网络工作组
请求评论：3261
过时：2543
类别：标准轨道

J. Rosenberg
dynamicsoft
H. Schulzrinne
哥伦比亚大学
G. Camarillo
爱立信
A. Johnston
WorldCom
J. Peterson 佩顿·J
Neustar
R. Sparks
dynamicsoft
M. Handley 汉德利
ICIR
E. Schooler
AT&T
六月2002

SIP: 会话初始化协议

状态说明

此文档指定了一个用于互联网标准跟踪的协议。
互联网社区，并讨论和提出建议
改进。请参阅“ 互联网 ”当前版本。
官方协议标准（STD 1）的标准化状态
此协议的状态。本备忘录的分发不受限制。

版权声明

摘要

本文件描述了会话初始化协议（SIP），一个
应用层控制（信令）协议用于创建
修改和终止包含一个或多个参与者的会话。
这些会话包括互联网电话通话、多媒体
分布和多媒体会议。

SIP邀请用于创建会话，携带会话描述
允许参与者就一组兼容的媒体类型达成一致。
SIP利用称为代理服务器的元素来帮助路由请求
到用户的当前位置，验证和授权用户
服务，实现提供商呼叫路由策略，并提供
功能提供给用户。SIP 还提供了一种注册功能，
允许用户上传他们当前的地理位置以供代理使用
服务器。SIP 在多个不同的传输协议之上运行。

Table of Contents

1 简介 8
2 SIP功能概述 9
3 术语 10
4 操作概述 10
5 协议结构 18
6 定义 20
7 SIP消息 26
7.1 请求 27
7.2 响应 28
7.3 报头字段 29
7.3.1 报头字段格式 30
7.3.2 报头字段分类 32
7.3.3 紧凑形式 32
7.4 机体 33
7.4.1 消息体类型 33
7.4.2 消息体长度 33
7.5 框架SIP消息 34
8 通用用户代理行为 34
8.1 UAC 行为 35
8.1.1 生成请求 35
8.1.1.1 请求URI 35
8.1.1.2 旨在 36
8.1.1.3 从 37
8.1.1.4 通话标识符 37
8.1.1.5 CSeq 38
8.1.1.6 最大转发数 38
8.1.1.7 通过 39
8.1.1.8 联系方式 40
8.1.1.9 支持和需求 40
8.1.1.10 额外消息组件 41
8.1.2 发送请求 41
8.1.3 处理响应 42
8.1.3.1 事务层错误 42
8.1.3.2 不可识别的响应 42
8.1.3.3 通孔 43
8.1.3.4 处理3xx响应 43
8.1.3.5 处理4xx响应 45
8.2 UAS行为 46
8.2.1 方法检查 46
8.2.2 报头检查 46
8.2.2.1 请求与Request-URI 46
8.2.2.2 合并请求 47
8.2.2.3 需要..... 47
8.2.3 内容处理 48
8.2.4 应用扩展 49
8.2.5 处理请求 49

8.2.6 生成响应 49
8.2.6.1 发送临时响应 49
8.2.6.2 标题和标签 50
8.2.7 无状态UAS行为 50
8.3 重定向服务器 51
9 取消请求 53
9.1 客户行为 53
9.2 服务器行为 55
10 注册 56
10.1 概述 56
10.2 构建注册请求 57
10.2.1 添加绑定 59
10.2.1.1 设置联系地址的过期间隔 60
10.2.1.2 联系地址的偏好 61
10.2.2 移除绑定 61
10.2.3 获取绑定 61
10.2.4 刷新绑定 61
10.2.5 设置内部时钟 62
10.2.6 发现注册机构 62
10.2.7 传输请求 62
10.2.8 错误响应 63
10.3 处理注册请求 63
11 查询功能 66
11.1 选项请求的构建 67
11.2 选项请求的处理 68
12 对话 69
12.1 对话的创建 70
12.1.1 UAS行为 70
12.1.2 UAC 行为 71
12.2 对话内的请求 72
12.2.1 UAC 行为 73
12.2.1.1 生成请求 73
12.2.1.2 处理响应 75
12.2.2 UAS 行为 76
12.3 对话终止 77
13 启动会话 77
13.1 概述 77
13.2 UAC 处理 78
13.2.1 创建初始 INVITE 78
13.2.2 处理 INVITE 响应 81
13.2.2.1 1xx 响应 81
13.2.2.2 3xx 响应 81
13.2.2.3 4xx、5xx 和 6xx 响应 81
13.2.2.4 2xx 响应 82
13.3 UAS处理 83
13.3.1 INVITE 处理 83
13.3.1.1 进度 84
13.3.1.2 INVITE 被重定向 84

13.3.1.3 INVITE 被拒绝 85
13.3.1.4 INVITE 被接受 85
14 修改现有会话 86
14.1 UAC 行为 86
14.2 UAS行为 88
15 会话终止 89
15.1 使用 BYE 请求终止会话 90
15.1.1 UAC 行为 90
15.1.2 UAS 行为 91
16 代理行为 91
16.1 概述 91
16.2 有状态代理 92
16.3 请求验证 94
16.4 路由信息预处理 96
16.5 确定请求目标 97
16.6 请求转发 99
16.7 响应处理 107
16.8 处理计时器 C 114
16.9 处理传输错误 115
16.10 取消处理 115
16.11 无状态代理 116
16.12 代理路由处理摘要 118
16.12.1 示例 118
16.12.1.1 基本SIP梯形 118
16.12.1.2 遍历严格路由代理 120
16.12.1.3 重写Record-Route报头字段值 121
17 交易 122
17.1 客户交易 124
17.1.1 INVITE 客户端事务 125
17.1.1.1 INVITE 事务概述 125
17.1.1.2 正式描述 125
17.1.1.3 ACK请求的构建 129
17.1.2 非 INVITE 客户端事务 130
17.1.2.1 非INVITE事务概述 130
17.1.2.2 正式描述 131
17.1.3 将客户端交易与响应匹配 132
17.1.4 处理传输错误 133
17.2 服务器事务 134
17.2.1 INVITE 服务器事务 134
17.2.2 非 INVITE 服务器事务 137
17.2.3 将匹配请求与服务器事务对应 138
17.2.4 处理传输错误 141
18 交通运输 141
18.1 客户 142
18.1.1 发送请求 142
18.1.2 接收响应 144
18.2 服务器 145
18.2.1 接收请求 145

18.2.2 发送响应 146
18.3 框架 147
18.4 错误处理 147
19 常见消息组件 147
19.1 SIP和SIPS统一资源指示符 148
19.1.1 SIP和SIPS URI组件 148
19.1.2 字符转义要求 152
19.1.3 示例SIP和SIPS URI 153
19.1.4 URI比较 153
19.1.5 从URI形成请求 156
19.1.6 将SIP URI和tel URL相关联 157
19.2 选项标签 158
19.3 标签 159
20 标题字段 159
20.1 接受 161
20.2 接受编码 163
20.3 接受语言 164
20.4 警报信息 164
20.5 允许 165
20.6 认证信息 165
20.7 授权 165
20.8 通话标识符 166
20.9 通话信息 166
20.10 联系方式 167
20.11 内容处置 168
20.12 内容编码 169
20.13 内容语言 169
20.14 内容长度 169
20.15 内容类型 170
20.16 CSeq 170
20.17 日期 170
20.18 错误信息 171
20.19 过期 171
20.20 来自 172
20.21 回复引用 172
20.22 最大转发数 173
20.23 最小到期日 173
20.24 MIME版本 173
20.25 组织 174
20.26 优先级 174
20.27 代理认证 174
20.28 代理授权 175
20.29 代理要求 175
20.30 记录路由 175
20.31 回复地址 176
20.32 需要..... 176
20.33 重试时间 176
20.34 路线 177

20.35 服务器 177
20.36 主题 177
20.37 支持的 178
20.38 时间戳 178
20.39 至 178
20.40 不支持 179
20.41 用户代理 179
20.42 通过 179
20.43 警告 180
20.44 WWW-Authenticate 182
21 响应代码 182
21.1 临时 1xx 182
21.1.1 100 尝试 183
21.1.2 180 铃声 183
21.1.3 181 正在转发呼叫 183
21.1.4 182 队列 183
21.1.5 183 会话进度 183
21.2 成功的2xx 183
21.2.1 200 OK 183
21.3 重定向 3xx 184
21.3.1 300 多选 184
21.3.2 301 永久移动 184
21.3.3 302 暂时移动 184
21.3.4 305 使用代理 185
21.3.5 380 替代服务 185
21.4 请求失败 4xx 185
21.4.1 400 错误请求 185
21.4.2 401 未授权 185
21.4.3 402 需要支付 186
21.4.4 403 禁止访问 186
21.4.5 404 未找到 186
21.4.6 405 方法不允许 186
21.4.7 406 不接受 186
21.4.8 407 代理身份验证需要 186
21.4.9 408 请求超时 186
21.4.10 410 已找到 187
21.4.11 413 请求实体过大 187
21.4.12 414 请求URI太长 187
21.4.13 415 不支持的媒体类型 187
21.4.14 416 不支持的 URI 方案 187
21.4.15 420 错误扩展 187
21.4.16 421 扩展所需 188
21.4.17 423 时间间隔过短 188
21.4.18 480 临时不可用 188
21.4.19 481 调用/事务不存在 188
21.4.20 482 循环检测 188
21.4.21 483 跳数过多 189
21.4.22 484 地址不完整 189

21.4.23 485 疑难 189
21.4.24 486 忙碌状态 189
21.4.25 487 请求终止 190
21.4.26 488 不可接受此处 190
21.4.27 491 请求待处理 190
21.4.28 493 不可解密 190
21.5 服务器故障 5xx 190
21.5.1 500 服务器内部错误 190
21.5.2 501 未实现 191
21.5.3 502 网关错误 191
21.5.4 503 服务不可用 191
21.5.5 504 服务器超时 191
21.5.6 505 版本不支持 192
21.5.7 513 消息过大 192
21.6 全局失败 6xx 192
21.6.1 600 处处忙碌 192
21.6.2 603 下降 192
21.6.3 604 任何地方都不存在 192
21.6.4 606 不可接受 192
22 HTTP认证的使用 193
22.1 框架 193
22.2 用户到用户认证 195
22.3 代理到用户身份验证 197
22.4 摘要认证方案 199
23 S/MIME ..... 201 中文翻译：23 S/MIME 201
23.1 S/MIME证书 201
23.2 S/MIME 密钥交换 202
23.3 保护 MIME 主体 205
23.4 使用S/MIME的SIP头部隐私和完整性：
隧道SIP 207
23.4.1 SIP的完整性和保密性属性
标题 207
23.4.1.1 完整性 207
23.4.1.2 保密性 208
23.4.2 隧道完整性及认证 209
23.4.3 洞穿加密 211
24 示例 213
24.1 注册 213
24.2 会话设置 214
25 SIP协议的扩展BNF 219
25.1 基本规则 219
26 安全考虑：威胁模型和安全
使用建议 232
26.1 攻击和威胁模型 233
26.1.1 注册劫持 233
26.1.2 模拟服务器 234
26.1.3 消息体篡改 235
26.1.4 会话终止 235

26.1.5 服务拒绝和放大 236
26.2 安全机制 237
26.2.1 传输和网络层安全 238
26.2.2 SIPS URI 方案 239
26.2.3 HTTP 认证 240
26.2.4 S/MIME ..... 240 中文翻译：26.2.4 S/MIME 240
26.3 实施安全机制 241
26.3.1 SIP实施者的要求 241
26.3.2 安全解决方案 242
26.3.2.1 注册 242
26.3.2.2 跨域请求 243
26.3.2.3 对等请求 245
26.3.2.4 DoS 防护 246
26.4 限制 247
26.4.1 HTTP摘要 247
26.4.2 S/MIME 248
26.4.3 TLS 249
26.4.4 SIPS URI 249
26.5 隐私 251
27 IANA 考虑事项 252
27.1 选项标签 252
27.2 警告代码 252
27.3 报头字段名称 253
27.4 方法与响应代码 253
27.5 "消息/sip" MIME 类型。 254
27.6 新内容处置参数注册 255
28 从RFC 2543的变更 255
28.1 主要功能变更 255
28.2 小型功能更改 260
29 规范性引用 261
30 262 有关信息参考文献 .....
一个定时器值表 265
致谢 266
作者地址 267
完整版权声明 269

1 简介

互联网有许多应用需要创建
会话的管理，其中会话被视为
数据在参与者协会之间的交换。
实现这些应用的做法使得其复杂化
参与者：用户可以在端点之间移动，他们可能
可由多个名称访问，并且它们可能使用几种方式通信
不同媒体 - 有时同时。 众多协议有
被编写了各种形式实时多媒体的著作
会话数据，例如语音、视频或文本消息。 会话
启动协议（SIP）与这些协议协同工作

启用互联网端点（称为用户代理）以发现一个
另一个并就他们会对会话进行表征达成一致
想分享。对于定位潜在的会议参与者，以及
对于其他函数，SIP 允许创建一个 {v*} 基础设施。
网络主机（称为代理服务器）用户代理可以向其发送
注册、会议邀请和其他请求。SIP是
一个敏捷、通用型创建、修改工具
终止独立于底层传输工作的会话
协议且不依赖于正在进行的会话类型
已建立。

2 SIP功能概述

SIP 是一种应用层控制协议，可以建立，
修改并终止多媒体会话（会议）等
互联网电话通话。SIP 也可以邀请参与者加入
已存在的会话，例如多播会议。 媒体可以
添加到（并从）现有会话中。SIP
透明支持名称映射和重定向服务，其中
支持个人移动性 [27] - 用户可以保持单一
外部可见标识符，无论其网络位置如何。

SIP支持建立和终止多媒体的五个方面
通讯：

- 用户位置：确定要使用的终端系统
  通信；
- 用户可用性：确定被叫方的意愿
  参与通信的一方；
- 用户能力：媒体和媒体参数的确定
  待使用；
- 会话设置: "响铃"，在{v*}处建立会话参数
  被叫方和主叫方；
- 会话管理：包括转移和终止
  会话、修改会话参数和调用
  服务。

SIP不是一个垂直集成的通信系统。SIP是
相当一个可以与其他IETF协议一起使用的组件
构建一个完整的多媒体架构。通常，这些
架构将包括实时传输等协议
协议（RTP）（RFC 1889 [28]）用于传输实时数据
提供QoS反馈，实时流媒体协议（RTSP）（RFC
2326 [29] 用于控制流媒体传输，媒体

网关控制协议（MEGACO）（RFC 3015 [30]）用于控制
网关到公共交换电话网络（PSTN），以及
会话描述协议（SDP）（RFC 2327 [1]）用于描述
多媒体会话。因此，应将SIP与{v*}结合使用
为了向用户提供完整的服务，与其他协议一起
用户。然而，SIP的基本功能和操作为
不依赖于这些协议中的任何一个。

SIP不提供服务。相反，SIP提供原语，
可用于实现不同的服务。例如，SIP 可以
定位用户并向其当前位置交付一个不透明物体。
如果此原始程序用于传递用 {v*} 编写的会话描述
SDP，例如，端点可以就一个 {v*} 的参数达成一致
会话。如果使用相同的原始方法来传递照片，
呼叫者以及会话描述，一项“ 呼叫者ID ”服务可以
易于实现。正如这个例子所示，一个原始操作是
通常用于提供多种不同的服务。

SIP不提供如主席控制等会议控制服务
或者投票，并未规定如何管理会议。
SIP可以用来启动使用某些其他会议的会话
控制协议。 由于SIP消息及其建立的会话
可以通过完全不同的网络，SIP 不能，并且可以
不，提供任何类型的网络资源预留功能。

服务提供的性质使得安全性尤其
重要。为此，SIP提供了一套安全服务，
包括拒绝服务预防、身份验证（用户和{v*}）
用户和代理到用户），完整性保护，以及加密和
隐私服务。

SIP与IPv4和IPv6都兼容。

3 术语

在此文档中，关键词“ 必须 ”、“ 禁止 ”、“ 必需 ”。
"SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT"
推荐"、"可能"和"可选"应解释为
描述于BCP 14、RFC 2119 [2] 中，并指示需求级别
符合的SIP实现。

4 操作概述

本节介绍了使用简单操作的SIP基本操作。
示例。本节具有教程性质，不包含
任何规范性陈述。

第一个示例展示了SIP的基本功能：{v*}的位置
终点，沟通欲望的信号，会话协商
会话建立参数，以及会话的拆除
已建立。

图1显示了SIP消息交换的典型示例，其中{v*}保持不变。
两位用户，Alice 和 Bob。 （每条消息都标有字母
"F" 和一个用于文本参考的数字。) 在这个例子中，Alice
使用她的PC上的SIP应用程序（称为软电话）进行通话
鲍勃通过互联网使用他的SIP电话。同时显示的是两个SIP代理
服务器代表Alice和Bob进行会话的代理
建立。这种典型安排通常被称为
"SIP 梯形" 如由虚线所表示的几何形状所示
在图1中。

爱丽丝使用他的SIP身份 “ 呼叫 ” 鲍勃，这是一种统一资源类型
标识符（URI）称为SIP URI。SIP URI在章节中定义。
19.1. 它具有与电子邮件地址类似的形式，通常
包含用户名和主机名。在这种情况下，它是
sip:bob@biloxi.com，其中biloxi.com是Bob的SIP域名
服务提供商。Alice有一个SIP URI sip:alice@atlanta.com。
爱丽丝可能输入了鲍勃的URI，或者可能点击了一个超链接
或一个地址簿条目。SIP还提供了一个安全的URI，
称为SIPS URI。例如：sips:bob@biloxi.com。一个呼叫
制作成SIPS URI保证安全、加密的传输
(即TLS)用于承载从主叫方到
调用方的域。从那里，请求被安全地发送到
被调用者，但具有依赖于策略的安全机制
被调用者的域。

SIP基于类似于HTTP的请求/响应事务模型。
每次交易都包含一个调用特定操作的请求
方法或函数在服务器上，以及至少一个响应。在
这个例子中，交易从Alice的软电话开始发送
一个发送给Bob的SIP URI的INVITE请求。INVITE是一个示例
关于指定请求者（Alice）所采取操作的SIP方法
希望服务器（Bob）进行取用。INVITE请求包含一个数字
头部字段。头部字段是命名属性，提供
额外关于消息的信息。其中包含在
邀请包含通话的唯一标识符，目的地
地址，爱丽丝的地址，以及关于会话类型的资料
Alice希望与Bob建立的。 The INVITE（消息F1在
图1)可能看起来像这样：

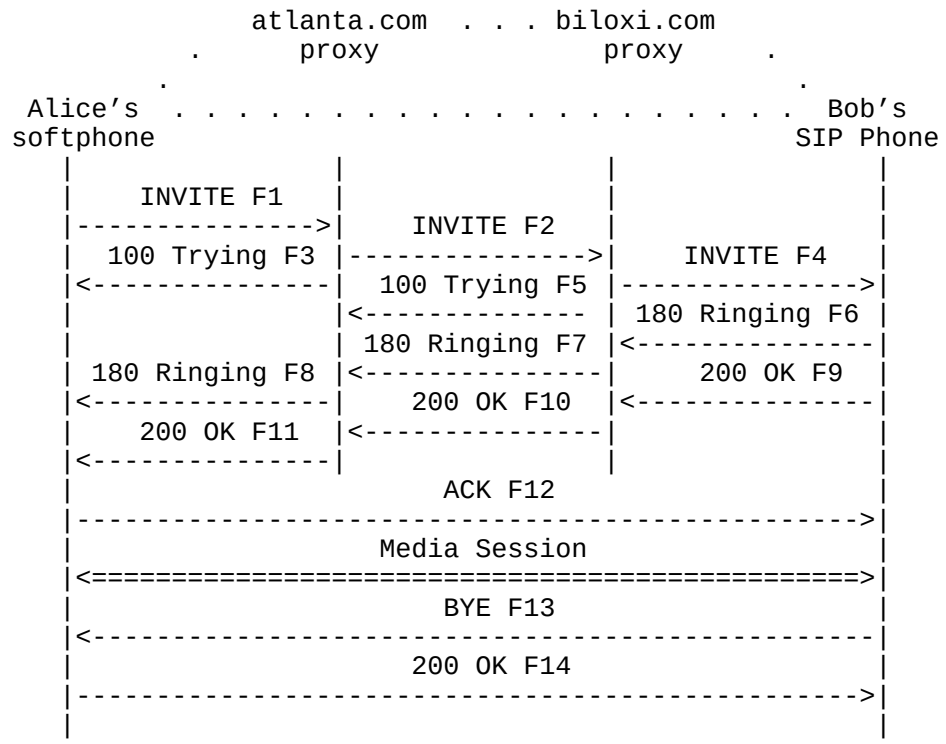

图 1 SIP会话设置示例与SIP陷阱 ezoid

邀请 sip:bob@biloxi.com SIP/2.0
通过：SIP/2.0/UDP pc33.atlanta.com;分支=z9hG4bK776asdhds
最大转发数：70
收件人：Bob <sip:bob@biloxi.com>
从：Alice <sip:alice@atlanta.com>;tag=1928301774
呼叫ID：a84b4c76e66710@pc33.atlanta.com
CSeq: 314159 邀请
联系：<sip:alice@pc33.atlanta.com>
内容类型: application/sdp
内容长度：142

(爱丽丝的SDP未显示)

文本编码消息的第一行包含方法名称
(INVITE). 以下行是一个头部字段的列表。 This
示例包含一个最小必需集。头部字段是
以下简要描述：

通过包含Alice的地址（pc33.atlanta.com）
期待收到对此请求的回复。它还包含一个 {v*}。
分支参数，用于标识这笔交易。

包含一个显示名称（Bob）和一个SIP或SIPS URI
(sip:bob@biloxi.com) 请求最初是针对的
有向的。显示名称在RFC 2822 [3]中描述。

从也包含一个显示名称（Alice）和一个SIP或SIPS URI
(sip:alice@atlanta.com) 表示请求的发起者。
此报头字段还包含一个包含随机字符串的标签参数
(1928301774) 该号码由软电话添加到URI中。它被用于
用于识别目的。

呼叫ID包含此呼叫的全局唯一标识符，
由随机字符串和软电话的组合生成
主机名或IP地址。To标签和From标签的组合。
并且 Call-ID 完全定义了一个点对点 SIP 关系
在Alice和Bob之间，被称为对话。

CSeq 或命令序列包含一个整数和方法名称。
会话中每个新的请求都会增加CSeq编号。
这是一个传统的序列号。

联系包含一个表示直接路由的SIP或SIPS URI
联系Alice，通常由一个在完全限定域名下的用户名组成
域名（FQDN）。虽然FQDN是首选，但许多终端系统仍然使用
未注册域名，因此允许IP地址。
虽然 Via 头部字段告诉其他元素将数据发送到
响应，Contact头字段告诉其他元素发送到何处
未来请求。

Max-Forwards用于限制请求可以在其上进行的跳数
目的地的方式。它由一个整数组成。
每次跳转减少一个。

内容类型包含对消息体的描述（未显示）。

内容长度包含消息体的八位（字节）计数。

SIP头字段完整集合在第20节中定义。

会话的详细信息，例如媒体类型、编解码器等
采样率，不是使用SIP进行描述的。相反，而是使用正文部分
SIP消息包含会话描述，以某种编码方式表示
其他协议格式。其中一种格式是会话描述
协议（SDP）（RFC 2327 [1]）。此SDP消息（未在图中显示）

示例) 以类似于的方式通过SIP消息承载 {v*}
电子邮件携带的文档附件，或网页
页面被携带在HTTP消息中。

自软电话不知道鲍勃的位置或SIP
服务器位于 biloxi.com 域名下，软电话向发送 INVITE
SIP服务器为Alice的域名atlanta.com提供服务。地址
亚特兰大.com SIP服务器可能已在爱丽丝的配置中
软电话，或者它可能已被DHCP发现，例如。

亚特兰大.com SIP服务器是一种称为代理的SIP服务器
服务器。一个代理服务器接收SIP请求并将它们转发
代表申请人。在此示例中，代理服务器接收
邀请请求并发送一个100（尝试中）响应回Alice的
软电话。100（尝试）响应表示INVITE请求已发送
已收到并且代理正在代表她进行路由
对目的地的INVITE请求。SIP中的响应使用三位数
代码后跟描述性短语。此响应包含
与 To、From、Call-ID、CSeq 和 Via 中的 branch 参数相同的
邀请，允许Alice的软电话将此响应与
发送的INVITE。亚特兰大.com代理服务器定位代理
服务器位于biloxi.com，可能通过执行特定类型的DNS
(域名服务)查找以找到提供服务的SIP服务器
biloxi.com 域名。这在[4]中有描述。因此，它
获取biloxi.com代理服务器的IP地址并转发，
或代理，那里的INVITE请求。在转发请求之前，
亚特兰大.com代理服务器添加了一个额外的Via头字段
值包含其自身的地址（INVITE 已经包含
爱丽丝在第一个Via的地址）。biloxi.com代理服务器
接收到的INVITE并回以100（尝试中）响应
亚特兰大.com代理服务器以表明它已收到
邀请并正在处理请求。代理服务器咨询
数据库，通常称为位置服务，其中包含
Bob当前的IP地址。 （我们将在下一节中看到如何）
此数据库可以被填充。) Biloxi.com代理服务器添加
另一个 Via 头字段值及其自己的地址到 INVITE
将其代理到鲍勃的SIP电话。

鲍勃的SIP电话收到INVITE并提醒鲍勃有来电
Alice来电，以便Bob决定是否接听电话，
即，鲍勃的电话响了。鲍勃的SIP电话在180中指示这一点。
(响铃)响应，通过两个代理路由回
反向方向。每个代理使用 Via 头字段来
确定响应的发送位置并移除自己的地址
顶部。因此，尽管DNS和位置服务查找进行了
需要路由初始的INVITE，180（振铃）响应可以
返回给调用者，无需查找或无需状态

维护在代理中。这也具有期望的属性，
每个看到 INVITE 的代理也将看到所有对该请求的响应
邀请。

当Alice的软电话接收到180（响铃）响应时，它传递
此信息告诉爱丽丝，或许使用音频回铃音或通过
在爱丽丝的屏幕上显示一条消息。

在这个例子中，Bob决定接听电话。当他拿起电话时
手机，他的SIP电话发送一个200（OK）响应以表示
呼叫已接听。200（OK）包含消息体
与Bob会话类型的SDP媒体描述
愿意与Alice建立。因此，存在一个两阶段
SDP消息交换：Alice发送了一条给Bob，Bob也发送了一条
返回爱丽丝。 这个两阶段交换提供基本的协商
功能，并基于SDP的简单出价/回答模型
交换。如果鲍勃不想接听电话或正忙于
另一个调用，本应发送错误响应而不是
200 (OK)，这将导致没有媒体会话
已建立。SIP响应代码的完整列表在章节中。
21. 200（OK）（如图1中的消息F9）可能看起来像这样
鲍勃发送出去：

SIP/2.0 200 正确
通过：SIP/2.0/UDP server10.biloxi.com
;分支=z9hG4bKnashds8;接收=192.0.2.3
通过：SIP/2.0/UDP bigbox3.site3.atlanta.com
;分支=z9hG4bK77ef4c2312983.1;接收=192.0.2.2
通过：SIP/2.0/UDP pc33.atlanta.com
;分支=z9hG4bK776asdhds ;接收=192.0.2.1
收件人：Bob sip:bob@biloxi.com>;tag=a6c85cf
从：Alice <sip:alice@atlanta.com>;tag=1928301774
呼叫ID：a84b4c76e66710@pc33.atlanta.com
CSeq: 314159 邀请
联系：<sip:bob@192.0.2.4>
内容类型: application/sdp
内容长度：131

(鲍勃的SDP未显示)

响应的第一行包含响应代码（200）和
原因短语（OK）。剩余的行包含标题字段。
The Via, To, From, Call-ID, and CSeq header fields are copied from
the INVITE 请求。 （有三个 Via 头字段值 - 一个
由Alice的SIP电话添加，由atlanta.com代理添加，以及
一个由biloxi.com代理添加的。）Bob的SIP电话添加了一个标签
参数到To报头字段。此标签将被{v*}所包含。
两个端点进入对话框，并将包含在所有未来的

请求和响应在此调用中。 联系头部字段
包含一个Bob可以通过其SIP电话直接联系到的URI。
内容类型和内容长度指的是消息体（不是 {v*}）
显示) 包含 Bob 的 SDP 媒体信息。

除了本页中显示的DNS和位置服务查找外
示例，代理服务器可以做出灵活的“ 路由决策 ”来
确定发送请求的位置。例如，如果Bob的SIP电话
返回了一个486（此处忙碌）响应，biloxi.com代理服务器
可以代理将INVITE代理到鲍勃的语音邮件服务器。一个代理服务器可以
同时向多个位置发送一个INVITE。
并行搜索的类型被称为分叉。

在这种情况下，200（OK）通过两个代理重新路由。
是收到爱丽丝的软电话，然后停止回铃音
并且表示电话已被接听。 最后，Alice的
软电话向Bob的SIP电话发送确认消息，ACK
确认已收到最终响应（200（OK））。在此
示例，ACK 直接从爱丽丝的软电话发送到鲍勃的SIP
电话，绕过两个代理。这发生是因为端点
已从联系头部字段中学习彼此的地址
通过INVITE/200（OK）交换，当时并不知道
初始INVITE已发送。两个代理执行了查找。
不再需要，因此代理从调用流程中退出。
完成用于建立连接的 INVITE/200/ACK 三方握手
SIP会话。会话设置的详细信息请参阅第13节。

爱丽丝和鲍勃的媒体会话现已开始，他们发送媒体
数据包使用它们在SDP交换中同意的格式。
通常，端到端媒体数据包的路径与
SIP信令消息。

在会话期间，Alice或Bob可能会决定更改
媒体会话的特性。这是通过
发送包含新媒体描述的重新邀请。
邀请引用现有对话，以便另一方知道
这是要修改现有会话而不是建立
新会话。对方发送200（OK）以接受更改。
请求者以ACK响应200（OK）。如果其他
党派不接受变更，他发送一个错误响应，例如
488（此处不可接受），但也收到一个ACK。然而，
重邀请失败不会导致现有呼叫失败 -
会话继续使用之前协商的
特性。会话修改的详细信息请参阅第节
14.

在通话结束时，Bob 首先断开连接（挂断电话）
生成一个BYE消息。此BYE消息直接路由到Alice的
软电话，再次绕过代理。爱丽丝确认已收到
BYE 200（OK）响应，终止会话
BYE 交易。不发送 ACK - 只有在
对INVITE请求的响应的响应。 这其中的原因有
稍后将对INVITE的特殊处理进行讨论，但与此相关
SIP中的可靠性机制，它可能需要的时间长度
一个需要接听的电话，以及分叉。因此，
请求处理在SIP中通常被归类为INVITE或非-{v*}
邀请，指除邀请外的所有其他方法。 详细信息
在会话终止方面，请参阅第15节。

第24.2节详细描述了图1中显示的消息。

在某些情况下，对于SIP信令路径中的代理来说，这可能是有用的
查看端点在持续时间内的所有消息
会话。例如，如果biloxi.com代理服务器希望
在初始INVITE之后继续留在SIP消息路径中，它就会
添加到INVITE中一个称为Record-的必需路由头字段
路由包含解析到主机名或IP地址的URI
代理。此信息将由鲍勃的SIP
电话和（由于Record-Route头部字段被回传）
200（OK）Alice的软电话并存储了整个持续时间
对话。Biloxi.com代理服务器随后将接收并代理
ACK，BYE，以及200（OK）回复BYE。每个代理可以独立
决定接收后续消息，并且这些消息将通过
通过所有选择接收它的代理。此功能是
常用于提供通话中功能的代理。

注册是SIP中另一种常见操作。注册是一
方式，biloxi.com服务器可以学习Bob的当前位置。
在初始化时，以及周期性间隔，Bob的SIP电话发送
将注册消息发送到名为biloxi.com域的SIP服务器
注册器。注册消息将Bob的SIP或SIPS URI关联
(sip:bob@biloxi.com) 与他目前使用的机器一起
已记录（在Contact头部字段中作为SIP或SIPS URI传达）。
注册员将此关联（也称为绑定）写入
数据库，称为位置服务，其中它可以被使用
代理在biloxi.com域名中。 通常，这是一个注册商服务器的
域名与该域的代理服务器位于同一位置。它是一个
重要的概念是区分SIP服务器类型的区别
是逻辑的，不是物理的。

鲍勃不仅限于从单一设备注册。例如，
他家里的SIP电话和办公室的那部都能发送
注册。此信息存储在同一位置

服务和允许代理执行各种类型的搜索
定位Bob。同样，可以注册多个用户在一个
同一时间单个设备。

位置服务只是一个抽象概念。它通常
包含允许代理输入URI并接收的信息
集合，包含零个或多个URI，指示代理将请求发送到何处
请求。注册是创建此类信息的一种方式，但
不是唯一的方法。可以在{v*}处配置任意映射函数。
管理员的决定权。

最后，需要注意的是，在SIP中，注册被用于
为路由传入的SIP请求且在授权中不起作用
出站请求。授权和身份验证在此处理
SIP 要么基于请求的挑战/响应
机制，或使用第X节中讨论的底层方案
26. 中文翻译：26.

SIP消息的完整集合，用于此注册示例
在24.1节中。

SIP中的附加操作，例如查询功能
SIP服务器或客户端使用OPTIONS，或取消挂起的
请求使用 CANCEL，将在后续章节中介绍。

5 协议结构

SIP是一个分层协议，这意味着它的
行为用一组相当独立的术语来描述
处理阶段之间只有松散耦合的阶段。
协议行为被描述为分层，目的是
演示，允许描述跨{v*}的常见函数
元素在一个单独的部分中。它不规定一个实现
在任何方式下。当我们说一个元素 “ 包含 ” 一层时，我们的意思是
它符合该层定义的规则集。

并非每个由协议指定的元素都包含每一层。
此外，由SIP指定的元素是逻辑元素，不是
物理的。一个物理实现可以选择作为不同的
逻辑元素，甚至可能是基于每笔交易的。

SIP的最低层是其语法和编码。其编码是
使用扩展的巴科斯-诺尔范式语法（BNF）指定。
25节中指定了完整的BNF；SIP的概述
消息的结构可以在第7节中找到。

第二层是传输层。它定义了客户端
发送请求并接收响应以及服务器如何接收
请求并通过网络发送响应。所有SIP元素
包含一个传输层。传输层在{v*}中描述。
第18节。

第三层是交易层。交易是
SIP的基本组件。一个事务是由一个请求发送的
客户端事务（使用传输层）到服务器
交易，以及从该请求中发送的所有响应
服务器事务返回给客户端。事务层处理
应用层重传，响应与请求的匹配，
并且应用层超时。任何用户代理客户端的任务
(UAC) 通过一系列交易来完成。
讨论交易可以在第17节找到。用户代理
包含一个事务层，就像有状态代理一样。无状态
代理不包含事务层。事务层
具有客户端组件（称为客户端事务）和
服务器组件（称为服务器事务），每个组件
由一个构建为有限状态机的模型表示
处理特定请求。

交易层之上的层被称为交易用户
(TU). 每个SIP实体，除了无状态代理外，都是一个
交易用户。当TU希望发送请求时，它创建一个
客户端事务实例并将其与请求一起传递
目标IP地址、端口以及要发送的传输协议
请求。一个创建客户端事务的TU也可以取消它。
当客户端取消交易时，它请求服务器停止
进一步处理，恢复到之前存在的状态
交易已启动，并生成特定的错误响应
该交易。这是通过一个取消请求来完成的，其中
构成其自身的交易，但引用了待进行的交易
已取消（第9节）。

SIP元素，即用户代理客户端和服务器，无状态
并且状态化代理和注册表，包含一个核心
区分彼此。核心，除了无状态的
代理，是交易用户。 而UAC和UAS的行为
核心取决于方法，对于所有方法都有一些共同规则
方法（第8节）。对于UAC，这些规则支配着{v*}的构建
关于一个请求；对于UAS，它们规范了请求的处理
生成响应。 由于注册在...中起着重要作用
SIP，一个处理REGISTER的UAS，被赋予特殊名称
注册员。第10节描述了UAC和UAS的核心行为。
注册方法。第11节描述了UAC和UAS的核心行为。
选项方法，用于确定UA的功能。

某些其他请求是在对话中发送的。对话是
点对点SIP关系，存在于两个用户代理之间且持续存在
一段时间。 对话促进了消息的排序和
正确的请求路由到用户代理之间。 The INVITE
方法是在本规范中定义的唯一建立方式
对话。当UAC发送一个处于上下文中的请求时
对話，它遵循第8节中讨论的常见UAC规则，但
同样，讨论中间对话请求的规则。第12节讨论对话
并且介绍了它们的建设和维护程序，
除了在对话框内构建请求之外。

SIP中最重要的一种方法是INVITE方法，它被用于
建立参与者之间的会话。会话是
参与者集合，以及他们之间的媒体流，用于
通信的目的。第13节讨论了会话的
启动，导致一个或多个SIP对话。第14节
讨论了该会话的特性是如何通过修改的
在对话中使用INVITE请求。最后，第15节
讨论了会话是如何终止的。

第8、10、11、12、13、14和15节的程序处理如下
完全使用UA核心（第9节描述了取消，其中
适用于UA核心和代理核心）。第16节讨论了
代理元素，便于在用户之间路由消息
代理

6 定义

以下术语对SIP具有特殊意义。

记录地址：记录地址（AOR）是一个SIP或SIPS URI
指向具有位置服务的域，该服务可以映射
另一个URI的URI，用户可能在那里可用。
通常，位置服务是通过以下方式填充的：
注册。AOR通常被认为是“ 公众
用户地址"的。

用户代理对对碰：用户代理对对碰（B2BUA）是一种
逻辑实体，接收请求并对其进行处理
用户代理服务器（UAS）。为了确定请求
应予回答，它充当用户代理客户端（UAC）并
生成请求。与代理服务器不同，它维护对话
状态并必须参与所有在对话框中发送的请求
它已经建立。由于它是UAC和一个{v*}的连接。
UAS，无需对其行为进行明确定义。

调用：调用是一个非正式术语，指的是某些沟通
在节点之间，通常为以下目的设置
多媒体对话。

调用腿：对对话框[31]的另一种称呼；在此处不再使用
规范。

调用状态：如果一个代理保留了状态，则称为调用状态
对发起的INVITE到终止的BYE的对话
请求。一个有状态的代理总是事务状态的，
但反之不一定成立。

客户端：客户端是指任何发送SIP请求的网络元素
并且接收SIP响应。客户端可能交互也可能不交互
直接与人类用户交互。用户代理客户端和代理是
客户。

会议：包含以下多媒体会议（见下文）
多个参与者。

核心：核心指代特定类型的特定功能
SIP实体，即特定于有状态或无状态的
代理，用户代理或注册机构。所有核心，除了那些用于
无状态代理，是事务用户。

对话：对话是两个{v*}之间的对等SIP关系
UAs that persists for some time. A dialog is established by
SIP消息，例如对INVITE请求的2xx响应。
对話由呼叫标识符、本地标签和{v*}识别。
远程标签。对话框在RFC中以前被称为呼叫腿
2543.

下游：事务内消息转发的一个方向
那指的是用户请求流的方向
代理客户端到用户代理服务器。

最终响应：终止SIP事务的响应，作为
与不提供临时响应的相反。所有2xx，3xx，
4xx、5xx和6xx响应是最终的。

标题：标题是SIP消息的一个组件，用于传达
信息关于消息。它被结构化为一个序列
头部字段。

表头字段：表头字段是SIP消息的一个组成部分
标题。一个标题字段可以出现为一个或多个标题字段
行。标题字段行由标题字段名称和零
或更多头部字段值。一个头部字段上的多个值

给定报头字段行由逗号分隔。一些报头
字段只能有一个标题字段值，并且作为
结果，始终以单个标题字段行出现。

表头字段值：表头字段值是一个单一值；
标题字段由零个或多个标题字段值组成。

家庭域：为SIP用户提供服务的域。
通常，这是URI中存在的域
注册的记录地址。

信息 响应：与临时响应相同 问题。

发起者，主叫方，呼叫者：发起会话的一方
（和对话）带有INVITE请求。 主叫方保留此
角色从它发送建立初始INVITE的时间起
一个对话，直到该对话终止。

邀请：一个 INVITE 请求。

受邀者，被邀请用户，被叫方，被叫人：指被邀请的方
接收用于建立连接的 INVITE 请求
新会话。被调用方从那时起保留此角色。
接收INVITE直到对话终止
由该INVITE建立的。

位置服务：位置服务被SIP重定向或
代理服务器以获取被叫方的可能信息
位置（s）。它包含地址-的绑定列表。
记录键到零个或多个联系地址。绑定
可以以多种方式创建和删除；本规范
定义了一个用于更新绑定的 REGISTER 方法。

循环：一个到达代理的请求，被转发，后来
返回到相同的代理。当它到达时，第二个
时间，其请求URI与第一次相同，以及其他
标题字段影响代理操作的部分保持不变，所以
该代理将对以下内容做出相同的处理决策
请求第一次提出的请求。循环请求是错误，
并且检测和处理它们的程序是
由协议描述。

松散路由：如果一个代理遵循
本规范中定义的处理程序
路由报头字段。这些过程将
请求的目的地（存在于Request-URI中）从

需要访问的代理集合
(存在于路由头字段中)。一个符合 {v*} 的代理
这些机制也被称为松散路由器。

消息：作为协议一部分，SIP元素之间发送的数据。
SIP消息要么是请求，要么是响应。

方法：该方法是一个请求旨在的主要功能
在服务器上调用。该方法包含在请求中
消息本身。示例方法包括 INVITE 和 BYE。

出站代理：一个从客户端接收请求的代理，即使
尽管它可能不是由Request-URI解析的服务器。
通常，UA会手动配置出站代理，
或可以通过自动配置协议了解一个。

并行搜索：在并行搜索中，一个代理发出多个
接收传入请求后对可能的用户位置进行请求
请求。而不是发出一个请求然后等待
在发出下一个请求之前的最终响应，如在
顺序搜索，并行搜索在无需的情况下发出请求
等待先前请求的结果。

临时响应：服务器用来指示的响应
进度，但这并不会终止一个SIP会话。1xx
响应为临时，其他响应被视为
最终。

代理，代理服务器：一个充当双方中介的实体
服务器和客户端，用于发起请求
代表其他客户。代理服务器主要扮演
路由的作用，这意味着它的任务是确保一个
请求发送到另一个“ 更接近 ”目标实体的实体
用户。代理对于强制执行策略（对于
示例，确保用户允许进行通话）。A
代理解释，并在必要时重写特定部分
在转发之前发送一个请求消息。

递归：当客户端生成一个3xx响应时，它会递归处理
新的请求到联系人头部中的一个或多个URI
响应中的字段。

重定向服务器：重定向服务器是一种用户代理服务器，它
生成它接收到的请求的3xx响应，将其引导
客户端联系一组备选URI。

注册员：注册员是一个接受注册请求的服务器
并且将接收到的信息放入那些请求中
该域处理的位置服务。

常规交易：任何具有 {v*} 的交易都是常规交易
一种除INVITE、ACK或CANCEL之外的方法。

请求：客户端发送到服务器的SIP消息，用于
目的调用特定操作。

响应：从服务器发送到客户端的SIP消息，用于
指示客户端发送到服务器的请求的状态
服务器。

回铃音：回铃音是由呼入方产生的信号音
党派的申请表明被叫方正在
警报（响铃）。

路由集：路由集是一组有序的SIP或SIPS URI
which represent a list of proxies that must be traversed when 表示在遍历时必须穿越的代理列表
发送特定请求。可以学习到路由集。
通过记录路由等头部信息，或者可以配置。

服务器：服务器是一种网络元素，它接收请求在
为了服务它们并发送响应给那些
请求。服务器示例包括代理、用户代理服务器，
重定向服务器和注册商。

顺序搜索：在顺序搜索中，一个代理服务器尝试
每个按顺序的联系方式，继续到下一个
仅在前一个生成最终响应后。一个2xx
或 6xx 类最终响应总是终止一个序列
搜索。

会话：从SDP规范中： "多媒体会话是
多媒体发送器和接收器的集合以及数据流
从发送者流向接收者。一个多媒体会议是
一个多媒体会话的示例。"（RFC 2327 [1]）（一个会话
如定义的SDP可以包含一个或多个RTP会话。) 如
定义后，被调用者可以被不同的人邀请多次。
调用，到同一会话。如果使用SDP，则是一个会话
由SDP用户名、会话ID的连接定义
网络类型、地址类型和原始地址元素
字段。

SIP 事务：SIP 事务发生在客户端和服务器之间
服务器并包含从第一个请求发送的所有消息
从客户端到服务器，直到最终的（非1xx）响应

从服务器发送到客户端。如果请求是INVITE
并且最终响应是非2xx，交易也
包含对响应的ACK。对2xx响应的ACK为
一个INVITE请求是一个独立的交易。

螺旋：螺旋是一个被路由到代理的SIP请求，
转发并再次到达那个代理，但
这次不同之处将导致不同的结果
处理决策比原始请求。通常，这
表示请求的 Request-URI 与其之前的
到达。螺旋不是一个错误条件，与循环不同。
通常原因为此是呼叫转接。用户拨打
joe@example.com. example.com 代理将其转发到 Joe 的
PC，进而将其转发到bob@example.com。
请求被代理回example.com代理。然而，
这不是一个循环。由于请求是针对一个
不同用户，它被视为螺旋，且是有效的
条件。

有状态代理：维护客户端和的逻辑实体
服务器事务状态机，由本规范定义
在处理一个请求期间，也称为事务
有状态代理。有状态代理的行为进一步
定义在第16节中。一个（事务）有状态的代理不是
与调用有状态代理相同。

无状态代理：一个不维护状态的逻辑实体
客户端或服务器事务状态机在此定义
规范当它处理请求时。一个无状态的代理
向下传递它接收到的每个请求以及每个
响应它从上游接收到的。

严格路由：如果一个代理遵循
RFC 2543的路径处理规则以及许多先前工作中的
进度版本的此RFC。该规则导致代理
销毁当存在路由头时Request-URI的内容
字段存在。在此不使用严格路由行为。
规范，倾向于宽松的路由行为。代理
执行严格路由的也被称为严格路由器。

目标刷新请求：在一个{v*}内发送的目标刷新请求
对远程进行修改的请求定义为对话
对话的目标。

交易用户（TU）：协议处理层的
位于事务层之上。事务用户包括
UAC核心，UAS核心和代理核心。

上游：事务中消息转发的一个方向
那指的是响应从用户流向的方向
代理服务器返回给用户代理客户端。

URL编码：根据RFC 2396编码的字符字符串，
第2.4节[5]。

用户代理客户端（UAC）：用户代理客户端是一个逻辑实体
创建一个新请求，然后使用客户端
事务状态机发送它。UAC的作用持续
仅在该交易期间。换句话说，如果
一段软件发起请求，它充当UAC
该交易时长。如果它收到一个请求
稍后，它扮演用户代理服务器的角色
处理该交易。

UAC 核心功能：UAC 所需的一组处理函数
位于事务层和传输层之上。

用户代理服务器（UAS）：用户代理服务器是一个逻辑实体
生成对SIP请求的响应。该响应
接受、拒绝或重定向请求。此角色持续
仅在该交易期间。换句话说，如果
一段软件对请求做出响应，它充当{v*}
该交易时长。如果它生成一个请求
稍后，它扮演用户代理客户端的角色
处理该交易。

UAS 核心功能：在 UAS 中所需的一组处理函数
位于事务层和传输层之上。

用户代理（UA）：一个可以同时作为用户的逻辑实体
代理客户端和用户代理服务器。

UAC和UAS的作用，以及代理和重定向服务器，是
按每笔交易定义。例如，用户
代理在发送初始INVITE时充当UAC
请求并作为UAS在接收到被叫方的BYE请求时。
同样，相同的软件可以作为一台代理服务器为其中一个
请求并作为下一个请求的重定向服务器。

代理、位置和注册服务器如上定义的是逻辑的
实体；实现可以将其组合成一个单一的应用程序。

7 SIP消息

SIP是一种基于文本的协议，并使用UTF-8字符集（RFC 2279）
[7]).

