计算机与通信学报，2017，5，98-111
http://www.scirp.org/journal/jcc
ISSN在线：2327-5227
ISSN印刷：2327-5219

超文本传输协议版本2（HTTP / 2）的隐写术

Biljana Dimitrova, Aleksandra Mileva

马其顿共和国什蒂普GoceDelčev大学计算机科学学院
Email: aleksandra.mileva@ugd.edu.mk

如何引用本文：Dimitrova，B.和Mileva，A.（2017）超文本传输协议版本2（HTTP / 2）的隐写术。Journal of Computer and Communications，5，98-111.
https://doi.org/10.4236/jcc.2017.55008

收到：2017年2月22日
接受：2017年3月28日
发布时间：2017年3月31日

摘 要

网络隐写技术由利用网络协议隐藏数据的不同隐写技术组成。我们介绍了九个新的秘密通道，这些通道利用了新的标准HTTP / 2，无论其传输载体（TLS还是明文TCP）都可以使用。这些隐蔽通道使用具有双重性质的协议功能（例如，无法用两种方式表示填充）；或非强制性的功能（如流优先级和依赖性）；或随机值字段（作为PING帧有效载荷字段）；或没有严格的规则，如何获取某些字段的新值（作为流标识符）。据我们所知，这是关于在HTTP / 2中隐藏数据的第一项研究。此外，我们对可以使用HTTP / 1.x创建的现有隐蔽通道进行了小幅调查，并分析了它们是否可以与HTTP / 2一起使用。

关键字

网络隐秘术，网络安全，信息隐藏，隐蔽通道

1. 介绍

网络隐写术是一种在通信网络中以合法传输方式隐藏秘密数据而不破坏使用的隐藏数据载体的艺术[1]. 通常，它尝试将不同的网络协议部署为运营商，同时尝试隐藏来自网络设备的隐藏数据。因此，现代隐写术的主要研究领域是网络隐写术，也是隐蔽的渠道。隐蔽通道是进程可以利用其以违反系统安全策略的方式传输信息的任何通信通道[2]. 可以非法使用基于网络的隐蔽渠道来协调分布式拒绝服务攻击或恶意软件的传播（例如蠕虫W32。Morto使用DNS记录与其命令和控制服务器进行通信），以进行恐怖分子之间的秘密通信

和罪犯，工业间谍活动，但在法律上也是如此，以规避某些国家/地区使用互联网的限制（例如，Infranet[3]），安全的网络管理通讯[4],版权保护等

网络隐写方法的一种可能分类如下：[5],三个大类分开的地方：

- 修改协议数据单元（PDU）的方法，包括带有来自协议头或/和协议有效载荷的协议控制信息的字段。
- 通过PDU重新排序，故意丢失，使用数据包间延迟，修改时间戳等方法来修改PDU流结构的方法。
- 混合方法，涉及前两种方法的组合。

网络协议作为秘密数据的载体的最佳选择是最流行和使用最广泛的协议，因此，超文本传输协议（HTTP）成为自然选择。尽管在过去的20年中Web发生了巨大的发展，但从1997年的RFC 2068的标准化和1999年的RFC 2616的改进以来，HTTP / 1.1一直保持到2014年。应对新的Web技术时使用HTTP / 1.1。2014年6月，IETF的HTTPbis工作组发布了六个部分的更新规范（RFC 7230-5），并于2015年5月在RFC 7540中发布了新的主要版本HTTP /2。HTTP/ 2是一个二进制协议，它带来了与以前的版本相比，有很多改进和好处，例如：

a) 多路复用和并发：可以在
与单独的流相同的TCP连接，并且它们的响应可能在相同的流中乱序接收。此功能消除了客户端和服务器之间需要多个TCP连接的需要。

b) 服务器推送：如果服务器知道需要一些资源，并且稍后会在给定的网站上请求，则服务器可以发送这些资源而无需请求，客户端将缓存这些资源直到稍后；

c) 标头压缩：使用特殊的帧和压缩可以大大减少HTTP标头的大小；

d) 流的依赖关系和优先级：客户端可以向服务器指示哪些流比其他流更重要，并且需要首先交付。

HTTP / 2是考虑到安全性而构建的TCP / IP协议套件的最新成员，但在他的设计中仍然存在许多双重性，可用于构建隐蔽通道。这些二重性来自于某些功能可以通过多种方式获得的可能性，或者该功能的部署不是强制性的。有趣的是，HTTP / 2的设计者学习了如何通过将标头字段设置为零来处理使用带有随机填充或保留字段的标头字段的隐式通道，或者如何通过对标头的帧进行排序来处理使用PDU重新排序的隐式通道。固然重要，但他们仍然留下了许多方法来建立隐秘渠道。在本文中，我们介绍了几个可以使用HTTP / 2创建的秘密通道。主要的HTTP / 2功能

