



(12) 发明专利申请

(10) 申请公布号 CN 105340242 A

(43) 申请公布日 2016.02.17

(21) 申请号 201480035501.1

(51) Int. Cl.

(22) 申请日 2014.04.21

H04L 29/08(2006.01)

(30) 优先权数据

H04L 29/06(2006.01)

1307340.8 2013.04.23 GB

(85) PCT国际申请进入国家阶段日

2015.12.23

(86) PCT国际申请的申请数据

PCT/EP2014/001052 2014.04.21

(87) PCT国际申请的公布数据

WO2014/173521 EN 2014.10.30

(71) 申请人 古如罗技微系统公司

地址 芬兰土尔库市里南路 34 号

(72) 发明人 托马斯·迈克尔·卡开宁

沃特里·哈卡莱依宁 奥西·卡雷沃

(74) 专利代理机构 北京新知远方知识产权代理

事务所(普通合伙) 11397

代理人 张超颖

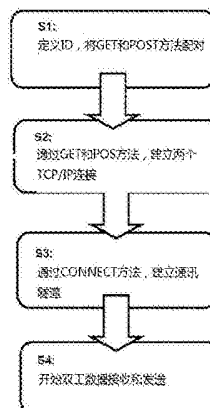
权利要求书1页 说明书9页 附图3页

(54) 发明名称

使用 HTTP 的双向实时通讯系统

(57) 摘要

提供了一种经由支持基于 HTTP 的通讯的系统建立通讯连接的方法,包括步骤:(a) 采用与 HTTP 相关联的 GET 和 POST 方法的组合,使用通讯系统建立系统的两个节点之间的双向实时通讯连接;(b) 通过采用与 HTTP 关联的 CONNECT 方法对双向通讯连接进行 TCP/IP 和 / 或 UDP 隧道化。还提供了相应的通讯系统。



1. 一种通讯系统,操作于支持基于 HTTP 的通讯,其中,所述通讯系统采用与 HTTP 相关联的 GET 和 POST 方法的组合,建立系统的两个节点之间的双向实时通讯连接,其中,经由网络连接的数据交换的实现是采用区块化的方式和 / 或作为一系列的数据块,其特征在于,通过网络连接通讯的用于数据区块和 / 或数据块的最大片段大小 (MSS) 是作为支持通讯连接的通讯网络容量的方程而被优化的。

2. 根据权利要求 1 所述的通讯系统,其特征在于,所述双向通讯连接是通过采用与 HTTP 关联的 CONNECT 方法而 TCP/IP 和 / 或 UDP 隧道化的。

3. 根据权利要求 1 所述的通讯系统,其特征在于,通讯连接包括一个接收连接和一个发送连接,用来提供双向通讯,其中,连接被保持开放,直到一个空的区块和 / 或一个空的多部分数据块被接收。

4. 根据权利要求 1 所述的通讯系统,其特征在于,通讯连接对通过其通讯的数据使用加密。

5. 根据权利要求 1 所述的通讯系统,其特征在于,通讯连接为以下至少之一提供通讯:图形数据、图像数据、视频数据、音频数据、非结构化数据。

6. 一种经由支持基于 HTTP 的通讯的通讯系统建立通讯连接的方法,包括步骤:

(a) 采用与 HTTP 相关联的 GET 和 POST 方法的组合,使用通讯系统建立系统的两个节点之间的双向实时通讯连接;

(b) 采用区块化的方式和 / 或作为一系列的数据块,经由通讯连接进行数据交换;

(c) 对通过网络连接通讯的用于数据区块和 / 或数据块的最大片段大小 (MSS) 进行优化,所述优化是将其作为支持通讯连接的通讯网络容量的方程。

7. 根据权利要求 6 所述的方法,其特征在于,所述方法包括步骤,通过采用与 HTTP 关联的 CONNECT 方法对双向通讯连接进行 TCP/IP 和 / 或 UDP 隧道化。

8. 根据权利要求 6 所述的方法,其特征在于,所述通讯连接包括一个接收连接和一个发送连接,用来提供双向通讯,其中,连接被保持开放,直到一个空的区块和 / 或一个空的多部分数据块被接收。

9. 根据权利要求 6 所述的方法,其特征在于,所述通讯连接对通过其通讯的数据使用加密。

10. 根据权利要求 6 所述的方法,其特征在于,所述通讯连接为以下至少之一提供通讯:图形数据、图像数据、视频数据、音频数据、非结构化数据。

11. 一种记录于机器可读数据存储介质上的软件产品,所述软件产品执行于计算硬件,用以实现根据权利要求 6 的方法。

12. 根据权利要求 11 所述的软件产品,其特征在于,所述软件产品用 HTTP 表达,被执行于按照 HTTP 运行的通讯网络的服务器。

使用 HTTP 的双向实时通讯系统

技术领域

[0001] 本公开涉及通讯系统,尤其是使用实时超文本协议(Hypertext Transfer Protocol,HTTP)用于通讯图形数据、图像数据、视频数据、音频数据等各种类型的数字数据的通讯系统。进一步,本公开还涉及操作上述用于通讯各类数据的通讯系统的方法。再进一步,本公开还涉及记录于机器可读存储介质上的软件产品,其中,软件产品可执行于计算硬件以实现上述方法。

