

## 第10章

---

# HTTP-NG



在这本书即将完稿的时候，HTTP 正在庆祝它的第十个生日。这十年是这个因特网协议成就辉煌的十年。现在，世界上绝大多数的数字流量都是由 HTTP 传输的。

但随着 HTTP 迈入青少年时期，它也面临着一些挑战。从某些方面来说，HTTP 应用的步伐已经超越了其设计。现在，人们将 HTTP 作为各种不同应用程序的基础，并将其运行在很多不同的联网技术之上。

本章对 HTTP 未来的一些发展趋势和所要面临的挑战，以及被称为 HTTP-NG 的下一代 HTTP 结构方案进行了介绍。尽管 HTTP-NG 工作组已经解散了，而且看起来也不太可能得到快速的应用，但它还是给出了 HTTP 未来一些潜在的发展方向。

## 10.1 HTTP 发展中存在的问题

HTTP 最初被设想为一种简单的技术，用于访问分布式信息服务器上链接的多媒体内容。但在过去的十年中，HTTP 及其衍生产品起到了更为广泛的作用。

HTTP/1.1 现在提供了可以追踪文档版本的标记和指纹，提供了一些方法来支持文档的上传以及与可编程网关之间的交互，还提供对多语言内容、安全及认证功能、降低流量的缓存功能、减小时延的管道功能、降低启动时间提高带宽使用效率的持久连接，以及用来进行部分更新的访问范围功能的支持。HTTP 的扩展及衍生产品具有更为广泛的功能，可以提供对文档发布、应用程序服务、任意的消息服务、视频流以及无线多媒体访问的支持。HTTP 正在成为分布式多媒体应用程序的“操作系统”。

尽管 HTTP/1.1 的设计经过了充分的考量，但随着 HTTP 被越来越多地用作复杂远程操作的统一载体，HTTP/1.1 已经开始显现出了一些局限性。HTTP 的发展中至少存在 4 方面的问题。

- 复杂性

HTTP 相当复杂，而且其特性之间是相互依存的。由于存在一些复杂的、相互交织的要求，以及连接管理、报文处理和功能逻辑之间的混合作用，要想正确地实现 HTTP 软件肯定是非常痛苦、很容易出错的。

- 可扩展性

HTTP 很难实现递增式扩展。很多流传下来的 HTTP 应用程序中都没有自主的功能性扩展技术，使协议的扩展无法兼容。

- 性能

HTTP 中有些部分效率不高。其中很多低效特性会随着高时延、低吞吐量的无线

访问技术的广泛使用而变得更加严重。

- 传输依赖性

HTTP 是围绕 TCP/IP 网络协议栈设计的。尽管没有限制说不能使用替代协议栈，但在这方面所做的工作非常少。HTTP 要为替代协议栈提供更多的支持，才能作为一个更广阔的报文发送平台应用于嵌入式和无线应用程序之中。

## 10.2 HTTP-NG的活动

1997 年夏天，万维网联盟启动了一个特殊项目，调查并提出一个新的 HTTP 版本，以修正与复杂性、可扩展性、性能及传输依赖性有关的一些问题。这个新的 HTTP 被称为 HTTP：下一代系统（HTTP-NG）。

在 1998 年 12 月举行的一次 IETF 会议上，提出了一组 HTTP-NG 建议。这些建议勾勒出了一组可能的 HTTP 主要发展方向。这项技术还未被广泛采用（可能永远也不会被广泛采用了），但 HTTP-NG 确实在扩展 HTTP 系统方面做出了最认真的努力。下面我们来看看 HTTP-NG。

## 10.3 模块化及功能增强

可以用三个英语单词来描述 HTTP-NG 的主题：“模块化及功能增强”（modularize and enhance）。如图 10-1 所示，HTTP-NG 工作组建议将协议模块化为三层，而不是将连接管理、报文处理、服务器处理逻辑和协议方法全都混在一起。