第2节介绍了这些概念和概念。第3节介绍了可以使用HTTP / 1.x创建的不同的现有隐蔽通道，此外，还说明了它们是否适用于新版本。第4节主要介绍了HTTP / 2中的九个新隐蔽通道组，无论其传输载体（TLS还是明文TCP）都可以使用。

2. HTTP / 2如何工作？

HTTP / 2保留了与HTTP / 1.1相同的语义，并且未对其基本概念和功能进行任何更改。通过更改语法传达这些语义的方式，它为HTTP / 1.1请求/响应提供了一种优化的传输机制。两种协议之间的主要区别在于HTTP / 2是二进制协议。HTTP / 2连接是客户端和服务器之间的TCP连接，由三个元素组成：

- 流：在端点之间传送消息的双向流；
- 消息：由一个或多个帧组成的逻辑HTTP消息；
- 帧：承载特定类型数据的最小通信单元。

HTTP / 2连接可以承载许多独立的双向流，其中许多流可以并行交换消息。每个消息都被分成较小的帧，这些帧被发送到端点。经由HTTP / 2传输的每个帧都与一个流相关联，并且所有流都分配给唯一且不能被其他流使用的流标识符。有10种不同类型的帧，包括固定的9字节首部和可变长度的有效负载，具体取决于帧类型。每个帧都由标题和有效负载组成。每个帧头还包含以下字段：24位长度（帧有效载荷），8位类型（帧），8位标志，1位保留字段（R）和31位流标识符。这是通过插头和其组件之间的连接线呈现的图1。

HTTP / 2使用新的压缩程序HPACK压缩报头元数据，以减少开销，并提高了性能。HTTP / 2中使用的帧类型为：数据，标头，优先级，第一流，设置，推送承诺，Ping，GOAWAY，WINDOWS更新和持续。

RST STREAM和GOAWAY帧包含错误代码，用于指示与任何特定流或整个连接有关的错误。从一个流发送帧的顺序非常重要，因为接收方在接收帧的顺序内进行处理。

客户端通过首先向服务器发送请求以确定服务器是否支持HTTP / 2来启动HTTP / 2连接。对于“ http”和“ https” URI，此过程有所不同，其中使用不同的标识符进行协议的标识。字符串“ h2”用于标识在TLS上使用HTTP / 2，字符串“ h2c”用于在明文TCP上用于HTTP / 2。当客户端想要使用常规的非加密通道时，它必须使用HTTP升级机制来协商协议。客户端首先发送HTTP / 1.1请求，该请求包含带有“ h2c”令牌和HTTP2-Setting标头字段的Upgrade标头字段（图2).