背景技术

[0002] 一般而言,超文本协议(HTTP)广泛用于实现当前的互联网。该协议是用于分布的、协作的超媒体信息系统的—个应用协议。在实现中,HTTP是操作用于建设网络的对象的一个多线性集,使用逻辑连接来定义网络。连接就是通常所称的“超链接”,它定义节点之间的网络关系。

[0003] 在为互联网而实现的客户服务模型中,HTTP是作一个需求响应协议来工作。在该模型中,网络浏览器被用于实现客户端,在服务器上执行的软件应用可以主管(host)一个网站。在操作中,一个特定的客户端提交一个HTTP请求消息给服务器,服务器做出响应的方式是提供如HTML文件和其它内容等资源,或者代表客户端执行数据处理工作,或者甚至是返还一个响应消息给客户端。上述的网络浏览器可以由各种方式来实现,例如,作为一个用户代理、作为一个网络爬虫、或者是任何可执行于计算硬件上的其它软件,所述的计算硬件访问、使用或者显示源于互联网的数据内容。

[0004] HTTP允许即时的(immediate)网络单元来使能客户端与服务器之间的通讯。互联网的高流量网站通常使用网络缓存服务器,其代表上游服务器来传送内容,以改善数据和/或服务传送的响应时间。此外,设在专用网络的边界的HTTP代理服务器是有益的,它用来促进与没有全球可路由地址的客户端的通讯,即,通过外部服务器来中继消息。

[0005] 通过使用统一资源标识(Uniform Resources Identifier,URI),通常也称之为统一资源定位器(Uniform Resources Locator,URL),HTTP资料可被识别和定位于一个给定的网络。URI与超链接用超文本标识语言(HTML)来表达,HTML能够形成彼此互连的超文本文件。

[0006] HTTP会话是通过网络需求响应交易的序列来实现的。例如,HTTP客户端建立与服务器的一个特殊端口的传输控制协议(Transmission Control Protocol,TCP)连接来发起一个请求,发回一个状态线(例如,“HTTP/1.1200OK”)连同一个相关的消息而做出响应。该相关消息的主体通常是被请求的资源,尽管所返回的也可能是一个错误消息。

[0007] HTTP定义一些方法,通常被便利地称之为“动词”,用于指示针对一个被识别的资源的一个期望的动作。资源可以是数据文件或来自一个可执行对象的输出,驻留在一个或多个服务上。表1提供了HTTP方法,也称之为HTTP动词的示例。

[0008] 表1、HTTP方法(HTTP动词)

[0009]

“动词”	详情
GET	请求具体资源的呈现，其中，使用“GET”的请求应当仅调取数据。
HEAD	请求一个响应，所述的响应与从 GET 可获得的是一样的，只是缺失响应的主体，HEAD 通常被用于以有效的方式调取元数据。
POST	请求一个给定的服务器接收请求所附带的一个实体，把它作为被 URL 所确定的给定的网络资源的一个新的从属。
PUT	请求将被附带的实体进行存储，根据被提供的 URI(URL)来存储。如果 URI 指向一个已经存在的资源，那项资源被修改。
DELETE	请求删除具体的资源
TRACE	导致一个收到的请求被回应给一个给定的客户端
OPTIONS	返回 HTTP 方法，所述方法被与一个给定的 ULR 相关联的服务器支持。
CONNECT	把被请求的链接转换成透明的 TCP/IP 隧道，例如，通过上述的非加密的 HTTP 代理来促成 TLS 和 SSL 加密的通讯 (HTTPS)，默认的是，HTTP 连接是非加密的，HTTPS 连接是加密的。
PATCH	请求将部分的修改应用于一个给定的资源

[0010] 因此，当前的网络浏览器所采用的主要的传送协议是前述的 HTTP。在特殊的浏览器软件应用中，几个相关的“生态系统”、它们所使用的软件，在没有 HTTP 的情况下无法运行。如上所述，HTTP 是基于请求的，参见表 1，所述的请求是被传送的，响应于这些请求，接到这些请求后，HTML 页或二进制数据（如图像或音频流 / 文件）通常被提供。

[0011] 考虑到互联网的复杂性，互联网通常的延迟 (latency) 会在运行中出现。在要求数据交换的情况中，这些延迟能引发问题，如，需要双向（双工）的通讯，这是需要实时响应的，比如视频图像和 / 或音频的传送与接收只能容忍很小的延迟。通过互联网的双向通讯的已知技术是通过互联网协议的语音服务 (VoIP)，以及基于互联网的视频会议，例如，当今使用 SKYPE (SKYPE 是个注册商标) 或类似软件而提供的服务。

[0012] 已知的是，采用“WebSockets”协议（该协议在网址链接 <http://tools.ietf.org/html/rfc6455> 中有介绍），来应对一个具体类型的通讯需要，它实现了下述的通讯特性：

[0013] (i) WebSocket 用在 HTTP/HTTPS 隧道内，在此情况下，为端口 80/443 打开了防火墙，因为它们现在通常被用在网络浏览器中。