一个SIP消息要么是客户端向服务器发送的请求，要么是
服务器对客户端的响应。

请求（第7.1节）和响应（第7.2节）消息都使用
RFC 2822 [3]的基本格式，即使语法有所不同
字符集和语法特定。 （SIP允许头部字段，
无效的RFC 2822标题字段，例如。）两种类型
消息由一个起始行、一个或多个头部字段以及一个 {v*} 组成。
空行表示标题字段的结束，以及一个可选的
消息体。

通用消息 = 开始行
*消息头
CRLF CRLF CRLF CRLF CRLF CRLF CRLF CRLF CRLF CRLF CRLF CRLF CRLF CRLF
[消息体]
开始行 = 请求行 / 状态行

起始行、每条消息头行以及空行必须
由回车换行序列（CRLF）终止。注意
空行必须存在，即使消息体不存在也是如此。

除了上述字符集的差异外，SIP的大部分内容
消息和头部字段语法与HTTP/1.1相同。
比在这里重复语法和语义，我们使用[HX.Y]来指代
到当前HTTP/1.1规范（RFC 2616 [8]）的第X.Y节。

然而，SIP 不是 HTTP 的扩展。

7.1 请求

SIP请求通过具有一个起始请求行来区分。
行。请求行包含一个方法名称、一个请求URI和空格。
协议版本由单个空格（SP）字符分隔。

请求行以CRLF结束。除在
行尾CRLF序列。不允许线性空白（LWS）
在任何元素中。

请求行 = 方法 SP 请求URI SP SIP版本 CRLF

方法：本规范定义了六种方法：REGISTER用于
注册联系信息、INVITE、ACK和CANCEL的
设置会话，BYE 用于终止会话，以及
选项用于查询服务器关于其功能的信息。SIP
扩展，在标准跟踪RFC中记录的，可以定义
附加方法。

请求URI：请求URI是一个SIP或SIPS URI，如以下所述
第19.1节或通用URI（RFC 2396 [5]）。它表示
用户或服务，该请求正在被发送至。
请求URI不得包含未转义的空格或控制字符
字符不得包含在“ <> ”中。

SIP 元素可能支持除以下方案之外的请求 URI：
"sip" 和 "sips"，例如 RFC 中的 "tel" URI 方案
2806 [9]. SIP 元素可以使用任何方式将非-SIP URI 进行翻译。
机制可供使用，导致SIP URI，SIPS URI，
或者某些其他方案。

SIP-Version: 请求和响应消息都包含
当前使用的SIP版本，并遵循[H3.1]（将HTTP替换为
通过SIP，并将HTTP/1.1替换为SIP/2.0）有关版本
排序、合规要求以及版本升级
数字。为了符合本规范，
应用程序发送SIP消息时必须包含SIP版本
关于 "SIP/2.0"。SIP-Version 字符串不区分大小写，
但是实现必须发送大写字母。

与HTTP/1.1不同，SIP将版本号视为一个字面量
字符串。实际上，这应该没有区别。

7.2 响应

SIP 响应通过具有状态行与请求区分
作为它们的起始行。状态行由协议版本组成
随后是一个数字状态码及其相关的文本短语，
每个元素之间由单个空格字符分隔。

不允许在最后的CRLF序列中包含CR或LF。

状态行 = SIP版本 SP 状态码 SP 原因短语 CRLF

状态码是一个3位整数结果码，表示
结果尝试理解和满足一个请求。
原因短语旨在提供一个简短的文本描述，{v*}保持不变。
状态码。状态码旨在供自动机使用，
whereas the Reason-Phrase is intended for the human user. A client
不需要检查或显示原因短语。

当此规范建议具体措辞以说明原因
短语，实现可以选择其他文本，例如，在
源语言指示Accept-Language头字段中的
请求。

状态码的第一个数字定义了响应的类别。
最后两位数字没有任何分类作用。 对于这个
原因，任何状态码在100到199之间的响应是
被称为“ 1xx响应 ”，任何状态码为
在200到299之间作为“ 2xx响应 ”，以此类推。SIP/2.0允许
六个第一位的值：

1xx：临时 -- 已收到请求，正在继续处理
请求；

2xx: 成功 -- 动作已成功接收，理解,
并且接受；

3xx: 重定向 -- 需要采取进一步行动
完成请求；

4xx: 客户端错误 -- 请求包含错误的语法或无法
满足于本服务器；

5xx: 服务器错误 -- 服务器未能完成显然
有效请求；

6xx：全局失败 -- 请求在任何地方都无法得到满足
服务器。

第21节定义了这些类并描述了各个代码。

7.3 报头字段

SIP头字段在语法上与HTTP头字段相似
并且语义。特别是，SIP头字段遵循[H4.2]
消息头语法定义及规则
扩展头部字段到多行。然而，后者是
指定在HTTP中，具有隐式空白和折叠。
规范符合RFC 2234 [10]并仅使用显式
空格和折叠作为语法的一个组成部分。

[H4.2] 还指定了相同字段的多个头部字段
名称的值是逗号分隔的列表，可以合并成一个
表头字段。这也适用于SIP，但具体规则是
不同，因为语法不同。具体来说，任何SIP
标题其语法形式为

表头 = "header-name" HCOLON 表头值 *(COMMA 表头值)

允许将相同名称的报头字段合并为一个逗号-
分隔列表。联系头部字段允许逗号分隔
除非标题字段值为"*"，否则列入列表。

7.3.1 报头字段格式

表头字段遵循与以下提供的相同通用表头格式：
第2.2节 of RFC 2822 [3]。每个报头字段由一个字段
名称后跟冒号（":"）和字段值。

字段名称: 字段值

消息头中指定的第25节的形式语法
允许在两侧有任意数量的空白字符
冒号；然而，实现应避免字段之间有空格
姓名和冒号，冒号和之间用一个空格（SP）隔开
字段值。

```
Subject:            lunch
Subject      :      lunch
Subject            :lunch
Subject: lunch
```

因此，上述所有都是有效且等价的，但最后一个是
首选形式。

表头字段可以通过在每个之前续行来扩展到多行
额外行，至少包含一个空格或水平制表符（HT）。该行
中断符和下一行的开头空白处是
视为单个SP字符。因此，以下内容是
等效：

主题：我知道你在那里，拿起电话和我说话！
主题：我知道你在那里，
拿起电话
和我说话！

头部字段中不同字段名的相对顺序是不
显著。然而，建议将标题字段{v*}
需要用于代理处理（Via，路由，Record-Route，Proxy-Require，
Max-Forwards，以及Proxy-Authorization，例如）出现在...
消息顶部以方便快速解析。 相对
字段名相同的表头字段行顺序很重要。
多个具有相同字段名的标题字段行可能存在。
只有当该报头字段的整个字段值时才发送消息
定义为逗号分隔的列表（即，如果遵循语法
定义在第7.3节中)。它**必须**能够组合多个
表头字段行合并为一个 "字段名: 字段值" 对，无需
改变消息的语义，通过附加每个后续
字段值从第一个开始，每个值用逗号分隔。 例外情况
此规则包括WWW-Authenticate、Authorization、Proxy-
认证，以及Proxy-Authorization头部字段。多个头部

字段行中可能存在这些名称，但由于
他们的语法不符合第7.3节中列出的通用形式，
它们**必须**不得合并为一个单独的报头字段行。

实现必须能够处理多个报头字段行
与任何组合的单行一个值或
逗号分隔值形式。

以下组标题字段行有效且等效：

路由：<sip:alice@atlanta.com>
主题：午餐
路由：<sip:bob@biloxi.com>
路由：<sip:carol@chicago.com>

路由：<sip:alice@atlanta.com>，<sip:bob@biloxi.com>
路由：<sip:carol@chicago.com>
主题：午餐

主题：午餐
路由：<sip:alice@atlanta.com>，<sip:bob@biloxi.com>，
<sip:carol@chicago.com>

每个以下块都是有效的，但与以下块不等价
其他：

路由：<sip:alice@atlanta.com>
路由：<sip:bob@biloxi.com>
路由：<sip:carol@chicago.com>

路由：<sip:bob@biloxi.com>
路由：<sip:alice@atlanta.com>
路由：<sip:carol@chicago.com>

路由：<sip:alice@atlanta.com>，<sip:carol@chicago.com>，
<sip:bob@biloxi.com>

每个头字段值的格式是按头名称定义的。
将始终是TEXT-UTF8字节的不可见序列，或
组合空格、标记、分隔符和引号字符串。
许多现有的报头字段将遵循以下一般形式：
值后跟一个分号分隔的参数名序列，
参数-值对：

字段名称 字段值 *(;参数名=参数- 值)

尽管可以附加任意数量的参数对到
表头字段值，任何给定的参数名不得出现更多
一次以上。

当比较头部字段时，字段名始终区分大小写-
无反应。除非在定义中另有说明，否则不适用。
特定报头字段，字段值，参数名称以及参数
值不区分大小写。标记始终不区分大小写。
除非另有说明，以引号字符串表示的值是
区分大小写。例如，

C Translator Text:<sip:alice@atlanta.com>；过期=36 00
联系人：
翻译文本：
等价于

联系：<sip:alice@atlanta.com>；ExPiReS=3600

和

内容处置：会话；处理=可选

等价于

c 内容处置：会话；处理=选项 NA
L 翻译文本：NAL
以下两个报头字段不等价：

警告：370 devnull "选择一个更大的管道"
警告：370 devnull "选择更大的管道"

7.3.2 报头字段分类

一些标题字段仅在请求或响应中才有意义。这些
被称作请求头字段和响应头字段,
分别。如果消息中出现了一个与{v*}不匹配的标题字段
其类别（例如响应中的请求头字段），它必须
被忽略。第20节定义了每个标题的分类
字段。

7.3.3 紧凑形式

SIP提供了一种机制来表示常见的头部字段名称。
缩写形式。这可能在消息否则时有用。
变得太大，无法携带在可用的运输工具上
(当使用UDP时，超过最大传输单元（MTU），对于
示例）。这些紧凑形式在第20节中定义。一个紧凑
表头字段名的较长形式可以使用form进行替换
任何时间都不改变消息的语义。一个标题

字段名称可以在同一文档中同时以长格式和短格式出现
消息。实现必须接受长格式和短格式
每个标题名称的。

7.4 体质

请求，包括在此扩展中定义的新请求
规范，除非另有说明，可能包含消息体。
身体解释取决于请求方法。

对于响应消息，请求方法和响应状态
代码确定任何消息体的类型和解释。
响应可能包含一个主体。

7.4.1 消息体类型

消息体的互联网媒体类型必须由以下提供
内容类型报头字段。如果正文经过任何编码
例如压缩，那么这必须通过Content-来指示
编码头字段；否则，必须省略内容编码。
如果适用，消息体的字符集表示为
内容类型报头字段值的部分。

The "multipart" MIME type defined in RFC 2046 [11] MAY be used within
消息正文。实现发送请求
包含多部分消息体的内容必须发送一个会话描述
作为非多部分消息体，如果远程实现请求
此通过一个不包含多部分的Accept头字段。

SIP消息可以包含二进制体或体部分。当没有
显式字符集参数由发送者提供，媒体子类型
"文本"类型被定义为具有默认字符集值
UTF-8

7.4.2 消息体长度

身体长度（以字节为单位）由Content-Length头提供
字段。第20.14节描述了此头文件所需的内容
字段详情。

HTTP/1.1的"分块"传输编码不得用于SIP。
(注意：分块编码修改了消息的主体，以便
将其作为一系列块进行传输，每个块都有自己的大小
指示器。）

7.5 建立SIP消息框架

与HTTP不同，SIP实现可以使用UDP或其他不可靠协议
数据报协议。每个这样的数据报携带一个请求或
响应。参见第18节关于不可靠使用约束的内容。
运输。

实现处理流导向的SIP消息
传输必须忽略出现在起始行之前的任何CRLF
[H4.1].

内容长度报头字段值用于定位内容的结尾
每个流中的SIP消息。它始终存在于SIP
消息通过面向流的传输发送。

8 通用用户代理行为

用户代理代表一个终端系统。它包含一个用户代理
客户端（UAC），生成请求，以及用户代理服务器
(UAS)，对此做出响应。一个UAC能够生成一个
基于某些外部刺激（用户点击按钮）的请求
或PSTN线路上的信号）并处理响应。一个UAS是
能够接收请求并根据{v*}生成响应
用户输入，外部刺激，程序执行结果，或
一些其他机制。

当UAC发送请求时，请求会通过一些数量的
代理服务器，将请求转发到UAS。当
UAS生成一个响应，该响应被转发到UAC。

UAC 和 UAS 程序强烈依赖于两个因素。首先，基于
关于请求或响应是在对话内部还是外部，
并且第二，基于请求的方法。讨论了对话框
在12节中彻底讨论；它们代表一种对等关系
在用户代理之间并通过特定的SIP方法建立，例如
作为INVITE。

在这一节中，我们讨论了UAC的方法无关规则。
UAS处理对话之外的请求时的行为。
当然包括那些自身就建立了一个 {v*} 的请求
对話。

安全程序，用于对话之外的请求和响应
在第26节中进行了描述。具体来说，存在用于 {v*} 的机制。
UAS 和 UAC 进行相互认证。 一组有限的隐私
功能也通过使用{v*}加密身体来支持
S/MIME

8.1 UAC 行为

This se 这是se动作涵盖UAC行为之外的di alog.

8.1.1 生成请求

一个由UAC提出的有效SIP请求至少必须包含
以下报头字段：To，From，CSeq，Call-ID，Max-Forwards，
并且 Via；所有这些标题字段在所有SIP中都是必填项
请求。这六个头部字段是基本构建
SIP消息的块，因为它们共同提供了大部分
关键消息路由服务包括地址处理
消息，响应的路由，限制消息传播，
消息排序，以及交易的唯一标识。
这些头部字段是除了强制请求行之外的附加信息，
包含方法、请求URI和SIP版本。

对话外发送的请求示例包括一个到 {v*} 的 INVITE
建立会话（第13节）以及用于查询的OPTIONS
功能（第11节）。

8.1.1.1 请求-URI

消息的初始 Request-URI 应设置为 {v*} 的值
To字段中的URI。一个值得注意的例外是REGISTER
方法；设置REGISTER的Request-URI的行为在 {v*}
第10节。也可能出于隐私原因而不希望这样做。
方便将这些字段设置为相同的值（尤其是如果这些字段需要设置为相同的值时）
源自UA预计在{v*}期间请求URI将会改变
中转）。

在某些特殊情况下，存在预存路由
集合可以影响消息的 Request-URI。一个预存的路径
集合是一组有序的URI，用于标识服务器链，以
哪个UAC将发送超出对话框的出站请求。
通常，它们由用户或服务提供商在UA上配置
手动，或通过某些其他非SIP机制。当提供者
希望配置一个具有出站代理的UA，建议这样做
这通过提供预存在的路由集来完成
一个单独的URI，即出站代理的URI。

当存在预存的路线集时，其程序为
填充第X节中详细说明的Request-URI和Route头字段
12.2.1.1 必须遵循（即使没有对话），使用
期望的请求URI作为远程目标URI。

8.1.1.2 至

The To header field first and foremost specifies the desired 目标头字段首先和最重要的是指定所需的
逻辑请求的“ 接收者 ”或记录地址
用户或资源，这是此请求的目标。这可能或可能不
不是请求的最终接收者。 "To" 头字段
可能包含SIP或SIPS URI，但它也可能使用其他URI
方案（例如，tel URL（RFC 2806 [9]））在适当的时候。
所有SIP实现必须支持SIP URI方案。任何
实现支持TLS的必须支持SIPS URI方案。
The To header field allows for a display name. 目标头字段允许显示名称。

一个UAC可能学会如何为特定填充“ 收件人 ”头字段
以多种方式请求。通常用户会建议使用 To
标题字段通过人机界面，可能输入URI
手动或从某种地址簿中选择它。经常，
用户不会输入完整的URI，而是一个数字字符串
或字母（例如，“ bob ”）。由UA自行决定
选择如何解释这个输入。使用字符串来形成
用户部分SIP URI表示UA希望该名称为
在域中解决到at符号右侧（RHS）
SIP URI（例如，sip:bob@example.com）。使用字符串来
从SIPS URI的用户部分推导出，表示UA（用户代理）希望
安全通信，并且该名称应在以下中进行解析
域名移至at符号的右侧。右侧通常会经常是
请求者的主域名，允许主域进行
处理发出的请求。这对于像{v*}这样的功能很有用。
"快速拨号"需要解释用户部分在主页中的情况
域。当UA不希望指定时，可以使用tel URL。
该域应解释的电话号码
用户输入。相反，每个请求通过的域
通过将会获得那个机会。例如，一个用户在
机场可能通过出口代理登录并发送请求
机场。如果他们输入“ 411 ”（这是当地电话号码）
美国电话簿查询服务），需要被
由机场的外发代理解释和处理，不是
用户的域。在这种情况下，tel:411将是正确的
选择。

A 请求必须在对话之外，不得包含 To 标签；标签在
请求的“ 收件人 ”字段标识了对话的对方。由于
未建立对话，没有标签存在。

有关To头字段的更多信息，请参阅第20.39节。
以下是一个有效的To标题字段的示例：

收件人：Carol <sip:carol@chicago.com>

8.1.1.3 从

The From header field indicates the logical identity of the initiator From头字段表示发起者的逻辑标识
请求中，可能是用户的记录地址。类似于“ 收件人 ”
表头字段，它包含一个URI和可选的显示名称。
由SIP元素使用以确定应用哪些处理规则
一个请求（例如，自动呼叫拒绝）。因此，它是
非常重要的是，从URI中不包含IP地址或FQDN
主机上的UA运行，因为这些不是逻辑的
名称。

The From header field allows for a display name. A UAC SHOULD use
显示名称 "匿名"，以及语法正确但
否则无意义的URI（如 sip:thisis@anonymous.invalid），如果
客户端的身份应保持隐藏。

通常，填充请求中From头字段的价值
由特定UA生成的是由用户或由用户预配置的
用户本地域的管理员。如果特定的UA是
由多个用户使用，它可能具有可切换的配置文件
包含一个与被分析用户身份相对应的URI。
请求的接收者可以验证请求发起者的身份
为了确定它们是它们From头字段中所声称的
声明他们是（有关认证的更多信息，请参阅第22节）。

The From field MUST contain a new "tag" parameter, chosen by the UAC. 翻译文本：From字段必须包含由UAC选择的新
查看第19.3节以获取有关选择标签的详细信息。

有关“ From ”头字段的高级信息，请参阅第20.20节。
示例：

从: "Bob" <sips:bob@biloxi.com> ;标签=a48s
从：sip:+12125551212@phone2net.com;标签=887s
从：匿名 <sip:c8oqz84zk7z@隐私.org>；标签=hyh8

8.1.1.4 通话标识

呼叫ID头字段充当一个唯一标识符，用于分组
一起一系列消息。它对于所有请求都必须相同
并且由任一UA在对话中发送的响应。它应当相同
在每次从UA的注册中。

在一个由任何对话框之外的UAC创建的新请求中，Call-ID
表头字段必须由UAC选择为全局唯一
标识符在时间和空间上除非被方法特定覆盖
行为。所有SIP用户代理都必须有一种方式来保证{v*}的呼叫
ID头字段它们产生的将不会意外生成
任何其他UA。注意，当请求在特定情况下重试后

请求修正的失败响应（对于
示例，认证的一个挑战），这些重试的请求是
未考虑新的请求，因此不需要新的Call-ID
表头字段；参见第8.1.3.5节。

使用密码学随机标识符（RFC 1750 [12]）在
生成 Call-IDs 是 推荐的。 实现可以 使用
form "本地ID@主机". Call-IDs 区分大小写，并且只是
逐字节比较。

使用密码学随机标识符提供了一些
防护会话劫持并降低其可能性
非故意 Call-ID 冲突。

无需配置或人机界面即可选择 {v*}
请求的Call-ID报头字段值。

关于Call-ID头部字段的更多信息，请参阅第
20.8.

示例： Translated Text：示例：

呼叫标识符：f81d4fae-7dec-11d0-a765-00a0c91e6bf6@foo.bar.com

8.1.1.5 CSeq

CSeq头字段用作识别和排序的方式
交易。它由一个序列号和一个方法组成。
方法必须与请求的方法匹配。对于非-REGISTER请求
在对话框之外，序列号值是任意的。
序列号值必须可以表示为32位无符号
整数且必须小于2**31。只要它遵循上述
指南，客户可以使用它喜欢的任何机制来选择
CSeq报头字段值。

第12.2.1.1节讨论了请求CSeq的构建
在对话框中。

示例： Translated Text：示例：

CSeq: 4711 邀请

8.1.1.6 最大转发数

Max-Forwards报头字段用于限制数据包跳转的次数
请求可以在前往目的地的途中传输。它由一个
整数在每个跳数中减一。如果Max-Forwards
值在请求到达目的地之前达到0，它将
被拒绝，并返回483（跳数过多）错误响应。

每个请求都必须包含一个Max-Forwards头部字段，UAC必须插入。
起源于一个值，该值应应为70。这个数字被选择以
足够大以保证请求不会被
在任何SIP网络中，当没有环路时掉落，但不是很大
关于循环发生时消耗代理资源。较低的值
应谨慎使用，并且仅在拓扑结构为 {v*} 的网络中使用
已知由UA。

8.1.1.7 通过

The Via header field indicates the transport used for the transaction Via头字段表示用于事务的传输
并识别响应应发送的位置。 一个 Via
表头字段值仅在将要传输的传输之后添加
已选择用于到达下一跳的路径（这可能涉及{v*}）。
使用[4]中的过程。

当UAC创建请求时，它必须在该请求中插入一个Via。
请求。头部字段中的协议名称和协议版本
必须分别是SIP和2.0。 Via头字段值必须
包含一个分支参数。此参数用于识别
由该请求创建的交易。此参数同时被两者使用。
客户端和服务器。

The branch parameter value MUST be unique across space and time for
所有由UA发送的请求。此规则的例外是 CANCEL
并且ACK对于非2xx响应。如以下所述，一个CANCEL请求
将具有与请求相同的分支参数值
取消。如第17.1.1.3节所述，对于非2xx的ACK
响应也将具有与INVITE相同的分支ID
响应它确认的。

分支ID参数的唯一性属性，以便
其作为交易ID的使用，不是RFC 2543的一部分。

该分支ID由符合本规范的元素插入
规范必须始终以字符 "z9hG4bK" 开头。
7个字符用作魔法饼干（认为7个足够了）
确保一个较旧的RFC 2543实现不会选择这样的
值)，以便接收请求的服务器可以确定
分支ID是按照本描述的方式构建的

规范（即全局唯一）。在此要求之外，
分支令牌的精确格式是实现定义的。

The Via header maddr, ttl, and sent-by components will be set when
请求由传输层处理（第18节）。

通过代理的处理在16.6节第8项中描述。
第16.7节项目3。

8.1.1.8 联系

The Contact header field provides a SIP or SIPS URI that can be used 联系人头字段提供了一个SIP或SIPS URI，可用于
联系该特定实例的UA以进行后续请求。
联系头部字段必须存在且必须恰好包含一个SIP
或SIPS URI在任何可能导致建立连接的请求中
对话。对于本规范中定义的方法，这包括
仅邀请请求。对于这些请求，作用域为
联系是全局的。也就是说，联系报头字段值包含
URI，其中UA希望接收请求，以及此URI
MUST 必须有效，即使在任何后续请求中使用也是如此
对話框。

如果请求URI或顶级路由头字段值包含SIPS
URI，联系头字段必须包含一个SIPS URI。

关于Contact头部字段的更多信息，请参阅第
20.10.

8.1.1.9 支持和需要

如果UAC支持可应用于SIP的扩展
服务器响应中，UAC 应包含一个 Supported 报头
字段在请求中列出选项标签（第19.2节）的选项
扩展。

The option tags listed MUST only refer to extensions defined in
标准跟踪RFC。这是为了防止服务器坚持
客户端实现非标准、供应商定义的功能，以便
接收服务。由实验定义的扩展
信息性RFC明确排除与以下内容的用法
请求中支持的头部字段，因为它们也经常被用来
文档供应商定义的扩展。

如果UAC希望坚持让UAS理解一个扩展，
UAC将对请求应用以处理请求，它
必须将一个包含请求中列出内容的 Require 标头字段插入到请求中。
选项标签为此扩展。如果UAC希望应用
扩展请求并坚持任何代理

遍历理解该扩展，它必须插入一个Proxy-Require
请求中包含该选项标签的标题字段
扩展。

与支持的报头字段一样，要求选项标签中的 {v*} 保持不变。
并且Proxy-Require头字段必须仅引用定义的扩展
在标准跟踪的RFC中。

8.1.1.10 附加消息组件

在创建了一个新的请求之后，并且描述了头部字段
以上已正确构建，任何额外的可选标题
字段被添加，以及任何特定于该方法的头字段也被添加。

SIP请求可能包含MIME编码的消息体。 无论如何
请求包含的正文类型，某些头部字段必须
被制定来表征身体的内容。 对于进一步的
关于这些头部字段的信息，请参阅第20.11至20.15节。

8.1.2 发送请求

请求的目标随后被计算。除非有
本地策略另有规定时，目的地必须确定
通过应用[4]中描述的DNS程序如下。如果
第一个元素在路由集中表示一个严格路由器（导致
在形成如第12.2.1.1节所述的请求时，
请求的Request-URI必须应用程序。
否则，将程序应用于第一个路由头字段
请求中的值（如果存在），或请求的Request-URI
如果没有存在路由头字段。这些程序产生
有序的地址、端口和尝试使用的传输集合。独立
其中URI用作[4]中程序的输入，如果
请求URI指定了一个SIPS资源，UAC必须遵循
[4]中的程序，如同输入URI是一个SIPS URI。

本地策略可以指定一个尝试的替代目的地集。
如果请求URI包含SIPS URI，任何替代目的地
必须使用TLS进行联系。除此之外，没有限制
在替代目的地，如果请求不包含路由头
字段。这为现有的路线提供了一个简单的替代方案
将作为指定出站代理的方式。然而，那种方法
配置出站代理不建议；一个预存在的
使用单个URI设置的路径应该被使用。如果请求
包含一个路由头字段，请求应发送到
从其最高值派生的位置，但可能被发送到任何
服务器，UA确信将遵守路由和请求-URI
此文档中指定的策略（与RFC中的策略相反）
2543). 尤其是配置了出站代理的UAC应该

尝试将请求发送到第一个指示的位置
路由报头字段值而不是采用发送策略
所有发送到出口代理的消息。

这确保了不添加Record-Route的出站代理
表头字段值将从后续路径中消失
请求。它允许无法解析第一个路由的端点
URI用于委托该任务给出口代理。

UAC 应遵循 [4] 中定义的状态管理程序
元素，尝试每个地址，直到联系到服务器。每次尝试
构成一笔新交易，因此每笔交易都不同
最高 Via 头字段值，带有新的分支参数。
此外，Via头字段中的传输值设置为
无论确定的目标服务器使用何种运输方式。

8.1.3 处理响应

响应首先由传输层处理，然后传递
到事务层为止。事务层执行其
处理后将响应传递给TU。大多数
响应处理在TU中是方法特定的。然而，有
某些一般行为与方法无关。

8.1.3.1 事务层错误

在某些情况下，事务层返回的响应将
不是SIP消息，而是一个事务层错误。当
超时错误来自事务层，它必须
被视为已收到408（请求超时）状态码。
如果传输层报告了致命的传输错误
通常，由于UDP中的致命ICMP错误或连接失败
TCP)，该条件必须被视为503（服务不可用）
状态码

8.1.3.2 不可识别的响应

A UAC 必须将其不识别的任何最终响应视为
等同于该类别的x00响应代码，并且必须能够
处理所有类别的 x00 响应代码。例如，如果 a
UAC 收到未识别的响应代码 431，它可以安全地
假设其请求存在问题，并对其进行处理
作为如果它收到了一个400（错误请求）响应代码的响应。
UAC 必须将任何不同于 100 的临时响应与其他处理方式不同
无法识别为183（会话进度）。UAC必须能够
处理100和183个响应。

8.1.3.3 通孔

如果响应中存在多个 Via 头部字段值，则
UAC 应该丢弃该消息。

存在先于的附加 Via 标头字段值
请求发起者建议，该消息是
误路由或可能损坏。

8.1.3.4 处理3xx响应

收到重定向响应后（例如，301 响应
状态码），客户端应使用Contact头中的URI(s)
字段以根据重定向制定一个或多个新请求
请求。此过程类似于代理递归的过程
3xx 类响应，如第 16.5 节和 16.6 节所述。一个客户端
以包含恰好一个URI的初始目标集开始，该URI为 {v*}
原始请求的请求URI。如果客户端希望制定
基于对该请求的3xx类响应的新请求，它放置
尝试放入目标集的URI。受以下限制条件约束：
此规范中，客户端可以选择它放置哪些联系URI
进入目标集。与代理递归一样，客户端处理
3xx类响应不得将任何给定的URI添加到目标集更多
比一次。如果原始请求中包含SIPS URI在请求-
URI，客户端可以选择递归到一个非SIPS URI，但应该
通知用户重定向到不安全的URI。

任何新的请求可能会收到包含{v*}的3xx响应。
原始URI作为联系人。 可以配置两个位置。
相互重定向。将任何给定的URI放置在目标集合中
仅一次可防止无限重定向循环。

随着目标集的增长，客户端可以生成新的请求到
URI 任意顺序。一种常见的机制是按“ q ”对集合进行排序。
参数值来自Contact头字段值。 对{v*}的请求。
URI 可能是顺序生成或并行生成。一种方法是
按递减的q值分组顺序处理，并处理URI
在每个q值组中并行。另一种是仅进行串行
按递减的q值顺序处理，任意选择之间
等q值联系人。

如果联系列表中的地址失败，如定义所述，{v*}
在下一段中，元素移动到下一个地址中
列表，直到列表耗尽。如果列表耗尽，那么
请求失败。

失败应通过失败响应代码（代码
大于399); 对于网络错误，客户端事务将
报告任何传输层故障给事务用户。 注意
某些响应代码（详见8.1.3.5）表示
请求可以重试；重试的请求不应
考虑到的故障。

当收到特定联系地址的故障时，
客户端应尝试下一个联系地址。这将涉及
创建一个新的客户端事务以发送新的请求。

为了根据3xx中的联系地址创建一个请求
响应，UAC 必须将目标集合中的整个 URI 复制到
请求URI，除了“ 方法参数 ”和“ 头部 ” URI
参数（参见第19.1.1节以了解这些参数的定义）。
它使用“ header ”参数来创建头部字段值，用于
新请求，覆盖与以下相关的头字段值 {v*}
根据第X节中的指南进行重定向请求
19.1.5.

请注意，在某些情况下，已被翻译的标题字段
在联系地址中传达的，可能反而可以附加到现有的
原始重定向请求中的请求头字段。 作为一个
一般规则，如果标题字段可以接受逗号分隔的列表
值，然后新的标题字段值可以附加到任何
现有值在原始重定向请求中。如果头部
字段不接受多个值，原始值中的
重定向请求可能会被头部字段值覆盖
在联系地址中传达。例如，如果联系
地址以以下值返回：

sip:用户@主机?主题=foo&呼叫信息=<http://www.foo.com>

然后，原始重定向请求中的任何Subject头字段是
被覆盖，但HTTP URL仅附加到任何现有
呼叫信息报头字段值。

建议UAC重用相同的To、From和Call-ID
在原始重定向请求中使用，但UAC也可能选择
更新新请求的Call-ID头部字段值，对于
示例。

最后，一旦构建了新的请求，它就使用 {v*} 发送。
一个新客户交易，因此必须在
顶级Via字段，如第8.1.1.7节所述。

在其他所有方面，收到重定向后发送的请求
响应应重新使用原始的报头字段和主体
请求。

在某些情况下，联系头部字段值可能在UAC中被缓存
根据接收到的状态码临时或永久地
存在一个过期间隔；参见第21.3.2节和
21.3.3.

8.1.3.5 处理4xx响应

某些4xx响应代码需要特定的UA处理，
与方法无关。

如果是一个401（未授权）或407（需要代理身份验证）
响应已接收，UAC 应遵循授权
22.2和22.3节中的程序以重试请求
凭证

如果收到413（请求实体过大）响应（第{v*}节）
21.4.11)，请求包含了一个比UAS更长的主体
愿意接受。如果可能，UAC 应该重试
请求，或者省略正文或使用较短长度的一种。

如果收到415（不支持的媒体类型）响应（第
21.4.13)，请求中包含UAS不支持的多媒体类型。
UAC 应该重试发送请求，这次只使用
内容包含在响应的Accept头部字段中列出的类型，
使用Accept-Encoding头字段中列出的编码
响应，并且带有在 Accept-Language 中列出的语言
响应。

如果收到416（不支持的URI方案）响应（第
21.4.14)，请求URI使用了不被支持的URI方案
服务器。客户端应重试请求，这次使用SIP
统一资源标识符

如果收到420（错误扩展）响应（第21.4.15节），则
请求包含了一个列出 {v*} 的 Require 或 Proxy-Require 报头字段
选项标签，用于表示由代理或UAS不支持的功能。 UAC
应该重试请求，这次省略掉在 中列出的任何扩展
响应中不支持的头字段。

在所有上述情况下，请求通过创建一个新的来重试
请求进行适当的修改。此新请求
构成一笔新交易，并且应当具有与{v*}相同的值
呼叫标识符（Call-ID）、接收方（To）和发送方（From）的上一请求，但CSeq应该
包含一个比上一个高1的新序列号。

与其他4xx响应一样，包括尚未定义的，进行重试
可能或可能不可能，这取决于方法和用例。

8.2 UAS 行为

当外部请求由UAS处理时，有
处理规则集合，独立于方法遵循。
第12节提供了关于UAS如何判断一个请求的指导。
内部或外部对话框。

请注意，请求处理是原子的。如果请求被接受，
所有与其相关的状态更改都必须执行。
已拒绝，所有状态变更**必须**不执行。

UASs 应当按照步骤的顺序处理请求
在此节中跟随（即，从认证开始，然后
检查方法、头部字段等。
本节剩余部分）。

8.2.1 方法检查

一旦请求被验证（或跳过验证），则
UAS 必须检查请求的方法。如果 UAS 识别
但是不支持请求的方法，它必须生成405
(方法不允许) 响应。生成响应的流程
在第8.2.6节中描述。UAS还必须添加一个允许
表头字段到405（方法不允许）响应。 允许
表头字段必须列出UAS支持的方法集
生成消息。 允许头字段呈现于
第20.5节。

如果服务器支持该方法，则继续处理。

8.2.2 报头检查

如果UAS不理解请求中的一个标题字段（即，
表头字段在本规范或任何规范中均未定义
支持扩展），服务器必须忽略该报头字段并
继续处理消息。一个UAS应忽略任何格式错误的
标题字段，这些字段对于处理请求不是必需的。

8.2.2.1 请求和Request-URI

请求的“至”报头字段标识原始请求的收件人
由“From”字段中指定的用户标识。 原始
接收者可能是也可能不是处理请求的UAS，因为
呼叫转接或其他代理操作。一个UAS可以应用任何
策略它希望确定是否接受当目标为的请求

表头字段不是UAS的标识。然而，它是
建议UAS即使不识别也接受请求
URI方案（例如，tel: URI），在“收件人”报头字段中，或
如果To头字段没有针对已知或当前用户
这个UAS。另一方面，如果UAS决定拒绝
请求，它应该生成一个带有403（禁止）状态码的响应
代码并将其传递给服务器事务进行传输。

然而，Request-URI标识了将要处理{v*}的UAS。
请求。如果请求URI使用的是UAS不支持的计划，
它应该拒绝带有416（不支持的URI方案）的请求
响应。如果 Request-URI 没有标识一个地址，该地址
UAS愿意接受对{v*}的请求，它应该拒绝该请求
使用404（未找到）响应。通常，一个使用{v*}的UA
注册方法将其记录地址绑定到特定联系人
地址将看到请求，其Request-URI等于该联系
地址。其他潜在接收 Request-URIs 的来源包括
请求和响应中由UA发送的“联系”报头字段
建立或刷新对话。

8.2.2.2 合并请求

如果请求在“收件人”头字段中没有标签，UAS核心必须
检查请求与当前交易的一致性。如果来源标签，
呼叫ID和CSeq与正在进行的那些完全匹配
交易，但请求与该交易不匹配（基于
关于第17.2.3节中的匹配规则，UAS核心应
生成一个482（循环检测）响应并将其传递给服务器
交易。

同一次请求已多次到达UAS，随后
不同的路径，很可能是由于分支造成的。UAS处理
第一个此类请求收到并响应以482（循环
检测到的) 到其余部分。

8.2.2.3 需求

假设UAS决定它是适当的处理元素
请求，它检查是否存在 Require 头字段。

The Require header field is used by a UAC to tell a UAS about SIP 需要头字段由UAC用于告知UAS关于SIP的信息
扩展，UAC期望UAS支持以
正确处理请求。其格式在章节中描述。
20.32. 如果一个UAS不理解在选项标签中列出的{v*}
需要头部字段，它必须通过生成响应来响应
状态码 420（无效扩展）。UAS 必须添加一个不支持的
表头字段，并在其中列出它不理解的那些选项
在请求的“Require”头部字段中。

请注意，在SIP CANCEL中不得使用Require和Proxy-Require
请求，或者在一个发送给非2xx响应的ACK请求中。这些
标题字段如果存在于这些请求中，则必须忽略。

一个针对2xx响应的ACK请求必须只包含那些Require和
代理-需要值，这些值存在于初始请求中。

示例： Translated Text：示例：

UAC->UAS: INVITE sip:watson@bell-telephone.com SIP/2.0
需要：100rel

UAS->UAC: SIP/2.0 420 错误扩展
不支持：100rel

此行为确保客户端-服务器交互将
立即进行，当双方都理解所有选项时
边，并且只有在选项不被理解时才会减速（如在{v*}中）。
示例上方）。对于匹配良好的客户端-服务器对，
交互迅速进行，节省了通常所需的往返一次
通过谈判机制。此外，它还消除了歧义
当客户端需要服务器不具备的功能时
理解。一些功能，例如呼叫处理字段，仅
对终端系统感兴趣。

8.2.3 内容处理

假设UAS理解客户端所需的任何扩展，
UAS 检查消息体以及头部字段，和
描述它。如果存在任何类型（由 {v*} 指示）的实体。
内容类型)，语言（由内容语言指示）或
编码（由Content-Encoding指示）无法理解，并且
该身体部位不是可选的（如内容-所示）
处置头字段），UAS 必须拒绝请求并返回 415 状态码
(不支持的媒体类型) 响应。响应必须包含一个
接受列出了它理解的所有身体类型的Accept头字段，
在请求包含不支持类型的正文的情况下
UAS。如果请求包含无法理解的编码内容
由UAS，响应必须包含一个Accept-Encoding头字段
列出UAS理解的编码。如果请求
包含UAS无法理解的语言的内容
响应必须包含一个Accept-Language头部字段，指示{v*}。
UAS所理解的语言。 除了这些检查之外，机体处理
取决于方法和类型。有关更多信息，请参阅
处理特定内容的首部字段，参见第7.4节
作为第20.11至20.15节。

8.2.4 应用扩展

一个希望在生成时应用某些扩展的UAS
响应不得这样做，除非支持该扩展
指示在请求中的“ 支持 ”标题字段中。
期望的扩展不被支持，服务器应仅依赖于
基线SIP和客户端支持的任何其他扩展。 在
罕见情况下，服务器无法处理请求
没有扩展，服务器可能会发送421（需要扩展）
响应。此响应表示无法进行适当的响应
生成，无需特定扩展支持。所需的
扩展（s）必须在请求头字段中包含
响应。此行为不建议，因为它通常
中断互操作性。

任何应用于非421响应的扩展必须在其中列出
需要包含在响应中的头部字段。当然，服务器
必须不应用在“ 支持 ”头字段中未列出的扩展。
请求。因此，在的“ Require ”头字段中
响应将仅包含在标准中定义的选项标签
跟踪RFC。

8.2.5 处理请求

假设前一小节中的所有检查都通过，
UAS处理变得方法特定。第10节涵盖了
注册请求，第11节涵盖OPTIONS请求，第13节
覆盖了INVITE请求，第15节覆盖了BYE请求。