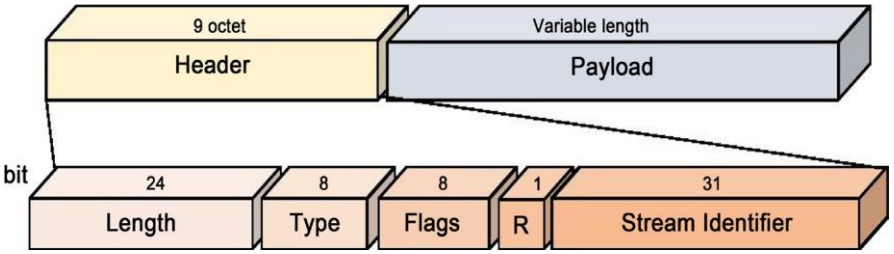

图1. 框架布局。

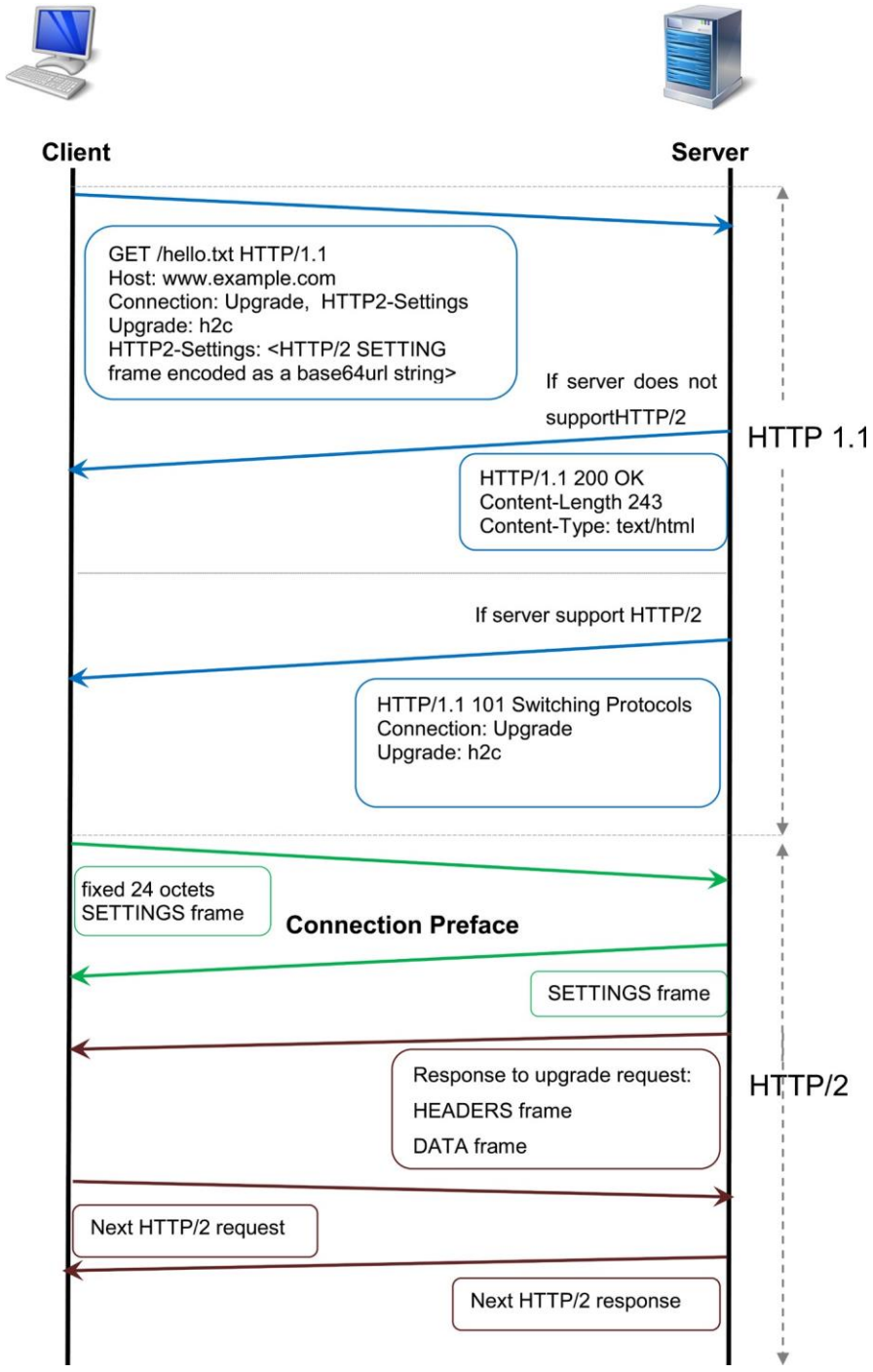

图2. 从HTTP / 1.1升级到HTTP / 2。

当服务器不支持HTTP / 2时，它将以HTTP 1.1响应进行响应，其中不存在Upgrade标头字段。相反，通过发送101交换协议响应，服务器确认升级并开始发送HTTP / 2帧。作为对协议的最终确认，客户端和服务器必须通过发送不同的连接序言来建立连接的设置。客户端连接序言以固定的24个八位位组序列开头，后跟一个可能为空的SETTINGS框架，服务器使用连接序言来回答，该序言由一个可能为空的SETTINGS框架组成。建立连接后，客户端和服务器交换任何类型的帧。通过发送GOAWAY帧，端点可以正常关闭连接，并且在从先前建立的流中接收到所有帧之后，将不再发送数据。为了关闭任何特定的流，任何一个端点都可以发送：RST STREAM帧或包含END STREAM标志的帧。

3. HTTP / 2早期版本中的隐蔽通道

协议描述中HTTP的一个有趣特征是对URI字符串，HTTP标头或HTTP消息正文的大小没有限制。通常，在不同的实现中会引入不同的限制。例如，Apache服务器接受HTTP标头，最大大小为8 KB，而IIS则最大为8 KB或16 KB，具体取决于版本。

通过设计，HTTP / 2中抑制了许多现有的HTTP / 1.x隐写方法。例如，在[6] [7] 使用HTTP / 1.x以与单个空格字符相同的方式处理标头中存在的任何数量的后续线性空格字符（可选换行[CLRF]，空格[SP]和制表符[HT]）的事实（例如[HT]可以是二进制1，[SP]可以是二进制0）。在HTTP / 2中，此要求是必须的，即必须将具有无效标头名称和标头字段值中不允许的字符的请求和响应视为格式错误。中间节点不得转发格式错误的请求或响应。此外，由于标头名称在HTTP / 1.x中不区分大小写，因此可以为隐式通道的标头字段值使用不同的大小写[8]. 这不能在HTTP / 2中完成，因为标头字段名称必须先转换为小写，然后才能在HTTP / 2中进行编码。Dyatlov和Castro提出的其他三种方法[8] 使用HTTP头字段重新排序的代码，可能的存在/不存在（例如，Accept-Encoding头字段）以及HTTP消息正文也可以扩展为HTTP / 2。