[0014] (ii) WebSocket 被用在全双工连接模式，其中，只有一个 TCP 连接能够实时双向通讯，即，通过改变数据传送的方向，用一个连接来传送和接收数据。

[0015] 尽管如此，这样的 WebSocket 是依赖于端口的，这带来了不期望的限制。

发明内容

[0016] 本发明旨在提供一个通讯系统，它能够通过 HTTP 通讯网络，以一种改进的方式提供双向的数据通讯。

[0017] 此外，本发明旨在提供一种改进的运行通讯系统的方法，用于通过 HTTP 通讯网络提供双向的数据通讯。

[0018] 根据本发明的第一方面，提供一种通讯系统，支持基于 HTTP 的通讯，其中，所述通讯系统采用与 HTTP 相关联的 GET 和 POST 方法的组合，建立系统的两个节点之间的双向实

时通讯连接,其中,经由网络连接的数据交换的实现是采用区块化的方式和 / 或作为一系列的数据块,其特征在于通过网络连接通讯的用于数据区块和 / 或数据块的最大片段大小 (MSS) 是作为支持通讯连接的通讯网络容量的方程而被优化的。

[0019] 该通讯系统的优点在于,它能够提供实时双向通讯,降低了延迟。

[0020] 可选地,CONNECT 方法能够被用于三种不同类型的场景:

[0021] (i) 一个连接被隧道化进入一个目标;这是默认的场景;

[0022] (ii) 通过本地主机,一个连接被隧道化进入一个目标,结果是,数据从本地服务中的一个发送进程中被传输给一个转发进程,数据从转发进程被发送给目标;这样一个方法是有益的,因为它能够阻止防病毒软件分析数据,这种分析会无意中拦阻或干预数据;

[0023] (iii) 一个连接被隧道化进入一个转发代理服务器,转发代理服务器然后将数据重新导向到目标;这样一个方法有益于在负载平衡系统中被采用,即,由客户端导致的网络负载被最优地分发到目标。例如,在骨干网中发送数据快于通过直接连接的发送。

[0024] 可选地,在所述通讯系统中,通讯连接包括一个接收连接和一个发送连接,用来提供双向通讯,其中,连接被保持开放,直到一个空的区块和 / 或一个空的多部分数据块被接收。

[0025] 可选地,在所述通讯系统中,通讯连接对通过其通讯的数据使用加密。

[0026] 在通讯系统中,通讯连接为以下至少之一提供通讯:图形数据、图像数据、视频数据、音频数据、非结构化数据。

[0027] 根据第二方面,本发明提供一种方法,经由支持基于 HTTP 的通讯的通讯系统建立通讯连接,其中,所述方法包括步骤:

[0028] (a) 采用与 HTTP 相关联的 GET 和 POST 方法的组合,使用通讯系统建立系统的两个节点之间的双向实时通讯连接;

[0029] (b) 采用区块化的方式和 / 或作为一系列的数据块,经由网络连接进行数据交换;

[0030] (c) 对通过网络连接通讯的用于数据区块和 / 或数据块的最大片段大小 (MSS) 进行优化,所述优化是将其作为支持通讯连接的通讯网络容量的方程。

[0031] 可选地,在所述方法中,通讯连接包括一个接收连接和一个发送连接,用来提供双向通讯,其中,连接被保持开放,直到一个空的区块和 / 或一个空的多部分数据块被接收。

[0032] 可选地,在所述通讯系统中,通讯连接对通过其通讯的数据使用加密。

[0033] 在通讯系统中,通讯连接为以下至少之一提供通讯:图形数据、图像数据、视频数据、音频数据、非结构化数据。

[0034] 根据第三方面,本发明提供了一种记录于机器可读数据存储介质上的软件产品,所述软件产品执行于计算硬件,用以实现根据本发明的第二方面的方法。

[0035] 可选地,所述软件产品用 HTTP 表达,被执行于按照 HTTP 运行的通讯网络的服务器。

[0036] 本发明的优点在于,其通讯系统能够提供双向全双工通讯,不管是加密或不加密,它使用已知的 HTTP 传输协议,执行于通讯系统中的软件或硬件防火墙和 / 或防病毒软中,额外的设置不是必要的。

[0037] 此外,本发明的益处还在于,它提高了通讯应用的功能和可靠性,因而简化了与系统相关的技术维护问题,如数据安全设置。

[0038] 可以理解,在不脱离所附权利要求限定的本发明范围的情况下,本发明的特征易于组合为多种形式。

附图说明

[0039] 现在将参考以下附图,仅通过示例来描述本公开的实施例,在附图中:

[0040] 图 1 是采用 HTTP 的通讯网络的示意图

[0041] 图 2 是本发明的方法的步骤的示意图;

[0042] 图 3 是本发明的方法的另一套步骤的示意图。

[0043] 在附图中,使用带下划线的数字来表示该带下划线的数字位于其上的项目或该带下划线的数字与之相邻的项目。不带下划线的数字与通过将该不带下划线的数字和一项目相连的线所标识的项目相关联。当一个数字不带下划线并且伴随有相关联的箭头时,使用该不带下划线的数字来标识该箭头所指向的一般项目。