8.2.6 生成响应

当UAS希望对一个请求构建响应时，它遵循
以下子节中详细说明的一般程序。
针对特定响应代码的附加行为，其中
此部分未详细说明，可能也需要。

一旦所有与创建响应相关的程序都已
已完成，UAS将响应返回给服务器
交易请求来源。

8.2.6.1 发送临时响应

一个主要非方法特定的生成指南
响应是，UASs 应当 **不** 发布一个临时响应
非邀请请求。相反，UAS应生成最终响应以
尽快发送非-INVITE请求。

当生成一个100（尝试）响应时，任何时间戳头字段
存在于请求中的内容必须复制到这个100（尝试中）
响应。如果生成响应有延迟，UAS
应在响应中的时间戳值中添加一个延迟值。
the value ǒ u xiào zhí zhí hán su ǒ y ī n wèi shí ji ā n de bié zhì zhèng zhèng zhí zhí zhí zhí zhí zhí zhí zhí zhí zhí zhí zhí zh
响应和请求接收，以秒为单位。

8.2.6.2 标题和标签

响应的“ From ”字段必须等于“ From ”报头字段
请求。响应的 Call-ID 报头字段必须等于
请求的 Call-ID 报头字段。 CSeq 报头字段 of the
响应必须等于请求的CSeq字段。 Via头
响应中的字段值必须等于 Via 报头字段值
在请求中并必须保持相同的顺序。

如果请求中包含一个To标签，则To报头字段
在响应中必须等于请求中的。然而，如果目标地址
请求中的标题字段未包含标签，收件人中的URI
响应中的标题字段必须等于To标题中的URI
字段；此外，UAS 必须在 To 头部字段中添加一个标签
响应（除100（尝试）响应外，在
哪个标签可能存在)。这用于识别UAS。
响应，可能产生一个对话ID的组件。
相同的标签必须用于对该请求的所有响应，包括最终的
并且临时（再次排除100（尝试））。程序为
标签的生成在19.3节中定义。

8.2.7 无状态UAS行为

无状态UAS是指不维护事务状态的UAS。
它通常回复请求，但丢弃任何会 {v*} 的状态
通常在发送响应后由UAS保留。
无状态UAS收到一个请求的重传，它重新生成
响应并重新发送它，就像是在回复第一次一样
请求实例。一个UAS不能是无状态的，除非请求
处理该方法始终会产生相同的响应
如果请求相同。这排除了无状态注册器，
例如。无状态UAS不使用事务层；它们
直接从传输层接收请求并发送响应
直接到传输层。

无状态UAS角色主要用于处理未经认证的
请求发出挑战响应的请求。如果
未经验证的请求被状态化处理，然后是恶意的
洪水般的未经验证的请求可能会产生大量

交易状态可能会减慢或完全停止呼叫处理
在一个UAS中，有效地创建拒绝服务条件；对于
更多信息请参阅第26.1.5节。

最无状态UAS最重要的行为如下：

- o 无状态UAS不得发送临时（1xx）响应。
- o 无状态UAS绝不能重传respo nses.
- o 无状态UAS必须忽略ACK请求。
- o 无状态UAS必须忽略取消请求。
- o 对于无状态的响应，必须生成标题标签
方式 - 以一种将生成相同标签的方式
相同请求持续。有关标签构建信息
查看第19.3节。

在所有其他方面，无状态UAS的行为与
一个有状态的UAS。UAS可以以有状态或无状态的方式运行
每个新请求的模式。

8.3 重定向服务器

在某些架构中，可能希望减少处理
负载在负责路由请求的代理服务器上，以及
提高信号路径的鲁棒性，通过依赖重定向。

重定向允许服务器推送请求的路由信息
在回复客户的回应中，从而将自己排除在外
此交易进一步的通讯循环，同时仍在协助
在定位请求的目标时。当请求的发起者时。
请求接收到重定向，它将根据 {v*} 发送新的请求
它接收到的URI（s）。通过从核心传播URI。
网络到其边缘，重定向允许网络有相当大的
可扩展性。

重定向服务器在逻辑上由一个服务器事务构成
层和一个可以访问位置服务的交易用户
某些类型（有关注册机构和位置的更多信息，请参阅第10节）
服务）。此位置服务实际上是一个数据库
包含单个URI与一组一个或多个的映射
替代位置，其中可以找到该URI的目标。

重定向服务器不发出任何自己的SIP请求。之后
接收到的请求不是 CANCEL，服务器要么拒绝
请求或收集替代位置的列表

位置服务并返回一个3xx类别的最终响应。
格式良好的CANCEL请求，它应该返回2xx响应。
响应结束SIP事务。重定向服务器维护
事务状态，用于整个SIP事务。它是
客户检测重定向之间转发循环的责任
服务器。

当一个重定向服务器向请求返回3xx响应时，它
填充（一个或多个）替代位置的列表到
联系头部字段。 Contact头部的一个“ expires ”参数
字段值也可以提供以指示{v*}的寿命
联系数据。

The Contact header field contains URIs giving the new locations or
用户名尝试列表，或可能只需指定额外的运输
参数。一个 301（永久移动）或 302（临时移动）
响应也可能给出相同的{v*}和用户名
针对初始请求进行定位但指定额外运输
参数，例如尝试不同的服务器或组播地址，或
SIP传输从UDP更改为TCP或反之亦然。

然而，重定向服务器**必须**不将请求重定向到等于{v*}的URI
到请求URI中的那个；相反，只要URI执行
不指向自身，服务器可以代理请求到
目标URI，或者可能拒绝它并返回404错误。

> 如果客户端正在使用出站代理，并且该代理实际上
> 重定向请求，可能出现无限重定向的情况
> 循环。

请注意，一个“ 联系 ”头字段值也可能引用不同的 {v*}
资源比最初所叫的更多。例如，一个SIP呼叫
连接到PSTN网关可能需要传递特殊信息
公告，例如“ 您拨打的号码已更改。 ”

A Contact 响应头字段可以包含任何合适的 URI
指示被叫方可被联系的位置，不仅限于SIP
统一资源标识符。例如，它可能包含电话、传真或irc（如果
它们被定义)或mailto:（RFC 2368 [32]）URL。第26.4.4节
讨论将SIPS URI重定向到其他URI的启示和局限性
非SIPS URI。

The "expires" parameter of a Contact header field value indicates how
只要URI有效。参数的值是一个数字
指示秒数。如果此参数未提供，则使用{v*}的值。
过期头字段决定了URI的有效时长。
无效的值应视为等同于3600。

这提供了与RFC的适度向后兼容性
2543，允许在此报头字段中使用绝对时间。
绝对时间被接收，它将被视为格式错误，并且
然后默认为3600。

重定向服务器必须忽略不理解的功能
(包括未识别的标题字段，任何未知选项标签在
需要，甚至方法名称）并继续进行重定向
相关请求。

9 取消请求

上一节讨论了生成{v*}的一般UA行为
请求和所有方法请求的处理响应。 在
本节，我们讨论一种通用方法，称为 CANCEL。

取消请求，正如其名所示，用于取消之前的
客户端发送的请求。具体来说，它要求UAS停止
处理请求并生成相应的错误响应
请求。 CANCEL 对已由 UAS 处理的请求无影响
已给出最终响应。因此，这最有用
取消那些服务器需要很长时间才能处理的请求
响应。因此，对于INVITE请求，最好使用CANCEL。
可能需要很长时间才能生成响应。在这种情况下，一个UAS
收到一个对INVITE的取消请求，但尚未发送的
最终响应，将“ 停止响铃 ”，然后响应INVITE
带有特定的错误响应（一个487）。

取消请求可以由代理和用户构建和发送
代理客户端。第15节讨论了在什么条件下UAC
将取消一个邀请请求，第16.10节讨论了代理
使用 CANCEL。

一个有状态的代理响应 CANCEL，而不是简单地转发
从下游元素接收到的响应。 对于那个
原因，CANCEL 被称为“ 逐跳 ”请求，因为它是一个
在每个状态代理跳转处做出响应。

9.1 客户行为

一个 CANCEL 请求不应该发送以取消除 {v*} 之外的其他请求
邀请。

由于除了INVITE之外的其他请求都会立即得到响应，
发送一个针对非-INVITE请求的CANCEL会始终创建一个
竞态条件。

以下步骤用于构建一个 CANCEL 请求。
请求URI、Call-ID、To、CSeq的数字部分和From报头
字段在 CANCEL 请求中必须与那些在 中的字段相同
请求正在取消，包括标签。由一个 CANCEL 构造的
客户端必须只有一个与以下匹配的 Via 报头字段值
请求中顶值被取消。使用相同的值
对于这些标题字段允许将 CANCEL 与之匹配
请求取消（第9.2节说明了此类匹配如何发生）。
然而，CSeq头字段的“ 方法 ”部分必须有一个值
取消。这使得它可以被识别和处理。
交易本身（见第17节）。

如果正在取消的请求包含一个路由头字段，则
请求取消（CANCEL）必须包含该路由头字段（Route）的值。

这是必要的，以便无状态代理能够路由 CANCEL
请求适当。

取消请求不得包含任何 Require 或 Proxy-Require
表头字段。

一旦构造了 CANCEL，客户端应检查它
已收到对请求的任何回应（临时或最终）
被取消（以下简称“ 原始请求 ”）。

如果未收到临时响应，则必须发出 CANCEL 请求
不能发送；相反，客户端必须等待{v*}的到来
临时响应，在发送请求之前。如果原始
请求已生成最终响应，取消不应
已发送，因为它是一个有效的无操作，因为取消（CANCEL）没有影响
请求已生成最终响应的请求。
客户端决定发送 CANCEL，它创建一个客户端事务
对于 CANCEL 并将 CANCEL 请求及其传递过去
目标地址、端口和传输。目标地址，
端口和用于取消的运输必须与所使用的相同
发送原始请求。

如果允许在收到响应之前发送 CANCEL
对于前一个请求，服务器可以接收 CANCEL
在原始请求之前。

请注意，与原始请求对应的交易
并且 CANCEL 交易将独立完成。然而，一个
UAC 取消请求不能依赖于收到 487（请求
终止)对原始请求的响应，作为一个RFC 2543-
符合规定的UAS不会产生这样的响应。如果不存在
最终请求的原始响应在64*T1秒内（T1是）

定义在第17.1.1.1节中)，客户端应随后考虑
原始交易已取消且应销毁客户端
处理原始请求的交易。

9.2 服务器行为

取消方法请求服务器端的TU取消一个
待处理交易。TU确定交易为
已通过发出 CANCEL 请求取消，然后假设
请求方法绝对不是 CANCEL 或 ACK，并应用
交易匹配程序第17.2.3节。匹配
交易是需要取消的。

服务器上处理 CANCEL 请求取决于请求的类型
服务器。一个无状态的代理会转发它，一个有状态的代理可能会
响应它并生成一些自己的取消请求，以及一个UAS
将对此做出响应。参见第16.10节关于取消的代理处理。

无人机系统首先根据通用无人机系统处理 CANCEL 请求
处理在第8.2节中描述。然而，由于CANCEL请求
逐跳且不能重新提交，它们不能被质疑
通过服务器以获取适当的凭据进行授权
表头字段。注意，取消（CANCEL）请求不包含{v*}。
需要头部字段。

如果UAS未找到与取消匹配的交易
根据上述程序，它应该响应 CANCEL
带有481（调用腿/交易不存在）。如果交易
对于原始请求仍然存在，UAS 的行为在
接收一个 CANCEL 请求取决于它是否已经发送了
原始请求的最终响应。如果有的话，取消
请求对原始请求的处理没有影响，无
对任何会话状态无影响，对生成的响应也无影响
对于原始请求。如果UAS尚未发布最终响应
对于原始请求，其行为取决于方法{v*}。
原始请求。如果原始请求是INVITE，则UAS
应立即用487（请求
终止）。一个取消请求对处理没有影响
与在本规范中定义的任何其他方法进行的交易。

无论原始请求的方法如何，只要{v*}
取消匹配了一个现有交易，UAS回答了取消
请求本身返回200（OK）响应。此响应是
根据第8.2.6节所述的程序构建
注意到对 CANCEL 的响应的 To 标签以及 To 标签
在响应原始请求中应当相同。
响应 CANCEL 被传递到服务器事务中
传输

10 注册

10.1 概述

SIP提供了一种发现功能。如果用户想要启动
与另一个用户的会话中，SIP必须发现当前主机({v*})
目标用户可到达的。此发现过程是
经常由SIP网络元素如代理服务器完成
并且重定向负责接收请求的服务器，
确定发送位置，基于对位置的知晓
用户，然后将其发送到那里。为此，SIP网络
元素咨询一个名为位置服务的抽象服务，
为特定域提供地址绑定。这些
地址绑定将传入的SIP或SIPS URI映射，sip:bob@biloxi.com，
例如，到一个或多个“ 某种程度更接近 ”的 URI。
期望用户，例如 sip:bob@engineering.biloxi.com。
最终，一个代理将咨询一个将位置映射的定位服务。
接收到的URI发送到用户代理，其中包含所需的收件人
目前居住。

注册在位置服务中为特定位置创建绑定
域，将一个地址记录URI与一个或多个相关联
联系地址。因此，当该域的代理收到
请求其Request-URI与记录地址匹配的请求，代理
将请求转发到注册的联系人地址
记录地址。 通常情况下，只注册一个地址是有意义的
域名位置服务中的记录地址
该记录地址将被路由到该域名。 在大多数
案例，这意味着注册的域将需要
匹配地址记录的URI中的域。

有许多方式可以获取位置服务的内容。
已建立。一种方式是行政上。在上面的例子中，
鲍勃众所周知是工程部门的成员，通过
对企业数据库的访问。然而，SIP提供了一种机制
为了一个UA显式地创建一个绑定。此机制被称为
注册。

注册意味着向一种特殊类型发送一个REGISTER请求。
UAS被称为注册机构。注册机构充当前端，用于
域名位置服务，基于{v*}的读取和写入映射
注册请求的内容。此位置服务随后
通常由负责路由的代理服务器咨询
请求该域。

一个整体注册过程的示意图在{v*}中给出。
图2。注意注册员和代理服务器是逻辑角色
该文本可以通过网络中的单个设备播放；为了

清晰度，这两个在此图中是分开的。另外请注意
UAs 可能通过代理服务器发送请求以到达
注册员如果这两个是独立的元素。

SIP不强制规定实现的具体机制
位置服务。唯一的要求是某个注册机构
域必须能够读取和写入数据到位置服务，
并且为该域名提供的代理服务器或重定向服务器必须能够
读取相同的数据。一个注册机构可以与一个
特定域的SIP代理服务器。

10.2 构建REGISTER请求

注册请求包括添加、删除和查询绑定。一个注册
请求可以在一个记录地址和其中一个或多个之间添加一个新的绑定
更多联系方式。代表特定人士注册
记录地址可以由适当授权的第三方执行
党派。客户端还可以删除之前的绑定或查询
确定当前为地址-的绑定
记录

除注明外，REGISTER请求的构建和
客户端发送REGISTER请求的行为与以下相同
一般UAC行为在8.1节和17.1节中描述。

一个注册请求不会建立对话。一个UAC可能包括一个
注册请求中基于预存在的路由报头字段
路由设置为第8.1节所述。记录路由报头字段
在REGISTER请求或响应中无意义，必须忽略
如果存在。特别是，UAC不得创建一个新的路由集
基于是否存在 Record-Route 报头字段
任何对注册请求的响应。

以下头部字段，除Contact外，必须包含在
注册请求。一个联系头字段可以包含：

请求URI：请求URI命名了位置域名
服务注册的对象（例如，
"sip:chicago.com"。 "userinfo" 和 "@" 组件的
SIP URI 必须不存在。

收件人：The To header field contains the address of record whose
注册需要创建、查询或修改。
表头字段和Request-URI字段通常不同，因为
前一个包含用户名。此记录地址必须
是一个SIP URI或SIPS URI。

从：来自头字段包含记录的地址
负责人注册的人员。值是
与To头部字段相同，除非请求是第三方-
党派登记。

调用-ID：所有来自UAC的注册应使用相同的调用-ID
注册发送到特定目的地的报头字段值
注册商。

如果相同的客户端使用不同的 Call-ID 值，则
注册器无法检测是否延迟的REGISTER请求
可能已经顺序错误到达。

CSeq: The CSeq值保证REGISTER的正确排序
请求。一个 UA 必须为每个请求增加 CSeq 值一次。
注册具有相同 Call-ID 的请求。

联系：注册请求可能包含一个带有联系头字段的标题
零个或多个包含地址绑定的值。

UAs 必须不发送新的注册（即包含新的 Contact
表头字段值，与重传相反）直到它们已经
收到前一个的注册员最终回复或
之前的REGISTER请求已超时。

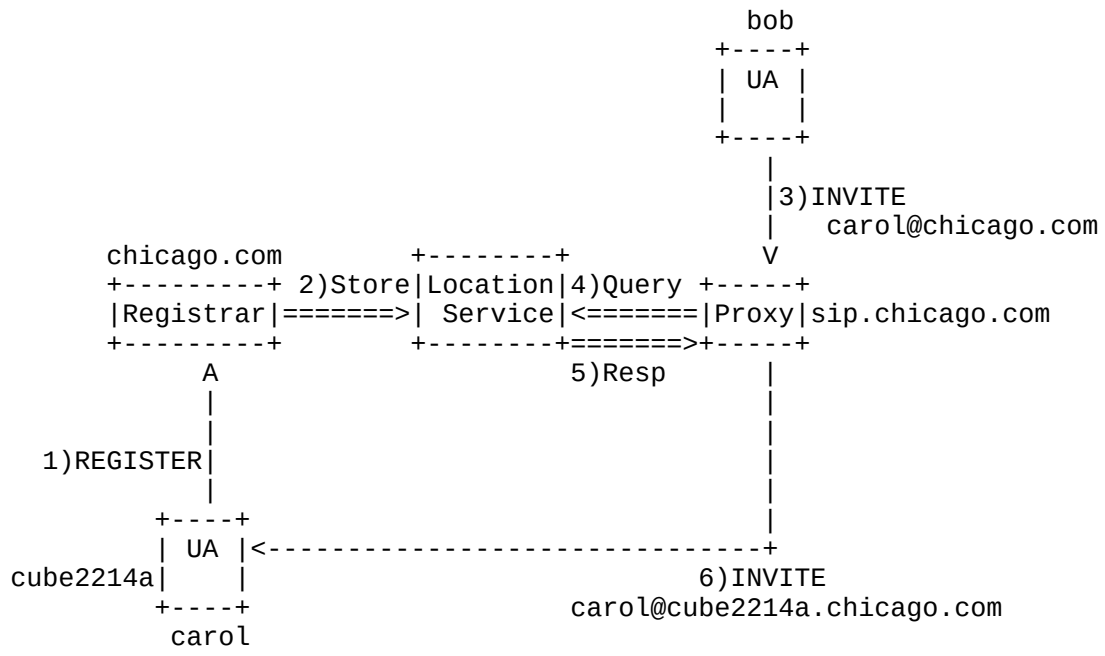

图 2：REGISTER 示例

以下 Contact 标头参数在 ... 中具有特殊含义
注册请求：

动作：RFC 2543 中的 "动作" 参数已被弃用。
UACs 应该不使用 "action" 参数。

过期: "过期"参数指示UA将保持多长时间有效
绑定需有效。值是一个数字
指示秒数。如果此参数未提供，则
值用于Expires头字段。
实现可以处理大于2**32-1的值
(4294967295秒或136年)相当于2**32-1。
无效值应视为等同于3600。

10.2.1 添加绑定

向注册机构发送的REGISTER请求包括联系信息
地址（复数）SIP请求应发送到这些地址记录
已转发。记录地址包含在“ 收件人 ”标题字段中
的注册请求。

请求的“ 联系 ”报头字段值通常包括
SIP 或 SIPS URI 识别特定的 SIP 终端（例如，
"sip:carol@cube2214a.chicago.com)，但他们**可能**使用任何URI方案。
SIP UA可以选择注册电话号码（使用tel URL，
RFC 2806 [9]) 或电子邮件地址（带有 mailto URL，RFC 2368 [32])
作为记录地址的联系人，例如。

例如，Carol，记录地址为 "sip:carol@chicago.com"，
将注册到chicago.com域名的SIP注册器。她
注册随后将由位于chicago.com的代理服务器使用
域用于路由Carol的记录地址的SIP请求
端点。

一旦客户端在注册机构建立了绑定，它可以发送
随后的注册包含新的绑定或对{v*}的修改
现有绑定，如需。对REGISTER的2xx响应
请求将包含在“ 联系 ”标题字段中，一个完整的列表，其中包含 {v*}
已注册为此记录地址的绑定
注册商

如果注册请求的“ 收件人 ”标题字段中的记录地址
是一个 SIPS URI，那么请求中的任何 Contact 头字段值
也应为SIPS URI。客户端应仅注册非SIPS URI
在SIPS地址记录下，当资源的安全性
由联系地址表示的由其他方式保证。
这可能适用于调用除SIP之外协议的URI。
或由除TLS以外的协议安全保护的SIP设备。

注册不需要更新所有绑定。通常，一个UA
仅更新其自身的联系地址。

10.2.1.1 设置联系地址的过期间隔

当客户端发送一个注册请求时，它可能会建议一个过期时间
间隔，表示客户端希望多长时间
注册需有效。（如第10.3节所述，）
注册商根据其本地选择实际时间间隔
策略。）

有两种方式可以让客户建议一个到期时间
绑定间隔：通过Expires头部字段或
"expires" 联系头参数。后者允许过期
每个绑定基础上的建议区间，当超过一个时
绑定是在单个REGISTER请求中给出的，而前者
建议为所有Contact头部字段值设置一个过期间隔
不包含“ expires ”参数的。

如果未提供表达建议到期时间的任一机制
存在于一个注册表中，客户端表明其希望
服务器选择。

10.2.1.2 联系地址偏好

如果在一个REGISTER请求中发送了多个联系人，则
注册UA打算将这些联系中的所有URI关联起来
表头字段值，其中包含在“ 收件人 ”中出现的记录地址
字段。此列表可以通过“ q ”参数进行优先级排序。
联系头部字段。"q"参数表示相对
偏好于特定的Contact头字段值，与
其他对此记录地址的绑定。第16.6节进行了描述
如何代理服务器使用此首选项指示。

10.2.2 移除绑定

注册是软状态，除非刷新，否则会过期，但可以
也必须明确删除。客户端可以尝试影响
注册机构根据第X节所述选择的过期间隔
10.2.1. 一个UA请求立即移除一个绑定
指定该联系地址的过期间隔为 "0"
一个注册请求。 用户代理（UAs）应支持此机制，以便
绑定可以在它们的过期间隔过去之前被移除。

The REGISTER-specific Contact header field value of "*" applies to
所有注册，但必须**不**使用除非过期头
字段存在，值为 "0"。

使用 "*" 联系头部字段值允许注册 UA
删除与记录地址关联的所有绑定
不知道它们的精确值。

10.2.3 获取绑定

任何注册请求的成功响应包含完整的列表
现有绑定，无论请求中是否包含
联系头字段。如果在一个中不存在联系头字段。
注册请求，绑定列表保持不变。

10.2.4 刷新绑定

每个UA负责刷新其拥有的绑定
之前建立的。 一个 UA 不应刷新由其设置的绑定
其他 UAs。

200（OK）响应来自注册商，包含一个联系人列表
字段枚举所有当前绑定。UA比较每个
联系地址以查看是否创建了联系地址，使用
比较规则在第19.1.4节中。如果是这样，它将更新到期
时间间隔根据expires参数或，如果不存在，则为
过期字段值。然后UA为每个{v*}发出一个注册请求。
在其绑定在到期间隔之前过期之前。它可能
将多个更新合并为一个注册请求。

A UA 应该在所有注册期间使用相同的 Call-ID
单次引导周期。注册刷新应发送到相同的
网络地址作为原始注册，除非重定向。

10.2.5 设置内部时钟

如果对REGISTER请求的响应包含日期头字段，
客户端可以使用此报头字段来了解当前时间
为了设置任何内部时钟。

10.2.6 发现注册机构

UAs可以使用三种方式来确定发送地址
注册：通过配置，使用记录地址，和
多播。一个UA可以被配置，其方式超出了本文档的范围。
规范，带有登记处地址。如果没有配置
注册地址，UA 应该使用地址的主机部分-
记录为请求URI，并在那里处理请求，使用
正常SIP服务器位置机制[4]。例如，UA为
用户 "sip:carol@chicago.com" 将注册请求发送至
sip:chicago.com

最后，一个UA可以被配置为使用多播。多播
注册地址为知名的 "所有SIP服务器"
多播地址 "sip.mcast.net"（IPv4：224.0.1.75）。
已分配的已知IPv6组播地址；此类分配
当需要时将单独进行文档说明。SIP UAs 可以为 {v*} 监听
该地址并使用它来了解其他位置
本地用户（见[33]）；然而，他们没有响应请求。

多播注册在某些环境中可能不合适，
例如，如果多个企业共享同一区域
网络。

10.2.7 传输请求

一旦构建了REGISTER方法，并且确定了目标位置，{v*}
消息已识别，UACs遵循以下描述的程序
第8.1.2节将REGISTER传递给事务层。

如果事务层因为注册而返回超时错误
未产生响应，UAC 不应立即重新尝试
注册到同一注册机构。

> 立即重试很可能也会超时。等待一些
> 合理的超时条件时间间隔
> 纠正可减少网络不必要的负载。 无特定
> 区间是强制规定的。

10.2.8 错误响应

如果UA收到423（间隔太短）响应，它可以重试
在所有联系过期间隔后进行的注册
地址在REGISTER请求中等于或大于的
423响应头字段Min-Expires中的过期间隔
(间隔过短) 响应。

10.3 处理注册请求

注册器是一种响应注册请求并维护{v*}的UAS。
一个可由代理服务器访问和重定向的绑定列表
服务器在其管理域内。一个注册机构处理
根据第8.2节和第17.2节的要求，但它接受
仅注册请求。注册机构不得生成6xx响应。

注册员可以根据需要将注册请求进行重定向。
通用用法是针对一个监听多播的注册机构
接口将多播注册请求重定向到其自身的单播
与302（临时移动）响应的接口。

注册者必须忽略Record-Route报头字段，如果它是
包含在注册请求中。注册机构不得包含一个
响应注册请求的任何响应中的Record-Route报头字段。

> 一个注册机构可能会收到一个经过代理的请求，该请求
> 将REGISTER视为未知请求，并添加了Record-
> 路由报头字段值。

注册员必须知道（例如，通过配置）该集合
域（s）的集合，它维护绑定。注册请求必须
由注册机构按接收顺序进行处理。
注册请求也必须原子性地处理，这意味着一个
特定注册请求要么完全处理，要么不处理
所有。每个REGISTER消息必须独立于任何
其他注册或绑定更改。

当收到一个REGISTER请求时，注册员遵循以下步骤：

1. 注册员检查Request-URI以确定其
具有对在域中标识的绑定的访问权限
请求URI。如果不，并且如果服务器也充当代理
服务器，服务器应将请求转发到指定的地址
域，遵循代理消息的一般行为
在第16节中描述。

2. 为确保注册机构支持任何必要的
扩展，注册表必须处理 Require 报头字段
值如第8.2.2节所述。

3. 注册员应验证UAC。验证机制为
SIP用户代理的认证在第22节中描述。
注册行为绝对不会覆盖通用
认证框架用于SIP。如果没有认证
机制可用时，注册员可以采取发件人地址
作为请求发起者的断言身份。

4. 注册员应确定认证用户是否
授权修改此记录地址的登记。
例如，注册机构可能会咨询一个授权
数据库将用户名映射到记录地址列表
对于该用户有修改绑定的授权。如果
认证用户无权修改绑定，
注册员必须返回403（禁止）并跳过
剩余步骤。

在支持第三方注册的架构中，一个
实体可能负责更新注册
与多个记录地址相关联。

5. 注册员从“ 收件人 ”标题中提取记录地址
请求的字段。如果记录地址无效
对于Request-URI中的域，注册机构必须发送
404（未找到）响应并跳过剩余步骤。The URI
必须转换为规范形式。为此，所有
URI 参数必须删除（包括用户参数），并且
任何转义字符都必须转换为它们的非转义形式
表单。结果用作绑定列表的索引。

6. 注册员检查请求是否包含联系人
表头字段。如果不，则跳到最后一步。如果
联系头部字段存在，注册商检查是否有
是一个包含特殊值 "*" 的联系人字段值
并且一个过期字段。如果请求有额外的联系
字段或非零的过期时间，则请求是
无效，并且服务器必须返回400（无效请求）和
跳过剩余步骤。如果不，登记员检查是否
呼叫ID与每个绑定存储的值一致。如果
不是，它**必须**解除绑定。如果它同意，它**必须**
仅当请求中的CSeq更高时移除绑定
比该绑定存储的值大。否则，更新
必须中止，并且请求失败。

7. 注册员现在处理每个联系地址中的联系
表头字段依次。对于每个地址，它确定
到期间隔如下：

- 如果字段值有一个“ expires ”参数，则该值
必须视为请求的到期时间。

- 如果没有这样的参数，但请求有一个
过期头字段，该值必须被视为
请求过期时间。

- 如果两者都不存在，必须使用本地配置的默认值
作为所需到期日处理。

注册员可以选择一个小于请求的到期时间
过期间隔。当且仅当请求的过期
时间间隔大于零 且 小于一小时 且
小于注册机构配置的最小值，注册机构可以
拒绝带有响应码423（间隔太长）的注册
简短）。此响应必须包含一个Min-Expires头字段
该声明了注册机构的最小到期间隔
愿意遵守。然后跳过剩余步骤。

允许注册员设置注册间隔
保护它免受过度频繁的注册刷新
当限制它需要维护的状态时和
降低注册信息过时的可能性。
注册的过期间隔常用于
服务创建。例如，是一个跟随服务，其中
用户可能只能在终端短暂可用
周期。因此，注册商应接受简短
注册；只有当请求满足以下条件时才应拒绝：
间隔如此之短，以至于刷新会降低注册商
性能。

对于每个地址，注册员随后搜索列表中的
当前绑定使用URI比较规则。如果
绑定不存在，暂且添加。
绑定确实存在，注册员检查 Call-ID 值。如果
现有绑定中的Call-ID值与
请求中的Call-ID值，绑定必须被移除，如果
到期时间为零，否则更新。
相同的，注册表比较CSeq值。如果该值
比现有结合的更高，它必须更新或
移除如上所述的绑定。如果不，更新必须
已终止且请求失败。

此算法确保来自同一的乱序请求
UA 被忽略。

每个绑定记录记录了 Call-ID 和 CSeq 值，来自
请求。

绑定更新必须提交（即，使其可见于
代理服务器或重定向服务器）当且仅当所有绑定
更新和添加成功。如果其中任何一个失败（对于
示例，因为后端数据库提交失败），所以
请求必须失败，并返回500（服务器错误）响应以及所有
暂定绑定更新必须删除。

8. 注册员返回一个200（OK）响应。响应必须
包含 Contact 标头字段值，列举所有当前
绑定。每个 Contact 值必须具有一个 "expires"
参数表示其过期间隔的选择
注册员。响应应包含一个日期头字段。

11 查询功能

SIP方法OPTIONS允许UA查询另一个UA或代理
服务器的能力。这允许客户端发现
关于支持的方法、内容类型、扩展的信息，
编解码器等，不“ 响铃 ”对方。例如，在
一个客户端在INVITE列表中插入一个Require头部字段，列出
选项，不确定目标UAS是否支持
客户端可以使用OPTIONS查询目标UAS，以查看是否
选项在支持的报头字段中返回。所有用户代理（UAs）必须支持
选项方法。

目标请求的OPTIONS请求通过Request-URI来识别，
可以识别另一个UA或SIP服务器。如果OPTIONS是
面向代理服务器，请求URI设置时没有用户
部分，类似于为REGISTER请求设置Request-URI的方式。

另外，一个接收带有 Max- 的 OPTIONS 请求的服务器
对于0的前向头字段值，可以响应请求
无论请求URI如何。

这种行为在HTTP/1.1中很常见。这种行为可以被使用
作为一个“ traceroute ”功能来检查{v*}的能力
通过发送一系列OPTIONS请求来发送单个跳转服务器
使用递增的Max-Forwards值。

与通用UA行为一样，事务层可以
返回超时错误，如果OPTIONS没有响应。这可能会
指示目标不可达，因此不可用。

一个OPTIONS请求可以作为已建立的对话的一部分发送。
查询可能稍后使用的对等方功能
对話。

11.1 OPTIONS 请求的构建

一个OPTIONS请求是使用SIP的标准规则构建的
请求如第8.1.1节所述。

一个 Contact 标头字段可以在 OPTIONS 中存在。

An Accept header field SHOULD be included to indicate the type of 接受头字段应当包含以指示类型
消息体，UAC希望接收的响应内容。通常，
这是设置为用于描述媒体的格式
UA的功能，例如SDP（application/sdp）。

响应OPTIONS请求的结果假定是作用域到
原始请求中的请求URI。然而，只有当是OPTIONS
作为部分已建立的对话发送的是否保证未来
请求将由生成 OPTIONS 的服务器接收
响应。

示例 OPTIONS 请求：

选项 sip:carol@chicago.com SIP/2.0
通过：SIP/2.0/UDP pc33.atlanta.com;分支=z9hG4bKhjhs8ass877
最大转发数：70
收件人：<sip:carol@chicago.com>
从：Alice <sip:alice@atlanta.com>;tag=1928301774
呼叫标识符：a84b4c76e66710
CSeq: 63104 OPTIONS
联系：<sip:alice@pc33.atlanta.com>
接受: application/sdp
内容长度：0

11.2 处理 OPTIONS 请求

响应OPTIONS请求使用标准规则构建
对于第8.2.6节中讨论的SIP响应。响应代码
选定的MUST与在请求中本应选择的相同
已是一个INVITE。也就是说，如果UAS是，则会返回200（OK）。
准备接受呼叫，如果这里忙碌，则会返回一个486（忙碌）
UAS正忙，等等。这允许使用OPTIONS请求
确定UAS的基本状态，这可以是一个指示
是否UAS会接受一个INVITE请求。

一个在对话中接收到的 OPTIONS 请求生成一个 200（OK）
响应与在对话外构建的响应相同
没有对对话产生任何影响。

此使用OPTIONS因代理差异而存在限制
处理 OPTIONS 和 INVITE 请求。当一个分叉的 INVITE 可以
导致返回多个200（OK）响应，一个分叉
OPTIONS 将只会产生单个 200 (OK) 响应，因为它
由代理使用非-INVITE处理。参见第16.7节
对于规范性细节。

如果对 OPTIONS 的响应由代理服务器生成，则
代理返回200（OK），列出服务器的功能。
响应不包含消息体。

允许、接受、Accept-Encoding、Accept-Language和受支持头
字段应在针对OPTIONS的200（OK）响应中存在
请求。如果响应由代理生成，则允许头
字段应当省略，因为它由于代理是方法而具有歧义
无神论者。 联系头部字段可能在200（OK）中存在
响应并具有与3xx响应相同的语义。即，
他们可能列出一组替代名称和达到的方式
用户。一个警告头字段可能存在。

消息体可以发送，其类型由以下确定
接受 OPTIONS 请求中的 Accept 头字段（application/sdp 是
默认情况下，如果不存在Accept头字段）。如果类型
包含一个可以描述媒体功能的，UAS 应该
在响应中包含一个主体以实现该目的。有关详情
此类实体的构建在应用/sdp的情况下是
在[13]中描述。

示例由UAS（对应于）生成的OPTIONS响应
请求在第11.1节中）：

SIP/2.0 200 正确
通过：SIP/2.0/UDP pc33.atlanta.com;分支=z9hG4bKhjhs8ass877
;接收=192.0.2.4
收件人：<sip:carol@chicago.com>;tag=93810874
从：Alice <sip:alice@atlanta.com>;tag=1928301774
呼叫标识符：a84b4c76e66710
CSeq: 63104 OPTIONS
联系：<sip:carol@chicago.com>
联系：<邮箱：carol@chicago.com>
允许：INVITE, ACK, CANCEL, OPTIONS, BYE
接受: application/sdp
Accept-Encoding: gzip 接受编码: gzip
接受语言: en
支持：foo
内容类型: application/sdp
内容长度：274

(SDP未显示)

12 对话

用户代理的一个关键概念是对话框。对话框
表示两个用户代理之间的点对点SIP关系
该持续一段时间。 对话促进了{v*}的排序。
用户代理之间的消息和请求的正确路由
它们之间。 对话表示一个上下文，其中包含{v*}。
解析SIP消息。第8节讨论了与方法无关的UA
处理对话之外的请求和响应。
章节讨论了如何使用这些请求和响应
构建一个对话框，然后是随后的请求和响应
在对话框中发送。

在每个UA上，通过一个对话ID识别一个对话框，该ID由以下组成
一个Call-ID值，一个本地标签和一个远程标签。每个对话ID
UA参与对话的不是同一个。具体来说，当地
标签在一个UA上与对等UA上的远程标签相同。
标签是不透明的令牌，有助于生成唯一的
对话ID。

一个对话框ID也与所有响应以及任何相关联
请求包含在“ 收件人 ”字段中的标签。计算规则为
消息的对话ID取决于SIP元素是否为UAC
或UAS。对于UAC，对话ID的Call-ID值被设置为
消息的Call-ID，远程标签设置为To中的标签
消息字段，本地标签设置为From中的标签

消息字段（这些规则适用于请求和
响应）。正如人们所期望的UAS，呼叫ID值
对話ID設置為消息的Call-ID，遠端標籤設置
到消息的“ From ”字段中的标签，并且本地标签被设置
到消息的“ 收件人 ”字段中的标签。

一个对话框包含某些用于进一步消息的状态片段
对话内的传输。此状态由对话组成
ID，一个本地序列号（用于对UA发出的请求进行排序）
其同等级的），一个远程序列号（用于对其请求进行排序的）
对UA进行对等，一个本地URI，一个远程URI，远程目标，一个布尔值
标志称为 "secure"，以及一个路由集，它是一个有序列表
统一资源标识符。路由集是需要遍历的服务器列表
向对等方发送请求。一个对话框也可以处于“ 早期 ”状态。
状态，当它以临时响应创建时发生，
然后过渡到 "已确认" 状态，当出现 2xx 最终
响应到达。对于其他响应，或者如果没有响应到达
所有都在那个对话框中，早期对话框终止。

12.1 对话的创建

对話通過生成非失敗性回應來創建
对具有特定方法的请求。在此规范中，仅
2xx 和 101-199 响应带有 To 标签，其中请求是
邀请，将建立对话。一个由非最终状态建立的对话。
响应请求处于“ 早期 ”状态，这被称为
早期对话。扩展可以定义其他创建{v*}的方法
对话框。第13节提供了更多特定于的详细信息。
邀请方法。 这里，我们描述了创建对话的过程
状态不依赖于方法。

UAs 必须按照描述将值分配给对话ID组件
以下。

12.1.1 UAS行为

当UAS响应请求并返回一个建立{v*}的响应时
对话（例如2xx到INVITE），UAS必须复制所有Record-Route
请求头字段值放入响应中（包括{v*}）
统一资源标识符（URIs）、统一资源标识符参数以及任何Record-Route报头字段参数，
无论它们是否为UAS所知)并且必须保持
值的顺序。UAS 必须在报头中添加一个 Contact 字段。
响应。 Contact头字段包含一个地址，其中包含{v*}。
UAS希望被联系以处理对话中的后续请求
（包括在INVITE情况下对2xx响应的ACK）。
通常，此URI的主机部分是IP地址或FQDN。
主机。 Contact头部字段中提供的URI必须是SIP
或SIPS URI。如果启动对话的请求包含了一个

SIPS URI 在 Request-URI 或顶级 Record-Route 头字段中
值，如果有的话，或者如果没有则联系头部字段
记录-路由报头字段，响应中的联系报头字段
MUST 是一个 SIPS URI。该 URI 应具有全局范围（即，它是全球性的）。
相同的URI可以在此对话框之外的消息中使用）。同样地，
URI在INVITE消息的Contact头部字段中的范围不
仅限于这个对话框。因此，它可以用在消息中
到UAC，即使在这个对话框之外。

UAS 然后构建对话的状态。此状态必须
维护整个对话期间。

如果请求通过TLS到达，并且请求URI包含SIPS
URI，"安全"标志设置为TRUE。

路由集必须设置为Record-Route中的URI列表
请求的标题字段，按顺序取用并保留所有URI
参数。如果不存在Record-Route头字段，
请求，路由集必须设置为空集。此路由集，
即使为空，也覆盖未来任何现有的路由设置
请求在此对话框中。远程目标必须设置为URI
从请求的“ 联系 ”标题字段中。

远程序列号必须设置为序列的值
请求CSeq报头字段中的数字。本地序列
数字必须为空。对话ID的调用标识符组件
必须设置为请求中Call-ID的值。
标签组件的对话ID必须设置为“ 收件人 ”字段中的标签
在响应请求（其中始终包含一个标签）时，并且
远程标签组件的对话框ID必须设置为来自的标签
从请求的字段中。一个UAS必须准备接收一个
请求中From字段没有标签，在这种情况下，标签是
被认为具有空值。

> 这是为了与RFC 2543保持向后兼容，其中
> 未强制要求从标签。

远程URI必须设置为From字段中的URI，并且
本地URI必须设置为“ 收件人 ”字段中的URI。

12.1.2 UAC 行为

当UAC发送一个可以建立对话的请求（例如
INVITE) 它必须提供一个具有全局范围（即，{v*}）的SIP或SIPS URI。
同SIP URI可用于此对话之外的消息中）在
请求的联系人头字段。如果请求有 Request-
URI 或最顶层的 Route 报头字段值，带有 SIPS URI，的
联系头部字段必须包含一个SIPS URI。

当UAC收到建立对话的响应时，它
构建对话的状态。此状态必须保持
对于对话的持续时间。

如果请求是通过TLS发送的，并且请求URI包含了一个
SIPS URI，"安全"标志设置为TRUE。

路由集必须设置为Record-Route中的URI列表
响应头字段，以反向顺序取用并保留
所有URI参数。如果不存在Record-Route头字段，
响应，路由集必须设置为空集。此路由
集合，即使为空，也覆盖了为未来设置的任何现有路由集合
请求在此对话框中。远程目标必须设置为URI
从响应的“ 联系 ”标题字段中。

本地序列号必须设置为序列的值
请求CSeq头部字段中的数字。远程序列
数字必须为空（它在远程UA发送时建立）
请求在对话框内）。调用标识符组件的
会话ID必须设置为请求中的Call-ID值。
本地标签组件的对话框ID必须设置为标签{v*}。
请求中的“ From ”字段，以及远程标签组件的
会话ID必须设置为响应中“ 收件人 ”字段的标签。
UAC 必须准备接收在 To 中不带标签的响应
字段，在这种情况下，标签被视为具有空值。

这是为了与RFC 2543保持向后兼容，其中
未强制要求To标签。

远程URI必须设置为“ 收件人 ”字段中的URI，并且本地
URI 须设置为“ From ”字段中的 URI。

12.2 对话内的请求

一旦两个UA之间建立了对话，任一UA都可以
根据需要，在对话中可以启动新交易。UA
发送请求将承担事务的UAC角色。
UA接收请求后将扮演UAS角色。请注意，这些可能
与交易期间UAs所持的不同角色
建立了对话。

请求在对话中可能包含Record-Route和Contact头
字段。然而，这些请求不会导致对话框的路由集
需要修改，尽管它们可能修改远程目标URI。
具体来说，不是目标刷新请求的请求不
修改对话框的远程目标URI，并请求目标
刷新请求。对于已建立的对话框。

邀请，唯一定义的目标刷新请求是重新邀请（见
第14节）。其他扩展可能定义不同的目标刷新
请求以其他方式建立的对话框。

请注意，ACK 不是一个目标刷新请求。

目标刷新请求仅更新对话框的远程目标URI，
并且不是由Record-Route记录形成的路由集。 更新
后者将引入严重的向后兼容性问题
符合RFC 2543的系统。

