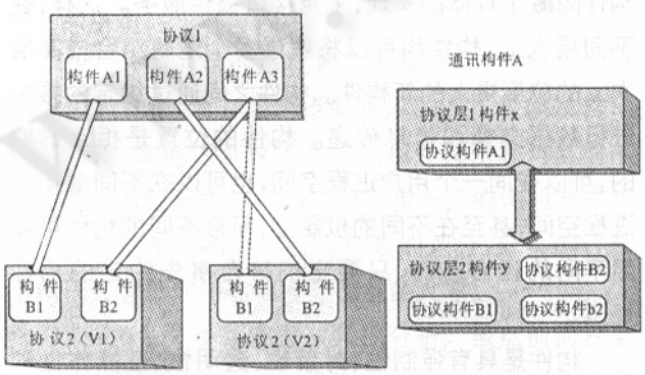

图 1　各协议构件关系

采用构件化方法实现 TCP/IP 网络协议栈的优点在于可以动态的将它们插入或卸出使用这些构件的应用程序或组件。为了这种功能，就必须考虑到许多方面的问题，关键之处在于以下三点：①如何划分构件的粒度；②如何设计构件接口；③提高网络通信的效率。

第一点，从粒度上来看，构件的粒度越小，协议划分的就越细，协议构件就越多；构件粒度越大，协议划分的越粗，协议构件就越少。一般来说，底层的协议粒度较小，顶层的粒度较大。协议构件粒度的大小，决定了协议构件模块化、信息封装性、局部化的程度。目前软件工程领域对构件的粒度没有统一的要求，由于构件是一个高内聚的软件包，只要符合软件工程中高内聚的原则，则构件的粒度大小可不限。例如根据用户的需要，如果用户是开发一个新的协议栈，可以采用大

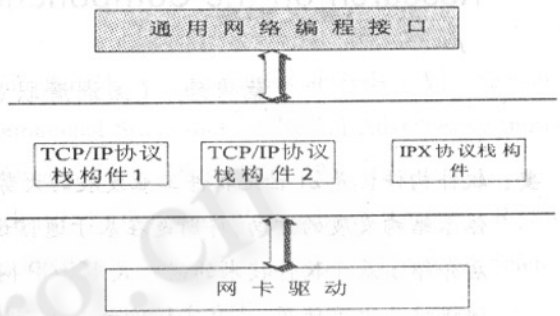

图 2　构件化协议栈框图

粒度的构件划分方式（如图 2 所示），即把整个网络协议栈作为一个大的构件，继承预先定义的构件。

第二点，构件接口提供了构件封装、构件交互、构件与环境的隔离及构件设施等机制。其提供给构件交互规则，并基于这些规则实现类似容器的标准环境。这样构件对外发生作用或构件间的交互，都是通过规范定义的接口进行。构件接口要实现构件的自由替换，首先必须设计良好的构件接口。所有的构件实现应该继承该接口。设计是网络协议构件化的基础，构件接口设计的好坏直接关系到网络协议构件化的性能。构件接口在构件开发平台、构件库和操作系统之间形成一套同意的接口，实现了构件组合之间、构件与操作系统之间、构件与构件库之间的简单清晰的数据传递标准规范，使构件更加透明，数据传输更加规范，构件管理更加方便。按照构件接口开发出的构件具有交好的灵活性和扩展性和相对独立性，易于独立升级、动态加载和跨平台、跨进程、跨网络使用。构件接口的设计原则是尽量保持接口的通用性和简洁性。当有新的功能添加时，可以通过继承实现新的接口方便地加入原有的框架，并不影响原来程序的正常使用。从而实现软件的无缝升级。

第三点，一个高效的通信协议需要满足以下三个方面：上下文切换和定时器的轻型完成；多个协议具有统一的使用界面；在网络设备，核心层和用户层之间有高效的缓冲机制，避免不必要的拷贝。

如果操作系统提供了快捷的上下文切换功能以及高效的进程间通信，那么整个网络协议栈可以作为一个用户态的进程对用户进程提供服务。如果进程上下文切换比较复杂，则整个网络协议栈可以作为动态链接库链入用户进程地址空间，但是这时需要对多进程共享的数据加锁进行同步访问控制。另外一种选择是将网络协议栈装入内核空间，这样也可以避免频繁的进程上下文切换带来的系统开销。