具体实施方式

[0044] 整体上,参见图 1,它描述一个系统和相关的方法,系统的一部分在总体上用 5 表示,所述方法能够减少延迟,它们是针对用于双向实时通讯,所采用的 HTTP 的描述符合 RFC2621、RFC2068、RFC1945 等标准。通常,HTTP 不是被设计用于使能第一和第二节点 10A、10B 之间的实时双向通讯,其中一个给定的客户端能够同时发送实时数据和实时接收,按以下方式进行:

[0045] (i) 在两个节点 10A、10B 之间所采用的通讯连接 20 支持加密格式的双向通讯;

[0046] (ii) 病毒保护软件 30 不干预正通过通讯连接 20 被发送和接收的内容 40;

[0047] (iii) 防火墙 60 不能阻止网络流量,除非发生互联流量总体阻塞的情况,即“WWW 流量”被阻止,例如,为了实现安全的财务交易而采用了银行连接;

[0048] (iv) 网络设备,例如桥接器和路由器,不能够分析和干预要通过通讯连接 20 而通讯的数据。

[0049] 本发明的实施例能够通过采用以下的特性而实现 (i) 至 (iv) 的功能:

[0050] (a) 采用两种互不相同的类型的 GET 和 POST 方法,参见上述的表 1,其中,GET 方法构建一个经由通讯连接 20 接收连接,POST 方法构建一个经由通讯连接 20 的发送连接;

[0051] (b) 采用与现在的 HTTP 中所用的一样的 CONNECT 方法,两个连接被隧道化;

[0052] (c) 采用一种形式的区块化的 (chunked) 或多部分的传输编码,下文将对此详述。

[0053] 传统地,HTTP 用于互联网会话,其中,GET 和 POST 方法以彼此独立的方式被采用,例如,GET 方法用于从网络服务器请求 HTML 内容,网络服务器用网络浏览客户端的主机,其中,为 GET 方法的连接保持开放,直至所有的响应数据被从主机发送至客户端。而且,采用了一个连接程序,它与 POST 方法一样,参见表 1,除非数据是从客户端被发送至主机。

[0054] 在以下描述的实施例中,以这样的方式执行通讯:以半双式的方式使用一个给定的封装 (socket),它区别于象前述的 WebSocket 这样的已知方法的实施例。在实施例中,数据的发送和 / 或接收比全双工连接更有效,因为网络界面卡不需要在接收与发送之间切换它们的输入 / 输出 (I/O) 状态。在已知的技术方案中所采用的这种切换消耗系统资源,因而也降低潜在的通讯速度。

[0055] 在以下描述的实施例中,只在 HTTP GET 和 POST 方法的初始化之后采用封装,或者是接收模式或者是发送模式。接下来,所用的网络适配器只需要工半双工的状态工作,从而节约网络基础设施与设备资源,因为在协商的 HTTP GET 和 / 或 POST 方法 headers(头部)之后直至出现连接的结束,连接仅仅以要么发送要么接收的状态工作。而且还有其它的益处,例如,防火墙和路由器,即集线器和转换器,接收更少的转换负荷,因而不会像只使用一个双工连接的现有双工通信方法那么快地中断。所以,这里所描述的实施例较之于前述的 WebSocket,对资源的使用更有效。

[0056] 前述的已知 WebSocket 很容易被防火墙分析认为是属于无法识别的连接类型而被断开连接,从而阻止或限制它们的使用,不管相关联的连接是隧道化与否。在本发明实施例中描述了一个根据 HTTP 协议的 GET 或 POST 连接功能,防火墙不能限制或阻止使用这些方法的通讯。

[0057] 在下文的实施例中,采用的 UDP 协议估计比 TCP 快三倍。可选地,这些实施例可以使用对等(peer-to-peer, P2P) 连接,它允许在应用层实现连接。

[0058] 这里所描述的实施例不同于已知的 HTTP 实现方式,在已知的 HTTP 实现中,缺少 GET 和 POST 方法之的任何连接,与之相区别的是,本发明的实施例采用 GET 与 POST 方法按新的形式相融合的方案,从而提供实时全双工的数据通讯。这里所提及的全双工数据通讯的实现方式是,使用一个接收连接和一个发送连接。一个接收连接或一个发送连接能够使用一个半双工连接模式或者一个全双工连接模式。

[0059] 虽然下述实施例的描述是基于 TCP 协议,可以理解,UDP(User Datagram Protocol) 协议也可以作为一个替代方案而被使用。虽然 TCP 和 UDP 都依赖于底层的 IP(Internet Protocol) 协议,UDP 数据报和 TCP 片段都以 IP 包(packet) 的形式发送,UDP 的不同之处在于,它是无连接协议,通过使用一个网络地址翻译(NAT),它有可能实现应用之间的对等通讯,不仅仅是一个本地局域网(LAN) 内部,而且在外部的互联网也是这样。通过使用这种方法,在系统 5 中通过服务器传输数据的需要就可以避免了,显著地节省了通讯网络容量。在系统 5 中使用 UDP 所产生的一个额外的益处是,在使用网络容量方面,与 TCP 相比,它的效率大约是三倍,因为 UDP 不是受控协议。此外,在 IPv4 和 IPv6 通讯网络中,用来实现系统 5 的以字节量度的 MSS 容量更大,因为 UDP 报头小于相应的 TCP 报头。