12.2.1 UAC 行为

12.2.1.1 生成请求

一个对话框内的请求是通过使用许多来构建的
状态作为对话部分存储的组件。

请求中的“ 收件人 ”字段中的URI必须设置为远程URI
从对话状态。请求的To头字段中的标签
必须设置为对话框ID的远程标签。 的From URI为
请求必须设置为从对话状态中的本地URI。 标签
在请求的“ From ”报头字段中必须设置为本地标签
的对话框ID。如果远程或本地标签的值为空，
标签参数必须从“ 到 ”或“ 从 ”报头字段中省略，
分别地。

使用原始文档中“ 从 ”和“ 到 ”字段中的URI
请求在后续请求中是向后进行的
与RFC 2543的兼容性，该规范使用了URI进行对话
识别。 在本规范中，仅使用标签。
对话识别。 预期必须进行反射
原始的“ 到 ”和“ 从 ”URI在中对话请求中将保持不变
已弃用在本规范的后续修订中。

请求的Call-ID必须设置为对话的Call-ID。
请求必须在对话中严格单调
递增且连续的CSeq序列号（每次递增一）
在每个方向上（当然不包括ACK和CANCEL，它们的编号
等于请求被确认或取消。因此，如果
本地序列号不为空，本地值的
序列号必须加一，并且此值必须
放入CSeq报头字段中。如果本地序列号是
空，必须根据指南选择一个初始值
第8.1.1.5节。CSeq报头字段值中的方法字段
必须匹配请求的方法。

32位长度，客户端可以在单个
呼叫，每秒一个请求，大约136年后需要
环绕。序列号的初始值被选择
因此，在同一个调用中后续请求不会绕回
周围。一个非零初始值允许客户端使用时间-
基于初始序列号。例如，一个客户端可以，
选择32位秒钟中的31个最重要的比特
初始序列号。

UAC使用远程目标和路由集来构建请求URI
请求的路由头字段。

如果路由集为空，UAC必须放置远程目标URI
进入Request-URI。UAC不得添加路由头字段到
请求。

如果路由集不为空，并且路由集中的第一个URI
包含lr参数（见第19.1.1节），UAC必须放置
远程目标URI到请求URI中，并且必须包含一个路由
标题字段包含按顺序排列的路由集值，包括所有
参数。

如果路由集不为空，并且其第一个URI不包含
lr参数，UAC必须放置路由集的第一个URI
进入Request-URI，去除任何不允许的参数
在请求URI中。UAC必须添加一个包含路由头字段的{v*}。
路线集值余数按顺序排列，包括所有
参数。然后UAC必须将远程目标URI放入
路由报头字段作为最后一个值。

例如，如果远程目标是 sip:user@remoteua 并且路由
集合包含：

<sip:proxy1{v*}，{v*}sip:proxy2{v*}，{v*}sip:proxy3；lr{v*}，{v*}sip:proxy4{v*}

请求将以以下 Request-URI 和路由形成
表头字段：

方法 sip:proxy1
路由：<sip:proxy2>，<sip:proxy3；lr>，<sip:proxy4>，<sip:user@remoteua>

如果路由集的第一个URI不包含lr
参数，代理指示不理解路由
本文件中描述的机制并将按指定方式行动
RFC 2543，将Request-URI替换为第一个Route报头
字段值它在转发消息时接收。放置
请求URI保留在路由头字段末尾

信息在该Request-URI中穿过严格路由器（它将
当请求达到宽松时，将返回到 Request-URI。
路由器）。

A UAC 应该在任何目标刷新中包含一个 Contact 报头字段
在对话中的请求，除非需要更改， {v*}
the URI 应该与之前请求中使用的相同
对话。如果“安全”标志为真，则该URI必须是SIPS URI。
如第12.2.2节所述，目标中的联系头部字段
刷新请求更新远程目标URI。这允许UA
提供一个新的联系地址，如果其地址在以下期间发生变化，应提供
对話的持续时间。

然而，非目标刷新请求的请求不会影响
远程目标URI对话框。

请求的其余部分按照第8.1.1节所述形成。

一旦请求被构建，服务器的地址是
计算并发送请求，使用相同的程序
请求对话之外（第8.1.2节）。

第8.1.2节中的程序通常会导致
请求发送到顶部路由指示的地址
表头字段值或如果没有路由表头字段则为请求URI
当前。在一定的限制条件下，它们允许请求
待发送至备用地址（例如默认出站
代理未在路由集中表示）。

12.2.1.2 处理响应

UAC将接收来自事务请求的响应
层。如果客户端事务返回超时，则处理为
作为一个408（请求超时）响应。

UAC接收请求并返回3xx响应时的行为
在对话框内与请求已发送到对话框外相同 {v*}
对话。此行为在8.1.3.4节中描述。

注意，然而，当UAC尝试替代位置时，它
仍然使用用于对话框的路由集来构建路由头
的请求。

当UAC收到针对目标刷新请求的2xx响应时，它
必须将对话框的远程目标URI替换为从的URI
响应中的联系头部字段，如果存在的话。

如果对对话框内请求的响应是481
(调用/交易不存在) 或 408 (请求超时)，用户账户控制（UAC）
应终止对话。如果{v*}，UAC也应终止对话。
没有任何响应收到对于请求（客户端
交易将通知TU关于超时。))

对于由INVITE发起的对话，终止对话包括
发送一个 BYE。

12.2.2 UAS行为

请求在对话框中发送，与其他请求一样，是原子的。如果
特定请求被UAS接受，所有状态变化
与之相关的是执行的。如果请求被拒绝，则没有任何
状态变化是执行的。

请注意，一些请求，例如INVITEs，会影响多个部分
状态。

UAS将从交易层接收请求。如果
请求在“ 收件人 ”标题字段中有一个标签，UAS核心计算{v*}。
请求对应的对话标识符并将其与比较
现有对话框。如果存在匹配，则这是一个中对话框请求。
在这种情况下，UAS首先应用相同的处理规则，对于{v*}
请求在对话之外，在第8.2节中讨论。

如果请求在“ 收件人 ”头字段中有一个标签，但对话
标识符与任何现有对话框不匹配，UAS可能存在
崩溃并重新启动，或者它可能收到了一个请求
不同（可能失败的）UAS（UASs可以构建To标签）
因此，一个UAS可以识别该标签是为它自己的UAS准备的
提供恢复）。另一种可能性是传入
请求已被简单误转。根据“ 收件人 ”标签，UAS可能
或者接受或拒绝请求。接受对 {v*} 的请求。
可接受标签提供鲁棒性，因此对话框可以持续
即使通过崩溃。 想要支持此功能的用户代理（UAs）必须
考虑一些问题，例如选择单调
增加CSeq序列号，即使在重启后也能重建
路由集，并接受超出范围的RTP时间戳和序列号
数字。

如果UAS希望拒绝请求，因为它不想
重新创建对话框，它必须以481响应请求
(调用/事务不存在) 状态码并将其传递到
服务器事务。

请求在任何方面都不改变对话状态的可能为
在对话框中接收（例如，一个OPTIONS请求）。它们是
作为如果它们是在对话外部接收到的那样进行处理。

如果远程序列号为空，它**必须**设置为值
请求中CSeq头部字段值中的序列号。
如果远程序列号不为空，但序列号
请求的序列号低于远程序列号时，请求
该设备出现故障，必须以500（服务器内部错误）状态拒绝。
错误) 响应。如果远程序列号不为空，则
请求的序列号大于远程
序列号，请求顺序。它是可能的，对于{v*}。
CSeq序列号应高于远程序列号
多于一个。这不是错误条件，UAS 应该
准备接收和处理CSeq值大于{v*}的请求
比上一个接收到的请求高一个。UAS 必须然后设置
远程序列号到序列号值的映射
请求中的CSeq报头字段值。

如果代理挑战由UAC生成的请求，则UAC有
重新提交带有凭证的请求。重新提交的请求
将有一个新的CSeq编号。UAS将永远看不到第一个
请求，因此，它将注意到CSeq号码空间中存在一个差距。
这样的差距不代表任何错误条件。

当UAS收到目标刷新请求时，它必须替换
对话的远程目标URI与来自“ Contact ”报头字段的URI
在该请求中，如果存在的话。

12.3 对话终止

与方法无关，如果在一个对话框之外生成请求
非2xx的最终响应，通过任何早期对话创建的
临时对该请求的响应已终止。该机制
对于终止已确认的对话框，方法是特定的。在此
规范，BYE方法终止一个会话和对话
与之相关。详见第15节。

13 开始会话

13.1 概述

当用户代理客户端希望启动一个会话（例如，
音频、视频或游戏），它制定一个INVITE请求。
邀请请求要求服务器建立会话。此请求
可能由代理转发，最终到达一个或多个UAS
这些UASs将可能接受邀请。
经常需要查询用户是否接受

邀请。一段时间后，这些UAS可以接受邀请
通过发送2xx响应来（意味着要建立会话）。
如果邀请未被接受，则响应为3xx、4xx、5xx或6xx
发送，具体取决于拒绝的原因。在发送之前
最终响应，UAS还可以发送临时响应（1xx）到
建议UAC关于联系被叫用户的进展。

在可能收到一个或多个临时响应之后，UAC
将获得一个或多个2xx响应或一个非2xx最终响应。
因为接收最终结果可能需要花费很长时间
对INVITE的响应，INVITE的可靠性机制
交易与其他请求（如OPTIONS）的交易不同。
一旦收到最终响应，UAC需要发送一个ACK。
每个最终收到的响应。发送此ACK的流程
取决于响应类型。对于最终响应在300和
699，ACK处理在事务层完成，并遵循
一组规则（见第17节）。对于2xx响应，ACK是
由UAC核心生成。

一个2xx响应的INVITE请求建立了一个会话，并且它还
创建了一个在发出INVITE的UA和UA之间的对话框
生成2xx响应的。因此，当多个2xx
响应来自不同的远程UA（因为INVITE
分支），每个2xx建立一个不同的对话。所有这些对话
是同一调用的一部分。

本节提供了使用 {v*} 建立会话的详细信息。
邀请。支持邀请的UA必须也支持ACK、取消和
再见。

13.2 UAC 处理

13.2.1 创建初始 INVITE

由于初始的INVITE代表了一个对话之外的请求，
其构建遵循第8.1.1节的程序。附加
处理需要针对特定情况下的INVITE进行。

邀请中应包含一个允许头字段（第20.5节）。
它表示在UA中对话框内可以调用的方法
发送INVITE，持续整个对话期间。例如，一个
UA 具备在对话中接收 INFO 请求的能力 [34] 应该
包含一个列出INFO方法的Allow头部字段。

支持的头字段（第20.37节）应当存在于
邀请。它列举了UAC理解的所有扩展。

一个接受（第20.1节）报头字段可以在INVITE中存在。
它表示对UA可接受的Content-Types，包括
它收到的响应，以及随后发送的任何请求
在由 INVITE 建立的对话中 it。Accept 头字段
特别适用于指示支持各种会话
描述格式。

UAC 可能会添加一个过期头字段（第20.19节）以限制
邀请的有效性。如果指明的过期时间
标题字段已到达，且对于INVITE没有最终答案
已收到，UAC核心应生成一个{v*}请求。
邀请，如第9节所述。

A UAC 也可能认为添加以下内容很有用，包括主题（章节
20.36)、组织（第20.25节）和用户代理（第20.41节）
表头字段。 它们都包含与 INVITE 相关的信息。

UAC 可能会选择向 INVITE 添加消息体。章节
8.1.1.10 处理如何构建头部字段 -- Content-
类型包括但不限于 -- 需要描述消息体。

存在包含会话的消息体的特殊规则
描述 - 它们对应的 Content-Disposition 是 "session"。
SIP使用一个提议/应答模型，其中一个UA发送一个会话
描述，称为报价，其中包含一个提议的描述
的会话。该报价表明所需的通信方式
(音频，视频，游戏)，这些均值（如编解码器）的参数
类型) 和接收回答者媒体信息的地址。
其他UA响应另一个会话描述，称为
答案，表示哪些通信方式被接受，的
参数适用于这些均值，以及接收地址
媒体来自提供者。一项出价/回答交换在{v*}内。
对话的上下文，因此如果SIP INVITE导致多个
对話，每個都是獨立的出價/回答交換。 出價/回答
模型定义了何时可以提出报价和答案的限制
（例如，当有新的报价正在进行时，你不能提出新的报价）。
这导致了对提供和答案可以存在的限制
出现在SIP消息中。在本规范中，提供和应答
只能出现在INVITE请求和响应以及ACK中。用法
提供和答案的范围进一步受限。对于初始的INVITE
交易，规则如下：

o 初始出价必须为INVITE或，如果没有，则为
在UAS返回的第一个可靠的非故障消息中
UAC。在此规范中，即最终的2xx
响应。

如果初始出价在INVITE中，则回答必须在其中
可靠的从UAS返回到UAC的非故障消息
与该INVITE相关。对于本规范，即
仅对该INVITE的最终2xx响应。那个完全相同的
答案也可以放在任何临时回复中
在答案之前。UAC 必须处理第一个会话
描述它接收到的答案，并且必须忽略任何
会话描述在随后的对初始响应的回答中
邀请。

如果初始出价在第一个可靠的非失败
消息从UAS返回到UAC，答案必须在
对那条消息的确认（在本规范中，ACK）
对于2xx响应）。

在发送或收到对第一个出价的答复后，
UAC 可能根据规则在请求中生成后续报价
指定给该方法，但仅当它已收到答案时
对于任何之前的出价，并且没有向其发送任何出价
尚未得到答案。

o 一旦UAS发送或接收了初始回答
提供，它不得在任何响应中生成后续的提供
到初始的INVITE。这意味着基于此的UAS
规范本身永远无法生成后续的报价，直到
初始交易的完成。

具体来说，上述规则指定了两个符合UAs的交易所
仅对此规范而言 - 报价包含在INVITE中，并且
回答在2xx（也可能在1xx中，具有相同的）
值)，或者报价在2xx范围内，且答案为ACK。
所有支持INVITE的用户代理必须支持这两个交换。

会话描述协议（SDP）（RFC 2327 [1]）必须
由所有用户代理支持，作为描述会话的手段，以及其
使用构建报价和答案的必须遵循程序
定义于[13]。

所描述的报价-回答模型的限制仅适用于
到内容处置头字段值为 "session" 的主体。
因此，可能同时包含INVITE和ACK。
正文消息（例如，INVITE 携带一张照片（内容-
处置：渲染）和ACK一个会话描述（内容-
处置: 会话))

如果Content-Disposition头字段缺失，则正文为
内容类型 application/sdp 表示处置为 "会话"，而
其他内容类型表示 “ 渲染 ” 。

一旦创建了INVITE，UAC遵循以下程序
定义用于在对话之外发送请求（第8节）。
导致构建一个客户端事务，该事务将
最终发送请求并将响应交付给UAC。

13.2.2 处理 INVITE 响应

一旦INVITE被传递给INVITE客户端事务，
UAC等待INVITE的响应。如果INVITE客户端
事务返回超时而不是响应，TU充当
如果已收到描述的408（请求超时）响应，则
在8.1.3节中。

13.2.2.1 1xx 响应

零、一或多个临时响应可能在之一或
更多最终回复已收到。 有关{v*}的临时回复。
邀请请求可以创建“ 早期对话 ”。如果有一个临时响应
具有在“ 收件人 ”字段中的标签，并且如果响应的对话框ID为
不匹配现有对话，使用程序构建一个
定义在第12.1.2节中。

早期对话仅在UAC需要发送时才会需要。
请求在初始INVITE之前的对话中的对等方
交易完成。临时存在的报头字段
响应在对话处于早期状态时适用
例如，在临时响应中的一个允许头字段
包含在对话框中使用的方法，当此处于
早期状态）。

13.2.2.2 3xx 响应

3xx响应可能包含一个或多个Contact头部字段值
提供新的地址，被叫方可能在这些地址上可联系。
根据3xx响应的状态码（见第21.3节），
UAC 可能会选择尝试那些新地址。

13.2.2.3 4xx、5xx 和 6xx 响应

一个非2xx的最终响应可能为INVITE接收。4xx，
5xx 和 6xx 响应可能包含一个 Contact 报头字段值
指示错误附加信息的存放位置
可以在{v*}中找到。随后的最终响应（这些只会到达
在错误条件下）必须忽略。

所有早期对话在接收到后均视为终止。
非2xx最终响应。

在收到非2xx最终响应后，UAC核心
考虑INVITE事务已完成。 INVITE客户端
事务处理响应的ACK生成（见
第17节）。

13.2.2.4 2xx 响应

多个2xx响应可能针对单个INVITE到达UAC
请求由于分叉代理。每个响应通过以下方式区分：
标签参数在To报头字段中，每个代表一个
不同的对话，具有不同的对话标识符。

如果2xx响应中的对话标识符与对话标识符匹配
现有对话的标识符，该对话必须过渡到
"已确认"状态，以及为对话设置的路线必须
基于2xx响应重新计算，使用第X节中的程序
12.2.1.2. 否则，必须创建一个新的处于“ 已确认 ”状态的对话框
使用第12.1.2节中的程序构建。

> 请注意，唯一需要重新计算的状态是路由
> 集合。其他状态信息，例如最高的序列号
> (远程和本地) 在对话框中发送的内容不会被重新计算。
> 仅路由集用于向后兼容性重新计算。RFC
> 2543 没有强制要求在 {v*} 中镜像 Record-Route 报头字段
> 一个1xx，只有2xx。然而，我们无法更新整个状态
> 对话，因为在中途对话请求可能已经被发送过
> 早期对话，修改序列号，例如。

UAC核心必须为从接收到的每个2xx生成一个ACK请求
交易层。ACK的报头字段构造
与任何在对话中发送的请求相同（见第节）
12) 除CSeq和相关的头部字段外
身份验证。CSeq报头字段的序列号必须
与被确认的INVITE相同，但CSeq方法必须
be ACK. The ACK MUST contain the same credentials as the INVITE. If
the 2xx 包含一个报价（基于上述规则），ACK 一定要
携带答案在其主体中。如果2xx响应中的出价不是
可接受，UAC核心必须在ACK中生成一个有效的答案。
立即发送一个BYE。

一旦构建了ACK，就使用[4]中的程序。
确定目标地址、端口和传输。然而，
请求直接传递到传输层进行传输，
而不是客户端事务。这是因为在UAC核心
处理ACK的重传，而不是传输层。
ACK必须每次在{v*}传递给客户端传输时传递。
重传触发了ACK的2xx最终响应
到达。

UAC核心认为INVITE事务在64*T1秒后完成
在接收到第一个2xx响应之后。在此点，所有
早期尚未过渡到正式对话的对话
已终止。一旦INVITE事务被认为已完成，
UAC核心，不再期望收到新的2xx响应。

如果在对一个INVITE请求的任何2xx响应进行确认后，UAC执行
不想继续那个对话框，那么UAC必须终止
通过发送如第15节所述的BYE请求来进行对话。

13.3 UAS 处理

13.3.1 INVITE 处理

UAS核心将接收来自事务层的INVITE请求。
它首先执行第8.2节的请求处理程序，
在对话内外都适用的请求。

假设这些处理状态完成而没有生成
响应，UAS核心执行以下附加处理步骤：

1. 如果请求是一个包含Expires头的INVITE
字段，UAS核心为秒数设置计时器
指示在标题字段值中。当计时器触发时，
邀请被视为已过期。如果邀请
在UAS生成最终响应之前过期，一个487
(请求终止)响应应当生成。

2. 如果请求是中间对话请求，则方法是无关的
处理在第12.2.2节中描述的内容首先应用。
也可能修改会话；第14节提供详细信息。

3. 如果请求在To头字段中有一个标签，但对话
标识符与任何现有对话框都不匹配，UAS
可能已崩溃并重新启动，或可能已收到请求
对于不同的（可能失败的）UAS。第12.2.2节提供了
指南以在如此情况下实现稳健的行为。

从现在开始处理假设INVITE位于...之外
对話，因此用于建立新会话的目的。

The INVITE may contain a session description, in which case the UAS
正在向该会话提供一份报价。 有可能
即使用户已经是该会话的参与者，即使
the INVITE is outside of a dialog. This can happen when a user is
被多个其他用户邀请到同一个多播会议
参与者。如果需要，UAS 可以使用其中的标识符。
会话描述以检测此重复。例如，SDP

包含会话ID和版本号在原始（o）字段中。如果
用户已经是会话的成员，并且会话
会话描述中包含的参数没有变化，
UAS 可能静默地接受 INVITE（即发送 2xx 响应）
不提示用户）。

如果INVITE不包含会话描述，UAS是
被邀请参加一个会议，并且UAC已要求
UAS提供会话的报价。它必须提供报价
在其第一次非故障可靠消息返回给UAC。在此
规范，即对INVITE的2xx响应。

UAS可以指示进度、接受、重定向或拒绝
邀请。在这些所有情况下，它使用 {v*} 公式来制定回应。
第8.2.6节中描述的程序。

13.3.1.1 进度 {v*}

如果UAS无法立即回答邀请，它可以
选择以指示对UAC（例如，一种 {v*}的某种进度
指示电话正在响铃）。这是通过以下方式实现的
临时响应介于101至199之间。这些临时
响应建立早期对话，因此遵循程序
除第8.2.6节之外，第12.1.1节的那些。一个UAS可能
发送尽可能多的临时响应。每个都必须
指示相同的对话ID。然而，这些将不会被发送
可靠地。

如果UAS希望有更长时间来回答INVITE，
需要请求一个 “ 扩展 ” 以防止代理
从取消交易。 代理人有权取消
一个交易，当响应之间存在3分钟的间隔时
交易。为防止取消，UAS 必须发送非100
每分钟提供临时响应，以处理可能的情况
丢失的临时响应。

> 一个INVITE事务可以在延长的时间内进行。
> 用户被挂起，或在与PSTN系统互操作时
> 允许通信发生而不需要回答
> 呼叫。后者在交互式语音应答（IVR）中很常见
> 系统。

13.3.1.2 INVITE 被重定向

如果UAS决定重定向呼叫，则发送3xx响应。
300（多选），301（永久移动）或302（移动
暂时)响应应包含一个Contact头字段

包含一个或多个要尝试的新地址的URI。
响应传递给INVITE服务器事务，该事务将处理
与它的重传一起。

13.3.1.3 邀请被拒绝

一个常见的场景发生在被调用者目前不愿意或
能够在此端系统接收额外呼叫。 A 486（此处忙碌）
应在此场景下返回。如果UAS知道没有
其他端系统将能够接受此呼叫，一个600（忙
任何地方) 应发送响应。然而，这不太可能。
该UAS将能够一般地知道这一点，因此这一点
响应通常不会被使用。响应会被传递到
邀请服务器事务，该事务将处理其重传。

一个拒绝包含在INVITE中的出价的无人机系统（UAS）应返回488
(不可接受此处) 响应。此类响应应包括一个
警告头字段值解释为何拒绝该报价。

13.3.1.4 邀请被接受

UAS核心生成一个2xx响应。此响应建立了一个
对話，因此遵循第12.1.1节的程序。
除了第8.2.6节中的那些之外。

A 2xx 响应到 INVITE 应该包含 Allow 头字段和
支持的报头字段，并且可能包含Accept报头字段。
包括这些标题字段允许UAC确定
特性与UAS在持续期间支持的扩展
调用，无需探测。

如果INVITE请求包含了一个提议，并且UAS尚未
发送了答案，2xx 必须包含答案。如果 INVITE 发送了
不包含报价，2xx必须包含报价，如果UAS有
尚未发送报价。

一旦构造了响应，它就会被传递给INVITE
服务器事务。注意，然而，邀请服务器
交易一旦收到这个最终
响应并将其传递给传输。因此，这是必要的
定期将响应直接传递到传输，直到
ACK到达。2xx响应被传递给传输。
时间间隔从T1秒开始，每次翻倍
重传直到达到 T2 秒（T1 和 T2 在文中定义）
第17节）。当收到一个ACK请求时，响应重传将停止。
响应已接收。这不受任何传输方式的影响。
协议用于发送响应。

由于2xx是端到端重传的，因此可能存在中间跳数
UAS和UAC是UDP。为确保可靠地跨
这些跳数，即使响应被定期重传
UAS上的运输是可靠的。

如果服务器在64*T1秒内重新传输2xx响应
接收到一个ACK，对话被确认，但会话应该
终止。这是通过BYE实现的，如第节所述。
15.

14 修改现有会话

一个成功的INVITE请求（见第13节）建立了两个
对两个用户代理和一个会话使用报价-回答的对话
模型。第12节解释了如何修改现有的对话　使用{v*}。
目标刷新请求（例如，更改远程目标URI）
关于对话框）。本节描述如何修改实际
会话。此修改可能涉及更改地址或端口，
添加媒体流、删除媒体流等。这是
通过在相同对话中发送新的INVITE请求来完成
该会话已建立。在会话内发送的 INVITE 请求。
现有对话称为重新邀请。

请注意，单个re-INVITE可以修改对话和{v*}。
会话同时的参数。

调用者或被调用者都可以修改现有的会话。

用户代理（UA）在检测到媒体故障时的行为是一个问题
本地策略。然而，自动生成re-INVITE或BYE是不允许的。
不建议使用，以避免在网络中产生过多的流量时发生拥塞
是拥塞。在任何情况下，如果这些消息被发送
自动地，它们应该在某个随机间隔后发送。

注意，上面的段落指的是自动生成的
BYEs和re-INVITEs。如果用户在媒体故障时挂断，
UA会像往常一样发送一个BYE请求。

14.1 UAC 行为

相同的提供-回答模型适用于会话描述
邀请（第13.2.1节）适用于重新邀请。因此，一个UAC
想要添加媒体流，例如，将创建一个新的
提供包含此媒体流的报价，并在INVITE中发送该报价
请求其对等节点。需要注意的是，完整
会话描述被发送，不仅仅是变化。
支持在各种元素中处理无状态会话，并且
支持故障转移和恢复功能。当然，UAC 可能

发送不带会话描述的重新邀请，在这种情况下，第一个
可靠的重新邀请（re-INVITE）非故障响应将包含提议
(在本规范中，即是一个2xx响应)。

如果会话描述格式具有版本功能
数字，提供者应表明会话的版本
描述已更改。

The To, From, Call-ID, CSeq, and Request-URI of a re-INVITE are set
遵循现有常规请求相同的规则
对話，詳見第12節。

A UAC 可能选择不添加 Alert-Info 报头字段或带有
内容处置 "alert" 用于重新邀请，因为UASs不支持
通常在接收到重新邀请时提醒用户。

与INVITE不同，它可以进行分支，而重新INVITE永远不会分支，并且
因此，仅生成一个最终的响应。 之所以这样做的原因是
re-INVITE永远不会分叉是因为Request-URI标识了
目标作为它建立的对话的UA实例，而不是
识别用户的记录地址。

请注意，UAC不得在以下范围内启动新的INVITE事务：
对端在进行另一个INVITE事务时，任一端进行对话
方向

1. 如果存在一个正在进行的INVITE客户端事务，TU必须
等待直到事务达到完成或终止
状态在新INVITE发起之前。

2. 如果存在一个正在进行的INVITE服务器事务，TU必须
等待交易达到确认或终止状态
状态在新INVITE发起之前。

然而，一个UA可以在一个INVITE期间启动一个常规事务
事务正在进行中。一个UA也可以发起一个INVITE
在进行常规交易时进行的交易。

如果UA收到对re-INVITE的非2xx最终响应，则该会话
参数必须保持不变，就像没有发出重新邀请一样。
请注意，如第12.2.1.2节所述，如果非2xx最终
响应是 481（调用/交易不存在），或 408
(请求超时)，或者完全没有收到对{v*}的响应
邀请（即，INVITE客户端返回超时）
交易时，UAC将终止对话框。

如果UAC收到对re-INVITE的491响应，它应该启动
计时器值T选择如下：

1. 如果UAC是会话ID的Call-ID的所有者
(意味着它生成了这个值)，T有一个随机选择的值
在2.1至4秒之间，以10毫秒为单位。

2. 如果UAC不是对话ID的Call-ID的所有者，T
具有在0到2秒之间的随机选择值，单位
10 毫秒。

当计时器触发时，UAC 应再次尝试重新邀请，
如果它仍然希望进行该会话修改。
示例，如果呼叫已经通过 BYE 挂断，则重新 INVITE
不会发生。

传输 re-INVITE 规则以及生成 ACK 规则
对重新邀请的2xx响应与初始邀请相同
(第13.2.1节)。

14.2 UAS 行为

第13.3.1节描述了区分传入的程序的步骤。
重新邀请来自初始邀请的邀请和处理一个重新邀请
现有对话框。

一个在发送最终消息之前收到第二个INVITE的UAS
对第一个带有更低CSeq序列号的INVITE请求的响应
相同对话框必须返回500（服务器内部错误）响应给
第二次INVITE必须包含一个Retry-After头部字段，其中包含
随机选择介于0到10秒之间的值。

一个在对话中接收到INVITE的同时它已发送的INVITE的UAS
在该对话框进行中时必须返回491（请求挂起）
对收到的INVITE的响应。

如果UA收到一个现有对话的重新INVITE，它必须检查
任何版本标识符在会话描述中，或者如果有
没有版本标识符，查看会话描述的内容
如果它已更改。如果会话描述已更改，UAS
必须相应调整会话参数，可能是在询问之后
用户进行确认。

会话描述的版本化可用于适应
新到会场的参会者能力，添加或删除
媒体，或将单播会议更改为多播会议。

如果新的会话描述不可接受，UAS 可以拒绝
通过返回一个488（此处不可接受）响应来处理它
邀请。 此响应应包含一个警告头字段。

如果UAS生成一个2xx响应且从未收到ACK，则
应生成一个BYE来终止对话。

一个UAS可能选择不对重传请求生成180（响铃）响应。
邀请，因为UACs通常不会将此信息渲染到
用户。同样地，UASs 可能会选择不使用 Alert-Info
标题字段或响应中带有“ alert ”Content-Disposition的正文
到重新邀请。

一个提供2xx（因为INVITE不包含）的UAS
一个报价）应将报价构建成好像UAS正在提出
全新的呼叫，受发送报价的约束条件限制
更新现有会话，如[13]中所述，在SDP的情况下。
具体来说，这意味着它应该包含尽可能多的媒体格式
并且UA愿意支持的媒体类型。UAS必须
确保会话描述与其前一个重叠
会话描述在媒体格式、传输或其他参数中
需要来自同侪的支持。这是为了避免需要
对等方拒绝会话描述。然而，如果它
不可接受UAC，UAC应生成一个回答
有效的会话描述，然后发送一个BYE来终止
会话

15 会话终止

本节描述了终止会话的流程
由SIP建立。会话状态和{v*}状态
对話非常密切相关。当与一个会话启动时
邀请，每个来自不同UAS的1xx或2xx响应都会创建
对話，如果該回應完成提議/回答交換，則
也创建了一个会话。因此，每个会话都是“ 关联 ”的。
使用一个单独的对话框 - 导致其创建的那个对话框。
初始INVITE生成非2xx的最终响应，该响应终止
所有会话（如果有）以及所有对话（如果有）已被创建
通过响应请求。凭借完成 {v*}
事务，非2xx最终响应也阻止了进一步的会话
从作为INVITE的结果被创建。BYE请求是
用于终止特定会话或尝试会话。在此
情况，具体的会话是具有另一端对等UA的那个
对话框的侧面。当在对话框中接收到BYE时，任何会话
与该对话框相关的操作应终止。一个UA不得发送
BYE 对话外部。 通话方的 UA 可能会发送 BYE 以表示任一
已确认或早期对话，以及被叫方的UA可能会发送一个BYE
已确认的对话，但不得在早期对话中发送 BYE。

然而，被调用方的UA不得在已确认的对话中发送BYE
直到它收到对其2xx响应的ACK确认或直到服务器
交易超时。如果未定义其他SIP扩展
应用层状态与对话相关，BYE 也
结束对话框。

非2xx最终响应对INVITE对话的影响
会话使使用 CANCEL 吸引人。 CANCEL 尝试
强制对INVITE请求返回非2xx响应（特别是487）。
因此，如果一个UAC（用户账户控制）希望完全放弃其呼叫尝试，
它可以发送一个CANCEL。如果INVITE导致2xx最终响应
到INVITE，这意味着UAS接受了邀请，同时
取消操作正在进行中。UAC可能会继续处理会话
由任何2xx响应建立，或MAY使用BYE终止它们。

SIP 中“ 挂断 ”的概念定义不明确。
特定于特定用户界面，尽管很常见。
通常，当用户挂断时，这表示他们想要
终止建立会话的尝试，以及终止任何
会话已创建。对于调用者的UA，这表示
如果初始INVITE没有生成最终
响应，并在最终之后对所有已确认的对话说再见
响应。对于被叫方的UA，通常意味着BYE；
大概，当用户拿起电话时，一个2xx发生了
生成，因此挂断会在ACK之后导致BYE
已接收。 这并不意味着用户在 {v*} 之前不能挂断
收到ACK确认，这仅仅意味着他手机中的软件
需要保持状态一段时间以便清理
正确。如果特定的UI允许用户拒绝一个
在回答之前调用，403（禁止）是一个不错的选择
表达这一点。根据上述规则，不能发送BYE。

15.1 Terminating a session 使用 BYE 请求终止会话 st 翻译文本：st

15.1.1 UAC 行为

一个 BYE 请求的构建方式与其他任何请求相同
对話，如第12节所述。

一旦构建了BYE，UAC核心创建一个新的非INVITE
客户端事务，并将其 BYE 请求传递给它。UAC 必须
考虑会话已终止（因此停止发送或
监听媒体）一旦BYE请求传递到
客户端事务。如果BYE的响应是481
(调用/交易不存在) 或 408（请求超时）或无

所有对 BYE 的响应都收到（即，超时发生）
由客户端事务返回的），UAC 必须考虑
会话和对话框已终止。

15.1.2 UAS行为

无人机系统首先根据通用无人机系统处理BYE请求
处理在第8.2节中描述。一个接收BYE的UAS核心
请求检查是否与现有对话匹配。如果BYE不
匹配现有对话，UAS核心应生成一个481
(调用/交易不存在) 响应并将其传递到
服务器事务。

这条规则意味着一个由UAC发送且不带标签的BYE将被
已拒绝。这是对RFC 2543的变更，该RFC允许BYE
没有标签。

一个接收现有对话BYE请求的UAS核心必须遵循
12.2.2节中处理请求的程序。一旦完成，
UAS 应该终止会话（因此停止发送和
监听媒体）。唯一一种它可以选择不进行的情况是
多播会话，即使其他参与者也可能加入
参与者已终止其在对话中的参与
会话。无论它是否结束其在会话中的参与，
UAS核心必须对BYE生成一个2xx响应，并且必须传递
该服务器事务用于传输。

UAS 必须仍然对收到的任何待处理请求做出响应
对话。建议使用487（请求终止）响应
生成到那些挂起的请求。

16 代理行为

16.1 概述

SIP 代理是路由 SIP 请求到用户代理的元素
服务器和SIP响应到用户代理客户端。一个请求可能
遍历其前往UAS途中的多个代理。每个代理都将进行路由
决策，在转发给下一个处理之前修改请求
元素。响应将通过同一组代理路由
遍历请求的逆序。

代理是一个SIP元素的逻辑角色。当请求
到达，一个可以充当代理角色的元素首先决定
如果它需要独立响应请求。例如，对于{v*}
请求可能格式不正确或元素可能需要从
客户端在充当代理之前。该元素可以响应任何

适当的错误代码。当直接响应请求时，
元素正在扮演UAS的角色，必须按照以下描述进行操作
第8.2节。

一个代理可以为每个操作以有状态或无状态模式运行
新请求。当无状态时，代理充当简单的转发
元素。它将每个请求向下传递给单个元素
根据基于 {v*} 的定位和路由决策确定
请求。它简单地将其收到的每个响应向上游转发。
无状态代理一旦消息被丢弃关于消息的信息
已转发。一个有状态的代理会记住信息
(具体来说，交易状态)关于每个传入请求以及任何
请求它作为处理传入请求的结果发送。
使用此信息影响未来消息的处理
与该请求相关。一个有状态的代理可以选择 “ 分支 ”
一个请求，将其路由到多个目的地。任何请求
转发到多个位置时必须进行状态管理。

在某些情况下，代理可能会使用有状态的方式转发请求
传输（如TCP）而不具有事务状态。
实例，一个代理可以转发一个请求从一条TCP连接到
只要放置足够的 {v*}，就可以无状态地进行另一笔交易
消息中的信息以便能够向下转发响应
相同的连接请求到达的地方。请求被转发
在不同类型的运输之间，其中代理的TU必须采取
在确保一种运输方式可靠交付中发挥积极作用
必须以状态方式转发事务。

一个有状态的代理可以在任何时候转换为无状态操作
在处理请求期间，只要它没有做任何事情
否则将阻止其最初无状态
(分支，例如，或生成100个响应)。当
执行此类转换时，所有状态都被简单地丢弃。
代理不应发起一个 CANCEL 请求。

当无状态或状态行为时，涉及的大部分处理
对于请求是相同的。接下来的几个小节是编写的
从有状态代理的角度来看。最后一节调用
那些无状态代理表现不同的地方。

16.2 状态化代理

当处于状态时，代理仅是一个纯SIP事务处理引擎。
其行为在此以服务器和客户端为模型进行建模
交易定义在第17节中。有状态代理有一个服务器
与一个或多个客户端事务相关联的交易
高层代理处理组件（见图3），称为
代理核心。一个进入的请求由服务器处理

事务。来自服务器的请求事务传递给
代理核心。代理核心确定请求的路由位置，
选择一个或多个下一跳位置。一个出向请求
每个下一跳位置都由其关联的客户端进行处理
交易。代理核心收集来自客户端的响应
交易并使用它们向服务器发送响应
交易。

每个新的状态化代理为每个新的 {v*} 创建一个新的服务器事务
请求已接收。任何请求的重传将随后进行
由该服务器处理，根据第17节。代理核心
MUST作为UAS对待发送即时临时
在该服务器事务（如100 Trying）中，如所述
第8.2.6节。因此，有状态的代理不应生成100
(尝试)对非-INVITE请求的响应。

这是一个代理行为的模型，而不是软件模型。
实现可以自由采取任何复制以下内容的任何方法
外部行为此模型定义的。

对于所有新的请求，包括任何未知方法的请求，一个元素
意图代理请求必须：

1. 验证请求（第16.3节）

2. 预处理路由信息（第16.4节）

3. 确定请求的目标（第16.5节）

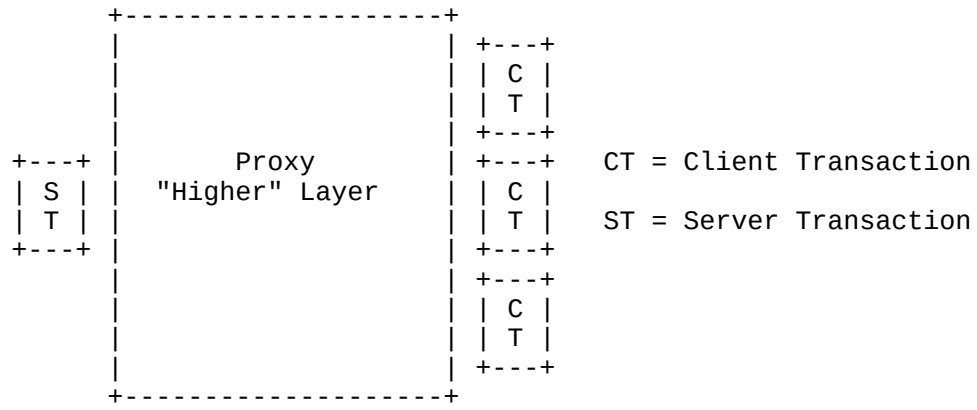

图 3：有状态代理模型

4. 将请求转发到每个目标（第16.6节）

5. 处理所有响应（第16.7节）

16.3 请求验证

在元素可以代理请求之前，它必须验证消息的
有效性。一个有效的消息必须通过以下检查：

1. 合理的语法

2. URI方案

3. 最大转发次数

4. (可选) 循环检测

5. 代理-要求

6. 代理授权

如果这些检查中的任何一个失败，该元素必须像用户代理一样表现
服务器（见第8.2节）并返回一个错误代码。

请注意，检测合并请求不需要代理。
不得将合并请求视为错误条件。端点
接收请求将解决如第{v*}节所述的合并
8.2.2.2.