248

- 第一层，报文传输层（message transport layer），这一层不考虑报文的的功能，而是致力于端点间报文的不透明传输。报文传输层支持各种子协议栈（比如无线环境下的协议栈），主要负责处理高效报文传输及处理方面的问题。HTTP-NG 项目组为本层提出了一个名为 WebMUX 的协议。
- 第二层，远程调用层（remote invocation layer），定义了请求 / 响应的功能，客户端可以通过这些功能调用对服务器资源的操作。本层独立于报文的传输以及操作的精确语义。它只是提供了一种标准的方法来调用服务器上所有的操作。本层试图提供一种像 CORBA、DCOM 和 Java RMI 那样的面向对象的可扩展框架，而不是 HTTP/1.1 中那种静态的、服务器端定义的方法。HTTP-NG 项目组建议本层使用二进制连接协议（Binary Wire Protocol）。
- 第三层，Web 应用层（Web application layer），提供了大部分的内容管理逻辑。所有的 HTTP/1.1 方法（GET、POST、PUT 等），以及 HTTP/1.1 首部参数都是在这里定义的。本层还支持其他构建在远程调用基础上的服务，比如 WebDAV。

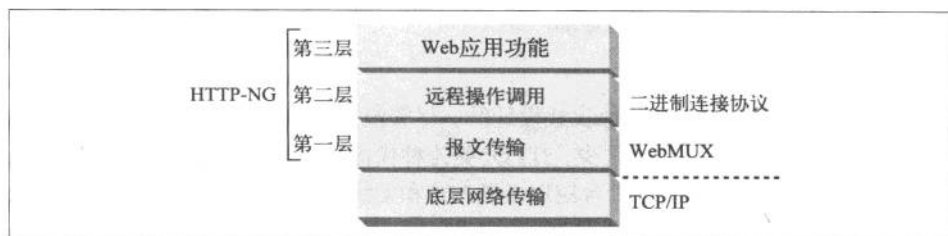


图 10-1 HTTP-NG 将功能都分散到各层之中实现

只要将 HTTP 组件模块化了，就可以对其进行改进，以提供更好的性能和更丰富的特性。

## 10.4 分布式对象

HTTP-NG 的很多基本原理和功能目标都是从 CORBA 和 DCOM 这样的结构化、面向对象的分布式对象系统中借鉴来的。分布式对象系统对可扩展性和功能特性都很有帮助。

从 1996 年开始，一个研究团体就在争论是否要将 HTTP 与更复杂的分布式对象系统聚合在一起。在 Web 中使用分布式对象模型有很多好处，更多与此相关的信息请查阅 Xerox PARC 早期名为“Migrating the Web Toward Distributed Objects”（“Web 向分布式对象的迁移”）的文章（<ftp://ftp.parc.xerox.com/pub/ilu/misc/Webilu.html>）。

将 Web 和分布式对象统一起来的雄心使得 HTTP-NG 的应用受到了某些社团的抵制。过去的一些分布式对象系统受到了重量级实现方案和形式上复杂性的影响。HTTP-NG 项目组也尝试去解决需求中提到的一些问题。

## 10.5 第一层——报文传输

我们从最底层开始，近距离地看看 HTTP-NG 这三层的功能。报文传输层关心的是报文的有效传输，不考虑报文的含义和目的。报文传输层为报文传输提供了一个 API，无论底层实际采用的是什么网络协议栈都可以使用。

本层关注的是提高报文传输的性能，其中包括：

- 对报文进行管道化和批量化传输，以降低往返时延；
- 重用连接，以降低时延，提高传输带宽；
- 在同一条连接上并行地复用多个报文流，在防止报文流饿死的同时优化共享连接；
- 对报文进行有效的分段，使报文边界的确定更加容易。