阿尔曼[9] 表示由于HTTP协议CONNECT方法的弱点，可以通过HTTP代理服务器进行任意连接。这些HTTP隧道不仅限于端口80和443，而且只要客户端在数据流周围扭曲适当的HTTP CONNECT标头，它们就可以在任何TCP端口上传递任何出站流量。有许多工具可以通过HTTP隧道传输不同的协议，例如Corkscrew[10]，它通过HTTP代理建立SSH隧道。这些是跨协议攻击的示例，当攻击者使客户端通过以下方式提交请求时

向服务器了解一个不同协议的一个协议，并且该请求在第二个协议中也有效。HTTP / 2的纯文本版本不能提供足够的保护，以防止此类攻击，但在RFC 7540中指出：“可以将带有HTTP / 2的ALPN标识符的TLS握手完成视为可以防止跨协议攻击”。

鲍尔[11] 建议使用一种协议“ Mute Posthorn”，该协议允许通过利用常规用户的网络浏览活动来创建匿名覆盖网络。该协议使用五种HTTP / HTML机制：重定向，Cookie，Referer标头，HTML元素和活动内容。

范·霍伦贝克[12] 实施了Wondjina工具，该工具使用HTTP ETag和If-None-Match标头字段创建双向秘密通道，该通道可让客户端验证其本地缓存副本是否仍为当前副本。提供特定文档后，Web服务器将被允许包含ETag头字段，该字段包含描述页面的字符串，而无需特殊说明其外观。首次检索时，客户端同时缓存页面及其ETag。If-None-Match主要用于条件GET请求中，以最小的事务开销实现对缓存信息的有效更新。当客户端希望更新一个或多个具有实体标签的已存储响应时，客户端在发出GET请求时应生成If-None-Match标头字段，其中包含当前缓存的ETag列表。作者还建议使用Content-MD5标头字段以一种方式为每个HTTP消息发送128位秘密数据。但是，此标头字段已从2014（RFC 7231）的协议规范中删除。

邓肯和马丁娜[13] 建议在HTTP响应中调制基于日期的字段（例如Date和Last-Modified）的最低有效位，并使用Content-Location标头字段，该字段旨在为当前正在访问的资源提供备用URL。Eßer和Freiling[14] 建议使用HTTP的秘密定时通道，其中Web服务器通过延迟响应（二进制1）或立即响应（二进制0）将秘密数据发送到客户端。

红外线[3] 是一个框架，该框架在某些国家/地区使用HTTP中的秘密通道来规避Internet中的审查。Infranet的Web服务器收到对隐秘网页的隐式请求，这些隐秘网页编码为对无害网页的HTTP请求序列，并使用隐写术将其隐藏在无害图像中的内容返回。

Graniszewski等提供的另一个HTTP 1.1及更高版本的秘密渠道[15]，使用HTTP标头中的Trailer字段隐藏数据。Trailer响应头字段允许发件人在分块的消息末尾包括其他字段，以提供在消息正文发送时可能动态生成的元数据，例如消息完整性检查，数字签名或后处理状态。

4. HTTP / 2中的隐蔽通道

有几种方法可以在HTTP / 2中创建新的秘密通道。为此，通常我们使用具有双重性质的协议功能，即相同的

可以通过多种方式获得功能，该功能不是强制性功能，存在随机值字段，或者没有严格的规则如何为某些字段获取新值。

4.1. 隐蔽通道使用填充

HTTP / 2帧中的三个帧，DATA，HEADERS和PUSH PROMISE帧使用填充作为安全性功能来掩盖消息的大小。提供它是为了减轻HTTP内的特定攻击（如BREACH），其中压缩的内容包括攻击者控制的明文和秘密数据。缓解这些攻击的另一种方法是禁用或限制压缩。使用时，填充八位位组必须设置为零，以防止其他攻击。使用填充时，第三个标志PADDED（0×8）设置为1，并且在帧有效载荷的开头有一个8位字段Pad Length，其中包含帧填充的长度（位置在末尾）以八位字节为单位）。仅当将PADDED标志设置为1时，“ Pad Length”和“ Padding”字段才存在。当不使用填充时，有两种表示效果相同：

- PADDED标志设置为0，并且
- 填充标志设置为1，填充长度字段设置为0。

这两种表示形式可以用作二进制零和一（图3）.在RFC 7540中，可以发现：“中间设备应该保留数据帧的填充，但是可以丢弃HEADERS和PUSH PROMISE帧的填充。中间设备更改帧的填充量的有效原因是为了改善对填充的保护。提供”。因此，对于DATA帧，没有中介会更改填充，并且可以将其用作客户端和服务器之间每个DATA帧的双向一位隐蔽通道。