1. 合理的语法检查

请求必须格式良好，足以由服务器处理
交易。 任何涉及这些剩余部分的组件。
请求验证步骤或请求转发部分必须
良好格式。任何其他组件，无论是否良好格式，都应该
被忽略且在消息转发时保持不变。对于
实例，一个元素不会因为以下原因拒绝请求
无效的日期标题字段。同样，代理也不会删除
无效的日期标题字段，在转发请求之前。

此协议旨在可扩展。未来的扩展可能
定义新方法和头字段的时间。一个元素必须
不拒绝代理一个请求，因为它包含一个方法或
标题字段它不知道的。

2. URI方案检查

如果请求URI包含一个其方案不被理解的URI,
代理，代理应拒绝带有416（不支持）的请求
URI 方案) 响应。

3. 最大转发数检查

Max-Forwards报头字段（第20.22节）用于限制
元素数量，SIP请求可以遍历的。

如果请求不包含 Max-Forwards 头字段，则此
检查通过。

如果请求包含一个带有字段的最大转发头字段
值大于零，检查通过。

如果请求包含一个带有字段的最大转发头字段
零值（0），该元素必须不转发请求。如果
请求为 OPTIONS，元素可以充当最终
接收方并按第11节回复。否则，该元素必须
返回一个483（跳数过多）响应。

4. 可选的循环检测检查

一个元素在转发之前可以检查是否存在转发环路
请求。如果请求包含一个带有已发送的Via头字段
通过值等于之前请求中放入的值
代理，请求已被此元素转发过。
请求要么是循环的，要么是合法地螺旋通过
元素。为了确定请求是否已循环，该元素可以
执行第8步中描述的分支参数计算
第16.6节在此消息中，并将其与参数进行比较
接收在该 Via 头字段中。如果参数匹配，则
请求已循环。如果它们不同，请求正在螺旋上升,
处理继续。如果检测到循环，则元素可以
返回一个482（循环检测）响应。

5. 代理-要求检查

未来对此协议的扩展可能会引入以下功能
需要由代理进行特殊处理。端点将包括一个
请求中使用这些功能的Proxy-Require头字段,
告诉代理除非功能是
理解了。

如果请求包含Proxy-Require头字段（第{v*}节）
20.29) 使用一个或多个选项标签，此元素不做
理解，该元素必须返回420（错误扩展）
响应。响应必须包含一个不支持的（章节
20.40) 头部字段列出该元素未使用的选项标签
理解。

6. 代理授权检查

如果元素在转发请求之前需要凭证，
请求必须按照第22.3节所述进行检查。
章节还定义了元素在检查时必须执行的操作
失败。

16.4 路线信息预处理

代理必须检查请求的Request-URI。如果
请求的请求URI包含一个此代理之前包含的值
放入Record-Route报头字段中（参见第16.6节第4项），
代理必须将请求中的 Request-URI 替换为最后一个
值来自路由头字段，并从该字段中移除该值
路由报头字段。代理必须随后继续处理，就像它已经收到 {v*}一样。
这个修改后的请求。

这只会发生在向其发送请求的元素时
代理（可能是一个端点）是一个严格的路由器。 这
重写接收是必要的，以实现向后兼容
与这些元素一起。它还允许此后的元素
规范以严格路由保留请求URI
代理（见第12.2.1.1节）。

此要求并不强制代理保持状态
检测它之前放置在Record-Route报头字段中的URI。
相反，代理只需要在那些URI中放置足够的信息
识别它们作为它们后来出现时提供的值。

如果请求URI包含一个maddr参数，代理必须检查
查看其值是否在代理的地址或域名集合中
已配置为负责。如果请求URI有一个maddr
参数由代理负责的值，以及请求
已使用指定的端口和传输方式（明确或通过）接收
默认)在请求URI中，代理必须删除maddr以及任何
非默认端口或传输参数，并继续处理，仿佛
这些值在请求中未曾出现。

一个请求可能带有与代理匹配的maddr，但在
端口或传输方式与URI中指示的不同。
一个请求需要使用指示的代理进行转发
港口和运输。

如果路由头字段中的第一个值指示此代理，
代理必须从请求中移除该值。

16.5 确定请求目标

接下来，代理计算请求的目标（们）。 目标集合
目标将由请求内容预先确定
或将从抽象位置服务中获取。每个目标
在集合中表示为URI。

如果请求的 Request-URI 包含一个 maddr 参数，则
请求URI必须作为唯一的目标放入目标集合中
URI，并且代理必须继续到第16.6节。

如果Request-URI的域指示一个域，则此元素是
不负责，请求URI必须放置到目标
将作为唯一目标设置，元素必须继续执行任务
请求转发（第16.6节）。

存在许多情况下，代理可能会收到
请求一个它不负责的域名。一个防火墙代理
处理出站电话（类似于HTTP代理处理出站电话的方式）
请求（requests）是这种情况可能发生的例子。

如果请求的目标集尚未预先确定为
上述描述表明，该元素负责
域在Request-URI中，并且该元素可以使用任何机制
它希望确定发送请求的位置。这些中的任何一个
机制可以被建模为访问一个抽象的位置服务。
这可能包括从位置服务获取信息
由SIP注册器创建，读取数据库，查询存在状态
服务器，利用其他协议，或简单地执行
算法替换在Request-URI上。当访问
由注册商构建的位置服务，请求URI必须
首先应按第10.3节所述进行规范化，然后才能使用
作为一个指标。这些机制的输出用于构建
目标集。

如果请求URI没有提供足够的信息，
代理确定目标集，它应返回485（模糊）
响应。此响应应包含一个Contact头字段
包含尝试的新地址的URI。例如，一个INVITE

to sip:John.Smith@company.com 可能在一个代理处存在歧义
位置服务列出了多个约翰·史密斯。参见第
21.4.23 有关详情。

任何关于请求或当前环境的信息
元素可以用于构建目标集。
实例，根据内容或
存在头部字段和主体，一天中的时间
请求到达时，请求到达的接口，
请求失败，或者甚至是元素的当前级别
利用率

随着通过这些服务定位潜在目标，它们的URI
被添加到目标集中。目标只能放置在
目标集一旦确定。如果目标URI已存在于集合中
(基于URI类型的等价定义)，它必须不
再次添加。

A代理不得向目标集添加额外的目标。
原始请求的请求URI未指示资源
代理负责。

一个代理只能在请求期间更改请求的 Request-URI。
如果它负责该URI，则转发。如果代理不
负责该URI，不会在3xx或416上递归
响应如下所述。

如果原始请求的Request-URI指示了一个资源，则
代理负责，代理可以继续添加目标到
请求转发开始后的集合。它可以使用任何
在处理过程中获得的信息以确定新的目标。
例如，一个代理可能选择将获得的联系信息整合到
将重定向响应（3xx）转换为目标集。如果代理使用
动态构建目标集时的信息来源（对于
实例，如果它咨询了一个SIP注册器），它应该监控
源用于处理请求的持续时间。 新位置
应随着可用性添加到目标集中。
以上，任何给定的URI不得添加到集合中超过一次。

仅允许将URI添加到集合中一次可以减少
不必要的网络流量，并在整合{v*}的情况下
重定向请求的联系人防止无限递归。

例如，一个平凡的定位服务是一个 "无操作"，其中
目标URI等于传入请求URI。请求被发送
到特定的下一跳代理进行进一步处理。在请求期间

转发第16.6节第6项，下一个跳转点的身份，
以SIP或SIPS URI的形式表示，作为最高路由插入
请求中的标题字段值。

如果请求URI指示此代理中不存在资源的资源
存在，代理必须返回404（未找到）响应。

如果应用上述所有操作后目标集仍然为空，则
代理必须返回一个错误响应，该响应应该是480
(暂时不可用) 响应。

16.6 请求转发

一旦目标集非空，代理可以开始转发
请求。一个有状态的代理可以按任何顺序处理该集合。
可以串行处理多个目标，允许每个客户端
交易完成后再开始下一笔。它可能开始
客户端与每个目标并行进行交易。它也可能
任意将集合划分为组，处理这些组
串行处理并并行处理每个组中的目标。

一个常见的排序机制是使用目标对象的qvalue参数
从“ 联系 ”标题字段获取（见第20.10节）。目标为
从最高q值处理到最低。q值相等的目标
可能并行处理。

一个有状态的代理必须有一个机制来维护目标集作为
响应被接收并将响应与每个转发的相关联
请求与原始请求相同。对于本模型而言，
这个机制是由代理层创建的“ 响应上下文 ”
在转发第一个请求之前。

对于每个目标，代理按照以下方式转发请求
步骤：

1. 复制收到的请求

2. 更新请求URI

3. 更新 Max-Forwards 头部字段

4. 可选地添加 Record-route 报头字段值

5. 可选添加额外的头部字段

6. 后处理路由信息

7. 确定下一跳地址、端口和传输

8. 添加一个Via头字段值

9. 如有必要，添加 Content-Length 头字段

10. 前进新请求

11. 设置计时器C

每个这些步骤的详细信息如下：

1. 复制请求

代理从接收到的请求的副本开始。
MUST最初必须包含所有标题字段从
收到请求。 处理中未详细说明的字段
以下描述的必须不得删除。副本应保持
请求中头部字段的排序方式。
代理不得重新排序具有公共字段的字段值
姓名（见第7.3.1节）。代理不得添加、修改、
或删除消息正文。

实际实现无需执行复制；主要
要求是每个下一跳的处理从以下开始
相同的请求。

2. 请求URI

请求URI在副本的开始行中必须被替换为
该目标的URI。如果URI包含任何参数
不允许在请求URI中使用，它们**必须**被删除。

这是代理角色的本质。这是机制
通过其中代理将请求路由到其目的地。

在某些情况下，接收到的 Request-URI 被放置到
目标集未经修改。对于该目标，
替换上面的操作实际上是一个无操作。

3. 最大转发次数

如果副本包含 Max-Forwards 头字段，则代理
必须将其值减一（1）。

如果副本不包含 Max-Forwards 头字段，则
代理必须添加一个字段值，该值应为70。

一些现有的UA将不会提供Max-Forwards头部字段
在一个请求中。

4. 记录路由

如果此代理希望继续在未来的请求路径上
在一个由该请求创建的对话框中（假设请求
创建一个对话框时，它必须插入一个Record-Route报头字段
将值复制到任何现有Record-Route报头之前
字段值，即使已经存在路由头字段。

请求建立对话可能包含预加载的路由
表头字段。

如果此请求已包含在对话中，代理应
在需要保持的情况下插入Record-Route报头字段值
在对话框中未来请求的路径上。在正常
端点操作，如第12节所述，这些记录
路由报头字段值不会对路由产生影响
端点使用的集合。

代理将保持在路径上，如果它选择不插入
记录-路由报头字段值已包含在已发送的请求中
部分对话。然而，它将被从路径中移除
当失败的端点重新构成对话时。

A代理可以将其中的Record-Route报头字段值插入到任何
请求。如果请求不启动对话，则
端点将忽略该值。详情请见第12节。
如何端点使用Record-Route报头字段值来
构建路由报头字段。

每个请求路径中的代理选择是否添加
独立记录路由报头字段值 - 存在
请求中的Record-Route报头字段并不强制
代理以添加一个值。

The URI placed in the Record-Route header field value MUST be a
SIP或SIPS URI。此URI必须包含一个lr参数（见
第19.1.1节）。此URI可能对每个{v*}不同
目标地址，请求将被转发到。URI不应
包含运输参数，除非代理有知识
（例如在私有网络中）下一个下游元素
这将支持后续请求的路径
运输。

该代理提供的URI将被某些其他元素使用
为了进行路由决策。这个代理通常没有方法
了解该元素的能力，因此它必须
仅限于SIP的强制元素
实现：SIP URI 和 TCP 或 UDP 传输之一。

The URI placed in the Record-Route header field MUST resolve to 记录路由头字段中放置的URI必须解析为
当插入元素时（或合适的替代品）
它应用了[4]的服务器位置程序，因此
后续请求达到相同的SIP元素。如果
请求URI包含SIPS URI或最顶层的路由头
字段值（在处理第6条后的后处理中）包含一个
SIPS URI，放置在Record-Route报头字段中的URI
MUST 是一个 SIPS URI。此外，如果请求未
接收通过TLS，代理必须插入一个Record-Route头
字段。以类似的方式，一个接收请求的代理
通过TLS，但生成一个不带SIPS URI的请求
请求URI或最顶层路由头字段值（在帖子后）
处理第6)项，必须插入Record-Route报头
字段不是SIPS URI。

一个安全边界的代理必须保持在边界上
在整个对话中。

如果放置在Record-Route报头字段中的URI需要被翻译
重写当它通过响应返回时，URI
必须足够不同，以便当时能够定位。 （请求
可能通过此代理螺旋，导致出现多个
记录-路由报头字段值被添加）。第8项
第16.7节推荐了一种机制来使URI
足够明显。

代理可能包括参数在Record-Route报头中
字段值。这些将在某些响应中回显。
请求如对INVITE的200（OK）响应。此类
参数可能有助于在消息中保持状态
而不是代理。

如果代理需要位于任何类型对话的路径中（例如
作为跨越防火墙的一方），它应该添加一个Record-Route
请求中每个方法的标题字段值，它不
理解，因为那种方法可能有对话语义。

The URI a proxy places into a Record-Route header field is only
有效于由事务创建的任何对话的整个生命周期
在它出现的地方。例如，一个会话状态代理，可以
拒绝接受包含该值的未来请求
请求URI在对话结束后。非对话-
状态代理，当然，没有对话何时发生的概念
已终止，但他们可能在其中编码足够的信息
与未来对话标识符进行比较的值
请求和可能拒绝与该信息不匹配的请求。
端点不得使用从记录路由获得的URI
标题字段在提供它的对话框外部。参见

第12节了解更多关于端点使用的相关信息
记录-路由报头字段。

记录路由可能在某些服务中是必需的，其中
代理需要观察对话中的所有消息。然而，它
减慢处理速度并损害可扩展性，因此代理
仅当需要特定服务时才应记录路由。

记录路由过程旨在为任何SIP工作
请求启动对话。INVITE是唯一此类
在此规范中请求，但对协议的扩展
可以定义其他。

5. 添加额外的头部字段

The proxy MAY add any other appropriate header fields to the 代理可以添加任何其他适当的头部字段到
复制此点。

6. 后处理路由信息

一个代理可能有一个本地策略，该策略强制要求一个请求
在交付之前访问一组特定的代理
目标。一个代理必须确保所有此类代理都是
松散的交换机。通常，这只能通过
确定如果代理在同一行政区域内
域。 这组代理由一组 URI 表示
(每个都包含lr参数)。此集合必须
推入复制前的路由头字段中
现有值，如果存在。如果存在路由头字段
不存在，它必须被添加，包含该URI列表。

如果代理有一个本地策略，要求请求
访问一个特定的代理，替代推送路由
值放入路由头字段是为了绕过转发
逻辑如下第10项，而是直接发送请求到
该特定代理的地址、端口和传输方式。
请求包含一个路由头字段，此替代方案绝不可
除非已知下一跳代理是宽松的，否则将被使用
路由器。否则，可以使用这种方法，但路由
插入机构因其坚固性而更受青睐，
灵活性、通用性和操作一致性。
此外，如果Request-URI包含SIPS URI，TLS必须
用于与该代理进行通信。

如果副本包含路由头字段，代理必须
检查其第一个值中的URI。如果该URI不
包含一个lr参数，代理必须按如下方式修改副本
遵循以下内容：

- 代理必须将请求URI放入路由头
字段作为最后一个值。

- 代理必须随后放置第一个路由报头字段值
将值从路由中移除
表头字段。

将 Request-URI 添加到 Route 头字段是其中一部分
一种用于传递该Request-URI中信息的机制
通过严格路由元素。 "弹出" 第一个路由
表头字段值格式化为Request-URI的消息
一种严格路由元素期望接收它的方式（带有其
请求URI中的自身URI和下一个要访问的位置
第一个路由报头字段值）。

7. 确定下一跳地址、端口和传输

代理可能有一个本地策略将请求发送到
特定IP地址、端口和传输，独立于
路由和请求URI的值。此类策略不得
如果代理不确定IP地址、端口和
传输对应于一个松散路由器的服务器。
然而，通过这种方式发送请求的机制
特定下一跳不建议使用；而是使用路由报头
字段应如上所述用于该目的。

在缺少此类主导机制的情况下，代理
应用[4]中列出的程序如下以确定
发送请求的位置。如果代理已重新格式化
请求发送到如所述的严格路由元素
第6步以上，代理必须应用那些程序到
请求的请求URI。否则，代理必须应用
路由头字段中第一个值的程序，如果
当前，否则为 Request-URI。这些程序将生成
有序的（地址，端口，传输）元组集合。
独立于作为输入使用的哪个URI
[4]中的程序，如果Request-URI指定了SIPS
资源，代理必须遵循[4]中的程序，就像它正在执行一样。
输入URI是一个SIPS URI。

如[4]中所述，代理必须尝试交付
消息发送到该集合中的第一个元组，并继续通过
按顺序设置，直到交付尝试成功。

对于每个尝试的元组，代理必须将消息格式化为
适用于元组并使用新的发送请求
客户端交易，如第8至10步所述。

由于每次尝试都使用一个新的客户端事务，因此它代表
一个新分支。因此，与Via一起提供的分支参数
表头字段在第8步插入时必须对每个不同
尝试。

如果客户端事务报告发送请求失败
或从其状态机超时，代理继续到
下一个有序集中的地址。如果有序集是
疲惫，请求无法转发到此元素
目标集。代理不需要在其中放置任何内容
响应上下文，但否则表现得像这个元素
目标集返回了一个408（请求超时）的最终响应。

8. 添加一个Via头字段值

代理必须将一个 Via 头部字段值插入到副本中
在现有的 Via 报头字段值之前。 的构建
此值遵循第8.1.1.7节相同的指南。
这表示代理将计算其自身的分支
参数，对于该分支将是全局唯一的，并且
包含必要的魔法饼干。注意，这表示
分支参数将因实例不同而不同
通过代理的螺旋或循环请求。

代理选择检测循环的具有额外的约束
在它们用于构建分支参数的值中。
一个选择检测循环的代理应该创建一个分支
参数可以通过实现分为两部分。
第一部分必须满足第8.1.1.7节的约束条件。
上述所述。第二个用于执行循环检测
区分环和螺旋。

循环检测是通过验证，当请求
返回代理，那些对 {v*} 有影响的字段
请求的处理方式没有改变。放置的值
在这个分支参数部分应反映所有
那些字段（包括任何路由、代理-要求以及代理-）
授权头字段）。这是为了确保如果{v*}
请求被路由回代理，其中一个这些字段
更改，它被视为螺旋而不是循环（见第节）
16.3). 创建此值的一种常见方法是计算一个 {v*}
加密哈希的To标签，From标签，Call-ID报头
字段，请求接收到的请求-URI（在
翻译)，最顶部的 Via 标题，以及序列号
从CSeq报头字段，除任何Proxy-Require之外
存在且可能存在的 Proxy-Authorization 头字段。

算法用于计算哈希的实现是依赖于实现的,
但是MD5（RFC 1321 [35]），以十六进制表示，是一个
合理的選擇。 （對於令牌來說，Base64是不允許的。）

如果代理想要检测循环，它将“ 分支 ”参数
供应必须依赖于影响所有加工信息的所有信息
一个请求，包括传入的 Request-URI 和任何头部
字段影响请求的接入或路由。这是
必须区分循环请求与那些的请求
路由参数在返回之前已更改
服务器。

请求方法不得包含在计算中
分支参数。特别是，取消（CANCEL）和确认（ACK）请求
（对于非2xx响应）必须与分支值相同
相应的请求他们取消或确认。该分支
参数用于在服务器上关联那些请求
处理它们（见第17.2.3节和第9.2节）。

9. 如有必要，添加 Content-Length 头字段

如果请求将通过流-发送到下一跳
基于传输，且副本不包含Content-Length头
字段，代理必须插入一个具有正确值的{v*}。
请求正文（见第20.14节）。

10. 前向请求

A stateful proxy 一定要为这个创建一个新的客户端事务
请求如第17.1节所述，并指示
交易使用地址、端口发送请求
运输步骤7中确定的。

11. 设置计时器C

为了处理一个INVITE请求永远无法处理的情况
生成一个最终响应，TU使用一个名为计时器的东西
计时器 C. 计时器 C 必须为每个客户端事务设置时
一个INVITE请求被代理。计时器必须大于3
分钟。第16.7节项目2讨论了此计时器的用法。
更新后包含临时回复，第16.8节讨论
处理触发时。

16.7 响应处理

当元素收到响应时，它首先尝试定位一个
客户端事务（第17.1.3节）与响应匹配。如果没有
当找到时，元素必须处理响应（即使它是一个
信息响应)作为一个无状态代理（如下所述）。如果
匹配成功，响应被交给客户端事务。

转发针对客户端事务（或更多）的响应
通常任何有关已发送相关请求的知识是
未找到可提高鲁棒性。特别是，它确保了
"晚" 2xx 对 INVITE 请求的响应被正确转发。

随着客户端事务将响应传递到代理层,
以下处理必须进行：

1. 找到适当响应上下文

2. 更新临时响应的计时器C

3. 移除最顶部的 Via

4. 将响应添加到响应上下文中

5. 检查此响应是否应立即转发

6. 当需要时，从以下选项中选择最佳最终响应：
响应上下文

如果每个客户端之后都没有转发最终响应
与响应上下文相关的交易已被终止,
代理必须从它所拥有的那些中选择并转发“ 最佳 ”响应
迄今为止已看到。

以下处理必须在每个响应上执行
转发。很可能每个请求都会有多个响应
将被转发：至少每个临时和一个最终响应。

7. 如有必要，聚合授权头字段值

8. 可选重写Record-Route报头字段值

9. 前传响应

10. 生成任何必要的 CANCEL 请求

每个上述步骤的详细信息如下：

1. 查找上下文

   代理定位到它之前创建的“ 响应上下文 ”
   转发使用所述密钥的原请求
   第16.6节。剩余的处理步骤发生在
   此上下文。

2. 更新临时响应的计时器C

   对于INVITE事务，如果响应是临时的
   响应状态码为101至199（包括101和199）
   但是100），代理必须重置该客户端的计时器C
   交易。计时器可以重置为不同的值，但
   此值必须大于3分钟。

3. 通过

   The proxy removes the topmost Via header field value from the
   响应。

   如果响应中不再存在任何 Via 头部字段值，则
   响应是为此元素准备的，不得转发。
   本节所述处理的剩余部分是
   未在此消息上执行，UAC处理规则
   在8.1.3节中描述的（运输）被替代描述。
   层处理已经发生）。

   这将在元素生成时发生，例如
   取消第10节中描述的请求。

4. 添加对上下文的响应

   收到的最终响应存储在响应上下文中
   直到服务器事务生成最终响应
   与此上下文相关。响应可能是一个候选人
   为了在该服务器上返回最佳最终响应
   交易。此响应中的信息可能需要。
   构建最佳响应，即使这个响应不会被选择。

   如果代理选择在3xx中的任何联系人上递归
   通过将它们添加到目标集中进行响应，它**必须**删除它们
   从添加响应之前的响应中
   上下文。然而，代理不应递归到非SIPS URI
   如果原始请求的 Request-URI 是 SIPS URI。如果

代理在所有3xx响应的联系人上递归，
代理不应将生成的无接触响应添加到
响应上下文。

在添加响应之前移除接触
上下文阻止上游的下一个元素重试
位置 此代理已尝试过的位置。

3xx 响应可能包含 SIP、SIPS 和非 SIP 的混合
统一资源标识符。代理可以选择递归处理SIP和SIPS URI
将余数放入响应上下文中以
返回，可能在最终响应中。

如果代理收到一个416（不支持的URI方案）响应
一个请求的 Request-URI 方案不是 SIP，而是该方案
在原始接收到的请求中是SIP或SIPS（即，是
代理将方案从SIP或SIPS更改为其他
当它代理请求时，代理应添加一个新的URI到
目标集。此URI应是一个SIP URI版本。
非SIP URI，刚刚尝试过的。在tel URL的情况下，
这是通过放置电话用户部分来实现的
将tel URL转换为SIP URI的用户部分，并设置
主机部分到先前请求发送的域名。
查看第19.1.6节以获取更多关于从tel形成SIP URI的详细信息。
URLs.

与3xx响应一样，如果代理通过{v*}在416上“递归”，
尝试使用SIP或SIPS URI，416响应不应
被添加到响应上下文中。

5. 检查转发响应

直到服务器事务收到最终响应，
以下响应必须立即转发：

- 任何除100（T以外的临时响应 尝试）

- 任何2xx响应

如果收到6xx响应，则不会立即转发，
但是，有状态的代理应该取消所有客户端挂起的
交易如第10节所述，并且**必须**不创建
任何在此上下文中的新分支。

这是对RFC 2543的变更，该规范强制要求代理
立即转发6xx响应。对于INVITE
交易，这种方法存在一个问题，即2xx响应
可能到达另一个分支，在这种情况下，代理将

必须转发2xx。结果导致UAC可以
接收一个6xx响应，随后是一个2xx响应，这应该
永远不允许发生。根据新规定，在
接收一个6xx，代理将发出一个CANCEL请求，其中
通常会导致所有未解决事项的487个响应
客户端交易，然后在那个点6xx是
转发至上游。

在服务器事务发送最终响应后，
以下响应必须立即转发：

- 任何对INVITE请求的2xx响应

A stateful proxy 不得立即转发任何其他
响应。特别是，一个有状态的代理不得转发
任何100（尝试）响应。那些候选响应
为以后作为“ 最佳 ”回复而收集的
如步骤“ 添加响应到上下文 ”中所述。

任何选择立即转发的响应必须进行处理
如步骤中所述 "聚合授权报头字段"
值"通过"记录-路由"。

此步骤与下一步骤结合，确保状态 {v*}
代理将精确地转发一个非INVITE的最终响应
请求，并且是恰好一个非2xx响应或一个或多个
2xx 响应一个 INVITE 请求。

6. 选择最佳回应

一个有状态的代理必须向响应发送一个最终响应
上下文的服务器事务如果没有最终响应已被
立即按照上述规则和所有客户端转发
在此响应上下文中的交易已被终止。

The stateful proxy MUST choose the "best" final response among
那些接收并存储在响应上下文中。

如果没有上下文中的最终响应，代理必须
发送一个408（请求超时）响应到服务器
交易。

否则，代理必须转发来自响应的响应
存储在响应上下文中。它必须从6xx中选择。
类响应，如果存在上下文中。如果没有6xx类
响应存在时，代理应从最低的 {v*} 选择
响应类存储在响应上下文中。代理可以
选择该选定类别中的任何响应。代理应

优先考虑提供影响 {v*} 的信息的回复
此请求的重新提交，例如401、407、415、420等
484 如果选择了4xx类。

一个接收503（服务不可用）响应的代理
除非它能确定任何 {v*}，否则不应将其向上游转发
随后的请求它可能代理的也将生成一个503。
换句话说，转发一个503意味着代理知道它
无法处理任何请求，不仅仅是针对Request-的请求
请求中生成的503错误的URI。如果唯一
响应接收到的状态码是503，代理应生成
一个500响应并将其转发到上游。

转发响应必须按照以下步骤进行处理
"聚合授权报头字段值" 通过 "Record"
路由"。

例如，如果一个代理将请求转发到4个位置，并且
收到503、407、501和404响应，它可能会选择
转发407（需要代理身份验证）响应。

1xx 和 2xx 响应可能涉及建立
对话框。当请求不包含To标签时，To标签
在响应中由UAC用于区分多个
对创建对话框请求的响应。代理不得
在1xx或2xx响应的“ 收件人 ”头字段中插入一个标签
如果请求中不包含一个。 代理不得修改
响应头字段中1xx或2xx响应的标签。

由于代理可能不会将标签插入到“ 收件人 ”头字段中
对请求未包含的请求返回1xx响应，它不能
自行发布非100%的临时回应。然而，它
可以分支请求到与以下元素相同的UAS
代理。此UAS可以返回其自身的临时响应，
进入与发起者的早期对话
请求。UAS不必是一个离散过程，从
代理。它可能是在同一系统中实现的虚拟UAS。
代码空间作为代理。

3-6xx 响应按跳数传递。当发出 3-6xx
响应，该元素实际上充当UAS，发布
它自己的响应，通常基于收到的响应
下游元素。一个元素应当保留To标签
当简单地将3-6xx响应转发到请求时，它执行
不包含 To 标签。

一个代理不得修改任何转发响应中的“ 到 ”标签。
一个包含To标签的请求。

虽然这对上游元素没有影响
代理替换了转发3-6xx响应中的To标签，
保留原始标签可能有助于调试。

当时代理从几个
响应，从它们中选择一个To标签是任意的，并且
生成一个新的 To 标签可能会使调试更容易。
发生时，例如，当组合401（未授权）和
407 (代理身份验证所需) 挑战，或结合
联系值来自未加密和未经身份验证的3xx
响应。

7. 聚合授权报头字段值

如果所选响应是401（未授权）或407（代理
认证要求），代理必须收集任何WWW-
从认证和Proxy-Authenticate头字段值中进行认证
所有其他401（未授权）和407（代理身份验证）
所需)至今在此响应上下文中收到的响应
并且在不做修改的情况下将它们添加到这个回复中之前
转发。 结果为401（未授权）或407（代理）
认证要求）响应可能有多个WWW-
验证 AND Proxy-Authenticate 头字段值。

这是必要的，因为任何或所有目的地
请求已转发到可能已请求凭证。
客户端需要接收所有这些挑战并提供
凭证，当它重试请求时，为每个它们。
动机在此行为中由第26节提供。

8. 记录路由

如果所选响应包含 Record-Route 报头字段
原始值由该代理提供，代理可以选择
重写转发响应前的值。
允许代理为自己提供不同的URI到
下一个上游和下游元素。代理可以选择
使用此机制，无论任何原因。例如，它很有用
对于多宿主主机。

如果代理通过TLS收到请求，并将其发送出去
在非TLS连接中，代理必须重写{v*}中的URI
记录-路由报头字段应为SIPS URI。如果代理
收到非TLS连接的请求，并将其发送出去
通过TLS，代理必须重写Record-Route中的URI
标题字段应为SIP URI。

新代理提供的URI必须满足相同的
对放置在Record-Route报头字段中的URI的限制
请求（参见第16.6节第4步）具有以下
修改：

The URI 不应包含传输参数，除非
代理知道下一个上游（与
下游)元素，该元素将在后续路径中
requests 支持该传输。

当代理决定修改 Record-Route 标头时
响应字段中，它执行的操作之一是
定位它所插入的Record-Route值。
请求螺旋上升，代理插入了一个Record-Route值
在螺旋的每次迭代中，定位正确的值在
响应（必须是反向迭代中的正确迭代）
方向)是棘手的。上述规则建议使用代理
希望重写 Record-Route 头字段值插入
足够不同的 URI 传入 Record-Route 报头字段
因此可以选择正确的进行重写。
推荐实现此功能的机制是代理要
为用户添加一个唯一的代理实例标识符
URI的部分。

当响应到达时，代理修改第一个
记录路由标识符与代理实例匹配。
修改结果导致URI缺少此数据片段
附加到URI的用户部分。在下一个
迭代，相同的算法（找到最顶部的Record-Route
表头字段值与参数)将正确提取
下一条 Record-Route 报头字段值由该
代理

不是每个请求的响应，当代理添加时
记录-路由报头字段值将包含一个记录-路由
表头字段。如果响应包含 Record-Route
表头字段，它将包含代理添加的值。

9. 前向响应

在执行步骤“ 聚合 ”中描述的处理后
授权头字段值"通过"记录-路由",
代理可以执行对{v*}的任何特定功能操作
已选响应。代理不得添加、修改或
删除消息正文。除非另有说明，代理
必须不得删除除Via以外的任何报头字段值
标题字段值，在第16.7节第3项中讨论。 在
特别地，代理不得删除任何“ 已接收 ”参数

它可能已添加到下一个 Via 标头字段值中，而
处理与此响应相关的请求。
代理必须将响应传递给服务器事务
与响应上下文相关。这将导致
响应现在正发送到指示的位置
最顶部的 Via 头部字段值。如果服务器事务是
不再可用于处理传输，该元素
必须无状态地转发响应，通过将其发送到
服务器传输。服务器事务可能指示
响应发送失败或在其状态中发出超时信号
机器。这些错误将记录以供诊断目的
根据适当情况，但协议要求无需采取补救措施
从代理处。

代理必须维护响应上下文，直到其所有
相关交易已被终止，即使之后
转发最终响应。

10. 生成 CANCELs

如果转发的响应是最终响应，代理必须
生成所有待处理客户交易的取消请求
与该响应上下文相关。一个代理也应该
生成所有待处理客户交易的取消请求
与当它收到6xx响应时关联的此响应上下文
响应。一个挂起的客户端事务是指尚未完成的交易。
收到临时回复，但无最终回复（它是
在先前的状态下）并且尚未有相关的取消
为它生成。生成 CANCEL 请求的描述在
第9.1节。

The requirement to CANCEL pending client transactions upon
转发最终响应并不能保证端点
不会接收到多个 200 (OK) 响应的 INVITE。 200
(OK) 在多个分支上可能生成响应之前
取消请求可以发送和处理。此外，它
合理的预期，未来的扩展可能会覆盖这一点
要求发出取消请求。

16.8 处理计时器 C

如果计时器C应该触发，代理必须重置计时器，使用{v*}
任何它选择的值，或终止客户端事务。如果
客户端事务已收到临时响应，代理
必须生成一个与该交易匹配的取消请求。
客户端事务尚未收到临时响应，代理
必须表现得好像事务收到了一个408（请求超时）
响应。

允许代理重置计时器使代理能够动态
扩展交易寿命，基于当前条件（例如
当计时器触发时（利用）。

16.9 处理传输错误

如果传输层在尝试时通知代理一个错误
发送一个请求（见第18.4节），代理必须表现得好像
已转发请求收到503（服务不可用）响应。

如果代理在转发响应时收到错误通知，它
删除响应。代理不应取消任何未完成的
客户端与此次响应上下文相关的交易由于这个
通知。

如果代理取消其未完成的客户交易，则单个
恶意或行为不当的客户可能导致所有交易失败
通过其Via头字段。

16.10 取消处理

一个有状态的代理可能会向它所拥有的任何其他请求生成一个 CANCEL
在任何时间生成（在收到临时响应的前提下）
该请求如第9.1节所述）。代理必须取消任何
待处理与响应上下文相关的客户端交易
它收到一个匹配的取消请求。

A stateful proxy 可能会为挂起的 INVITE 生成 CANCEL 请求
客户端事务基于INVITE中指定的期间
过期头字段已过时。然而，这通常是
不必要，因为涉及的端点将处理信号
交易结束。

当一个 CANCEL 请求在具有状态的代理中由其自身处理时
服务器事务，不会为其创建新的响应上下文。
相反，代理层在其现有的响应上下文中搜索
服务器处理与此请求关联的事务
取消。如果找到匹配的响应上下文，则该元素必须
立即返回一个200（OK）响应给CANCEL请求。
在这种情况下，该元素作为定义在中的用户代理服务器进行操作
第8.2节。此外，元素必须生成取消请求
对于上下文中描述的所有待处理客户端事务
第16.7节 步骤10.

如果未找到响应上下文，则该元素没有任何
对请求应用取消的知识。它必须无状态地声明。
转发 CANCEL 请求（它可能已无状态转发）
相关请求之前）。

16.11 无状态代理

当以无状态方式操作时，代理是一个简单的消息转发器。
无状态操作时执行的处理与相同
当表现有状态时。差异在此详细说明。

无状态代理没有任何关于事务的概念，或者
响应上下文，用于描述有状态代理行为。
相反，无状态代理接收消息，包括请求和
响应，直接来自传输层（见第18节）。
结果，无状态代理不会自行重新传输消息。
他们确实会转发他们收到的所有重传（他们确实
没有能力区分重传
原始消息）。此外，在无状态处理请求时，
一个元素不得生成其自身的100（尝试）或任何其他
临时响应。

A stateless proxy 一定要验证请求，如第 {v*} 节所述
16.3 Translated Text: 16.3

A stateless proxy 一定要遵循所描述的请求处理步骤
在16.4至16.5节中，有以下例外：

o 无状态代理必须从以下中选择一个且仅选择一个目标从以下
目标集。此选择必须仅依赖于字段。
服务器的信息和时不变属性。在
特别地，一个重传的请求必须转发到
每次处理时都指向相同的目的地。此外，
取消和非路由ACK请求必须生成相同的
选择它们相关的INVITE。

A stateless proxy 一定要遵循所描述的请求处理步骤
在16.6节中，有以下例外：

空间和时间上唯一分支ID的要求
适用于无状态代理。然而，一个无状态的
代理不能简单地使用随机数生成器来计算
分支ID的第一个组件，如第X节所述
16.6 项目 8. 这是因为请求的重传
需要具有相同的值，并且无状态代理无法区分
原始请求的重传。因此，
分支参数使其独特的组成部分必须是
每次重传请求被转发时都相同。因此
对于无状态代理，分支参数必须按如下方式计算：
一个关于消息参数的组合函数，这些参数是
重传时的不变量。

The stateless proxy MAY use any technique it likes to guarantee
唯一性跨事务的分支ID。然而，
以下程序建议使用。代理检查
接收到的最顶层Via报头字段中的分支ID
请求。如果它以魔法饼干开头，则第一个
分支ID的输出请求分量被计算
作为接收到的分支ID的哈希值。否则，第一个
分支ID的组件被计算为最顶层{v*}的哈希值
通过，To头字段中的标签，From头字段中的标签
字段，Call-ID报头字段，CSeq编号（但不包括
方法)，以及从接收到的请求中获取的 Request-URI。其中一个
这些字段将在两个不同的中始终变化
交易

所有在16.6节中指定的其他消息转换
必须导致重传的相同转换
请求。特别是，如果代理插入一个Record-Route
值或推URI到路由头字段中，它必须放置
相同的值在请求的重传中。至于
通过分支参数，这表示变换
必须基于时不变配置或
请求的重传不变性质。

o 无状态代理确定将请求转发到何处，作为
描述在第16.6节第10项中。
请求直接发送到传输层而不是
通过客户端事务。

由于无状态代理必须转发重传的请求到
相同的目的地并添加相同的分支参数到
每个它们，只能使用消息中的信息
自身和针对这些的时不变配置数据
计算。如果配置状态不是时不变的
（例如，如果路由表被更新）任何请求
可能受变化影响可能不会转发
无状态地在一个等于事务超时的间隔内
窗口在更改前后。处理方法
该时间间隔内受影响的请求是一个实现
决策。一个常见的解决方案是将它们向前交易
状态地。

无状态代理不得对 CANCEL 执行特殊处理
请求。它们按照上述规则处理，与其他任何请求一样
请求。特别是，无状态代理应用相同的路由
表头字段处理以取消对任何请求的应用
其他请求。

响应处理，如第16.7节所述，不适用于{v*}。
代理以无状态行为。当无状态响应到达时
代理，代理必须检查第一个发送者值
通过标题字段值。如果该地址与代理匹配，
(它等于这个代理之前请求中插入的值)
代理必须从响应中删除该报头字段值，并且
将结果转发到下一个Via标题中指示的位置
字段值。代理不得添加、修改或删除
消息正文。除非另有说明，代理不得删除
任何其他报头字段值。如果地址不匹配
代理，消息必须被静默丢弃。

16.12 代理路由处理摘要

在不存在相反的本地政策的情况下，处理 a
代理对包含路由头字段的请求执行的操作可以
总结如下步骤。

1. 代理将检查Request-URI。如果它指示一个
资源由该代理拥有，代理将用其替换
运行位置服务的结果。否则，
代理将不会更改Request-URI。

2. 代理将检查最顶部的路由头中的URI
字段值。如果它指示此代理，则代理将其移除
从路由报头字段（此路由节点已被
达到）。

3. 代理将请求转发到指示的资源
通过最顶部的路由头字段值中的URI或
请求URI（如果没有存在路由头字段）。代理
确定使用时的地址、端口和传输方式
通过应用[4]中的程序来转发请求
那个URI。

如果路径上没有遇到严格路由元素
请求，请求URI将始终指示目标
请求。

16.12.1 示例

16.12.1.1 基本SIP梯形

这是基本SIP梯形，U1 -> P1 -> P2 -> U2，具有
同时代理记录路由。以下是流程。

U1 发送：

邀请 sip:callee@domain.com SIP/2.0
联系：sip:caller@u1.example.com

到P1。P1是一个出站代理。P1不负责
domain.com，因此它在DNS中查找并将其发送到那里。它也
添加一个Record-Route报头字段值：

邀请 sip:callee@domain.com SIP/2.0
联系：sip:caller@u1.example.com
记录路由: <sip:p1.example.com;lr>

P2明白这一点。它负责domain.com，因此它运行了一个位置
服务并重写Request-URI。它还添加了Record-Route
表头字段值。没有路由表头字段，因此它解析
新的请求URI以确定请求发送的位置：

邀请 sip:callee@u2.domain.com SIP/2.0
联系：sip:caller@u1.example.com
记录路由: <sip:p2.domain.com;lr>
记录路由: <sip:p1.example.com;lr>

被调用者 在u2.domain.com获取此信息并响应w ith a 200 OK: ith a 200 OK:

SIP/2.0 200 正确
联系：sip：callee@u2.domain.com
记录路由: <sip:p2.domain.com;lr>
记录路由: <sip:p1.example.com;lr>

调用者u2还将其对话状态的远程目标URI设置为
sip:caller@u1.example.com 以及其路由设置为：

（<sip:p2.domain.com;lr>，<sip:p1.example.com;lr>）

这是由P2转发给P1到U1的正常流程。现在，U1设置其
对话状态远程目标URI为sip:callee@u2.domain.com及其
路由设置为：

(<sip:p1.example.com;lr>,<sip:p2.domain.com;lr>)

由于所有路由集元素都包含lr参数，U1
构建以下 BYE 请求：

再见 sip:被叫方@u2.domain.com SIP/2.0
路由：<sip:p1.example.com;lr>,<sip:p2.domain.com;lr>

与其他任何元素（包括代理）一样，它解决了
URI在最高路由头字段值中使用DNS来确定
发送请求的位置。这发送到P1。P1注意到它是
不负责Request-URI中指明的资源
它不会改变它。它确实看到它是第一个值。
路由报头字段，因此它移除该值，并转发
请求 P2：

再见 sip:被叫方@u2.domain.com SIP/2.0
路由：<sip:p2.domain.com;lr>

P2还注意到它不对由指示的资源负责
请求URI（它负责domain.com，不是
u2.domain.com)，所以它不会改变它。它确实看到了自己
第一个路由报头字段值，因此它将其删除并转发
基于对u2.domain.com的DNS查询的结果，以下内容：
请求URI:

再见 sip:被叫方@u2.domain.com SIP/2.0

16.12.1.2 严格路由代理遍历 y

翻译文本：
在这个场景中，通过四个代理建立了对话，每个代理
其中添加了Record-Route报头字段值。第三个代理
实现RFC 2543中指定的严格路由过程
许多正在进行中的作品。

U1->P1->P2->P3->P4->U2

到达U2的INVITE包含：

邀请 sip:callee@u2.domain.com SIP/2.0
联系：sip:caller@u1.example.com
记录路由: <sip:p4.domain.com;lr>
记录路由: <sip:p3.middle.com>
记录路由: <sip:p2.example.com;lr>
记录路由: <sip:p1.example.com;lr>

U2响应以200 OK。稍后，U2发送以下
基于第一个路由报头字段值的P4 BYE请求。

再见 sip:caller@u1.example.com SIP/2.0
路由：<sip:p4.domain.com;lr>
路由：<sip:p3.middle.com>
路由：<sip:p2.example.com;lr>
路由：<sip:p1.example.com;lr>

P4不对Request-URI中指示的资源负责
所以它就会让它保持原样。它注意到它是元素 {v*}。
首先路由报头字段值，因此移除它。然后准备
发送基于当前第一个Route头部字段值的请求
sip:p3.middle.com，但它注意到这个URI不包含
lr参数，所以在发送之前，它重新格式化请求为：

再见 sip:p3.middle.com SIP/2.0
路由：<sip:p2.example.com;lr>
路由：<sip:p1.example.com;lr>
路由：<sip:caller@u1.example.com>

P3 是一个严格路由器，因此它将以下内容转发到 P2：

再见 sip:p2.example.com;lr SIP/2.0
路由：<sip:p1.example.com;lr>
路由：<sip:caller@u1.example.com>

P2看到请求URI是它放入Record-Route的值
表头字段，因此在进一步处理之前，它重新编写请求
待定：

再见 sip:caller@u1.example.com SIP/2.0
路由：<sip:p1.example.com;lr>

P2 对 u1.example.com 不负责，因此它将请求发送到
基于路由头字段值的解析的P1。

P1在顶部路由标题字段值中注意到自身，因此它
删除它，结果为：

再见 sip:caller@u1.example.com SIP/2.0

由于P1不负责u1.example.com，且没有路由
表头字段，P1将根据{v*}将请求转发到u1.example.com
请求URI。

16.12.1.3 重写 Record-Route 头字段值

在这个场景中，U1 和 U2 位于不同的私有命名空间中，
他们通过代理P1进入对话，该代理作为网关
在命名空间之间。

U1->P1->U2

U1 发送：

邀请 sip:callee@gateway.leftprivatespace.com SIP/2.0
联系：<sip:caller@u1.leftprivatespace.com>

P1 使用 i ts 位置服务并发送以下 g 到 U2:

邀请 sip:callee@rightprivatespace.com SIP/2.0
联系：<sip:caller@u1.leftprivatespace.com>
记录路由: <sip:gateway.rightprivatespace.com;lr>

U2将此200（OK）发送回P1：

SIP/2.0 200 正确
联系：<sip：callee@u2.rightprivatespace.com>
记录路由: <sip:gateway.rightprivatespace.com;lr>

P1重新编写其Record-Route报头参数，以提供一个值，该值
U1将找到有用，并发送以下内容给U1：

SIP/2.0 200 正确
联系：<sip:callee@u2.rightprivatespace.com>
记录路由: <sip:gateway.leftprivatespace.com;lr>

稍后，U1向P1发送以下BYE请求：

再见 sip:被叫方@u2.rightprivatespace.com SIP/2.0
路由：<sip:gateway.leftprivatespace.com;lr>

哪个P1转发给U2，如下所示：

再见 sip:被叫方@u2.rightprivatespace.com SIP/2.0

17 交易

SIP是一个事务性协议：组件之间的交互采取
放置在一系列独立的消息交换中。具体来说，一个
SIP 事务由一个请求及其所有响应组成
该请求，其中包含零个或多个临时响应和
一个或多个最终响应。在交易的情况下，其中
请求是一个INVITE（称为INVITE事务），
交易仅在最终响应不是的情况下才包含ACK
一个2xx响应。如果响应是2xx，则ACK不被考虑
交易的一部分。

这个分离的原因根植于其重要性
将所有200（OK）响应发送给UAC的INVITE请求。
将它们全部交付给UAC，UAS独自承担责任

用于重新传输它们（见第13.3.1.4节），以及仅UAC
负责通过ACK（见第节）确认它们
13.2.2.4). 由于此ACK仅由UAC重传，因此它是
有效考虑了其自身的交易。

交易分为客户端和服务器端。客户端
称为客户端事务，服务器端称为服务器
事务。客户端事务发送请求，并且
服务器事务发送响应。客户端和服务器
交易是嵌入在任何数量中的逻辑函数
元素。特别是，它们存在于用户代理和有状态中
代理服务器。 考虑第4节中的示例。 在此示例中，
UAC执行客户端事务，并且其出站代理
执行服务器事务。出站代理也执行一个
客户端事务，向服务器事务发送请求
在入站代理中。该代理还执行客户端事务，
它进而将请求发送到UAS的服务器事务中。
这是在图4中显示的。