HTTP-NG 工作组将大部分精力都放在了为第一层的报文传输开发 WebMUX 协议上。WebMUX 是个高性能的报文协议，可以对报文进行分段，并在一条复用的 TCP 连接上交错地传输报文。本章会对 WebMUX 进行较为详细的介绍。

## 10.6 第二层——远程调用

HTTP-NG 结构的中间层提供了对远程方法调用的支持。本层提供了通用的请求 / 响应框架，客户端可通过此框架调用对服务器资源的操作。本层并不关心特定操作的实现及语义（缓存、安全性以及方法逻辑等），它只关心允许客户端远程调用服务器操作的接口。

现在已经有很多远程方法调用标准了（举几个例子来说，比如 CORBA、DOM 和 Java RMI），本层并不打算支持这些系统中所有好的特性。但它有一个明确的目标，就是要对 HTTP/1.1 所提供的 HTTP RMI 支持进行扩展。特别是，要以可扩展的面向对象方式提供更通用的远程过程调用支持。

HTTP-NG 小组建议本层采用二进制连接协议。这个协议支持一种高性能的可扩展技术，通过这种技术可以调用服务器上经过良好描述的操作，并将结果返回。本章稍后将对二进制连接协议进行较为详细的讨论。

250

## 10.7 第三层——Web应用

Web 应用层是执行语义和应用程序特定逻辑的地方。HTTP-NG 工作组避开了扩展 HTTP 应用特性的诱惑，专注于正规的基础建设工作。

Web 应用层描述了一个用于提供应用程序特定服务的系统。这些服务并不单一，不同的应用程序可能使用不同的 API。比如，HTTP/1.1 的 Web 应用构成的应用程序与 WebDAV 可能会共享一些公用的部分，但又会有所不同。HTTP-NG 结构允许多个应用共存于本层，共享底层特性，它还提供了一种添加新应用程序的机制。

Web 应用层的基本思想是提供与 HTTP/1.1 等价的功能和一些扩展接口，同时将其映射到一个可扩展的分布式对象框架中去。更多与 Web 应用层接口有关的内容可以参见 <http://www.w3.org/Protocols/HTTP-NG/1998/08/draft-larner-nginterfaces-00.txt>。

## 10.8 WebMUX

HTTP-NG 工作组花费了很多精力，为报文传输开发了 WebMUX 标准。WebMUX 是一个复杂的高性能报文系统，通过这个系统，可以在一个复用的 TCP 连接上并行

地传输报文。可以对以不同速度产生和消耗的独立报文流进行高效的分组，并将其复用到一条或少数几条 TCP 连接上去（参见图 10-2）。

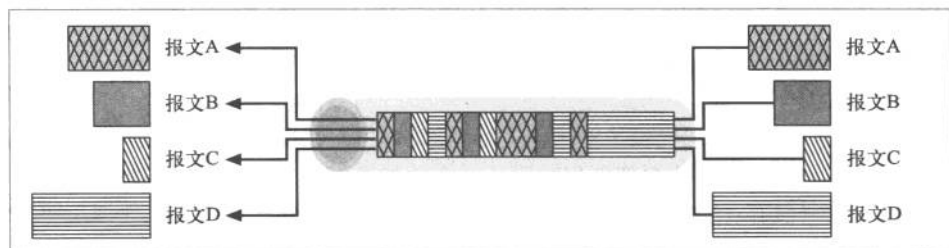


图 10-2 WebMUX 可以在一条连接上复用多条报文

WebMUX 协议的关键目标包括如下几条。

- 设计简单。
- 高性能。
- 复用——可以在一条连接上动态、高效地交错传递多个（使用任意高层协议的）数据流，不用因为等待那些速度很慢的生产者程序而延迟数据的传输。
- 基于信用的流量控制——数据是以不同的速率产生和消耗的，发送者和接收者的内存和可用的 CPU 资源都有所不同。WebMUX 使用的是“基于信用的”流量控制方案，接收者可以预先声明期望的数据接收速度，防止出现资源缺乏产生的死锁。
- 保持对齐——保持复用流中数据的对齐，这样才能有效地发送并处理二进制数据。
- 丰富的功能——接口足够丰富，能支持套接字 API。