4.2. 使用流标识符的隐蔽通道

流标识符由无符号的31位整数表示。在升级到HTTP / 2之前，值0×0为连接控制消息保留，而值0×1为HTTP / 1.1请求保留。奇数编号的流标识符用于客户端发起的流，偶数编号的流标识符用于服务器发起的流。任何新流的流标识符必须大于所有已打开或保留的流标识符。

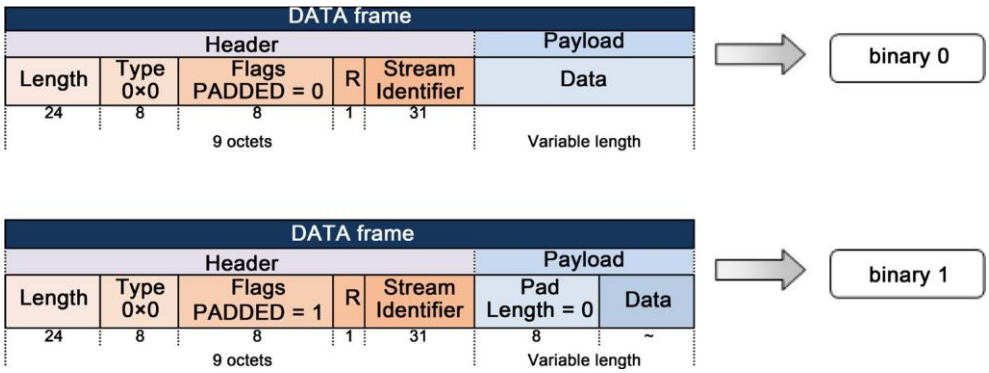

图3. 使用填充用隐式通道表示二进制0和1。

端点流。流标识符无法重复使用，因此，它们会被长期存在的连接耗尽，从而导致建立新的连接。流可以处于七个不同状态之一：“空闲”，“保留（本地）”，“保留（远程）”，“打开”，“半关闭（远程）”，“半关闭（本地）”和“关闭”。

假设MAX CSI是在给定时刻客户端启动的流的最大使用流标识符，而MAX SSI是服务器同时启动流的最大使用的流标识符。可以通过以下方式在客户端和服务器之间创建一个双向秘密通道（图4）:

- 如果客户端要向服务器发送二进制1，它将启动一个新流，该流的流标识符为MAX CSI + 2，对于二进制0，流的标识符为流标识符MAX CSI + 4。
- 如果服务器要向客户端发送二进制1，它将启动一个新流，该流的流标识符为MAX SSI + 2，对于二进制0，流的标识符为MAX SSI + 4。

这样，对于长寿命连接，一侧可以在$2^{29}-1$（对于所有二进制位0）和$2^{30}-1$（对于所有二进制位1）之间传输最大值。任何异常。使用SETTINGS帧内的SETTINGS MAX CONCURRENT STREAMS参数，可以在任一通信站点引入并发活动流数量的限制。最初，此参数的值没有限制，建议不要小于100。由于活动流仅出现在“打开”，“半关闭（远程）”或“半关闭（本地）”中的流状态。

4.3. 使用PING框架的隐蔽通道

PING帧可以从客户端和服务器的两端发送，它们仅与流标识符0×0关联。它们用于确定空闲连接是否仍然有效，并用于测量最小往返行程发件人的时间。不带ACK标志的PING帧必须通过发送PING帧作为应答，并设置ACK位来进行确认，