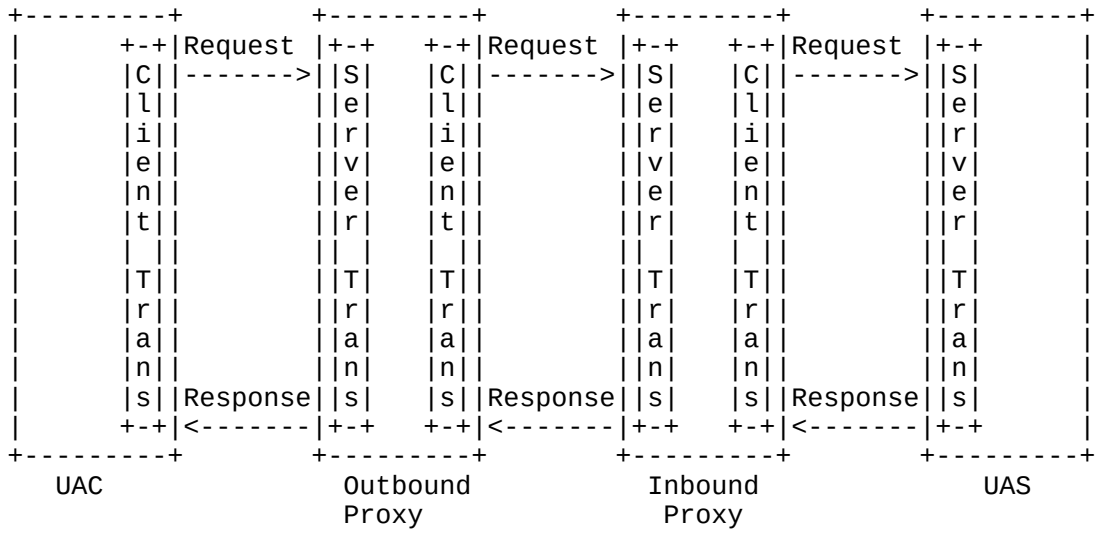

图 4：交易关系

无状态代理不包含客户端或服务器事务。
交易存在于UA或状态化代理的一侧，
并且另一侧的UA或状态化代理。至于SIP
关于事务，无状态代理实际上是
透明。客户端交易的目的在于接收一个
请求来自客户端嵌入的元素（称为
元素 "交易用户" 或 TU；它可以是 UA 或有状态的
代理），并可靠地将请求交付给服务器事务。

客户端事务也负责接收响应
将它们交付给TU，过滤掉任何响应
重传或禁止的响应（例如对ACK的响应）。
此外，在INVITE请求的情况下，客户端
事务负责生成任何的ACK请求
最终响应接受2xx响应。

同样，服务器事务的目的是接收
请求来自传输层并将它们交付给TU。
服务器事务过滤掉来自的任何请求重传
网络。服务器事务接受来自TU的响应。
将它们传输到传输层进行传输
网络。在INVITE事务的情况下，它吸收ACK
请求任何最终响应，除了2xx响应。

2xx 响应及其确认接收（ACK）受到特殊处理。
响应仅由UAS重传，其ACK也仅由UAS生成
由UAC执行。这种端到端处理是必需的，以便呼叫者
了解已接受通话的整个用户集。因为
对此特殊处理，2xx响应的重传
由UA核心处理，而不是事务层。同样，
生成2xx的ACK由UA核心处理。每个
代理沿着路径仅将每个2xx响应转发到INVITE
其对应的ACK。

17.1 客户交易

客户端事务通过以下方式提供其功能：
维护状态机。

The TU communicates with the client transaction through a simple 与客户端事务通过简单的方式通信
接口。当TU希望发起一笔新交易时，它
创建一个客户端事务并将发送的SIP请求传递给它
并且一个IP地址、端口号以及要发送到的传输方式。
客户端事务开始执行其状态机。有效
响应从客户端事务传递到TU。

有两种客户端事务状态机，取决于
关于TU传递的请求方法。 一种处理客户端
交易记录用于INVITE请求。此类机器被指
作为INVITE客户端事务的到。另一种类型处理客户端
所有请求（除INVITE和ACK外）的交易。这是
称为非-INVITE客户端事务。没有客户端
交易用于ACK。如果TU希望发送ACK，它传递一个
直接传输到传输层。

邀请事务与其他方法不同
由于其持续时间延长。通常，需要人工输入
为了响应一个INVITE。预期的长时间延迟
发送响应，论证三次握手。 在另一方面
手，其他方法的请求预期会迅速完成。
由于非-INVITE事务依赖于双向
握手，TUs 应立即响应非-INVITE 请求。

17.1.1 INVITE 客户端事务

17.1.1.1 INVITE 事务概述

The INVITE transaction consists of a three-way handshake. The client 翻译文本：INVITE 事务由三次握手组成。客户端
事务发送一个INVITE，服务器事务发送响应，
并且客户端事务发送一个ACK。对于不可靠的传输
(例如UDP)，客户端事务在
时间间隔从T1秒开始，之后每过一段时间就翻倍
重传。T1是往返时间（RTT）的估计，
默认为500毫秒。几乎所有的事务计时器
在此处描述的与T1成比例，改变T1会调整它们的值。
请求不会在可靠传输中重新传输。之后
接收1xx响应，任何重传都将完全停止，并且
客户端等待进一步响应。服务器事务可以
发送额外的1xx响应，这些响应不能可靠地传输
服务器事务。最终，服务器事务决定
发送最终响应。对于不可靠的传输，该响应
定期重传，并且对于可靠的传输，它是
发送一次。对于每个在客户端收到的最终响应
交易，客户端交易发送一个ACK，其目的是
这是为了抑制响应的重传。

17.1.1.2 正式描述

状态机用于INVITE客户端事务的示例如下
图5. 初始状态，“ 呼叫 ”，必须输入当TU
启动一个新的客户端事务，使用INVITE请求。
客户端事务必须将请求传递给传输层，以便
传输（见第18节）。如果正在使用不可靠的传输方式
已使用，客户端事务必须启动定时器A，其值为T1。
如果正在使用可靠的传输，客户端事务应
NOT 启动定时器 A（定时器 A 控制请求重传）。对于
任何运输，客户端事务必须以一个值启动计时器B
64*T1 秒（计时器B控制事务超时）。

当定时器A触发时，客户端事务必须重新传输{v*}。
通过将其传递到传输层进行请求，并且必须重置
计时器值为2*T1。重传的正式定义

在事务层上下文中是获取消息
之前发送到传输层并将其传递给传输
层再次。

当定时器A在2*T1秒后触发时，请求必须
重传一次（假设客户端事务仍然在此
状态）。此过程必须继续，以便请求是
重新传输，每次传输后间隔翻倍。
这些重传应在客户端
事务处于“ 调用 ”状态。

T1的默认值为500毫秒。T1是RTT的估计值
在客户端和服务器事务之间。元素可能（尽管它
不建议在封闭、私有的环境中使用较小的T1值（）
网络不允许通用互联网连接。T1 可能是
选择更大的，如果事先知道则建议这样做
（例如在高延迟访问链路上）RTT 较大。
无论T1的值是多少，重传的指数退避
本节中描述的必须使用。

如果客户端事务仍然处于“ 调用 ”状态，当计时器
B 发火，客户端事务应通知 TU 发生超时
已发生。客户端事务不得生成ACK。
64*T1的值等于发送七个所需的时间
请求在不可靠的传输情况下。

如果客户端事务在 {v*} 时收到一个临时响应
"调用"状态，它转换到"进行中"状态。在
进行中状态，客户端事务不应重新传输
请求任何更长时间。此外，临时响应必须
传递给TU。任何进一步的临时响应**必须**传递
在“ 进行中 ”状态时，直到达到TU。

当处于“ 呼叫 ”或“ 进行 ”状态时，接收到的
响应状态码为300-699的请求必须导致客户端
事务过渡到“ 已完成 ”。客户端事务
必须将接收到的响应传递到TU，并且客户端
交易必须生成一个ACK请求，即使传输是
可靠的（从响应中构建ACK的指南是）
在17.1.1.3节中给出，然后将ACK传递给传输
层用于传输。ACK必须发送到相同的地址，
端口，以及原始请求发送到的传输方式。
客户端事务进入时应启动定时器D
"完成"状态，其值至少为32秒的不稳定
运输，以及可靠运输的零秒值。
定时器D反映了服务器事务可以占用的时间
当使用不可靠的运输工具时保持“ 已完成 ”状态。
这等于INVITE服务器事务中的计时器H，其

默认为 64*T1。然而，客户端事务并不知道
服务器事务中使用的T1的值，因此是一个绝对最小值
32s 被用于代替以 T1 为基础的定时器 D。

任何在{v*}期间接收到的最终响应的重传
"完成"状态必须导致ACK重新传递到
传输层用于重传，但新接收到的响应
不得向上传递给TU。 响应的重传是
定义为任何与同一客户端交易匹配的响应
基于第17.1.3节的规定。

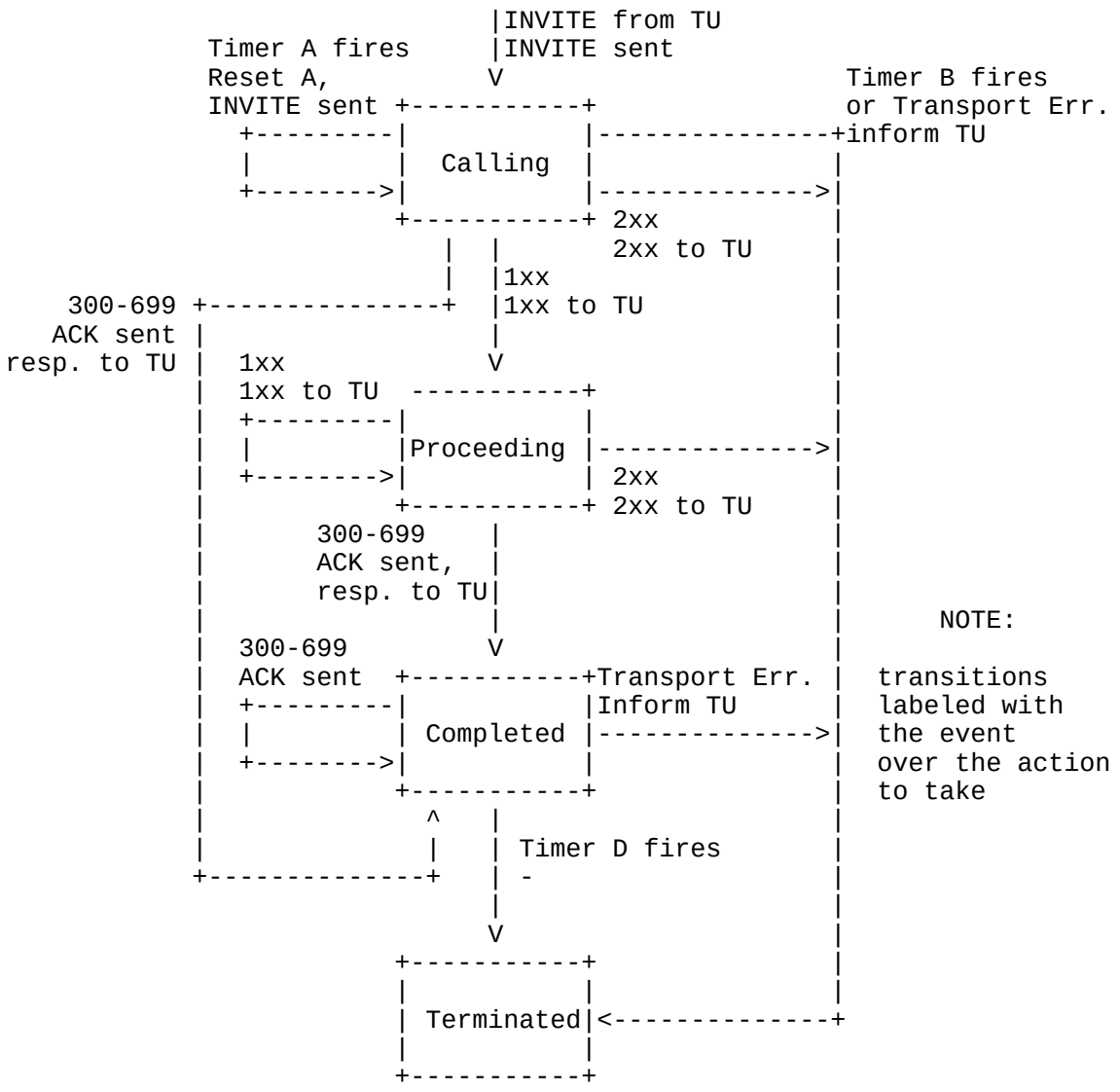

图 5：INVITE 客户端事务

如果计时器D在客户端事务处于“ 完成 ”状态时触发
状态，客户端事务必须移动到终止状态。

当处于“ 呼叫 ”或“ 进行 ”状态时，接收到的
2xx 响应必须使客户端事务进入
"已终止"状态，并且响应必须传递给TU。
此响应的处理取决于TU是否为代理

核心或UAC核心。UAC核心将处理ACK的生成
此响应，尽管代理核心始终转发200（OK）
上游。代理和UAC对200（OK）的不同处理
这是它未被处理的原因
交易层

客户端事务必须在进入时立即销毁。
"已终止"状态。这实际上是为了保证正确
操作。原因是，对INVITE的2xx响应被处理
不同；每个都通过代理转发，并且处理{v*}
在一个UAC中不同。因此，每个2xx都需要传递给代理
核心（以便转发）以及到UAC核心（以便
确认）。没有进行事务层处理。
每当传输接收到响应时，如果传输
层未找到匹配的客户端事务（使用以下规则：）
第17.1.3节），响应直接传递到核心。
匹配客户端事务被第一个2xx所销毁。
随后的2xx将找不到匹配项，因此将被传递到
核心。

17.1.1.3 ACK请求的构建

本节指定了在内部发送的ACK请求的构建。
客户端事务。一个生成2xx响应的ACK的UAC核心。
必须遵循第13节中描述的规则。

客户端事务构建的ACK请求必须包含
值，对于Call-ID、From和Request-URI相等
请求传递给传输的这些标题字段值
通过客户端事务（称之为“ 原始请求 ”）。 The To
标题字段在ACK中必须等于To标题字段在
响应被确认，因此通常会有所不同
原始请求中的“ TO ”报头字段通过添加
标签参数。ACK必须包含一个单独的Via头字段，并且
这必须等于原始的顶部 Via 头字段
请求。ACK中的CSeq报头字段必须包含相同的{v*}。
原始请求中存在的序列号值
但是方法参数必须等于 "ACK"。

如果正在确认响应的INVITE请求具有路由
表头字段，这些表头字段必须在ACK中出现。这是
为确保ACK可以通过任何下游正确路由
无状态代理。

尽管任何请求都可能包含一个主体，ACK中的主体是特殊的
由于如果内容不理解，则请求不能被拒绝。
因此，对于非2xx响应，在ACK中放置实体的位置**不推荐**。
但是，如果这样做，车身类型将限制为任何之前出现过的类型
the INVITE，假设对INVITE的响应不是415。如果
它曾是，ACK中的主体可以是Accept中列出的任何类型
415中的标题字段

例如，考虑以下请求：

邀请 sip:bob@biloxi.com SIP/2.0
通过：SIP/2.0/UDP pc33.atlanta.com;分支=z9hG4bKkjshdyff
收件人：Bob sip:bob@biloxi.com>
从：Alice <sip:alice@atlanta.com>;tag=88sja8x
最大转发数：70
呼叫标识符：987asjd97y7atg
CSeq: 986759 邀请

对于此请求的非2xx最终响应的ACK请求将
看起来像这样：

ACK sip:bob@biloxi.com SIP/2.0 Translated Text: ACK sip:bob@biloxi.com SIP/2.0
通过：SIP/2.0/UDP pc33.atlanta.com;分支=z9hG4bKkjshdyff
收件人：Bob sip:bob@biloxi.com>;tag=99sa0xk
从：Alice <sip:alice@atlanta.com>;tag=88sja8x
最大转发数：70
呼叫标识符：987asjd97y7atg
CSeq: 986759 ACK

17.1.2 非INVITE客户端事务

17.1.2.1 非INVITE事务概述

非-INVITE 事务不使用 ACK。它们很简单
请求-响应交互。对于不可靠的传输，请求
在起始时间为T1的间隔内重新传输，该间隔翻倍直到
它击中T2。如果收到临时响应，则重新传输
继续对不可靠的运输进行，但以T2为间隔。
服务器重新传输它发送的最后一个响应，这可以
是一个临时或最终响应，只有当重传的
请求已接收。这就是为什么需要请求重传的原因。
继续即使在临时响应之后；他们要确保
可靠的最终响应交付。

与INVITE事务不同，非INVITE事务没有特殊
处理2xx响应。结果是只有单个2xx
对非-INVITE的响应永远不会发送给UAC。

17.1.2.2 正式描述

非-INVITE 客户端事务的状态机显示在
图6. 它与INVITE的状态机非常相似。

当TU启动一个新客户端时进入“ 尝试 ”状态
交易带有请求。当进入此状态时，客户端
事务应设置定时器F在64*T1秒后触发。
必须传递给传输层进行传输。如果
不可靠的运输正在使用中，客户端事务必须设置定时器
E在T1秒内触发。如果计时器E在此状态下仍然触发，
定时器被重置，但这次值为 MIN(2*T1, T2)。
当计时器再次触发时，它重置为 MIN(4*T1, T2)。
过程继续，以便重传以指数方式发生
增加的间隔，上限为T2。T2的默认值为4秒，
并且它表示非-INVITE服务器事务的时间量
将需要时间来响应请求，如果它没有响应
立即。对于T1和T2的默认值，这导致
500 ms，1 s，2 s，4 s，4 s，4 s，等等

如果计时器F在客户端事务仍在进行时触发
"尝试"状态，客户端事务应通知TU关于
超时后，它应进入“ 已终止 ”状态。如果一条
临时响应在“ 尝试 ”状态下收到时，
响应必须传递给TU，然后是客户端事务
应移动到“ 进行中 ”状态。如果收到最终响应（状态
当处于“ 尝试 ”状态时接收到的代码（200-699）的响应
必须传递给TU，并且客户端事务必须过渡
到“ 完成 ”状态。

如果计时器E在“ 进行中 ”状态时触发，请求必须
传递到传输层进行重传，并且计时器E必须
重置为T2秒的值。如果在计时器F触发时处于
进行中状态，TU必须被告知超时情况，并且
客户端事务必须过渡到终止状态。如果有一个
最终响应（状态码200-699）在...时收到
进行中状态，响应必须传递给TU，并且
客户端事务必须过渡到“ 完成 ”状态。

一旦客户端事务进入“ 完成 ”状态，它必须设置
计时器K在T4秒后触发，用于不可靠的运输，为零
秒用于可靠的运输。存在“ 完成 ”状态以
缓冲可能收到的任何额外响应重传
(这就是为什么客户端事务只保留在那里，只为了

不可靠的运输）。T4表示网络的时间量
将用于清除客户端和服务器事务之间的消息。
T4的默认值为5秒。响应在以下情况下被视为重传：
它与相同的交易匹配，使用第X节中指定的规则
17.1.3. 如果在当前状态下定时器K触发，则客户端事务
必须过渡到“ 已终止 ”状态。

一旦交易处于终止状态，它**必须**被销毁
立即。

17.1.3 客户交易匹配响应

当客户端的传输层接收到响应时，它必须
确定哪个客户端事务将处理响应，以便
处理第17.1.1节和17.1.2节可以发生。
分支参数在顶部的Via报头字段中用于此
目的。一个响应在以下两种情况下与客户端事务匹配：
条件：

1. 如果响应中的分支参数值相同
顶部的Via头字段作为分支参数在顶部
通过创建事务的请求的报头字段。

2. 如果CSeq报头字段中的方法参数与 {v*}相匹配
请求创建事务的方法。
方法是必需的，因为一个 CANCEL 请求构成一个
不同的交易，但共享分支的相同值
参数。

如果通过组播发送请求，则可能它将
从不同的服务器生成多个响应。这些响应
所有都将具有最顶层Via中的相同分支参数，但会不同
在To标签中。根据规则收到的第一个响应
以上，将被使用，其他将被视为重传。
这不是错误；多播SIP仅提供基本的
"单跳发现式"服务，该服务仅限于处理
单个响应。 有关详细信息，请参阅第18.1.1节。

17.1.4 处理传输错误

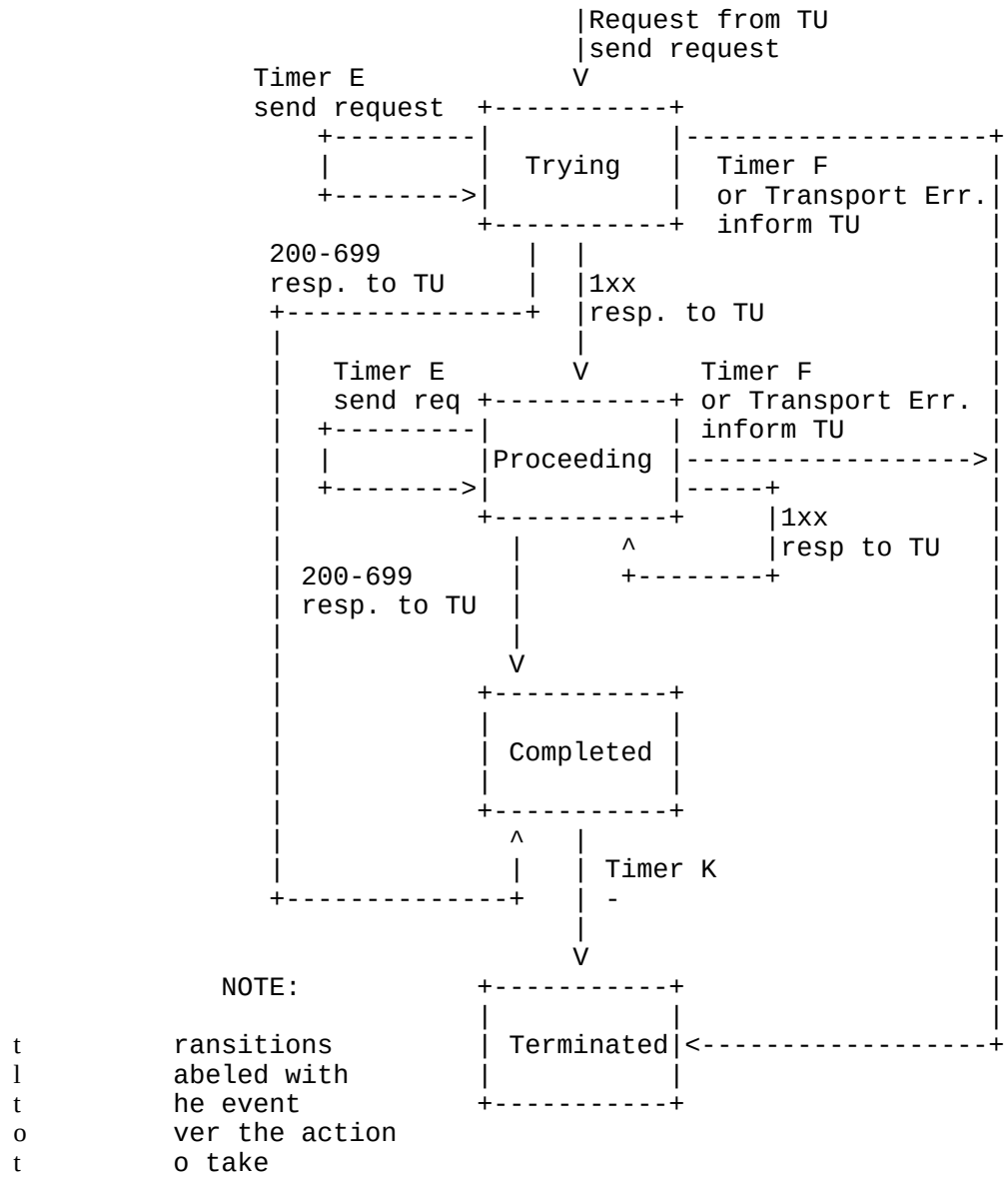

图 6：非 INVITE 客户端事务

当客户端事务向传输层发送请求时
如果需要发送，则如果传输层，则遵循以下程序
指示失败。

客户端事务应通知TU存在传输故障
已发生，并且客户端事务应直接过渡
到 "已终止" 状态。TU 将处理故障转移
机制如[4]所述。

17.2 服务器事务

服务器事务负责将请求发送到
TU 和可靠响应传输。它实现了
通过状态机实现。服务器事务由
核心当接收到请求且希望处理事务时
对于那个请求（这并不总是如此）。

与客户端事务一样，状态机取决于是否
收到的请求是一个INVITE请求。

17.2.1 INVITE 服务器事务

状态图显示INVITE服务器事务的如下
图7.

当为请求构造服务器事务时，它进入
进行中状态。服务器事务必须生成一个100
(尝试)响应，除非它知道TU将生成一个
200毫秒内提供临时或最终响应，在这种情况下，它可能
生成一个100（尝试）响应。此临时响应是
需要快速抑制重传请求以避免
网络拥塞。100（尝试）响应是构建的
根据第8.2.6节中的程序，除了{v*}之外，
在响应的 “ 收件人 ” 字段中插入标签（当没有时）
在请求中存在（）的情况从MAY降级为SHOULD NOT。
请求必须传递给TU。

TU 向服务器传递任意数量的临时响应
交易。只要服务器交易处于
进行中状态，这些都必须传递给传输
层用于传输。它们不能可靠地通过
事务层（它们不会通过它重传）并且不会引起
服务器事务状态的变化。如果请求
重传在 “ 进行中 ” 状态时接收，最
最近收到的来自TU的临时响应必须
传递给传输层进行重传。一个请求是
重传，如果它根据相同的基于服务器的交易匹配
第17.2.3节规则。

如果处于 “ 进行中 ” 状态时，TU 发送一个 2xx 响应到
服务器事务，服务器事务必须通过此
响应传输层的传输。它不是

服务器事务重传；2xx的重传
响应由TU处理。服务器事务必须随后
过渡到“ 已终止 ”状态。

当处于“ 进行中 ”状态时，如果TU通过一个响应{v*}
状态码从300到699至服务器事务，响应
必须传递给传输层进行传输，并且状态
机器必须进入“ 完成 ”状态。对于不可靠的传输，
计时器 G 设置在 T1 秒后触发，并且未设置在触发
可靠的运输。

这是对RFC 2543的变更，其中响应始终
重传，即使在可靠的传输中。

当进入“ 完成 ”状态时，计时器H必须设置为触发 {v*}
64*T1秒适用于所有运输。计时器H确定服务器
事务放弃重新发送响应。其值是
选择等于计时器B，客户端事务将持续的时间
继续重试发送请求。如果计时器 G 触发，则响应
被再次传递到传输层进行重传，并且
计时器 G 设置在 MIN(2*T1, T2) 秒后触发。从那时起，当
计时器 G 触发，响应再次传递给传输
传输，并且定时器G以加倍值重置，除非
该值超过T2，此时将其重置为{v*}的值
T2. 这与请求的重传行为相同
"尝试"状态的非-INVITE客户端事务。此外，
当处于“ 完成 ”状态时，如果请求重传
已接收，服务器应将响应传递给传输
重传

如果服务器事务期间收到一个ACK
"完成"状态，服务器事务必须过渡到
"已确认"状态。由于在此状态下忽略计时器G，因此任何
重传响应将停止。

如果计时器H在“ 完成 ”状态下触发，则表示{v*}
ACK从未收到。在这种情况下，服务器事务必须
过渡到“ 终止 ”状态，并且必须通知TU
交易失败已发生。

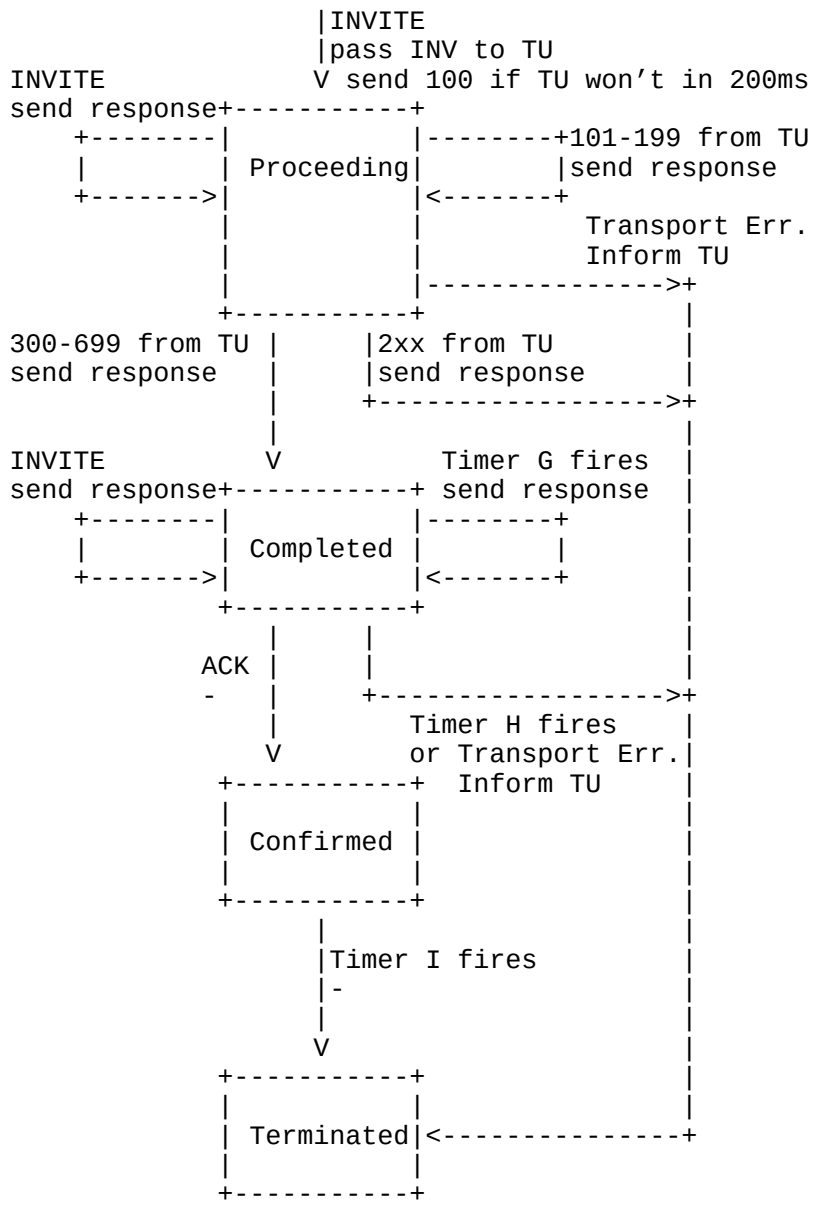

图 7：INVITE 服务器事务

"已确认"状态的目的在于吸收任何额外的ACK
消息到达，由最终重传触发
响应。当进入此状态时，定时器I被设置为在T4时触发
秒数对于不可靠的运输，以及零秒数对于可靠的
交通. 一旦定时器I触发，服务器必须过渡到
"已终止"状态。

一旦交易处于“ 已终止 ”状态，它必须
立即销毁。与客户端交易一样，这是必需的
确保对INVITE的2xx响应的可靠性。

17.2.2 非INVITE服务器事务

非-INVITE 服务器事务的状态机如图所示
图8.

状态机在“ 尝试 ”状态下初始化，并传递
一个初始化时非INVITE或ACK的请求。此请求是
提交给TU。一旦进入“ 尝试 ”状态，任何进一步的请求
重传被丢弃。如果请求是重传，则
与同一服务器事务匹配，使用在以下规则中指定的规则
第17.2.3节。

当处于“ 尝试 ”状态时，如果TU通过一个临时响应
服务器事务，服务器事务必须进入
进行中状态。响应必须传递给传输
层用于传输。任何进一步的临时响应
从“ 进行中 ”状态接收的TU必须传递
传输到传输层。如果发生重传，{v*}
请求在“ 进行中 ”状态时收到
最近发送的临时响应必须传递给传输
层用于重传。如果TU通过最终响应（状态
代码200-699)在“ 进行中 ”状态下发送到服务器时，
交易必须进入“ 完成 ”状态，并且响应必须
传递到传输层进行传输。

当服务器事务进入“ 完成 ”状态时，它必须设置
定时器J在64*T1秒后触发，用于不可靠的运输，为零
秒可靠传输。当处于“ 完成 ”状态时，
服务器事务必须将最终响应传递给传输
层，当请求需要重传时进行重传
已收到。任何其他由TU传递给服务器的最终响应
事务在“ 完成 ”状态下必须被丢弃。
服务器事务保持在此状态，直到计时器J触发
哪个点它**必须**过渡到“ 已终止 ”状态。

服务器事务必须在进入时立即被销毁。
"已终止"状态。

17.2.3 将匹配请求与服务器事务对应

当服务器从网络接收到请求时，它必须
与现有交易相匹配。这是在
以下方式。

请求中最顶部的Via头部字段中的分支参数
被检验。如果它存在并且以魔法cookie开头
"z9hG4bK"，该请求由客户端事务生成
符合本规范。因此，分支参数
将在该客户端发送的所有交易中都是唯一的。
请求与交易匹配，如果：

1. 请求中的分支参数等于其中的一个
顶 通过创建请求的头部字段来
交易，并且

2. 请求中顶部的 Via 的发送方值等于
一个在创建事务的请求中，并且

3. 请求的方法与创建该方法的那个相匹配
事务，除了ACK，请求的方法
创建事务的是 INVITE。

此匹配规则适用于INVITE和非INVITE事务
相似。

发送值用作匹配过程的一部分，因为
可能存在分支的意外或恶意复制
}来自不同客户端的参数。

如果顶Via头字段中的分支参数不存在，
如果没有包含魔法饼干，则以下步骤是
已使用。这些存在是为了处理与RFC 2543的向后兼容性
合规实现。

The INVITE request matches a transaction if the Request-URI, To tag,
从标签、Call-ID、CSeq 和顶级 Via 头字段匹配那些
邀请请求创建了事务。在此情况下，
邀请是对原始创建者的重传
交易。ACK请求与交易匹配，如果请求-
URI、来源标签、调用ID、CSeq编号（非方法）、以及顶级Via
表头字段与创建该INVITE请求的表头字段匹配
交易，并且ACK的To标签与的To标签匹配
服务器事务发送的响应。匹配基于
为每个这些报头字段定义的匹配规则。
在ACK匹配中包含标签的To头部字段
处理有助于区分2xx响应的ACK和其他响应的ACK

在代理服务器上，该代理可能已转发两个响应（这种情况可能发生
在异常条件下。具体来说，当一个代理分叉了一个请求时，
然后崩溃，响应可能被发送到另一个代理，
可能最终会将多个响应向上游转发）。一个ACK
请求与之前ACK匹配的INVITE事务匹配
被视为之前ACK的重传。

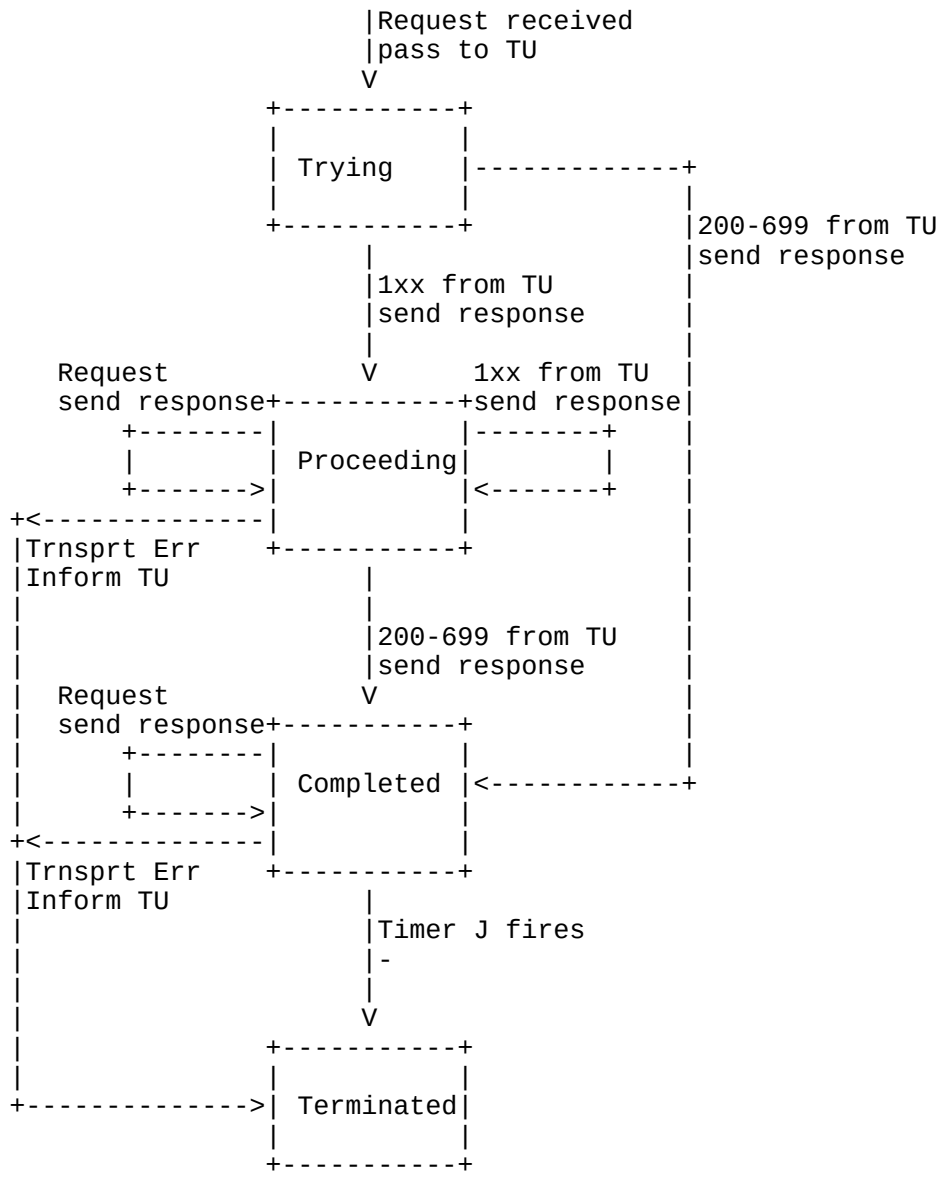

图8：非-INVITE服务器事务

对于所有其他请求方法，一个请求与一个事务相匹配
如果请求URI、To标签、From标签、Call-ID、CSeq（包括{v*}）
方法），并且顶级 Via 标头字段与请求的相匹配
创建了交易。匹配基于匹配进行

规则定义了每个这些报头字段。当遇到非-INVITE
请求匹配现有交易，它是一个重传
创建该交易的那个请求。

因为匹配规则包括请求URI，服务器无法
匹配一个响应到一笔交易。当TU传递一个响应到
服务器事务，必须将它传递给特定的服务器
交易针对的响应。

17.2.4 处理传输错误

当服务器事务向传输层发送响应时
待发送，如果运输，则遵循以下程序
层表示失败。

首先，遵循[4]中的程序，这些程序试图传递
备份的响应。如果这些都应该失败，基于
定义失败在[4]中，服务器事务应通知
当发生故障时，TU 应该过渡到
终止状态。

18 运输

传输层负责实际的传输
网络传输中的请求和响应。这包括
确定用于请求或响应的连接
面向连接的传输情况。

传输层负责管理持久
连接用于TCP和SCTP等传输协议，或TLS之上的连接
那些，包括对传输层开放的。这包括
客户端或服务器传输打开的连接，因此
连接在客户端和服务器传输功能之间是共享的。
这些连接由从地址中形成的元组索引。
端口和连接远端传输协议。当
一层打开连接，此索引设置为
目标IP、端口和传输。当连接是
被传输层接受，此索引设置为源IP
地址、端口号和传输。注意，因为源
端口通常是短暂的，但无法确定它是否
临时或通过[4]中的程序选择，接受连接
由传输层将频繁不会被重用。结果是
那两个在“ 对等 ”关系中使用的连接的代理
面向运输通常会有两个连接在使用中，一个
对于每个方向的交易发起。

建议保持连接开启一段时间
在发送最后一条消息后的实现定义持续时间
接收通过该连接。此持续时间应至少等于
最长的时间元素需要以带来
从实例化到终止状态的交易。这是为了
使交易更有可能在同一{v*}完成
连接它们被发起的（例如，请求，）
响应，以及在INVITE的情况下，对于非2xx响应的ACK）。
这通常意味着至少64*T1（参见第17.1.1.1节）
定义 T1）。然而，在某个元素中，它可能更大。
具有使用大值定时器C的TU（第16.6节第11点），
例如。

所有SIP元素必须实现UDP和TCP。SIP元素可以选择实现。
实现其他协议。

将TCP强制应用于UA是一个从RFC来的重大变化
2543. 它源于处理更大消息的需求，
必须使用TCP，如下文所述。因此，即使一个元素
从不发送大消息，它可能会收到一条，需要被
能够处理它们。

18.1 客户

18.1.1 发送请求

客户端负责发送{v*}。
请求和接收响应。传输层的用户
通过客户端传输请求，一个IP地址，端口，
运输，以及可能的多播目标TTL。

如果请求距离路径MTU小于200字节，或者如果它更大
超过1300字节且路径MTU未知，请求必须发送
使用RFC 2914 [43]拥塞控制传输协议，此类
作为TCP。如果这导致传输协议从
一个在顶部Via中指示的，顶部Via中的值必须是
已更改。这防止了UDP上消息的碎片化。
为较大消息提供拥塞控制。然而，
实现必须能够处理最大长度的消息
数据报包大小。对于UDP，此大小为65,535字节，包括
IP和UDP头部。

消息大小和MTU之间的200字节“ 缓冲区 ”
适应了SIP响应可能大于的事实
请求。这由于Record-Route的添加而发生。
表头字段值添加到对INVITE的响应中，例如。与
额外缓冲区，响应可以比大约大170字节
请求，并且在IPv4上仍然不会被分段（大约30字节）

被IP/UDP消耗，假设没有IPSec）。当选择1300时
路径MTU未知，基于1500字节的假设
以太网MTU。

如果元素因为这些消息大小而通过TCP发送请求
约束，并且该请求本应通过
UDP，如果尝试建立连接生成一个
ICMP协议不受支持，或导致TCP重置，该元素
应重试请求，使用UDP。这只是为了提供
向后兼容与RFC 2543兼容的实现
不支持TCP。预计此行为将
已弃用在此规范的未来修订版中。

一个向多播地址发送请求的客户端必须添加
"maddr" 参数到其 Via 头字段值中包含的
目标多播地址，对于IPv4，应添加“ ttl ”
参数值为1。IPv6组播的使用未定义
在此规范中，并将是未来的主题
标准化当需要时。

这些规则导致SIP中组播的有意限制。
其主要功能是提供一种类似于“ 单跳发现 ”的
服务，向一组同构服务器发送请求，
在它只需要处理来自任何一个的响应时
他们。此功能对注册最有用。事实上，
基于第17.1.3节中的交易处理规则，
客户端事务将接受第一个响应，并查看任何
其他人被视为重传，因为它们都包含相同的 Via
分支标识符。

在发送请求之前，客户端传输必须插入一个{v*}的值。
将 "sent-by" 字段转换为 Via 报头字段。此字段包含
一个IP地址或主机名，以及端口号。完全限定域名（FQDN）的使用
推荐。此字段用于在特定情况下发送响应
条件，如下所述。如果端口不存在，则使用默认
值的取决于传输方式。对于UDP、TCP和SCTP，它是5060。
5061 用于 TLS。

对于可靠的传输，响应通常在{v*}上发送。
连接请求接收的连接。因此，客户端
运输必须准备在同一 {v*} 上接收响应
连接用于发送请求。在错误条件下，
服务器可能尝试打开一个新的连接以发送响应。
处理此情况时，传输层也必须做好以下准备
接收来自源IP地址的传入连接
请求已发送，并在“ sent-by ”字段中包含端口号。它还

必须准备好在任何地址上接收传入的连接
端口，服务器将根据程序选择该端口
在[4]的第5节中描述。

对于不可靠的单播传输，客户端传输必须
准备接收来自源IP地址的响应
请求已发送（因为响应会发送回源地址）
并且 "sent-by" 字段中的端口号。此外，与
可靠的运输，在某些情况下，将发送响应
其他地方。客户端必须准备接收任何响应。
地址和端口号，服务器将根据这些信息进行选择
5节中描述的程序。