251

更多有关 WebMUX 协议的内容请参阅 <http://www.w3.org/Protocols/MUX/WD-mux-980722.html>。

## 10.9 二进制连接协议

HTTP-NG 小组建议使用二进制连接协议来提高下一代 HTTP 协议支持远程操作的能力。

HTTP-NG 定义了一些“对象类型”，并为每种对象类型分配了一组方法。为每种对象类型分配一个 URI，以便将对它的描述和它的方法宣传出去。通过这种方式，HTTP-NG 提供了一种比 HTTP/1.1 的扩展性更强，且面向对象的执行模型，HTTP/1.1 中所有的方法都是在服务器中静态定义的。

二进制连接协议通过一条有状态的连接承载了从客户端发往服务器的操作调用请求，以及从服务器发往客户端的操作结果应答。有状态的连接可以提供更高的效率。

请求报文中包含操作、目标对象和可选的数据值。应答报文带回了操作的最终状态、所对应请求的序列号（允许以任意顺序传递并行的请求和响应），以及可选的返回值。除了请求和应答报文之外，这个协议还定义了几种内部控制报文，用来提高连接的效率和强壮性。

更多有关二进制连接协议的内容请参阅 <http://www.w3.org/Protocols/HTTP-NG/1998/08/draft-janssen-httpng-wire-00.txt>。

## 10.10 当前的状态

1998 年底，HTTP-NG 小组认定现在将 HTTP-NG 建议引入 IETF 还为时尚早。工业界和各社团都还没有完全调整到 HTTP/1.1 上来，如果没有明确的转换计划就进行重大的 HTTP-NG 重构计划，将其重构为分布式对象模式可能会造成严重

252

对此，人们提出了两点建议。

- 不要指望能在一步之内实现整个 HTTP-NG 重构计划，建议重点关注 WebMUX 传输技术。但是，在编写此书的时候，人们还没有足够的兴趣去建立一个 WebMUX 工作组。
- 要去探讨能否将正式的协议类型修改得足够灵活（可能是通过 XML 实现的），以满足 Web 上应用的需要。对于可扩展的分布式对象系统来说，这一点尤其重要。这项工作仍在进行之中。

编写此书时，还没有什么主导力量在驱动 HTTP-NG 的应用。但随着 HTTP 应用的不断增多，随着它越来越多地作为各种应用程序的平台使用，以及无线因特网技术和面向消费者的因特网技术的不断应用，HTTP-NG 中提出的一些技术可能会在 HTTP 的青少年时期逐渐显现出其重要性。

## 10.11 更多信息

更多有关 HTTP-NG 的信息，请参见下面列出的详细规范和活动报告。

- <http://www.w3.org/Protocols/HTTP-NG/>  
HTTP-NG 工作组（建议），W3C 联盟 Web 站点。

- <http://www.w3.org/Protocols/MUX/WD-mux-980722.html>  
J. Gettys 和 H. Nielsen 编写的 “The WebMUX Protocol” ( “WebMUX 协议” )。
- <http://www.w3.org/Protocols/HTTP-NG/1998/08/draft-janssen-httpng-wire-00.txt>  
B. Janssen 编写的 “Binary Wire Protocol for HTTP-NG” ( “HTTP-NG 的二进制连接协议” )。
- <http://www.w3.org/Protocols/HTTP-NG/1998/08/draft-larner-nginterfaces-00.txt>  
D. Larner 编写的 “HTTP-NG Web Interfaces” ( “HTTP-NG Web 接口” )。
- <ftp://ftp.parc.xerox.com/pub/ilu/misc/Webilu.html>  
D. Larner 编写的 “Migrating the Web Toward Distributed Objects” ( “Web 向分布式对象的迁移” )。

253