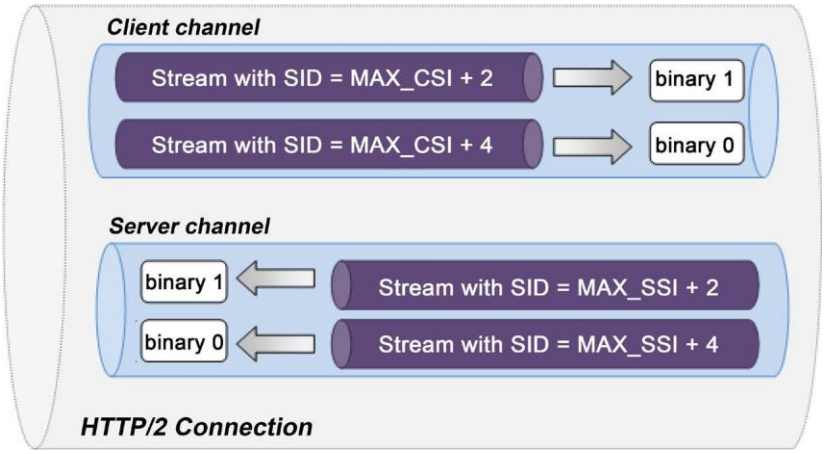

图4. 使用流标识符表示具有隐蔽通道的二进制0和1。

具有相同的有效负载并具有更高的优先级。PING帧的有效载荷可以是任何64位数字。没有限制，因此可以通过每个PING帧发送64位来创建隐蔽通道。

4.4. 使用流优先级和依存关系的隐蔽通道

HTTP / 2使用具有或不具有分配优先级的流，从而使它们成为或不依赖于其他流的竞争。当发送容量有限时，发送方将根据优先级选择要发送帧的流。另外，在每个依赖项上分配一个介于1和256之间的相对权重。如果多个流依赖于同一流，则该权重将用于确定分配给它们的可用资源的相对比例。所有流最初都非排他性地依赖于流0×0，而推入流最初依赖于它们的关联流。两种情况下的默认权重均为16。

通过一个HEADERS帧打开流，该帧还可以为流分配优先级。此后，可以通过PRIORITY帧随时更改给定流的优先级。

对于新的秘密渠道，关于优先级和依赖关系的其他详细信息并不重要。唯一重要的是，优先级排序过程对于其他通信终结点而言只是建议性的，而不是强制性的-无法保证一定会实现。因此，我们可以使用它来创建新的秘密通道。

如果HEADERS帧的PRIORITY（0×20）标志设置为1，则在帧有效载荷中，将显示Exclusive Flag（E），Stream Dependency和Weight字段。使新的流依赖于先前创建的流（也可以考虑其他先前的流），则每个HEADERS帧可以使用9位，E标志使用1位，Weight字段使用8位双向隐蔽通道，而不会引起可疑情况。因此，如果一个HTTP / 2连接在其整个生命周期中的一个方向上有n个流，则每个HTTP / 2连接可以在一个方向上发送9n位。

PRIORITY框架具有咨询作用，并指定流的发送者建议的优先级。它可以在任何流状态和任何时间发送，但在组成单个标头块的连续帧之间除外。同样，我们可以使用每个优先级帧9位，E标志1位和Weight字段8位来创建隐蔽通道。在一个HTTP / 2连接期间发送的PRIORITY帧数本身不应是异常。

4.5. 使用不同数量的特定种类帧的隐蔽通道

一个HTTP请求包括：

- 一个HEADERS帧，后跟零个或多个CONTINUATION帧，其中包含标题块；
- 零个或多个包含有效载荷主体的数据帧；
- （可选）一个HEADERS帧，后跟零个或多个CONTINUATION

包含尾部的框架。

HTTP响应的结构是相似的，在开头有额外的零个或多个HEADERS帧，每个帧之后是零个或多个CONTINUATION帧，其中包含信息性（1xx）HTTP响应的标头块。此外，当使用PUSH PROMISE帧时，标头块在该帧内开始，并且其后可以是零个或多个CONTINUATION帧。每个标题块都作为连续的帧序列发送，没有任何其他类型的交错帧或来自任何其他流的交错帧，并且保留了帧顺序。可以部署用于秘密通道创建的一个重要属性是DATA和CONTINUATION帧是可变长度的八位位组序列，并且可以以不同数量发送。因此，我们可以使用以下方法创建一个秘密通道：

- DATA帧的奇数个为二进制1，并且
- 偶数个DATA帧为二进制0。

通过这种方式，我们可以在每个方向上每个流最多发送一位，或者在每个方向上由n个流组成的HTTP / 2连接最多发送n位。同样，我们可以使用以下方法创建另一个秘密通道：

- CONTINUATION帧的奇数为二进制1，并且
- 连续帧的偶数为二进制0。

这样，如果每个流之后有k个不同的HEADERS和PUSH PROMISE帧，且后面带有CONTINUATION帧，则每个流最多可以将两个位发送到服务器，最多k个位发送给客户端。或者，每个HTTP / 2连接（由n个流组成）对服务器最多2n位，对客户端最多kn位。

4.6. 使用Cookie标头字段的隐式通道

为了获得更好的压缩效率，与HTTP / 1.x中的规则不同，HTTP / 2可以将cookie对从一个Cookie头字段中分离为多个Cookie头字段，每个cookie头字段中都包含一个或多个cookie对。因此，由于这种双重性，我们可以通过以下方式创建从客户端（支持HTTP / 2）到服务器的单向隐蔽通道：

- 当前只有一个Cookie头字段为二进制1，并且
- 当前有多个Cookie头字段为二进制0。

通常，为此，至少必须有两个cookie对。这样，每个HTTP / 2连接（由n个流组成），我们最多可以将n位发送到服务器。

4.7. 使用SETTINGS帧的隐蔽通道

SETTINGS帧用于连接前言阶段，用于由双方配置不同的特定于连接的参数，但是它们也可以在HTTP / 2连接期间的任何时间发送。SETTINGS帧中的参数值将替换这些参数的任何现有值，并由ACK位设置为1的空SETTINGS帧进行确认。SETTINGS帧的有效负载包含零个或多个参数，这些参数由无符号定义