[0060] 虽然下文会描述 TCP 用于 GET 和 POST 的情况,可以理解,这些连接中只有一个使用 TCP,而其它的都使用 UDP。此外,可以理解,GET 和 POST 连接都可以使用 UDP。

[0061] 可以理解,在发送或接收端的数据可以从电路交换数据变为基于 IP 的数据,相应地,从基于 IP 的数据变为电路交换数据,而且这并不背离本发明的方案的范围。

[0062] 在第一实施例中,参见附图 2,执行下述的一系列步骤:

[0063] 步骤 1(S1):到数据连接的一个客户端生成一个唯一的流标识(ID),其中,ID 用于将 GET 和 POST 方法组对到一起,从而用来实现数据连接的服务器知道 GET 和 POST 方法对(pair) 属于同一个客户端。下文将描述 ID 的更多细节。尽管如此,需要理解的是,当唯一流标识(ID) 被用来对发送和接收连接进行组合时,GET 和 POST 方法不对本发明构成限制。尽管流 ID 的主要目的是在服务器把客户端的发送和接收连接绑定在一起,它可以同时还被用于认证和识别客户端。这意味着服务器在继续进行处理之前可以丢弃那些有害的、错误的和 / 或无法识别的连接。这样的功能使得它有可能保护服务器并减小或去除由无法识

别的连接请求和不必要的计算所产生的服务器负荷。换言之,这使得系统保留资源,其益处是,节约能源,并减少在服务器设施中尤其是负载平衡系统中所需要的服务器的数量。

[0064] **步骤 2(S2)**:客户端建立两个到服务器的 TCP/IP 连接,例如在它的默认端“80”,此后,客户端用 CONNET 方法发送一个相联的报头。在操作中,CONNECT 方法将所请求的数据转换成透明的 TCP/IP 隧道,例如,用来像前的那样,通过一个非加密的代理,促成 TLS 和 SSL 加密通讯 (HTTP)。

[0065] 当实现步骤 1 和步骤 2 时,可以采用各种形式的加密,例如,SSL1.0、SSL2.0、SSL3.0、TLS1.0、TLS1.1、TLS1.2 或类似的加密形式。但是,上述的隧道有益地是透明的,以保证不同的“生态系统”之间的安全的通讯。另外,使用能够针对恶意攻击或干扰提供保护的硬件。用来实现本实施例的这样的透明的隧道连接可以防止能监视和分析数据流量的黑客、恶意软件、防病毒软件、防火墙软件或其它设备和 / 或软件侵入通过隧道通讯的数据。

[0066] **步骤 3(S3)**:取决于用于通讯隧道的接收或发送连接,GET 方法或 POST 方法的报头继续被发送和接收。报头包含针对由通讯隧道的提供的给定的通讯会话的必要的信息。

[0067] 此外,报头有益地采用传统形式的数据结构,虽然报头包括下述的参数:

[0068] (i) 流 ID 类的信息,用于受保护的 / 已建立的连接;

[0069] (ii) 传输编码,区块化的 (chunked) 或多部分的 (multi-part) 格式。

[0070] 包含在报头中的信息保证数据的传输和接收作为单独的数据块出现。有益地,数据的最大片段大小 (Maximum Segment Size, MSS) 被按照支持通讯隧道的网络的容量而优化,这时考虑到用于区块化的或多部分报头的字节的数量,使得在发送和接收数据时字节不丢失,由此提供安全可靠的数据交换。

[0071] 这种网络优化的实现是,从网络请求耦合连接到服务器的客户端的最大传输单位 (Maximum Transfer Unit, MTU) 值。由此可行的是,确定一个在通讯网络中最弱的通讯连接,然后为向与最弱连接相关联的客户端设备的传输设置最大片段大小 (MSS),按照最弱的连接能够接纳的速率设置。MSS 值可以由服务器通讯给系统中的其它客户端设备。这样的网络优化的实现采用包括下述方法的步骤:

[0072] **步骤 A**:系统确定一个将服务器耦合到客户端设备的最弱的数据连接;例如,为一个给定的数据连接的 MTU 值是 1500 字节。当该 MTU 值被减去 TCP 报头字节的值,即 40 字节,那么,1460 字节是可用的。该 1460 字节对应于 MSS。

[0073] **步骤 B**:通过采用被确定的最弱连接的 MSS,系统确定针对一个给定会话的 MSS。

[0074] **步骤 C**:可选地,系统采用的 Nagle 算法被禁用,从而防止系统内的拥塞控制,即,通过设置系统的封装 (socket) 中的 TCP_NODELAY 选择来实现,此设置使得 Nagle 算法被禁用。Nagle 算法的这种被禁用是期望的,因为,在相应的数据包被发送之前,Nagle 算法先等待一定数量的字节被加入到发送队列。当 Nagle 算法被禁用,如前所述,系统能够其大小仅由系统来决定的数据包。