3　TCP 协议模型

根据上述原则，我们可采用不同的策略、粒度来实现协议构件。

TCP 是一个单块式设计的协议，通过协商协议选项以改变协议的功能。如果需要它提供新的服务能力、支持新的下层服务，或者消除协议的缺点时，修改 TCP 协议便是不可避免的。请求修改的结果是使用更高级协议机制取代现存的机制，或者是引进新的或可替代的机制。为了做到向下兼容，老的版本和新的版本必须能够互操作，所以新的版本也必须支持老的功能。由于 TCP 是单块式设计的，它被看作为一个大的协议对象，每个引进的新选项需要对 TCP 对象进行修改，整个协议对选项的依赖关系变大，复杂性增加，正确性也越来越难以保证。建立在基于构件的软件开发模型上的构件化 TCP 协议模型便可以解决这一问题。

3.1 协议模型的结构

图 3 表明了构件化 TCP 的一个可能的体系结构，所组装的连接控制、DSS 和 DSR 对象封装了适当的功能，在考虑了下层服务的情况下，提供所需服务。其中 TCP 连接上下文对象代表

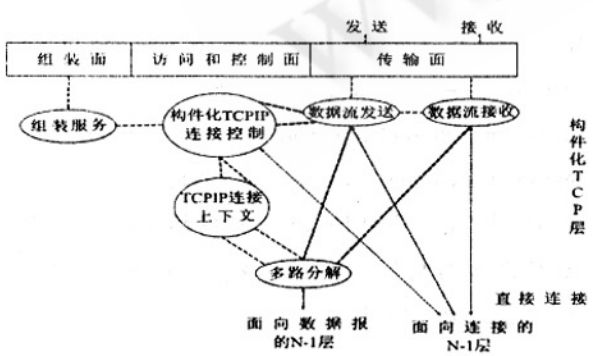

图 3　基于构件的 TCP 结构

一个标准的 TCP 连接。用来在整个连接周期内协商各种 TCP 构件控制信息。为了知道对方 TCP 是否是一个构件化 TCP 的实现，可以使用 TCP 连接上下文对象的选项设置，或一个众所周知的端口号得到对方 TCP 实现版本。TCP 连接上下文对象可以分解成完成分块、字节排序、加校验和、连接控制、差错/流量控制和捎带功能的对象。

协议对象通过 PDU 交互信息，PDU 在数据流中传输时会用到多路分解和多路复用。多路分解是将系统第 N 层的 DSR 分解为第 N +1 层的多个 DSS，多路复用正好相反。多路分解对象根据外来的 PDU 的源端口号和目的端口号，把它们转发给适当的协议对象。只有 TCP 连接上下文对象必须使用多路分解功能，所有其他的对象都可以直接链接到低层协议对象上去，以避免多路分解功能。新的功能是以构件和可复用的方式引入的，而不是用替换整个单块式 TCP 版本来实现的。

3.2 TCP 模块分解与组装

考虑到构件化 TCP 复用单块式 TCP 功能的可能性，原始的 TCP 被分解为一组协议对象，每个对象封装了以下协议功能之一：分块（发送者）、分块（接收者）、字节排序、加校验和、连接控制、多路分解、差错控制和窗口流控（发送者）、差错控制和窗口流控（接收者）、捎带。图 4 表明了 TCP 上下文分解后子对象的一种可能的组装方式。