16位标识符字段和无符号32位值字段。参数按照它们在有效负载中出现的顺序进行处理。协议规范中有六个定义的参数：

- SETTINGS HEADER TABLE SIZE（0×1）-用于解码头块的头压缩表的最大大小（以八位字节为单位）。初始值为4096个八位位组。
- 设置启用推（0×2）-启用/禁用服务器推送。初始值为1，表示允许服务器推送。
- 设置最大并发流（0×3）-发送方允许的最大并发流数。最初，此值没有限制。
- 设置初始窗口大小（0×4）-发送方的初始窗口大小，以八位字节为单位，用于流级别的流量控制。初始值为2^{16}-1个八位位组。
- SETTINGS MAX FRAME SIZE（发送方愿意接收的最大帧有效载荷的大小（0×5），以八位字节为单位）。初始值为2^{14}个八位位组。
- 设置最大报头列表大小（0×6）-发送方准备接受的报头列表的最大大小（八位字节）。最初，此值没有限制。

我们可以使用这些参数的不同值来定义隐蔽通道。由于他的布尔性质和对连接的影响，我们可以排除SETTINGS ENABLE PUSH参数。对于每个其他参数，我们可以定义

- 给定参数的偶数字段为二进制0，并且
- 给定参数的奇数值字段为二进制1。

此外，如果需要，我们可以定义一个更改这些值的间隔，以使协议正常工作。因此，我们可以在一个方向上每个SETTINGS帧发送5位。

4.8. 使用流量控制的隐蔽通道

可以在每个单独的流或整个连接上进行HTTP / 2中的流控制。这是逐跳进行的，而不是在整个端到端路径上进行的。接收者使用基于信用的方案将准备接收的数据量发送给发送者，发送者必须遵守这些限制。可以通过在连接序言的SETTINGS帧中包含SETTINGS INITIAL WINDOW SIZE的值来调整新流的初始窗口大小。之后，可以通过发送WINDOW UPDATE或SETTINGS帧随时更改窗口大小。只能使用WINDOW UPDATE框架更改连接流控制窗口。流和连接流控制窗口的默认值为65,535个八位位组。流控制的对象只是DATA帧。

WINDOW UPDATE帧的有效负载包含一个保留位字段和Windows Size Increment字段，该字段是无符号的31位整数（不允许使用0值），该值指示发送者除了现有流以外还可以传输的八位字节数。控制窗口。没有严格定义的方法

端点如何或何时通告帧的大小，因此可用于制作新的隐蔽信道。此外，将为不同的流发送单独的WINDOW UPDATE帧，并且对于可以发送这些帧的流状态没有限制。可以使用以下方法完成两个相邻跃点之间的一个双向隐蔽信道：

- Windows大小增量字段的偶数值为二进制0，并且
- Windows大小增量字段的奇数值为二进制1。

这样，如果在一个HTTP / 2连接中有k个流，则可以通过在每个时间随时为每个流发送一个单独的WINDOW UPDATE帧来在一个方向上发送k位，而不会引起任何异常。

4.9. 使用HPACK的隐蔽通道

HTTP / 2附带了一个新的压缩器，它消除了报头字段HPACK中的冗余。它威胁标头字段为名称-值对的有序集合，被视为八位字节序列，并可能重复。一个头字段可以使用静态或动态表编码为索引值（引用），也可以通过指定其名称和值将其表示为文字值。标头字段值始终以文字形式表示。另外，可以直接对文字值进行编码，也可以使用静态霍夫曼代码进行编码。RFC 7541仅描述了HPACK解码器应如何工作。HPACK解码器顺序处理标头块，以重建原始标头列表。

字符串文字表示形式具有三个字段：一位标志H（指示是否使用霍夫曼编码），字符串长度字段（其是用于对字符串文字进行编码的八位字节数）和字符串数据字段（具有字符串字面量。因此，我们可以使用以下方法创建一个秘密通道：

- 没有编码的字符串文字（字段H = 0）为二进制0，并且
- 编码为（字段H = 1）的字符串文字为二进制1。

如果标头块中有k个字符串文字，则每个标头块可以发送k位。对于每个TCP流，HTTP请求中最多可以有两个标头块（在开头和结尾），而HTTP响应中最多可以有两个标头块（在开头和结尾）。