对于多播，客户端传输必须准备好接收
对同一组播组和端口的请求的响应
被发送（即，它需要是它所属的多播组的成员）
已发送请求到。）

如果请求指向一个IP地址、端口和传输到
当现有连接打开时，建议这样做
连接可用于发送请求，但可以使用另一个连接
已打开并使用。

如果使用组播发送请求，它将被发送到该组
地址、端口号和TTL由传输用户提供。如果请求
使用单播不可靠传输发送，它被发送到IP
地址和端口号由传输用户提供。

18.1.2 接收响应

当收到响应时，客户端传输检查顶部
通过标题字段值。如果 "sent-by" 参数的值为
该报头字段值与以下值不对应：{v*}
客户端传输配置为插入请求中，响应
必须静默丢弃。

如果存在任何客户交易，则客户
运输使用第17.1.3节的匹配程序来尝试
匹配响应到现有交易。如果存在
匹配，响应必须传递给该事务。否则，
响应必须传递给核心（无论它是无状态的
代理、有状态的代理或UA）进行进一步处理。处理
这些“ 散乱 ”的回复依赖于核心（一个代理将）
将它们转发，而UA将丢弃，例如）。

18.2 服务器

18.2.1 接收请求

服务器应准备好接收任何IP地址上的请求，
端口和运输组合，可能是DNS查找的结果
在SIP或SIPS URI [4] 上，该URI是为了以下目的而分发
与该服务器通信。在此上下文中，“ 分发 ”
包括在REGISTER中在Contact头字段放置一个URI
请求或重定向响应，或在Record-Route报头字段中
一个请求或响应。一个URI也可以通过放置它来“ 分发 ”
在网页或名片上。同时，也建议服务器
监听默认SIP端口上的请求（TCP和UDP端口5060）
5061 用于TCP上的TLS（所有公共接口）。典型
异常情况将是私有网络，或者当有多个服务器
实例运行在同一个主机上。对于任何端口和接口
服务器监听UDP时，它必须监听相同的端口
并且为TCP提供接口。这是因为可能需要发送消息
使用TCP，而不是UDP，如果它太大。因此，
对换不成立。服务器不必监听UDP。
特定地址和端口，因为它正在监听那个相同的
地址和端口号用于TCP。当然，也可能有其他原因导致
服务器需要在特定的地址和端口上监听UDP。

当服务器传输接收到任何传输上的请求时，它
必须检查顶部Via中的“ sent-by ”参数的值
表头字段值。如果“ sent-by ”参数的主机部分
包含域名，或者如果它包含一个不同的IP地址
从数据包源地址，服务器必须添加一个“ 接收 ”
参数到该 Via 头字段值的。此参数必须
包含接收数据包的源地址。
用于协助服务器传输层发送响应，
由于必须从请求的源IP地址发送
来。

考虑服务器传输接收到的请求，其看起来像，
在部分：

邀请 sip:bob@Biloxi.com SIP/2.0
通过：SIP/2.0/UDP bobspc.biloxi.com:5060

请求接收到的源IP地址为192.0.2.4。
在向上传递请求之前，传输添加一个 "已接收"
参数，因此请求看起来部分如下：

邀请 sip:bob@Biloxi.com SIP/2.0
通过：SIP/2.0/UDP bobspc.biloxi.com:5060；接收=192.0.2.4

接下来，服务器传输尝试将请求与服务器匹配
交易。它使用在以下描述的匹配规则中这样做。
第17.2.3节。如果找到一个匹配的服务器事务，则
请求传递给该事务进行处理。如果没有匹配
找到，请求传递到核心，核心可能会决定
为该请求构建一个新的服务器事务。注意，当
一个UAS核心向INVITE发送2xx响应，服务器事务是
已摧毁。这意味着当确认（ACK）到达时，将不会有
匹配服务器事务，并且根据此规则，ACK是
传递到UAS核心，进行处理。

18.2.2 发送响应

服务器传输使用顶部的 Via 标头字段的值
为了确定发送响应的位置。它必须遵循
以下过程：

如果 "sent-protocol" 是一种可靠的传输协议，例如
TCP 或 SCTP，或者在这些之上的 TLS，响应必须使用 {v*} 发送
现有的原始请求源连接
创建该交易的，如果该连接仍然开启。
这需要服务器传输来维护一个关联
在服务器事务和传输连接之间。如果那样
连接已不再打开，服务器应打开
连接到“ 接收 ”参数中的IP地址，如果
当前，使用“ sent-by ”值中的端口或默认端口
端口号，如果没有指定端口号。
连接尝试失败，服务器应使用以下程序
在[4]中，为了确定服务器的IP地址和
打开连接并发送响应的端口。

否则，如果 Via 头字段值包含一个 "maddr"
参数，响应必须转发到列出的地址
那里，使用“ sent-by ”中指示的端口，如果没有指定则使用端口5060。
无存在。如果地址是多播地址，则
响应应使用“ ttl ”中指示的TTL发送
参数，或者如果不存在该参数，则具有TTL为1。

否则（对于不可靠的单播传输），如果顶部的 Via
具有 "接收" 参数，响应必须发送到
地址在“ 接收 ”参数中，使用指示的端口
在 "sent-by" 值中，或如果没有指定则使用端口 5060
明确地。如果这失败，例如，引发ICMP "端口
无法访问的"响应，[4]中第5节的程序
应用于确定响应发送位置。

否则，如果它未标记为接收器，则响应必须
发送到由 "sent-by" 值指示的地址，使用 {v*}
流程见[4]的第5节。

18.3 框架

在消息导向的传输（如UDP）的情况下，如果
消息包含一个Content-Length头部字段，消息体是
假定包含那么多字节。如果还有额外的字节在
传输数据包超出主体末尾，它们必须
废弃。如果传输数据包在结束前结束，
消息正文，这被视为错误。如果消息是
响应，它必须被丢弃。如果消息是一个请求，则
元素应生成一个400（错误请求）响应。如果消息
没有 Content-Length 头字段，假定消息体为
在传输数据包的末尾结束。

在流导向传输的情况下，例如TCP，内容-
长度报头字段表示主体的大小。
长度报头字段必须与面向流的传输一起使用。

18.4 错误处理

错误处理与消息是否为请求无关
响应。

如果传输用户请求通过 {v*} 发送消息
不可靠的传输，结果是ICMP错误，行为
依赖于ICMP错误的类型。主机、网络、端口或协议
无法访问的错误，或参数问题错误 应该导致
传输层通知传输用户发送失败。
源封禁和TTL超时ICMP错误应忽略。

如果传输用户请求通过可靠的
运输，结果为连接失败，运输
层应通知传输用户发送失败。

19 常见消息组件

SIP消息中存在某些组件，它们在各种
SIP消息中的位置（有时甚至在其外部）
优点需单独讨论。

19.1 SIP和SIPS统一资源指示符

一个SIP或SIPS URI标识一个通信资源。 像所有
统一资源标识符（URI）、SIP 和 SIPS URI 可以放置在网页、电子邮件消息中,
或印刷文献。它们包含足够的信息以
启动并维护与资源的通信会话。

通信资源的示例包括以下内容：

一个在线服务的用户

o 一款多行电话的显示

一个消息系统上的邮箱

o 网关服务中的一个PSTN号码

一个组织中的组（例如“ 销售 ”或“ 帮助台 ”）

一个SIPS URI指定资源需要通过安全方式接触。
表示，特别是，在UAC之间应使用TLS。
域名拥有URI。从那里，使用安全通信
达到用户，其中具体的安全机制取决于
域名策略。任何由SIP URI描述的资源都可以
升级为SIPS URI，只需更改方案即可，如果它是
希望与该资源安全通信。

19.1.1 SIP 和 SIPS URI 组件

" sip: " 和 " sips: " 方案遵循 RFC 2396 [5] 中的指南。
他们使用一种类似于mailto URL的形式，允许指定
SIP请求头字段和SIP消息体。这使得
可以指定会话的主题、媒体类型或紧急程度
由网页或电子邮件消息中的URI启动。
SIP或SIPS URI的正式语法在第25节中给出。
一般形式，在SIP URI的情况下是：

sip:用户:密码@主机:端口;uri-参数?头部

SIPS URI的格式相同，只是方案不同。
"sips" 而不是 sip。这些标记以及其中的一些标记
扩展具有以下含义：

用户：主机上特定资源的标识符
已解决。在此上下文中，“ 主机 ”一词经常指代{v*}。
到一个域。一个URI的"userinfo"由这个用户组成。
字段，密码字段以及跟在其后的 @ 符号。
URI的用户信息部分是可选的，并且当{v*}时可以不存在。

目标主机没有用户概念或当{v*}
主机本身是被识别的资源。如果存在@符号
存在SIP或SIPS URI中，用户字段必须不为空。

如果被寻址的主机可以处理电话号码，则
实例，一个互联网电话网关，电话-
在RFC 2806 [9]中定义的订阅者字段可以使用
填充用户字段。对于特殊转义规则，有专门的说明。
编码SIP和SIPS URI中的电话用户字段
在第19.1.2节中描述。

密码：与用户关联的密码。 当前的SIP和
SIPS URI 语法允许此字段存在，其使用是NOT
推荐，因为认证信息的传递
在明文（如URI）中已被证明是一个安全风险
在几乎所有使用过的情况下。例如，
在字段中传输PIN号码会暴露PIN。

请注意，密码字段只是用户的一个扩展
部分。不希望给予特殊
对字段密码部分的**重要性**可能仅仅是
将 "user:password" 作为单个字符串处理。

主机：提供SIP资源的计算机。主机部分包含
或者是一个完全限定的域名或数字IPv4或IPv6
地址。使用完全限定的域名形式是
推荐，尽可能时使用。

端口：请求要发送的端口号。

URI参数：影响从{v*}构建的请求的参数
URI。

URI参数添加在hostport组件之后，并且是
. 以分号分隔。

URI参数的形式为：

参数名称 "=" 参数值

尽管可能有任意数量的URI参数
包含在URI中，任何给定的参数名不得出现
一次以上。

这个可扩展机制包括传输、maddr、ttl、
用户、方法和lr参数。

运输参数确定运输机制,
用于发送SIP消息，如[4]中指定。SIP可以
使用任何网络传输协议。参数名称是
定义了UDP（RFC 768 [14]）、TCP（RFC 761 [15]）和SCTP
(RFC 2960 [16]). 对于SIPS URI，传输参数必须
指示一个可靠的运输方式。

The maddr 参数指示要使用的服务器地址
联系此用户，覆盖任何从地址派生的内容
主机字段。当存在maddr参数时，端口号
URI的传输组件适用于地址
指示在maddr参数值中。[4]描述了
正确的解释 transport、maddr 和 hostport 在
为了获取目标地址、端口和传输
用于发送请求。

maddr字段已被用作一种简单的松散源形式
路由。它允许URI指定必须使用的代理
已穿越到目的地的途中。继续使用
maddr 参数以此方式强烈不建议（的）
机制使其得以实现的已弃用)。实现
应改用本文中描述的Route机制
文档，如必要则建立预存的路线集
(参见第8.1.1.1节)。这提供了一个完整的URI来描述
要遍历的节点。

The ttl parameter determines the time-to-live value of the UDP
组播数据包并且仅当maddr是组播时使用
地址和传输协议是UDP。例如，要
指定一个使用多播调用alice@atlanta.com的调用
239.255.255.1 15 的 ttl，以下 URI 将会是
已使用：

sip:alice@atlanta.com;maddr=239.255.255.1;ttl=15

有效的电话用户字符串集合是以下集合的子集：
有效的用户字符串。用户URI参数存在是为了
区分电话号码和偶然出现的用户名
看起来像电话号码。如果用户字符串包含一个
电话号码格式化为电话用户，用户
参数值 "phone" 应该存在。即使没有这个
参数，SIP和SIPS URI的接收者可以解释为
电话号码的预@部分，如果当地有对电话号码的限制
命名空间允许用户名。

SIP请求从URI构建的方法可以是
指定与方法参数。

lr参数，当存在时，表示该元素
负责此资源的实现路由机制
在此文档中指定。此参数将在
URI代理放置到Record-Route报头字段值中，并且
可能出现在预存在的路由集中的 URI 中。

此参数用于实现与{v*}的向后兼容性
实现RFC 2543严格路由机制的系统
并且rfc2543bis草案至bis-05。一个准备元素
根据不包含此参数的URI发送请求
可以假设接收元素实现了严格路由
重新格式化消息以保留其中的信息在
请求URI。

由于uri参数机制是可扩展的，SIP元素
必须静默忽略它们不支持的任何uri参数
理解。

标题：请求构造中要包含的标题字段
从URI中。

SIP请求中的头部字段可以用“ ？ ”指定
机制在URI内部。头部名称和值是
编码为以&分隔的hname = hvalue对。
特殊hname "body" 表示相关的hvalue是
SIP请求的消息体。

表1总结了基于{v*}的SIP和SIPS URI组件的使用
上下文中出现的URI。外部列描述
URI出现在SIP消息之外的任何地方，例如在
网页或名片。标记为“ m ”的条目是必填的，那些
标记为 "o" 的为可选，标记为 "-" 的不允许。
元素处理URI时应当忽略任何不允许的组件，如果
它们存在。第二列指示默认值。
一个可选元素，如果不存在。"--"表示不存在。
元素要么不是可选的，要么没有默认值。

URI在Contact头部字段中的限制因不同情况而异
在标题字段出现的上下文中。一套适用于
消息建立和维护对话（INVITE及其200（OK））
响应）。另一个适用于注册和重定向
消息（注册，其200（OK）响应，以及3xx类响应）
任何方法）。

19.1.2 字符转义要求

	default	Req.-URI	To	From	reg./redir. Contact	dialog Contact/ R-R/Route	external
user	--	o	o	o	o	o	o
password	--	o	o	o	o	o	o
host	--	m	m	m	m	m	m
port	(1)	o	-	-	o	o	o
user-param	ip	o	o	o	o	o	o
method	INVITE	-	-	-	-	-	o
maddr-param	--	o	-	-	o	o	o
ttl-param	1	o	-	-	o	-	o
transp.-param	(2)	o	-	-	o	o	o
lr-param	--	o	-	-	-	o	o
other-param	--	o	o	o	o	o	o
headers	--	-	-	-	o	-	o

(1)：默认端口号取决于传输和方案。
默认为 5060，用于 sip：使用 UDP、TCP 或 SCTP。默认为
5061 用于 sip: 通过 TCP 使用 TLS 和 sips: 通过 TCP。

(2)：默认传输方式取决于方案。对于 sip:，它是 UDP。
对于{sips}，它是TCP。

表 1：SIP 头部 URI 组件的使用和默认值
字段值、请求URI和引用

SIP遵循RFC 2396 [5]的要求和指南，当
定义必须在SIP URI中转义的字符集，以及
使用其 ""%" HEX HEX"" 转义机制。来自RFC 2396 [5]：

> 任何给定URI实际保留的字符集
> 组件由该组件定义。一般来说，一个字符
> 如果URI的语义发生变化，则保留该字符
> 被替换为其转义后的US-ASCII编码 [5]。排除US-ASCII
> ASCII字符（RFC 2396 [5]），例如空格和控制字符
> 字符和用作URI分隔符的字符也必须
> 转义。URI不得包含未转义的空格和控制字符
> 字符。

对于每个组件，有效的 BNF 展开集恰好定义了
哪些字符可能以未转义的形式出现。所有其他字符必须
转义

例如，"@" 不在用户字符集中
组件，因此用户 "j@s0n" 至少必须对 @ 符号进行编码，
如 "j%40s0n"。

扩展第25节中的hname和hvalue标记显示，所有URI
保留字符在头部字段名称和值中必须被转义。

电话用户组件的子集具有特殊
转义考虑。 在以下字符集中，未保留的字符集为
RFC 2806 [9] 对电话用户描述包含多个
字符在需要转义的各个语法元素中
用于SIP URI。任何出现在电话用户中的字符
那些不出现在用户规则的BNF展开中的必须
被转义。

请注意，在宿主组件中不允许字符转义。
SIP或SIPS URI（其扩展中%字符无效）。
This is likely to change in the future as requirements for 这可能在将来发生变化，因为对 {v*} 的需求
国际化域名已确定。当前
实现不得试图通过将 {v*} 处理为来提高鲁棒性
接收到的主机组件中的转义字符被当作字面值处理
等同于它们的未转义对应项。所需的行为是
满足IDN要求可能显著不同。

19.1.3 示例 SIP 和 SIPS URI {v*}

sip:alice@atlanta.com
sip:alice:secretword@atlanta.com;transport=tcp
sips:alice@atlanta.com?主题=项目%20x&优先级=紧急
sip:+1-212-555-1212:1234@gateway.com;user=电话
sips:1212@gateway.com
sip:alice@192.0.2.4
sip:atlanta.com;方法=注册?到=alice%40atlanta.com
sip:alice;day=星期二@亚特兰大.com

The last sample URI above has a user field value of Translated Text: 上面的最后一个样本URI的用户字段值为
alice;day=星期二
分号在此字段中应不转义。 对于以下目的
此协议中，字段是透明的。该值的结构是
仅对负责资源的SIP元素有用。

19.1.4 URI 比较

一些本规范中的操作需要确定两个 {v*}
SIP或SIPS URI等价。在此规范中，注册机构
需要比较在REGISTER请求中的Contact URI中的绑定（见
第10.3节。SIP和SIPS URI进行比较以判断相等性
根据以下规则：

一个SIP和SIPS URI从不等价。

o SIP和SIPS URI的用户信息比较是大小写敏感的-
敏感。这包括包含密码的用户信息或
格式化为电话用户。所有其他用户的比较
URI组件不区分大小写，除非明确指定
定义否则。

参数和头部字段的排序是不重要的
在比较SIP和SIPS URI时。

o 除了“ 保留 ”集合中的字符（参见RFC 2396
[5]) 等价于它们的 ""%" HEX HEX" 编码。

一个DNS查询主机名的结果IP地址
不匹配该主机名。

对于两个URI相等，用户、密码、主机和端口
组件必须匹配。

A URI省略用户组件的将不会匹配一个URI，其中
包含一个。一个省略密码组件的URI将不会
匹配包含一个的URI。

一个省略了具有默认值的任何组件的URI将不会
匹配显式包含该组件的URI
默认值。例如，省略可选端口的URI
组件不会匹配显式声明端口5060的URI。
同样适用于传输参数，TTL参数，
用户参数，和方法组件。

定义 sip:user@host 不等同于
sip:user@host:5060 是对 RFC 2543 的一个变更。当推导
从URI中获取地址，预期得到等效地址
等效URI。URI sip:user@host:5060 将始终
解析到端口5060。URI sip:user@host可能解析到
其他端口通过[4]中详细描述的DNS SRV机制。

o URI uri-parameter components are compared as follows: o URI uri-参数组件的比较方式如下：

- 任何出现在两个URI中的uri参数必须匹配。

- 一个用户、ttl 或仅出现在一个中的方法 uri 参数
URI从不匹配，即使它包含默认值。

- 包含 maddr 参数的 URI 不会匹配 URI
不包含maddr参数。

- 所有仅出现在一个URI中的其他uri参数
忽略当比较URI时。

- URI头组件永远不会被忽略。任何存在的头组件必须在两个URI中都存在且匹配这两个URI匹配。匹配规则为每个报头字段定义在第20节中。

以下集合中的URI是等效的：

sip：%61lice@atlanta.com;transport=TCP
sip:alice@AtLanTa.CoM;传输=tcp

sip:carol@chicago.com
sip:carol@chicago.com;newparam=5
sip:carol@chicago.com;安全=开启

sip:biloxi.com;传输=tcp;方法=REGISTER?到=sip:bob%40biloxi.com
sip:biloxi.com;方法=注册;传输=tcp?到=sip:bob%40biloxi.com

sip:alice@atlanta.com?主题=项目%20x&优先级=紧急
sip:alice@atlanta.com?优先级=紧急&主题=项目%20x

以下每个集合中的URI均不相等：

SIP:ALICE@AtLanTa.CoM;传输=udp (不同的用户名)
sip:alice@AtLanTa.CoM;传输=UDP

sip:bob@biloxi.com (可以解析到不同的端口)
sip:bob@biloxi.com:5060

sip:bob@biloxi.com (可以解析为不同的传输方式)
sip:bob@biloxi.com;传输=udp

sip:bob@biloxi.com (可以解析为不同的端口和传输)
sip:bob@biloxi.com:6000;传输=tcp

sip:carol@chicago.com (不同的报头组件)
sip:carol@chicago.com?Subject=下次%20会议

sip:bob@phone21.boxesbybob.com (即使那是
sip:bob@192.0.2.4 phone21.boxesbybob.com 解析为)

请注意，等价性不是传递的：

- o sip:carol@chicago.com 和 sip:carol@chicago.com;security=on 是等效

- o sip:carol@chicago.com 和 sip:carol@chicago.com;安全=关闭等价

o sip:carol@chicago.com;安全=开启和
sip:carol@chicago.com;安全=关闭不等效

19.1.5 从URI形成请求

一个实现需要在直接形成请求时小心处理
从URI。来自名片、网页，甚至来自
协议内部源，如已注册联系人，可能包含
不适当的头部字段或主体部分。

一个实现必须包括提供的传输、maddr、ttl 或
用户参数在形成的请求的Request-URI中。如果URI
包含一个方法参数，其值必须用作方法的
请求。方法参数不得放置在
请求URI。未知URI参数必须放在消息中
请求URI。

一个实现应该处理任何头或体的存在
URI中的部分作为希望将它们包含在消息中的内容，以及
选择按组件逐个尊重请求。

一个实现不应该尊重这些显然危险的头部
字段：From、Call-ID、CSeq、Via 和 Record-Route。

实现不应尊重任何请求的Route头字段
值以避免无意中成为恶意行为的代理
攻击。

一个实现不应该遵守包含头字段请求的要求
可能导致它错误地宣传其位置或能力。
这些包括：Accept, Accept-Encoding, Accept-Language, Allow,
联系（在其对话框使用中），组织，支持，和用户-
代理

一个实现应该验证任何请求的准确性
描述性头部字段，包括：Content-Disposition，Content-
编码，Content-Language，Content-Length，Content-Type，日期，
Mime版本，和时间戳。

如果从给定的URI构建消息形成的请求是
无效的SIP请求，URI无效。实现必须
不进行发送请求。它应改为追求
由于在发生上下文中无效URI而采取的行动过程。

构建的请求可能以多种方式无效。
包括但不限于标题字段中的语法错误，
无效的URI参数组合，或错误
消息体描述。

发送由给定URI形成的请求可能需要能力
不可用于实现。URI可能指示使用了一个
未实现的传输或扩展，例如。一种实现
应拒绝发送这些请求，而不是修改它们
匹配其功能。实现不得发送请求
需要它不支持的功能扩展。

例如，这样的请求可以通过存在来形成
需要头部参数或方法URI参数，其中包含
未知或明确不支持的价值。

19.1.6 将SIP URI和tel URL相关联

当将tel URL（RFC 2806 [9]）转换为SIP或SIPS URI时，
整个tel URL的电话用户部分，包括任何
参数被放置在SIP或SIPS URI的用户信息部分。

因此，tel:+358-555-1234567;postd=pp22 变为

sip:+358-555-1234567;postd=pp22@foo.com;user=phone

或者
sips:+358-555-1234567;postd=pp22@foo.com;user=phone

不是
sip:+358-555-1234567@foo.com;postd=pp22;user=phone

或者

sips:+358-555-1234567@foo.com;postd=pp22;user=phone

通常，等效的 "tel" URL 转换为 SIP 或 SIPS URI 在
这种时尚可能不会产生等效的SIP或SIPS URI。
SIP和SIPS URI的用户信息是按大小写敏感进行比较的
字符串。tel URL 的不区分大小写的部分中的方差
重新排序tel URL参数不影响tel URL等价性，
但会影响由它们形成的SIP URI的等价性。

例如，

电话：+358-555-1234567；邮政编码：=pp22
电话：+358-555-1234567；POSTD=PP22

等价，而

sip:+358-555-1234567;postd=pp22@foo.com;user=phone
sip:+358-555-1234567;POSTD=PP22@foo.com;user=phone

不是。

同样地,

电话：+358-555-1234567；邮编：=pp22；订阅号：=1411
电话：+358-555-1234567；子网=1411；邮政=pp22

等价，而

sip:+358-555-1234567;postd=pp22;isub=1411@foo.com;user=phone
sip:+358-555-1234567;isub=1411;postd=pp22@foo.com;user=phone

不是。

为了减轻这个问题，构建电话用户元素
字段应折叠到SIP或SIPS URI的用户信息部分中
任何不区分大小写的电话用户部分转换为小写,
并且按参数字典顺序对电话用户参数进行排序
姓名，除了ISDN子地址和拨号后，它们先出现之外
按照该顺序。（tel URL的所有组件，除了future-）
扩展参数被定义为不区分大小写进行比较。）

根据这个建议，两者

电话：+358-555-1234567；邮政编码：=pp22
电话：+358-555-1234567；POSTD=PP22

成为

sip {v*} : +358-555-1234567;postd=pp22@foo.com;user= 电话

并且两者

电话：+358-555-1234567；时间戳=a.b；电话上下文=5
电话：+358-555-1234567；电话上下文=5；时间戳=a.b

成为

sip:+358-555-1234567;phone-context=5;tsp=a.b@foo.com;user=phone

19.2 选项标签

选项标签是用于指定新选项的唯一标识符
(扩展) 在SIP中。这些标签用于Require（第20.32节）。
代理要求（第20.29节），支持（第20.37节）和
不支持（第20.40节）的标题字段。请注意，这些选项
在那些头字段中以参数形式出现，在选项标记 = 令牌中
表（见第25节关于标记的定义）。

选项标签在标准跟踪的RFC中定义。这是一个变更
从以往实践，并设立以确保持续多
供应商互操作性（参见第20.32节和第20.32节中的讨论）
20.37). 一个用于确保易于使用的IANA选项标记注册表被使用。
参考。

19.3 标签

SIP的To和From头部字段中使用了“ tag ”参数
消息。它作为识别对话的通用机制。
这是Call-ID与两个标签的组合，其中一个来自
每个对话参与者。当UA发送请求到外部时
一个对话框，它只包含一个From标签，提供“ 一半 ”的对话框
ID. 对话由响应（s）完成，每个响应
贡献了To报头字段的下半部分。对于{v*}的分支
SIP请求意味着可以从一个
单个请求。这也解释了双边对话的必要性
标识符；如果没有接收者的贡献，
发起者无法区分已建立的多个对话
从单个请求。

当UA生成一个标签用于插入到请求中时
响应，它必须是全局唯一且加密随机
至少32位的随机性。此选择的属性
需求是UA将在From中放置不同的标签
标题比将其放入“ 收件人 ”标题中的INVITE
对同一INVITE的响应。这是为了让UA
邀请自身进入一个会话，这是调用“ 环回 ”的常见情况
在PSTN网关中。同样，针对不同通话的两个INVITE请求将
具有不同的 From 标签，并且对不同调用有两个响应
具有不同的 To 标签。

除了全球唯一性的要求之外，该算法对于 {v*} 的要求也需满足。
生成一个标签是特定实现的。标签在
容错系统，其中要在{v*}上恢复对话
故障后的备用服务器。一个UAS可以在这种情况下选择标签。
一种备份可以识别请求作为对话一部分的方式
失败的服务器，因此确定它应该尝试
恢复与该对话及其相关联的任何其他状态。

20 表头字段

The general syntax for header fields is covered in Section 7.3. This
章节列出了完整的头字段集合以及相关说明
语法、意义和用法。 在本节中，我们使用 [HX.Y]
引用当前HTTP/1.1规范RFC的X.Y节
2616 [8]。每个报头字段的示例已给出。

关于方法和代理相关的头部字段信息
处理总结在表2和表3中。

"where"列描述了请求和响应类型，其中
表头字段可以被使用。此列中的值是：

R: 标头字段仅可出现在请求中；

s: 头部字段仅可出现在响应中

r Translated Text s;

翻译文本：

2xx, 4xx等：数值或范围表示响应
代码，其中可以使用标题字段；

c: 头部字段从请求复制到响应。

一个在“ where ”列中的空条目表示该标题
字段可能存在于所有请求和响应中。

"代理"列描述代理可能执行的操作
表头字段：

a: 如果不存在，代理可以添加或连接头部字段。

m: 代理可以修改现有的报头字段值。

d: 代理可以删除一个头部字段值。

r: 代理必须能够读取头部字段，因此这
标题字段无法加密。

下六列与头部字段的存在相关
方法：

c: 条件；对头部字段的要求取决于
消息的上下文。

m: 标头字段是必需的。

m*: 标头字段应当发送，但客户端/服务器需要
准备好接收没有该标题字段的消息。

o: 头部字段是可选的。

t: 标题字段应当发送，但客户端/服务器需要
已准备好接收不带该标题字段的消息。

如果使用基于流的协议（如TCP）作为
运输后，标题字段必须发送。

*: 如果消息体不为空，则必须包含标题字段。
查看第20.14、20.15和7.4节以获取详细信息。

-: 标题字段不适用。

"可选"表示一个元素可以包含该标题字段。
请求或响应，并且UA（用户代理）可以忽略如果存在的头字段
在请求或响应中（此规则的例外是 Require）
标题字段在20.32中讨论）。一个“ 强制 ”的标题字段必须
存在于请求中，并且必须被接收请求的UAS理解
请求。一个强制性的响应头字段必须在
响应，并且头部字段必须被UAC理解
处理响应。 "不适用"表示该标题
字段必须在请求中不存在。如果放置了一个，
请求错误，它必须被接收该请求的用户代理（UAS）忽略
请求。同样，对于标记为“ 不适用 ”的标题字段
响应表示UAS不得将头部字段放置在
响应，并且UAC必须忽略响应中的头字段。

A UA 应忽略扩展头参数，这些参数不是
理解了。

一些常见报头字段名的紧凑形式也被定义了，用于
使用当整体消息大小有问题时。

联系人、来源和目的地标题字段包含一个URI。如果URI
包含逗号、问号或分号，URI 一定必须
括号内（< 和 >）。任何URI参数是
包含在这些括号内。如果URI没有被括在尖括号内。
括号，任何分号分隔的参数都是标题参数，
非URI参数。

20.1 接受

The Accept header field follows the syntax defined in [H14.1]. The 接受头字段遵循[H14.1]中定义的语法。
语义也相同，除了如果没有接受
标题字段存在时，服务器应假定一个默认值
应用/SDP.

一个空的Accept头字段表示没有可接受的格式。

示例： Translated Text：示例：

Header field	where	proxy	ACK	BYE	CAN	INV	OPT	REG
Accept	R		-	o	-	o	m*	o
Accept	2xx		-	-	-	o	m*	o
Accept	415		-	c	-	c	c	c
Accept-Encoding	R		-	o	-	o	o	o
Accept-Encoding	2xx		-	-	-	o	m*	o
Accept-Encoding	415		-	c	-	c	c	c
Accept-Language	R		-	o	-	o	o	o
Accept-Language	2xx		-	-	-	o	m*	o
Accept-Language	415		-	c	-	c	c	c
Alert-Info	R	ar	-	-	-	o	-	-
Alert-Info	180	ar	-	-	-	o	-	-
Allow	R		-	o	-	o	o	o
Allow	2xx		-	o	-	m*	m*	o
Allow	r		-	o	-	o	o	o
Allow	405		-	m	-	m	m	m
Authentication-Info	2xx		-	o	-	o	o	o
Authorization	R		o	o	o	o	o	o
Call-ID	c	r	m	m	m	m	m	m
Call-Info		ar	-	-	-	o	o	o
Contact	R		o	-	-	m	o	o
Contact	1xx		-	-	-	o	-	-
Contact	2xx		-	-	-	m	o	o
Contact	3xx	d	-	o	-	o	o	o
Contact	485		-	o	-	o	o	o
Content-Disposition			o	o	-	o	o	o
Content-Encoding			o	o	-	o	o	o
Content-Language			o	o	-	o	o	o
Content-Length		ar	t	t	t	t	t	t
Content-Type			*	*	-	*	*	*
CSeq	c	r	m	m	m	m	m	m
Date		a	o	o	o	o	o	o
Error-Info	300-699	a	-	o	o	o	o	o
Expires			-	-	-	o	-	o
From	c	r	m	m	m	m	m	m
In-Reply-To	R		-	-	-	o	-	-
Max-Forwards	R	amr	m	m	m	m	m	m
Min-Expires	423		-	-	-	-	-	m
MIME-Version			o	o	-	o	o	o
Organization		ar	-	-	-	o	o	o

表2：头部字段摘要，A--O

Header field	where	proxy	ACK	BYE	CAN	INV	OPT	REG
Priority	R	ar	-	-	-	o	-	-
Proxy-Authenticate	407	ar	-	m	-	m	m	m
Proxy-Authenticate	401	ar	-	o	o	o	o	o
Proxy-Authorization	R	dr	o	o	-	o	o	o
Proxy-Require	R	ar	-	o	-	o	o	o
Record-Route	R	ar	o	o	o	o	o	-
Record-Route	2xx,18x	mr	-	o	o	o	o	-
Reply-To			-	-	-	o	-	-
Require		ar	-	c	-	c	c	c
Retry-After	404,413,480,486		-	o	o	o	o	o
	500,503		-	o	o	o	o	o
	600,603		-	o	o	o	o	o
Route	R	adr	c	c	c	c	c	c
Server	r		-	o	o	o	o	o
Subject	R		-	-	-	o	-	-
Supported	R		-	o	o	m*	o	o
Supported	2xx		-	o	o	m*	m*	o
Timestamp			o	o	o	o	o	o
To	c(1)	r	m	m	m	m	m	m
Unsupported	420		-	m	-	m	m	m
User-Agent			o	o	o	o	o	o
Via	R	amr	m	m	m	m	m	m
Via	rc	dr	m	m	m	m	m	m
Warning	r		-	o	o	o	o	o
WWW-Authenticate	401	ar	-	m	-	m	m	m
WWW-Authenticate	407	ar	-	o	-	o	o	o

表3：标题字段摘要，P--Z；（1）：可能已复制
标签添加

接受: 应用 tion/sdp;level=1, application/x-private, text/html

20.2 接受编码

The Accept-Encoding header field is similar to Accept, but restricts
响应中可接受的 content-codings [H3.5]。参见
[H14.3]. SIP中的语义与在定义中定义的语义相同
[H14.3].

一个空的Accept-Encoding头部字段是允许的。它是
等同于Accept-Encoding: identity，即仅是身份
编码，即无编码，是允许的。

如果不存在Accept-Encoding头字段，服务器应当
假设默认值为恒等映射。

这与HTTP定义略有不同，其中指示
当不存在时，可以使用任何编码，但使用身份编码
是首选。

示例： Translated Text：示例：

Accept-Encoding: gzip 接受编码: gzip

20.3 接受语言

The Accept-Language header field is used in requests to indicate the
首选语言用于原因短语、会话描述或
状态响应作为响应体中的消息携带。如果没有
接受语言头字段存在时，服务器应假定所有
语言对客户来说是可接受的。

The Accept-Language header field follows the syntax defined in
[H14.4]. 根据“ q ”对语言进行排序的规则
参数也适用于SIP。

示例： Translated Text：示例：

接受语言: da, en-gb;q=0.8, en;q=0.7

20.4 警报信息

当存在于一个INVITE请求中时，Alert-Info报头字段
指定了UAS的替代铃声。当存在于180
(响铃)响应，Alert-Info报头字段指定了
替代UAC的回铃音。典型用法是用于代理
将此标题字段插入以提供独特的铃声功能。

The Alert-Info header field can introduce security risks. These
风险及其处理方式在第20.9节中讨论。
讨论了 Call-Info 头字段，因为风险是
相同。

此外，用户应能够禁用此功能
选择性。

这有助于防止因使用{v*}而可能导致的干扰
此报头字段由不可信元素组成。

示例： Translated Text：示例：

警告信息：<http://www.example.com/sounds/moo.wav>

20.5 允许

The Allow header field lists the set of methods supported by the UA
生成消息。

所有方法，包括ACK和CANCEL，UA必须理解
包含在Allow头部字段的列表中，当
当前。缺少 Allow 报头字段不得
解释为发送消息的UA不支持
方法。 而是意味着 UA 没有提供任何
关于它支持哪些方法的信息。

在响应除{v*}之外的方法中提供Allow头字段
选项减少所需消息的数量。

示例： Translated Text：示例：

允许：INVITE, ACK, OPTIONS, CANCEL, BYE

20.6 认证信息

认证-信息报头字段提供相互
HTTP摘要认证。一个UAS可能包含此报头字段
对一个成功认证的请求的2xx响应中
使用基于Authorization头字段的摘要。

语法和 语义遵循RFC 2中指定的 617 [17].

示例： Translated Text：示例：

认证信息：nextnonce="47364c23432d2e131a5fb210812c"

20.7 授权

The Authorization header field contains authentication credentials of
a UA. 第22.2节概述了授权头的使用
字段，并且第22.4节描述了当使用时的语法和语义
使用HTTP身份验证。

此报头字段，连同Proxy-Authorization，打破了通用
关于多个报头字段值的规则。尽管不是逗号-
分隔列表，此报头字段名可能出现多次，
并且**不得**使用通常的方式合并到单个标题行中
规则描述在第7.3节中。

在下面的示例中，Digest周围没有引号
参数：

授权: 摘要用户名="Alice"，域="atlanta.com"，
nonce="84a4cc6f3082121f32b42a2187831a9e"，
响应="7587245234b3434cc3412213e5f113a5432"

20.8 通话标识

呼叫ID报头字段唯一标识特定的邀请
或所有特定客户的注册。一个单一的多媒体
会议可以引发几个具有不同Call-IDs的调用，
例如，如果一个用户多次邀请单个个体
相同的（长期运行的）会议。Call-IDs 区分大小写。
逐字节简单地比较。

The compact form of the Call-ID header field is i. Call-ID头字段的紧凑形式是 i。

示例：

呼叫标识符：f81d4fae-7dec-11d0-a765-00a0c91e6bf6@biloxi.com
i:7b1f81d4fae-7dec-11d0-a765-00a0c91e6bf6@192.0.2.4

20.9 通话信息

The Call-Info header field provides additional information about the 呼叫信息报头字段提供了有关 {v*}的附加信息
呼叫方或被叫方，取决于它是否出现在请求中
响应。URI的目的由“目的”描述。
参数。 "图标"参数指定一个适合作为{v*}的图像
标志性的主叫或被叫方表示。 "info"参数
描述调用者或被调用者的一般情况，例如，通过网页
页面。 "card" 参数提供名片，例如，在
vCard [36] 或 LDIF [37] 格式。 可以注册额外的标记
使用IANA以及第27节中的程序。

使用 Call-Info 头字段可能会带来安全风险。如果
被调用者获取由恶意调用者提供的URI，被调用者
可能存在显示不适当或冒犯性内容的危险，
危险或非法内容等。因此，它是
建议UA仅渲染Call-Info中的信息
表头字段如果可以验证元素的真实性
起源于标题字段并信任该元素。此需求不必
是等价用户代理；代理可以将此报头字段插入到请求中。

示例： Translated Text：示例：

呼叫信息：<http://wwww.example.com/alice/photo.jpg>；目的=图标，
<http://www.example.com/alice/> ;目的=信息

20.10 联系 {v*}

A Contact header field value provides a URI whose meaning depends on
请求或响应的类型。

A 联系头部字段值可以包含一个显示名称，一个带有 {v*} 的 URI
URI参数，和头部参数。

本文件定义了联系人参数 "q" 和 "expires"。
这些参数仅在联系人存在于 Translated Text: 这些参数仅在联系人存在于 （此处未包含公式符号 {v*} 的翻译，因为
注册请求或响应，或3xx响应中。附加
参数可能在其他规范中定义。

当头部字段值包含显示名称时，URI
包括所有URI参数的是"<"和">"。如果没有"<"
并且 ">" 存在，URI 之后的所有参数都是头信息
参数，不是URI参数。显示名称可以是标记，或者是一个 {v*}
引用字符串，如果需要更大的字符集。

即使 "display-name" 为空，"name-addr" 格式也必须
如果 "addr-spec" 包含逗号、分号或问号，则使用
mark. 显示名称之间可能存在或不存在空白字符（LWS）。
"<"

这些规则用于解析显示名称、URI和URI参数，以及
标题参数也适用于“ 收件人 ”和“ 发件人 ”标题字段。

> 联系人头字段的作用类似于位置头字段
> 字段在HTTP中。然而，HTTP头字段只允许一个
> 地址，未加引号。由于URI可以包含逗号和分号
> 作为保留字符，它们可能会被误认为是标题或
> 参数分隔符，分别。

紧凑形式的 Contact 头字段是 m（表示“ 已移动 ”）。

示例：

> 联系： "沃森先生" <sip:watson@worcester.bell-telephone.com>
> ; q=0.7; 过期=3600,
> "沃森先生" <mailto:watson@bell-telephone.com> ;q=0.1
> m: <sips:bob@192.0.2.4>;过期=60

20.11 内容处置

内容处置头字段描述了消息主体的处理方式
或者，对于多部分消息，应将消息体部分进行解释
由UAC或UAS。此SIP报头字段扩展了MIME内容-
类型（RFC 2183 [18]）。

几个新的“ Content-Disposition ”头部的“ 处置类型 ”
由SIP定义。值"session"表示该部分是会话
描述一个会话，无论是通话还是早期（预通话）媒体。
值 "render" 表示身体部位应该显示或
否则渲染给用户。注意值 "render" 的使用。
而不是“ 内联 ”，以避免MIME体是 {v*} 的含义
显示为整个消息渲染的一部分（因为自{v*}以来）
SIP消息的MIME体通常不会显示给用户）。
为了向后兼容，如果Content-Disposition头字段
缺少，服务器应假定Content-Type的正文
application/sdp 是“ 会话 ”处置，而其他内容
类型是 "渲染"。

处置类型“ 图标 ”表示该身体部位包含一个
图片适合作为主叫方或被叫方的标志性表示
当一条消息被用户代理信息性地呈现时，这可能被呈现为信息
已收到，或在对话进行期间持续存在。
值 "alert" 表示身体部位包含信息，例如
作为一个音频剪辑，应由用户代理进行渲染
尝试提醒用户收到一个请求，通常是一个
请求启动对话；此警报体例如
在180次响铃后作为电话铃声播放
临时响应已发送。

任何具有“ disposition-type ”且将内容呈现到　的MIME正文
用户仅在消息被正确处理时才应被处理
已认证。

处理参数，处理-param，描述了UAS应该如何
响应接收到的消息体，其内容类型或处置方式
类型它不理解。该参数有定义的值
"可选" 和 "必需"。如果处理参数缺失，
值 "必需" 应该被假定。处理参数是
描述于RFC 3204 [19]。

如果此报头字段缺失，则MIME类型确定默认值
内容处置。如果没有指定，则假定“ 渲染 ”。

示例： Translated Text：示例：

内容处置：会话

20.12 内容编码

内容编码报头字段用作对 Translated Text: {v*}的修饰符
"媒体类型"。当存在时，其值指示了哪些附加
内容编码已应用于实体主体，因此
解码机制必须应用以获取媒体类型
由Content-Type头字段引用。Content-Encoding是
主要用来允许一个物体在压缩时不失去
其底层媒体类型的身份。

如果对实体主体应用了多个编码，则
内容编码必须按照它们被列出的顺序进行
应用。

所有内容编码值均不区分大小写。IANA充当
内容编码值令牌的注册表。参见[H3.5]。
内容编码语法的定义。

客户端可以在请求体中应用内容编码。
服务器可以应用内容编码到响应体中。
服务器必须仅使用Accept-Encoding头中列出的编码
请求中的字段。

内容编码头字段的紧凑形式是 e.
示例：

内容编码：gzip
e: 打包

20.13 内容语言

参见 [H14.12]。示例：

内容语言: fr

20.14 内容长度

内容长度标头字段表示消息的大小-
主体，以十进制字节数表示，发送给接收者。
应用应使用此字段来指示{v*}的大小。
消息体要传输的内容，无论媒体类型如何
实体。如果使用基于流的协议（如TCP）作为
运输，必须使用头部字段。

消息体的尺寸不包括分隔的CRLF
表头字段和主体。任何内容长度大于或等于
零是一个有效值。如果消息中没有主体，那么
内容长度报头字段值必须设置为零。

The ability to omit Content-Length simplifies the creation of 翻译文本：省略Content-Length的能力简化了创建过程
类似于CGI的脚本，动态生成响应。

头字段紧凑形式为 l。

示例：

内容长度：349
l: 173 翻译文本：l: 173

20.15 内容类型

内容类型报头字段指示媒体类型
消息体发送给收件人。 "媒体类型"元素是
定义于 [H3.7]。如果 {v*} 存在，则 Content-Type 报头字段必须存在。
身体不为空。如果身体为空，并且有 Content-Type
表头字段存在，表示特定的体存在
类型长度为零（例如，一个空音频文件）。

头部字段的紧凑形式是 c。

示例：

内容类型: application/sdp
c: text/html; charset=ISO-8859-4 翻译文本：c: text/html; charset=ISO-8859-4

20.16 CSeq

请求中的CSeq报头字段包含一个十进制序列
数字和请求方法。序列号必须
可表示为32位无符号整数。CSeq的方法部分是
区分大小写。CSeq报头字段用于对事务进行排序
在对话框中，提供一种唯一标识的方法
交易，以及区分新请求和请求
重传。如果两个CSeq报头字段相等，则认为它们相等。
序列号和请求方法相同。 示例：

CSeq: 4711 邀请

20.17 日期

日期报头字段包含日期和时间。与HTTP/1.1不同，
SIP仅支持最新的RFC 1123 [20]日期格式。
在 [H3.3] 中，SIP 将 SIP-date 中的时区限制为 "GMT"，而
RFC 1123 允许任何时区。RFC 1123 日期区分大小写。

日期报头字段反映了请求或响应的时间
首先发送。

日期报头字段可以被简单的端系统使用
电池供电时钟以获取当前时间的概念。
然而，在其GMT形式中，它要求客户端知道他们的偏移量
从GMT。

示例： Translated Text：示例：

日期：星期六，2010年11月13日 23:29:00 GMT

20.18 错误信息

错误信息报头字段提供了一个指向附加信息的指针
错误状态响应的信息。

SIP UACs 具有从弹出窗口到用户界面能力的范围
Windows 和 PC 软客户端上的音频到“ 黑 ”仅音频
电话或通过网关连接的端点。 而不是强制
服务器在发送错误和选择发送错误之间产生错误
状态码带有详细原因短语并播放音频
录制时，Error-Info报头字段允许两者都发送。
UAC 然后可以选择渲染哪个错误指示器
呼叫者。

A UAC 可能将错误信息头字段中的 SIP 或 SIPS URI 视为
如果是一个重定向中的联系人，则生成一个新的INVITE，结果
在一个已建立的录音公告会议中。一个非SIP URI
可能被渲染给用户。

示例：

SIP/2.0 404 您拨打的号码未在服务
错误信息：<sip:not-in-service-recording@atlanta.com>

20.19 过期

The Expires header field gives the relative time after which the 过期头字段给出了相对时间，在此之后
消息（或内容）过期。

The The精确意义取决于方法 t.