[0075] **步骤 4(S4)**:一旦 HTTP 请求报头已经被发送,一个相应的成功响应已经被接收自服务器,然后双工数据接收和发送就开始了。由此,两个与服务器连接被成功建立了,即,一个接收连接和一个发送连接,这些连接被保持在一个开放的状态,直到一个空的数据区块 (data chunk) 或空的多部分数据块 (multi-part data block) 被接受。

[0076] 下面通过 HTTP 码的形式来说明实施例的两个示例。

[0077] 示例 1:提供了 HTTP 码,其在执行的时候用于生成一个简单的被隧道化的在客户端与服务器之间的接收连接,其中,具有 IP 地址 192. 168. 0. 101 的端点 (peer) 连接于一个具有 IP 地址 192. 168. 0. 100 的主机。其中可以发现,在 HTTP 码中 GET 和 POST 方法的使用,以及区块化的 (chunked) 发送码被确定。

```
[0078] <connect>
[0079] <send>CONNECT 192. 168. 0. 100 :80 HTTP/1. 0\r\n
[0080] <send>Mozilla/5. 0 (Windows NT 5. 0)Gurulogic\r\n
[0081] <send>\r\n
[0082] <send>GET/readstream? streamid = 12345&param1 = value1&param2 = value2
HTTP/1. 1\r\n
[0083] <send>Host :192. 168. 0. 100\r\n
[0084] <send>Transfer-Coding :chunked\r\n
[0085] <send>User-Agent :Mozilla/5. 0 (Windows NT 5. 0)
[0086] <send>\r\n
[0087] <recc>HTTP/1. 1 200 OK\r\n
[0088] <reev>5AD\r\n
[0089] <recv>1453 bytes of
[0090] <recv>5AD\r\n
[0091] <recv>1453 bytes of
[0092] ...
[0093] <recv>5AD\r\n
[0094] <recv>1453 bytes of
[0095] <recv>0\r\n
[0096] <disconnect from 192. 168. 0. 100>
```

[0097] 示例 2:提供了 HTTP 码,其在执行的时候用于生成一个简单的被隧道化的在客户端与服务器之间的接收连接,其中,具有 IP 地址 192. 168. 0. 101 的端点 (peer) 连接于一个具有相应的 IP 地址 192. 168. 0. 100 的主机。其中可以发现,在 HTTP 码中 POST 和 CONNECT 方法的使用,以及区块化的发送码被确定。

```
[0098] <connect to 192. 168. 0. 100>
[0099] <send>CONNECT 192. 168. 0. 100 :80 HTTP/1. 0\r\n
[0100] <send>Mozilla/5. 0 (Windows NT 5. 0)Gurulogic\r\n
[0101] <send>\r\n
[0102] <send>POST/writestream? streamid = 12345&param1 = value1&param2 =
value2 HTTP/1. 1\r\n
[0103] <send>Host :192. 168. 0. 100\r\n
[0104] <send>Transfer-Coding :chunked\r\n
[0105] <send>User-Agent :Mozilla/5. 0 (Windows NT 5. 0)
[0106] <scnd>\r\n
[0107] <send>5AD\r\n
```

[0108] <send>1453 bytes of

[0109] <send>5AD\r\n

[0110] <send>1453 bytes of

[0111] ...

[0112] <send>5AD\r\n

[0113] <send>1453 bytes of

[0114] <send>0\r\n

[0115] <recv>HTTP/1.1 200 OK\r\n

[0116] 在这两个示例 1 和 2 中,假设 MSS 是 1460 字节,因此,优化过的区块的实际数据大小是 1453 字节。使用公式 1 所给出的公式来在系统中计算优化的区块的大小:

[0117] $MSS = (\text{区块报头的开头 (beginning of chunk header)} - (\text{区块报头的结尾 (end of chunk header)}))$ 公式 1

[0118] 区块报头的开头包括块数据的实际长度(例如,以十六进制标记)、一个或多个行字符的结尾,它们通常是回车(Carriage Return, CR)和换行(Line Feed, Lf)。区块的结尾类似于行字符的结尾,它使区块结束。

[0119] 参见图 2,可以理解,步骤 3(s3),即,使用 CONNECT 方法建立一个连接隧道,可以被省略,如图 3 所示。当不需要隧道的时候,连接隧道被省略。所以,关于图 2,也可以理解,连接隧道可以只为 GET 连接或 POST 连接而建立,即,在多个节点之间使用非对称隧道通讯设置,可选地,通讯隧道仅用于 GET 或 POST 连接。

[0120] 示例 3:MSS 优化只取决于由一个给定数据区块提供的一个给定的有效载荷,因为在处理过程中,在那个点,相应的 HTTP 区块报头已经被剥离了,数据块的有效载荷是 100%。现在,这样的 MSS 优化主要是基于如下的概念:最大传输单位(MTU)是一个单独的传输突发(burst),因此,那个层所能下载的最大的协议数据单位例如可以是 1500 字节,MSS(最大片段大小)所具有的数据大小等于 MTU 减去协议报头。在根据本发明的实施例的技术方案中,MSS 承载的数据量正好是受关注的网络的最弱连接所能发送的。所以,当采用本发明的技术方案,不出现将数据分割成更小的包的情形,提高了数据发送的速度和可靠性,也导致了更少的冲突与丢包,例如在 WiFi 网络中就是这样。