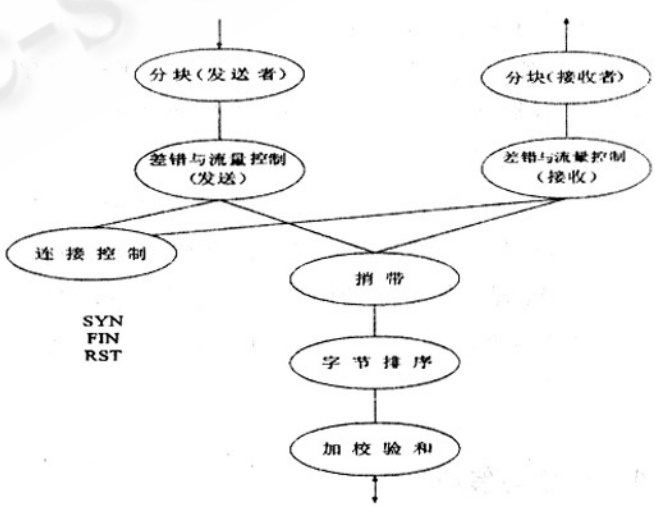

图 4　TCP 上下文分解

所有 TCP 协议对象定义和使用了一个全局的 PDU 格式。在模型设计中，这些对象还要考虑到多方面的因素对功能进行扩展。例如引进窗口比例选项，每个数据 PDU 携带时间戳选项字段，使用加速打开机制支持高效的请求/响应式通信等。组装 TCP 协议功能，使之与所请求的服务以及给定的下层网络性质严格匹配。这一点要求在连接建立期间，对相应的构件化 TCP 的服务能力进行协商。如果所有的参与者协商好需要的服务能力，就能以最好的方式组装 DSS 和 DSR 对象了。为了保持与老的单块式 TCP 实现互操作性，一个构件化 TCP 必须首先了解对方的 TCP 是否基于构件，即是否有能力进行服务组装。构件化 TCP 之间的协商，必须考虑到存在不同的 DSS 和 DSR 对象版本之间的差异。

3.3 可扩展性

遵照协议对象的设计原则，引进新的或可替代的协议功能和机制，不会影响已存在的协议功能和机制。另外，仅仅通过组装已存在的协议功能对象，也能提供新的服务。组装服务负责按照所要求的服务能力配置连接控制、DSS 和 DSR 对象。服务能力可以包含定义传输质量的服务质量、定义多点连接方式的服务模式、以及定义诸如动态增加新的参与者的连接控制功能的服务设施。

构件化的 TCP 协议，嵌入了标准的单块式 TCP 功能，并把任何选项作为可复用的协议对象设计和实现。每个 TCP 连接，能以专门的和最佳的 PDU 和状态机，配置恰当的 DSS 和 DSR 对象。这些 DSS 和 DSR 对象会有最佳的头部结构：如 32 位的窗口大小字段，而不是窗口比例选项；64 位序号字段，而不是那种 PAWS 机制；32 位的时间戳字段，而不是时间戳选项。此外，它们还能具有专门协议机制，用来支持请求/响应或者局部有序和局部可靠传递。这种方法保证了不同版本的互操作性，使 TCP 具备更好的可复用性和可扩展性。

4 总结与展望

构件化的协议可以组装模型进行协议动态配置，能够无缝实现构件的动态替换。既可以由构件支撑系统自动装配，又可以由系统管理员手动配置，大大增加了通讯设备的稳定性与服务质量。构件特征参数比较清晰的描述出了通信设备对服务对象的要求，使得服务对象能够更加具体化，更加符合通信设备的要求。

结合市场需求特点，在基于构件的嵌入式平台上开发面向通信设备的应用软件，嵌入式软件开发平台应用到更多的产品上。可以是 IPV6 路由器、GAR 低端路由器、CDMA 1X 基站系统、UAS、ADSL 宽带接入设备、SDH 光传输设备、视讯会议多点控制器、视讯多媒体终端、高端手机、3G 手机等产品上应用，扩大平台的产业化空间，创造巨大的经济效益，走出自主知识产权产业化道路。

参考文献

1 Wright G R, Stevens W R. TCP/IP Illustrated Vol 2: The Implementation. Boston MA USA, Addison Wesley, 1995.

2 Stevents W R. TCP/IP Illustrated Vol 1: The Protocol. Boston MAA, Addison Wesley, 1994.

3 Karn P. KA9Q TCP/IP Source Code. http://www.ka9q.net/code/ka9qnos/ , 1993.

4 R. Braden ,D. Borman , and C. Partridge. Computing the internet checksum. RFC 1071, Internet Engineering Task Force, September 1988.

5 W. Richard Stevens. TCP/ IP 详解. 卷一：协议[M]. 范建华，等，译. 北京：机械工业出版社，2000.

6 Gary R. Wright ,W. Richard Stevens. TCP/ IP 详解 卷二：实现[M]. 陆学莹，译. 北京：机械工业出版社，2000.

7 McCombie B. Embedded web servers now and in the future. Real - Time Magazine ,1998 ,1.

8 J. Postel. Internet protocol. RFC 791 , Internet Engi2neering Task Force , September 1981.

9 Comer ,D. E. 用 TCP/ IP 进行网际互联第一卷：原理、协议和结构(第四版)[M]. 林瑶，等，译. 北京：电子工业出版社，2001.