文字标头字段有三种不同的表示形式：带增量索引（以二进制序列01开头），不带索引（以二进制序列0000开始）和从不索引（以二进制序列0001开始）。如果动态表中不存在字段名称，则具有增量索引表示形式的文字标头字段会在动态表中添加新条目，并且没有索引表示形式的文字标头字段不会更改动态表。文字标头字段从不索引表示形式不会改变动态表，但是所有中介程序也必须使用相同的表示形式来对该标头字段进行编码。我们可以使用以下方法创建另一个秘密通道：

- 带增量索引或不带索引的文字头字段

tation为二进制0，并且

- 文字标头字段从不索引表示形式为二进制1。

如果标头块中有k个文字标头字段，则每个标头块可以发送k位。

5. 结论

像许多其他网络协议一样，HTTP / 2倾向于在其中隐藏数据。此外，它属于在接下来的几年中将被大量使用的网络协议，并且其流量不会引起任何怀疑。因此，重要的是找出可能的隐藏数据的方法，并设法减轻它们。本文讨论了第一部分，而其他人则试图找到一种缓解存在的隐秘渠道的解决方案。

此外，参与协议标准化过程的人员可以研究消除这些隐蔽渠道的可能性。例如，协议设计中的一种最佳实践应该是消除具有双重性质的功能，即不要以任何一种以上的方式完成任何事情。

这些隐蔽渠道的实现以及它们在实验室或真实环境中的测试将留作未来的工作。

参考文献

[1] Lubacz, J., Mazurczyk, W.和Szczypiorski, K. (2014) 网络隐写术原理和概述。IEEE通信杂志，第52期，第225-229页。 https://doi.org/10.1109/MCOM.2014.6815916

[2] 国防部 (1985) 国防部可信计算机系统评估标准。技术报告DoD 5200.28-STD。取代CSC-STD-001-83。http://csrc.nist.gov/publications/history/dod85.pdf

[3] Feamster, N., Balazinska, M., Harfst, G., Balakrishnan, H.和Karger, D. (2002) Infranet：规避网络审查和监视。第11届USENIX安全研讨会论文集，2002年8月8日至12日，旧金山，247-262。

[4] Forte, DV (2005) SecSyslog：一种基于隐蔽通道的安全日志记录方法。2005年11月7日至9日在台北举行的首届数字法证学系统方法国际研讨会 (SADFE 2005)，会议记录，248-263。https://doi.org/10.1109/SADFE.2005.21

[5] Mazurczyk, W., Lubacz, J.和Szczypiorski, K. (2008) 在VoIP中隐藏数据。第26届陆军科学会议论文集 (ASC 2008)，奥兰多，2008年12月1-4日。

[6] Kwecka, Z. (2006) 应用层隐蔽通道分析和检测。纳皮尔大学爱丁堡技术报告。https://pdfs.semanticscholar.org/f740/ca7afcb75d9c90c50894396dcfc08f824a91.pdf

[7] Heilman, S., Williams, J.和Johnson, D. (2016) HTTP UserAgents中的隐蔽通道。2016年6月8日至9日，奥尔巴尼，第11届信息保证年度研讨会 (ASIA 16) 会议录，第68-73页。

[8] Dyatlov, A.和Castro, S. (2003) 利用网络访问控制系统授权的数据流进行任意数据传输：HTTP协议上的隧道和隐蔽通道。美国灰色世界。http://gray-world.net/projects/papers/covert_paper.txt

[9] Alman，D.（2003）HTTP隧道虽然代理。SANS Institude. https://www.sans.org/reading-room/whitepapers/covert/http-tunnels-proxies-1202

[10] Padgett，P.（2001）开瓶器。https://www.mankier.com/1/corkscrew

[11] 鲍尔（Bauer，M.）（2003）HTTP中的新隐蔽通道：将不知情的Web浏览器添加到匿名集。隐私电子社会研讨会（WPES 2003）的会议记录，华盛顿特区，2003年10月30日，第72-78页。https://doi.org/10.1145/1005140.1005152

[12] Van Horenbeeck，M.（2006）网络上的欺骗：对隐蔽渠道的不同思考。澳大利亚信息战和安全会议记录，西澳大利亚州珀斯，2006年12月4-5日，第174-184页。

[13] Duncan，R.和Martina，JE（2010）使用Web协议的隐秘消息广播。Simguio Brasilerio de Seguranca的会议记录（SBSeg 2010），巴西福塔莱萨，2010年10月11日至15日，第61-70页。

[14] Eßer，HG和Freiling，FC（2005年），《HTTP协议》。曼海姆大学技术报告TR-2005-10。https://ub-madoc.bib.uni-mannheim.de/1136/1/tr_2005_10.pdf

[15] Graniszewski，W.，Krupski，J.和Szczypiorski，K.（2016）HTTP协议的隐秘通道。SPIE 10031，《光子学在天文学，通信，工业和高能物理实验2016中的应用》，2016年9月28日，100314Z。