[0121] MSS 优化的一个示例如下:

[0122]

客户端 1	客户端 1 与客户端 2 之间的 运营商	客户端 2
(网络 MTU 1500 字节)	(最弱网络的 MTU 600 Bytes)	(网络 MTU 1300 字节)

[0123] 开始连接创建 (Commencing Connection Creation):

[0124] 发送 ICMP-ping 命令来测试网络,发现如果 $MTU > 600$,客户端 1 与客户端 2 之间的通讯被阻止。因此,MTU 被设置为 600 字节,这意味着,在考虑到去掉 TCP 报头的 40 字节之后,MSS 是 560 字节。可以理解,UDP 协议中的报头较小的,如果使用 UDP,相应地,有效载荷就是较大的。

[0125] 客户端 1 向客户端 2 发送一个 3000 字节的包,被分成 6 个部分。这样的分割是简

单的,能够按照下述的公式来实现:

[0126] 字节的总数据被除以网络中的最小 MTU 减去开头和结尾的区块头,即 $3000 / (560 - (5 + 2)) = 5.42$ 个包,取整到最近的整数包数,除非其它数据在排队等待发送。

[0127] 包 1 :560 字节被发送,有效载荷是 553 字节,

[0128] 包 2 :560 字节被发送,有效载荷是 553 字节,

[0129] 包 3 :560 字节被发送,有效载荷是 553 字节,

[0130] 包 4 :560 字节被发送,有效载荷是 553 字节,

[0131] 包 5 :560 字节被发送,有效载荷是 553 字节,

[0132] 包 6 :560 字节被发送,有效载荷是 235 字节。

[0133] 如果包被未分割地发送,即一个 3000 字节的包,它有可能被网络途中运营商设备进行分段,这会消耗时间,也有可能引起问题,还有可能有必要重新发送丢失的包,所有这些都导致发送器在发送新包之前要等待,因为接收方不稳定的网络导致延迟。

[0134] 可以在不脱离由所附权利要求限定的本发明的范围的情况下对上面描述的本发明的实施例进行修改。用于描述和限定本发明的例如“包括”、“包含”、“合并”、“由……构成”、“具有”、“是”等表述意在理解为非排他形式,即允许存在未明确描述的项目、组件或元件。单数形式应理解为不排除多数。所附权利要求中的括号中的数字意在辅助对权利要求的理解,而不应被理解为以任何方式限制权利要求所请求的主题。

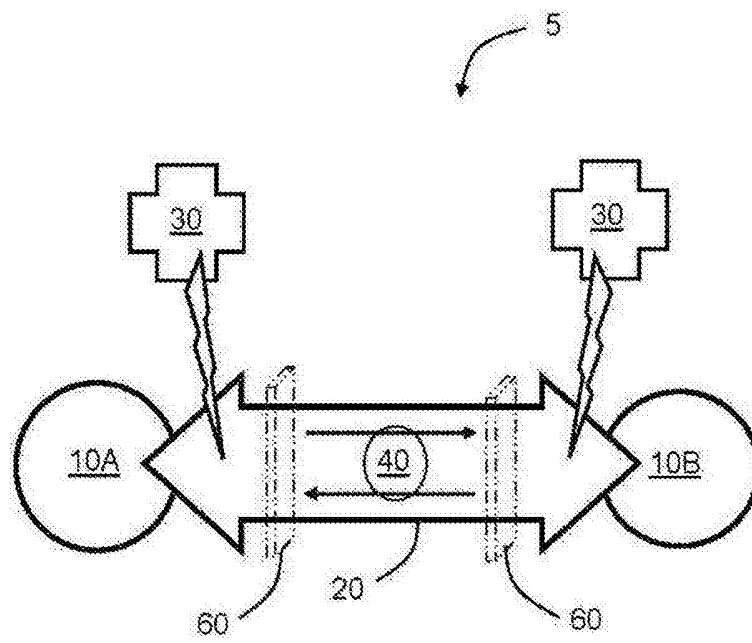


图 1

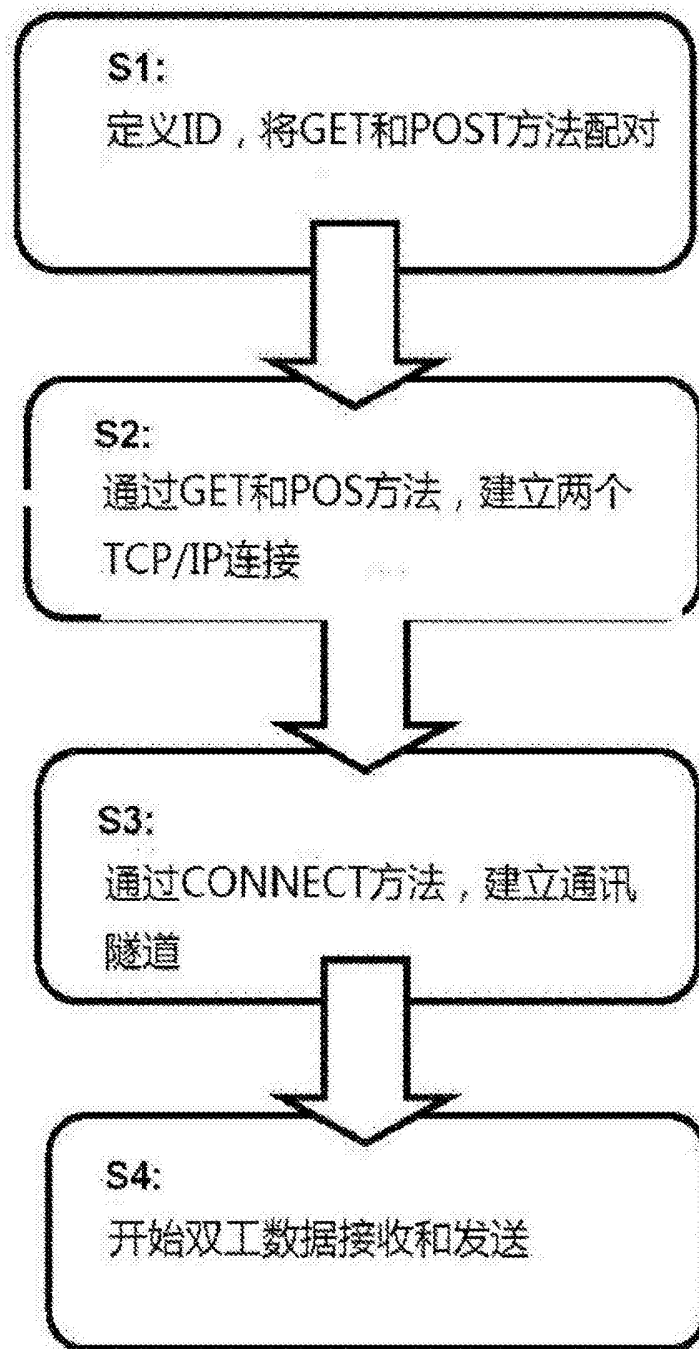


图 2

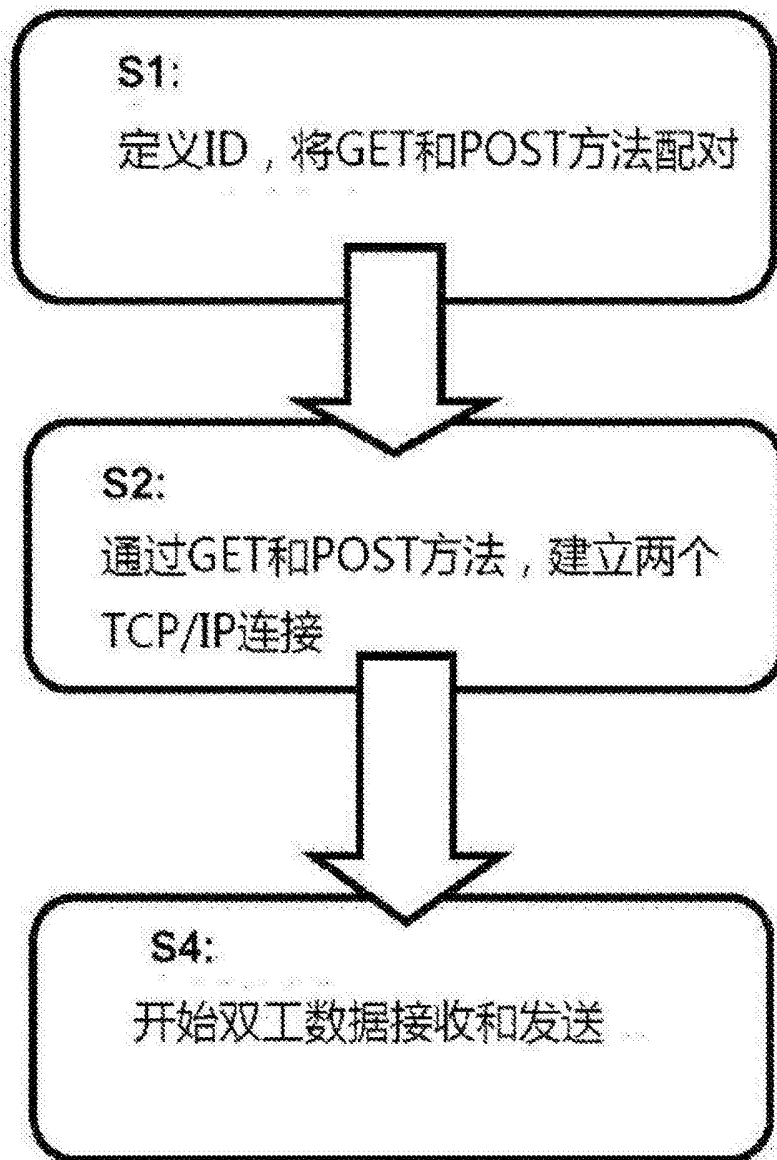


图 3