邀请中的过期时间不影响其持续时间
实际会话，可能由邀请产生。会话
描述协议可能提供在 {v*} 上表达时间限制的能力
会话持续时间，然而。

这个字段的值是秒的整数（以十进制表示）
在0和(2**32)-1之间，从收到请求开始测量。

示例： Translated Text：示例：

过期时间：5

20.20 从

The From header field indicates the initiator of the request. This Translated Text: From头字段表示请求的发起者。 这
可能与对话的发起者不同。请求由
被调用方到调用方使用From头部的被调用方地址
字段。

可选的 "display-name" 是指由人类用户进行渲染的
接口。系统应使用显示名称“ 匿名 ”。
客户端的身份应保持隐藏。即使显示-
名称为空，如果"addr-spec"不存在，则必须使用"name-addr"格式。
包含逗号、问号或分号。语法问题
在7.3.1节中讨论。

两个From头字段等效，如果它们的URI匹配，并且它们的
参数匹配。一个报头字段中的扩展参数，不是
存在于其他地方的忽略用于比较的目的。此
表示显示名称以及是否存在尖括号
不影响匹配。

查看第20.10节以了解解析显示名称、URI和{v*}的规则。
URI参数，和头部字段参数。

The compact form of the From header field is f.

示例：

来自: "A. G. Bell" <sip:agb@bell-telephone.com> ;tag=a48s
从：sip:+12125551212@server.phone2net.com;标签=887s
f: 匿名 <sip:c8oqz84zk7z@隐私.org>;标签=hyh8

20.21 回复主题

The In-Reply-To header field enumerates the Call-IDs that this call
引用或返回。这些Call-IDs可能已被缓存。
客户端随后包含在此头部字段中，用于返回调用。

这允许自动呼叫分配系统路由回拨
呼叫第一个呼叫的发起者。这也允许
调用者以过滤调用，以便仅返回他们调用的返回调用
原始文本：originated will be accepted. This field is not a substitute for 翻译文本：起源于的将被接受。 此字段不能
请求身份验证。

示例： Translated Text：示例：

In-Reply-To: 707 10@土星.bell-tel.com, 17320@土星.bell- tel.com 翻译文本：tel.com

20.22 最大转发次数

The Max-Forwards header field must be used with any SIP method to
限制可以转发请求的代理或网关数量
到下一个下游服务器。这也可以在以下情况下很有用。
客户端正在尝试追踪一个看似的请求链
链中失败或循环。

最大转发值是一个范围在0-255之间的整数，表示
剩余此请求消息允许被调用的次数
已转发。此计数在每个转发的服务器上递减
请求。建议的初始值是70。

此报头字段应由无法的元素插入
否则保证循环检测。例如，一个B2BUA应该
插入一个Max-Forwards头部字段。

示例： Translated Text：示例：

最大转发次数：6

20.23 最小到期

最小过期头字段传达最小刷新间隔
支持由该服务器管理的软状态元素。
包括由注册商存储的联系人头部字段。
表头字段包含从0到的十进制整数秒数
(2**32)-1. 423（时间间隔过短）中头部字段的用法
响应在10.2.8、10.3和21.4.17节中描述。 {v*}

示例： Translated Text：示例：

最小过期时间：60

20.24 MIME版本

参见 [H19.4.1]。

示例： Translated Text：示例：

MIME-Version: 1.0 Translated Text: MIME-版本: 1.0

20.25 组织

组织标题字段传达组织的名称到
SIP元素所发出的请求或响应所属的。

}区域M AY 可由客户端软件用于过滤 c 所有。

示例： Translated Text：示例：

组织：Bob的盒子

20.26 优先级

请求优先级报头字段表示请求的紧急程度，如{v*}所示。
感知客户端。 优先级报头字段描述了
优先级，SIP请求应具有的接收人类或
其代理。例如，它可能被纳入关于呼叫的决策中。
路由和接受。对于这些决策，一个不包含 {v*} 的消息。
优先头字段应被视为指定了优先级
"正常"的。优先级报头字段不影响{v*}的使用
通信资源，例如数据包转发优先级在
路由器或PSTN网关中的电路访问。头部字段可以
具有值 "非紧急"、"正常"、"紧急" 和 "紧急情况"，
但是，可以在其他地方定义附加值。这是建议的
仅当生命、肢体或
属性处于即将到来的危险之中。否则，没有语义
定义为此报头字段。

这些是RFC 2076 [38]中的值，增加了
紧急

示例：

主题：龙卷风正在向我们这边移动！
优先级：紧急

或者

主题：周末计划
优先级：非紧急

20.27 代理认证

一个Proxy-Authenticate头字段值包含一个认证
挑战。

此报头字段的使用定义在[H14.33]。参见第
22.3 关于其使用的更多详细信息。

示例： Translated Text：示例：

代理认证: 摘要领域="atlanta.com",
域="sip:ss1.carrier.com"，qop="auth",
nonce="f84f1cec41e6cbe5aea9c8e88d359",
不透明=""，过时=FALSE，算法=MD5

20.28 代理授权

The Proxy-Authorization header field allows the client to identify
自身（或其用户）到需要身份验证的代理。
代理授权字段值由包含凭证的凭据组成
用户代理的代理和/或认证信息
请求资源的领域。

查看第22.3节以了解此报头字段的使用定义。

此报头字段，连同授权，违反了一般规则
关于多个头部字段名称。尽管不是以逗号分隔的
列表，此报头字段名可能出现多次，并且必须
不能使用常规规则合并为单个标题行
在7.3.1节中描述。

示例： Translated Text：示例：

代理授权：摘要用户名="Alice"，域="atlanta.com",
nonce="c60f3082ee1212b402a21831ae",
response="245f23415f11432b3434341c022"

20.29 代理-要求

代理-要求头字段用于指示代理敏感
特性必须由代理支持。参见第20.32节。
更多关于此消息机制的信息以及一个使用示例。

示例： Translated Text：示例：

代理要求：foo

20.30 记录路由

记录路由报头字段由代理在请求中插入
将对话框中的未来请求强制路由通过代理。

其与路由头字段的使用示例在{v*}中描述
章节 16.12.1。

示例： Translated Text：示例：

记录路由: <sip:server10.biloxi.com;lr>,
<sip:bigbox3.site3.atlanta.com;lr>

20.31 回复至

回复到头字段包含一个逻辑返回URI，可能为
与“ From ”报头字段不同。例如，URI可能为
用于返回未接听的电话或未建立的会话。
希望保持匿名，标题字段应当省略
从请求中获取或以不透露任何信息的方式填充
私人信息。

即使 "display-name" 为空，"name-addr" 格式也必须
如果 "addr-spec" 包含逗号、问号或
分号。语法问题在第7.3.1节中讨论。

示例： Translated Text：示例：

回复至：Bob <sip:bob@biloxi.com>

20.32 需求

The Require header field is used by UACs to tell UASs about options 需要头字段由UAC用于告知UAS选项
该UAC期望UAS支持以处理
请求。尽管是一个可选的报头字段，但Require不得
忽略如果存在。

The Require header field contains a list of option tags, described in Translated Text: 需求头字段包含一个选项标签列表，描
第19.2节。每个选项标签定义了一个SIP扩展，该扩展必须
理解以处理请求。 通常，这用于
指示需要指定一组扩展头字段
理解了。一个符合本规范的UAC必须仅包括
选项标签对应于标准跟踪的RFC。

示例： Translated Text：示例：

需要：100rel

20.33 重试后

Retry-After头字段可以与500（服务器内部错误）一起使用。
错误) 或 503 (服务不可用) 响应来指示 {v*} 的持续时间
服务预计将无法向请求客户端提供服务
带有404（未找到），413（请求实体过大），480
(暂时不可用)，486（此处忙碌），600（忙碌），或603

(下降)响应以指示被叫方预计
再次可用。此字段的值是一个正整数
秒数（十进制表示）在响应时间之后。

可选注释可用于表示附加信息
关于回调时间。一个可选的 "duration" 参数
指示从 {v*} 开始被叫方将可联系的时间长度
可用初始时间。如果没有给出持续时间参数，则
服务假定无限期可用。

示例：

重试后：18000；持续时间=3600
Retry-After: 120（我在开会）

20.34 路线

路由头字段用于强制请求通过 {v*} 进行路由
已列出的代理集。 路由头部的使用示例
字段位于第16.12.1节。

示例： Translated Text：示例：

路由：<sip:bigbox3.site3.atlanta.com;lr>,
<sip:服务器10.biloxi.com;lr>

20.35 服务器

服务器报头字段包含有关所使用软件的信息
由UAS处理请求。

揭示服务器的具体软件版本可能会允许
服务器更容易受到针对软件的攻击
已知包含安全漏洞。实现者应确保服务器
表头字段一个可配置选项。

示例： Translated Text：示例：

服务器：HomeServer v2

20.36 主题

主题头字段提供摘要或指示性质
的调用，允许在不解析的情况下进行调用过滤
会话描述。会话描述不必使用
与邀请相同的主题指示。

The The紧凑形式的主题头字段 是 s。

示例： Translated Text：示例：

主题：需要更多盒子
s: 技术支持

20.37 支持

The Supported header field enumerates all the extensions supported by UAC 或 UAS。

支持的报头字段包含一个选项标签列表，描述如下
在19.2节中，这些由UAC或UAS理解。一个UA
符合本规范的必须仅包括选项标签
对应于标准跟踪的RFC。如果为空，则表示没有
扩展被支持。

支持的报头字段的紧凑形式是 k。

示例： Translated Text：示例：

支持：100rel

20.38 时间戳

时间戳报头字段描述了UAC何时发送了请求
UAS

查看第8.2.6节以获取如何生成响应的详细信息。
请求包含标题字段的请求。尽管没有
规范性行为在此定义，它使用标题，它
允许扩展或SIP应用获取RTT估计。

示例： Translated Text：示例：

时间戳：54

20.39 至

The To header 收件人指定逻辑接收者 请求。

可选的 "display-name" 是指由人类用户进行渲染的
接口。 "tag" 参数作为一个通用机制用于
对話識别。

查看 S 第19.3节查看“ tag ”参数的详细信息 eter.

比较 To 标题字段是否相等与 {v*} 相同
比较 From 头部字段。参见第 20.10 节以了解规则
用于解析显示名称、URI和URI参数以及头部字段
参数。

T 他紧凑形式的 To 头部字段是 t.

以下是一些有效的To标题字段的示例：

收件人：操作员 <sip:operator@cs.columbia.edu>;tag=287447
t: sip:+12125551212@server.phone2net.com t: sip:+12125551212@server.phone2net.com

20.40 不支持

The Unsupported header field lists the features not supported by the 不支持的报头字段列出了不支持的功能
UAS. 请参阅第20.32节以了解动机。

示例： Translated Text：示例：

不支持：foo

20.41 用户代理

用户代理头字段包含有关UAC的信息
发起请求。此报头字段的语义是
定义于 [H14.43]。

揭示用户代理的具体软件版本可能允许
用户代理更容易受到针对软件的攻击
该已知包含安全漏洞。实现者应
用户代理头字段是一个可配置选项。

示例： Translated Text：示例：

用户代理：软电话Beta1.5

20.42 通过

The Via header field indicates the request has taken this path so far
表示在路由响应中应遵循的路径。
The branch ID parameter in the Via header field values serves as a
交易标识符，并由代理用于检测循环。

A Via头字段值包含用于发送的传输协议
消息、客户端的主机名或网络地址，以及可能
端口号，它希望在此接收响应。 一个 Via
表头字段值也可以包含如 "maddr" 这样的参数
"ttl", "received", 和 "branch"，其含义和使用方式已描述

在其他部分。对于符合本规范的实现
规范，分支参数的值必须以{v*}开头
魔法饼干 "z9hG4bK"，如第8.1.1.7节所述。

在此定义的传输协议是 "UDP"、"TCP"、"TLS" 和 "SCTP"。
"TLS"表示基于TCP的TLS。当向SIPS URI发送请求时，
协议仍然指示 "SIP"，传输协议是 TLS。

通过：SIP/2.0/UDP erlang.bell-telephone.com:5060;分支=z9hG4bK87asdks7 通过：SIP/2.0/UDP 192
.0.2.1:5060 ;接收=192.0.2.207 ;分支=z9hG4bK77asjd

The compact form of the Via header field is {v*}.

在这个例子中，消息来自一个多宿主主机，
两个地址，192.0.2.1 和 192.0.2.207。 发件人猜错了
关于将使用哪个网络接口。Erlang.bell-
电话.com 注意到了不匹配并添加了一个参数到
上跳的Via头部字段值，包含地址
实际上这个包来自。

主机或网络地址以及端口号不是必需的。
遵循SIP URI语法。具体来说，两侧的空白字符（LWS）
":" 或 "/" 是允许的，如下所示：

通过：SIP / 2.0 / UDP first.example.com: 4000；ttl=16
;maddr=224.2.0.1 ;branch=z9hG4bKa7c6a8dlze.1 Translated Text: ;maddr=224.2.0.1 ;branch=z9hG4bKa7c6a8dlze.1

即使本规范要求分支参数为
存在于所有请求中，头字段BNF表示
它是可选的。这允许与RFC 2543元素进行互操作，
无需插入分支参数。

两个 Via 头部字段相等，如果它们的发送协议和发送者相同
字段相等，两者具有相同的参数集，并且
所有参数的值都相等。

20.43 警告

警告头字段用于携带附加信息
关于响应的状态。警告头字段值被发送
包含响应，并包含三位数警告代码、主机名和
警告文本。

"warn-text" 应该使用最自然的语言
使接收响应的人类用户能够理解。 这
决策可以基于任何可用的知识，例如
用户位置，请求中的 Accept-Language 字段，或

响应中的Content-Language字段。默认语言是i-
默认 [21]。

当前定义的 "warn-code" 如下所示，其中公式符号 {v*} 保持不变。
建议使用英文的warn-text以及它们的含义描述。
这些警告描述了由会话描述引起的失败。
第一个以“ 3 ”开头的警告代码的首位数字表示
SIP特定警告。警告300至329为保留
指示会话描述中关键词的问题，330
通过339是有关请求的基本网络服务的警告
在会话描述中，370至379是有关{v*}的警告
会话描述中请求的定量QoS参数，以及
390至399为杂项警告，不属于任何一个类别
上述类别中。

300 不兼容的网络协议：一个或多个网络协议
包含在会话描述中的内容不可用。

301 不兼容的网络地址格式：一个或多个网络
会话描述中包含的地址格式不
可用。

302 不兼容的传输协议：一个或多个传输
会话描述中描述的协议不是
可用。

303 不兼容的带宽单位：一个或多个带宽
测量单位包含在会话描述中
未理解。

304 媒体类型不可用：包含一个或多个媒体类型的内容中
会话描述不可用。

305 不兼容的媒体格式：包含一个或多个媒体格式
在会话描述中不可用。

306 属性未理解：一个或多个媒体属性
在会话描述中不支持。

307 会话描述参数未理解：一个参数
除了上面列出的以外，未被理解。

330 多播不可用：用户所在的位置
不支持组播。

331 单播不可用：用户所在的位置的站点
不支持单播通信（通常由于存在
关于防火墙的）。

370 网络带宽不足：会话中指定的带宽
描述或定义媒体的超出已知
可用。

399 其他警告：警告文本可以包括任意
信息要呈现给人类用户或记录。
系统接收到此警告时**必须**不得采取任何自动化
动作。

1xx 和 2xx 已被 HTTP/1.1 采用。

可以通过IANA定义额外的"warn-code"，如定义在
第27.2节。

示例：

警告：307 isi.edu "会话参数 ’ foo ’ 未理解"
警告：301 isi.edu "不兼容的网络地址类型 ‘ E.164 ’ "

20.44 WWW-Authenticate

A WWW-Authenticate头字段值包含一个身份验证
挑战。参见第22.2节以获取有关其使用的更多详细信息。

示例： Translated Text：示例：

WWW-Authenticate: Digest realm="atlanta.com",
域="sip:boxesbybob.com"，qop="auth",
nonce="f84f1cec41e6cbe5aea9c8e88d359",
不透明=""，过时=FALSE，算法=MD5

21 响应代码

响应代码与HTTP/1.1响应代码一致，并扩展了它们
代码。并非所有HTTP/1.1响应代码都适用，并且只有
那些适当的在这里给出。其他HTTP/1.1响应
代码不应使用。此外，SIP定义了一个新类别，6xx。

21.1 临时 1xx

临时响应，也称为信息响应，
指示所接触的服务器正在执行一些进一步的操作
并且尚未有明确的回应。服务器发送一个1xx
响应，如果它预计需要超过200毫秒才能获得最终结果
响应。请注意，1xx响应不可靠地传输。
它们永远不会让客户端发送ACK。临时（1xx）
响应可能包含消息体，包括会话描述。

21.1.1 100 尝试

此响应表明请求已被接收
下一跳服务器并且正在对某些未指定的动作进行操作
代表这次通话（例如，正在咨询数据库）。
这个响应，就像所有其他临时响应一样，停止了
重传由UAC发出的INVITE。100（尝试中）响应是
与其他临时响应不同，因为它从未
由有状态代理向上游转发。

21.1.2 180 铃声

UA 接收到的 INVITE 正在尝试提醒用户。 这
响应可用于启动本地回铃。

21.1.3 181 呼叫正在转发

服务器可以使用此状态码来指示调用正在进行中
转发到不同的目的地集合。

21.1.4 182 队列中

被叫方暂时不可用，但服务器已有
决定排队接听电话而不是拒绝它。当被叫方
变为可用时，它将返回适当的最终状态
响应。原因短语可以提供有关{v*}的更多详细信息。
通话状态，例如，“ 5个通话排队；预期等待
时间是15分钟"。服务器可以发出多个182（排队）
响应以更新呼叫者有关排队通话状态的信息。

21.1.5 183 会话进度

183（会话进度）响应用于传达信息
关于未分类通话的进展。
原因短语、报头字段或消息体可用于传达
更多关于呼叫进度的详细信息。

21.2 成功的 2xx

请求成功。

21.2.1 200 正常

请求已成功。 与请求一起返回的信息
响应取决于请求中使用的方

21.3 重定向 3xx

3xx响应提供有关用户新位置的信息，或
关于可能满足呼叫的替代服务。

21.3.1 300 多选

请求中的地址解析为几个选择，每个选择都有其
自己的特定位置，并且用户（或UA）可以选择一个首选
通信端点并将其请求重定向到该位置。

响应可能包含一个包含资源列表的消息体
特性及用户或UA可选择的地理位置
最合适的一个，如果允许 Accept 请求头的话
字段。然而，此消息尚未定义MIME类型
身体。

选择也应列为联系字段（第20.10节）。
不同于HTTP，SIP响应可以包含多个Contact字段或一个
地址列表在联系人字段中。用户代理（UAs）可以使用联系人头
字段值用于自动重定向或可能要求用户确认
一个选择。然而，本规范未定义任何标准
对于此类自动选择。

此状态响应适用于被叫方可以接通的情况
在几个不同的位置，服务器无法或更愿意
不代理请求。

21.3.2 301 永久移动

用户无法在Request-URI中的地址找到。
并且请求客户端应在新地址重试，该地址由
联系人头字段（第20.10节）。请求者应
更新任何本地目录、地址簿和用户位置缓存
使用此新值并将未来请求重定向到地址（们）
列出。

21.3.3 302 临时移动

请求客户端应在新地址（们）上重试请求
由“ Contact ”报头字段（第20.10节）给出。请求URI
的新请求使用 Contact 头字段中的值
响应。

有效期 Contact URI 的持续时间可以被指示
通过一个Expires（第20.19节）报头字段或一个过期
参数在Contact头部字段中。代理和用户代理（UAs）都可以
缓存此URI至过期时间。如果存在
无明确过期时间，地址仅有效一次
递归，并且**不得**为未来的交易进行缓存。

如果从“ Contact ”头字段缓存的URI失败，则请求-
URI从重定向请求中可以再尝试一次。

临时URI可能比以下内容过时：{v*}
到期时间，并且可能有一个新的临时URI可用。

21.3.4 305 使用代理

请求的资源必须通过以下代理访问：
联系字段。联系字段提供代理的URI。
接收者应通过重复此单个请求
代理。 305（使用代理）响应必须仅由UAS生成。

21.3.5 380 替代服务

呼叫未成功，但可能使用其他服务。

The alternative services are described in the message body of the 替代服务在消息体中描述。
响应。此类主体的格式在此处未定义，可能为
未来的标准化主题。

21.4 请求失败 4xx

4xx 响应是从特定服务器的明确失败响应
服务器。客户端不应在没有
修改（例如，添加适当的授权）。
然而，对不同的服务器提出相同的请求可能会成功。

21.4.1 400 错误请求

请求因语法错误无法理解。
原因短语应更详细地标识语法问题，对于
示例，"缺少 Call-ID 标头字段"。

21.4.2 401 未授权

请求需要用户身份验证。此响应由
UASs 和注册机构，而 407（需要代理身份验证）是
由代理服务器使用。

21.4.3 402 需要支付

预留供将来使用。

21.4.4 403 禁止访问

服务器理解了请求，但拒绝执行它。
授权将不会帮助，请求不应重复。

21.4.5 404 未找到

服务器有确凿信息表明用户不存在于
请求URI中指定的域。此状态也是
返回，如果请求URI中的域与任何域不匹配
请求接收者处理的域

21.4.6 405 方法不允许

请求行中指定的方法被理解，但不
允许由Request-URI标识的地址。

响应必须包含一个包含列表的 Allow 头字段，列表中包含
有效的指示地址的方法。

21.4.7 406 不接受

请求中指定的资源只能生成
响应实体具有不可接受的内容特征
根据请求中发送的Accept头字段。

21.4.8 407 需要代理认证

此代码类似于401（未授权），但表示{v*}
客户端必须首先使用代理进行身份验证。SIP 访问
身份验证在第26节和22.3节中解释。

此状态码可用于需要访问的
通信信道（例如，电话网关）而不是
被调用方需要认证。

21.4.9 408 请求超时

服务器无法在合适的时间内生成响应
时间，例如，如果它无法确定用户的地理位置
及时。客户端可以不修改地重复请求。
任何后续时间。

21.4.10 410 已消失

请求的资源在服务器上不再可用。
转发地址已知。此条件预计为
考虑为永久。如果服务器不知道，或者没有
功能以确定条件是否永久，
状态码 404（未找到）应被使用。

21.4.11 413 请求实体过大

服务器拒绝处理请求，因为请求
实体主体大于服务器愿意或能够处理的。
服务器可以关闭连接以防止客户端
继续请求。

如果条件是临时的，服务器应包含一个Retry-
在标题字段之后表示它是临时的以及之后是什么
客户端可以再次尝试时计时。

21.4.12 414 请求URI太长

服务器拒绝服务请求，因为请求URI
比服务器愿意解释的更长。

21.4.13 415 不支持的媒体类型

服务器拒绝服务请求，因为消息
请求体格式不支持服务器
请求的方法。服务器必须返回一个可接受的列表。
使用Accept、Accept-Encoding或Accept-Language头部的格式
字段，取决于具体问题的内容。 UAC
此响应的处理在8.1.3.5节中描述。

21.4.14 416 不支持的 URI 方案

服务器无法处理请求，因为URI的方案不正确
在请求URI中，对服务器来说是未知的。客户端处理
此响应在8.1.3.5节中描述。

21.4.15 420 错误扩展

服务器不理解在{v*}中指定的协议扩展
代理要求（第20.29节）或要求（第20.32节）头
字段。服务器必须包含一个不支持扩展的列表
在响应中不支持的标题字段。 UAC处理
此响应在8.1.3.5节中描述。

21.4.16 421 扩展所需

UAS需要特定的扩展来处理请求，但这个
扩展未在请求的“ 支持 ”头字段中列出。
响应此状态码时必须包含一个Require头字段
列出所需的扩展。

无人机系统（UAS）不应使用此响应，除非它确实无法提供任何
有用的服务客户。相反，如果有一个期望的扩展
未列在支持的头部字段中，服务器应处理
请求使用基线SIP功能和任何受支持的扩展
由客户。

21.4.17 423 时间间隔过短

服务器拒绝请求，因为{v*}的过期时间
请求刷新的资源太短。此响应
可以由注册商拒绝一个其联系
表头字段过期时间太小。 此响应的使用
并且相关的 Min-Expires 头字段在章节中描述
10.2.8, 10.3, 和 20.23.

21.4.18 480 暂时不可用

被调用方的终端系统成功建立了联系，但被调用方是
当前不可用（例如，未登录，登录但
在一个阻止与被叫方通信的状态下，或者有
激活了“ 勿扰 ”功能）。响应可以指示
更好的时间在Retry-After头字段中调用。用户可以
也将在其他地方可用（对此服务器而言未知）。原因是
短语应指明更精确的原因，说明被调用者为何
不可用。此值应由UA设置。状态486
(忙碌在此) 可用于更精确地指示特定
通话失败的原因。

此状态也由重定向或代理服务器返回
识别由Request-URI标识的用户，但不
当前为该用户有一个有效的转发位置。

21.4.19 481 调用/事务不存在

此状态表示UAS收到一个请求，该请求不
匹配任何现有对话或交易。

21.4.20 482 循环检测到

服务器已检测到循环（第16.3节项目4）。

21.4.21 483 跳数过多

服务器接收了一个包含最大转发数（第{v*}节）的请求。
20.22) 头部字段值为零。

21.4.22 484 地址不完整

服务器收到了一个请求，其请求URI不完整。
应提供附加信息于原因短语中。

此状态码允许重叠拨号。 与重叠
拨号，客户端不知道拨号的长度
字符串。它发送长度逐渐增加的字符串，提示
用户输入更多，直到它不再接收484（地址
不完整）状态响应。

21.4.23 485 疑难

请求URI不明确。响应可能包含一个列表。
可能的唯一地址在联系头部字段中。揭示
替代方案可能会侵犯用户或组织的隐私。
它必须能够配置服务器以响应状态404
(未找到) 或抑制列出可能的选择
模糊的请求URI。

示例对带有 Request-URI 的请求的响应
sip:lee@example.com:

SIP/2.0 485 模糊
联系：Carol Lee <sip:carol.lee@example.com>
联系：李平 <sip:p.lee@example.com>
联系：李·M·富特 < 邮箱：lee.foote@example.com >

某些电子邮件和语音邮件系统提供此功能。
状态码与3xx分开使用，因为语义是
不同：对于300，假设是同一人或服务
将通过提供的选项实现。虽然自动化
选择或顺序搜索对于3xx响应是有意义的，用户
干预对于485（模糊）响应是必要的。

2021.4.24 486 忙碌中

被调用方的终端系统成功建立了联系，但被调用方是
目前不愿意或能够接听此端额外的电话
系统。响应可以指示一个更好的调用时间。
重试-后报头字段。用户也可能可用

在其他地方，例如通过语音邮件服务。状态 600（忙碌）
任何地方) 应当使用，如果客户端知道没有其他端
系统将能够接受这个调用。

21.4.25 487 请求终止

请求被 BYE 或 CANCEL 请求终止。此响应
自身不会返回用于 CANCEL 请求的 {v*}。

21.4.26 488 不可接受此处

响应与606（不可接受）具有相同含义，但只有
适用于由Request-URI指定的特定资源及其
请求可能在其他地方成功。

消息体包含对媒体功能的描述可能为
存在于响应中，其格式根据Accept
表头字段在INVITE（或如果不存在，则为application/sdp），的
与对 OPTIONS 请求的 200 (OK) 响应的消息体相同。

2021.4.27 491 请求待处理

请求被一个具有待处理请求的UAS接收
相同的对话框。第14.2节描述了如何处理此类 “ 眩光 ” 情况
已解决。

2021.4.28 493 不可破译

请求被一个包含加密MIME的UAS接收
主体，对于该主体，收件人没有或不会提供
适当的解密密钥。此响应可能有单个主体
包含一个适当的公钥，该公钥应用于加密
MIME 主体发送到此 UA。此响应的使用详情
代码可在第23.2节中找到。

21.5 服务器故障 5xx

5xx 响应是当服务器自身发生故障时给出的失败响应
错误。

21.5.1 500 服务器内部错误

服务器遇到了一个意外的条件，阻止了它
满足请求。客户端可以显示特定的错误
条件并可能在几秒后重试请求。

如果条件是临时的，服务器可以指示何时
客户端可以使用Retry-After头字段重试请求。

21.5.2 501 未实现

服务器不支持满足以下功能的特性
请求。这是当UAS没有时适当的响应。
识别请求方法且无法支持它
任何用户。 （代理无论方法如何都转发所有请求。）

请注意，当服务器发送405（方法不允许）时
识别请求方法，但该方法不允许或
支持。

21.5.3 502 网关错误

服务器在充当网关或代理时收到了一个无效的
响应来自它尝试访问的下游服务器
满足请求。

21.5.4 503 服务不可用

服务器暂时无法处理请求，因为
临时过载或维护服务器。服务器可能
指示在 Retry-After 中何时客户端应重试请求
表头字段。如果未提供Retry-After，客户端必须表现得好像
它收到了一个500（服务器内部错误）响应。

一个客户端（代理或UAC）收到503（服务不可用）时应当
尝试将请求转发到备用服务器。它不应
将任何其他请求在指定的时间内转发到该服务器
在Retry-After头部字段中，如果存在。

服务器可能会拒绝连接或丢弃请求，而不是
响应503（服务不可用）。

21.5.5 504 服务器超时

服务器未及时收到外部服务器的响应
在尝试处理请求时访问。 408（请求
超时（Timeout）应在在未收到响应的情况下使用。
上游中Expires头部字段指定的周期
服务器。

21.5.6 505 版本不支持

服务器不支持或拒绝支持SIP协议
版本在请求中使用。服务器指示
无法或不愿意使用相同的请求完成
主要版本作为客户端，除此外还有此错误信息。

21.5.7 513 信息过大

服务器无法处理请求，因为消息长度
超出其能力范围。

21.6 全局故障 6xx

6xx响应表示服务器具有关于{v*}的确切信息
特定用户，而不仅仅是文中指出的特定实例
请求URI。

21.6.1 600 处处忙碌

被调用方的终端系统成功建立了联系，但被调用方是
忙碌且此时不想接电话。 响应
可能表示在Retry-After头字段中调用更好的时间。
如果被调用者不希望透露拒绝的原因
呼叫，被叫方使用状态码603（拒绝）代替。此状态
响应仅在客户端知道没有其他端点时返回
（例如语音邮件系统）将回答请求。否则，
486（此处忙碌）应返回。

21.6.2 603 下降

被调用方的机器成功建立了联系，但用户
明确表示不想或不能参加。 响应可以
指示在Retry-After头字段中调用更好的时间。
状态响应仅在客户端知道没有其他
终点将回答请求。

21.6.3 604 任何地方都不存在

服务器拥有用户在{v*}中指示的权威信息
请求URI在任何地方都不存在。

21.6.4 606 不可接受

用户代理成功建立了联系，但某些方面
会话描述，例如请求的媒体、带宽或
处理风格不可接受。

一个606（不可接受）响应表示用户希望
沟通，但不能充分支持所描述的会话。
606（不可接受）响应可能包含一个原因列表在一个
警告头字段描述了为什么描述的会话无法
受支持。警告原因代码列于第20.43节。

消息体包含对媒体功能的描述可能为
存在于响应中，其格式根据Accept
表头字段在INVITE（或如果不存在，则为application/sdp），的
与对 OPTIONS 请求的 200 (OK) 响应的消息体相同。

希望不需要频繁进行谈判，当
一个新用户被邀请加入一个已经存在的会议，
协商可能无法进行。取决于邀请
发起者决定是否对606（不可接受）采取行动
响应。

此状态响应仅在客户端知道没有
其他端点将回答请求。

22 HTTP认证的使用

SIP提供了一种无状态的、基于挑战的机制，用于
基于HTTP认证的认证。任何时间
那代理服务器或UA收到一个请求（除以下情况外）
在22.1节中给出)，它可能挑战请求的发起者
为确保其身份。一旦发起者已被
已识别，请求的接收者应确认是否
此用户无权提出该请求。 无
授权系统在此文档中建议或讨论。

本节中描述的“ 摘要 ”认证机制
仅提供消息认证和重放保护，不
消息完整性或保密性。 上述保护措施
超出摘要提供的部分需要被考虑以防止活跃
攻击者从修改SIP请求和响应。

请注意，由于其安全性较弱，"Basic" 的使用
认证已被弃用。服务器不得接受
凭证使用“ 基本 ”认证方案，服务器也
不得使用“ 基本 ”进行挑战。这是对RFC 2543的变更。

22.1 框架

SIP认证框架与HTTP认证框架紧密相似
(RFC 2617 [17]). 尤其是auth-scheme、auth-param的BNF
挑战、领域、领域值和凭证是相同的（尽管
"“ 基本 ”作为方案的使用是不允许的）。在SIP中，一个UAS
使用401（未授权）响应来挑战一个的身份
UAC. 此外，注册机构和重定向服务器可以使用
401 (未授权) 响应用于身份验证，但代理必须
NOT，而不是MAY使用407（代理身份验证所需）

响应。代理认证的包含要求为
代理授权、WWW-认证和授权
各种消息与RFC 2617 [17]中描述的相同。

由于SIP没有规范根URL的概念，因此
保护空间的概念在SIP中被不同解释。
领域字符串单独定义保护域。这是一个变更
从RFC 2543中，其中请求URI和域一起
定义了保护域。

此先前对保护域的定义引起了一些数量
自UAC发送的Request-URI与
请求URI可能由挑战服务器接收而不同，
并且确实最终的 Request-URI 形式可能无法为人所知
UAC。此外，之前的定义依赖于{v*}的存在
在请求URI中的SIP URI并似乎排除了其他替代方案
URI方案（例如，tel URL）。

用户代理或代理服务器的操作员，将进行身份验证
接收到的请求必须遵守以下指南：
创建用于他们服务器的领域字符串：

o 实体字符串必须全局唯一。建议
一个领域字符串包含一个主机名或域名，后面跟随的
推荐在RFC 2617第3.2.1节中[17]。

o Realm 字符串应提供一个可读的标识符
可以被渲染给用户。

例如：

邀请 sip:bob@biloxi.com SIP/2.0
授权: 摘要域="biloxi.com"，<...>

通常，SIP认证对特定域是有意义的，一个
保护域。因此，对于摘要认证，每个此类
保护域有其自己的用户名和密码集。如果
服务器对于特定请求不需要身份验证，它
可以接受默认用户名 "匿名"，该用户名没有密码
(密码为""). 同样，代表许多用户的UAC，例如
PSTN 网关，可能有它们自己的设备特定用户名和
密码，而不是特定用户的账户，用于他们的领域。

虽然服务器可以合法地挑战大多数SIP请求，但是
此文件定义了两个需要特殊处理的请求
处理认证：ACK和CANCEL。

在一种使用响应携带值的认证方案下
用于计算非ces（如Digest），出现了一些问题
任何无响应的请求，包括ACK。因此，
任何在INVITE中由服务器接受的凭证都必须
被该服务器接受以进行ACK。创建ACK消息的UAC
将复制所有的授权和代理授权
表头字段值，这些值出现在INVITE中，对应于ACK
对应。服务器不得尝试挑战一个ACK。

尽管 CANCEL 方法确实接受一个响应（一个 2xx），服务器必须
不尝试挑战取消请求，因为这些请求无法
应重新提交。 通常，一个 CANCEL 请求应该被接受。
服务器如果来自发送请求的同一跳
已取消（前提是存在某种运输或网络层）
安全关联，如第26.2.1节所述，已建立）。

当UAC收到挑战时，它应该向用户展示{v*}。
挑战中“ realm ”参数的内容（在{v*}中出现）
或者是一个WWW-Authenticate报头字段或Proxy-Authenticate报头
字段) 如果UAC设备尚未知道一个凭证
所讨论的领域。一个预先配置UAs的服务提供商
具有其领域凭证的应意识到用户将不会
有机会展示他们在这个领域的资质
当在预配置设备上受到挑战时。

最后，请注意，即使UAC可以定位到凭证，即使凭证是{v*}
与适当的领域相关，存在这些的潜力
凭证可能不再有效，或者挑战服务器
无论出于什么原因（尤其是
当提交“ 匿名 ”且无密码时）。在此情况下，
服务器可能重复其挑战，或者它可能以403响应
禁止。UAC不得重新尝试使用凭据请求
那些刚刚被拒绝的（尽管请求可能会重试）
nonce已过期）。

22.2 用户到用户认证

当UAS收到UAC的请求时，UAS可以验证
发起者在请求处理之前。如果没有任何凭证
(在授权头字段中)提供在请求中，的
UAS可以通过拒绝来挑战发起者提供凭证
请求带有401（未授权）状态码。

The WWW-Authenticate response-header field MUST be included in 401
(未授权) 响应消息。字段值由以下组成
至少一个挑战，表明身份验证方案（们）
参数适用于领域。

一个401挑战中WWW-Authenticate头字段的示例
是：

WWW-Authenticate: Digest
领域="biloxi.com",
qop="auth,auth-int",
nonce="dcd98b7102dd2f0e8b11d0f600bfb0c093",
不透明="5ccc069c403ebaf9f0171e9517f40e41"

当原始UAC接收到401（未授权）时，它应该，
如果可能，使用适当的凭据重新发起请求。
UAC可能需要在原始用户输入之前
进行中。一旦提供认证凭据
(直接由用户操作，或在内置密钥环中发现)，
UAs 应该缓存 To 头部给定值的凭证
字段和 "领域" 并尝试在下一个中使用这些值
请求该目的地。 UAs 可以以任何方式缓存凭证
他们想要。

如果无法找到域的凭据，UACs可能会尝试
重试请求，用户名为 "匿名" 且无密码（a
密码 of “ ”）。

一旦找到凭证，任何希望进行操作的UA（用户代理）
自身通过UAS或注册机构进行认证 -- 通常如此，但并非总是如此
必须的，在收到401（未授权）响应后 -- 可以做
因此，通过在请求中包含一个Authorization头字段。
授权字段值由包含凭证的组成 {v*}
认证信息 of the UA for the 资源域
也被请求以及支持所需的参数
身份验证和重放保护。

A n 示例中，授权头字段 d 是：

Translated Text: d 是： {v*}
授权: 摘要用户名="bob",
领域="biloxi.com",
nonce="dcd98b7102dd2f0e8b11d0f600bfb0c093"
uri="sip:bob@biloxi.com",
qop=认证,
nc=00000001,
cnonce="0a4f113b",
response="6629fae49393a05397450978507c4ef1",
不透明="5ccc069c403ebaf9f0171e9517f40e41"

当UAC在收到后重新提交带有其凭据的请求时
401（未授权）或407（需要代理身份验证）响应，
它必须像通常一样增加CSeq头字段值
当发送更新请求时。

22.3 代理到用户认证

同样，当UAC向代理服务器发送请求时，代理
服务器可能在请求之前对发起者进行身份验证
处理完毕。如果不存在凭证（在Proxy-Authorization头中）
字段)在请求中提供时，代理可以挑战
发起者通过拒绝请求并返回407状态码来提供凭据
(代理身份验证要求) 状态码。代理必须填充
407（代理认证所需）消息与代理-
验证适用于代理的报头字段值
请求的资源。

代理认证（Proxy-Authenticate）和代理授权（Proxy-Authorization）的使用并行于
在[17]中描述，有一个区别。代理不得添加值
到Proxy-Authorization报头字段。所有407（代理
认证要求）响应必须向上游转发
UAC遵循对任何其他响应的程序。它是
UAC 添加 Proxy-Authorization 头字段的职责
包含代理域凭证的值的公式表示为 {v*}
请求身份验证。

如果代理重新提交一个添加了Proxy-Authorization的请求
表头字段值，它需要在新中增加 CSeq
请求。然而，这会导致提交该请求的用户账户控制（UAC）。
原始请求丢弃来自UAS的响应，因为CSeq
值将不同。

当原始UAC接收到407（代理认证
(必需)它应该，如果它能够，重新发起该请求。
有效的凭证。它应遵循相同的程序，
显示上述用于响应的“ 领域 ”参数
到401。

如果无法找到域的凭据，UACs可能会尝试
重试请求，用户名为 "匿名" 且无密码（a
密码 of“ ”）。

UAC 应该也缓存用于重新发起的凭证
请求。

以下规则推荐用于代理凭证缓存：

如果UA收到401/407中的Proxy-Authenticate头字段值
对具有特定 Call-ID 的请求的响应，它应该
将那个域的凭据纳入所有后续请求
包含相同Call-ID的。这些凭据不得缓存
跨对话框；然而，如果UA配置了其领域为
本地出站代理，当存在时，则UA可以缓存

凭证跨对话框适用于该域。注意这确实意味着
一个对话框中的未来请求可能包含不是的凭证
所需由沿路由头路径上的任何代理使用。

任何希望向代理服务器进行身份验证的UA --
通常，但不一定是，在收到407（代理）
认证要求) 响应 -- 可通过包含代理-
请求中的授权头字段值。 代理-
授权请求头字段允许客户端识别
自身（或其用户）到需要身份验证的代理。
代理授权头字段值由凭证组成
包含用于代理的UA的认证信息
和/或请求资源的领域。

一个代理授权头字段值仅适用于代理
其领域由“ realm ”参数标识（此代理可能
之前已要求使用Proxy-Authenticate进行身份验证
字段）。当在链中使用多个代理时，一个Proxy-
授权头字段值不得被任何代理消耗
其领域与在该处指定的 "realm" 参数不匹配
值

请注意，如果一个不支持域的认证方案
在Proxy-Authorization报头字段中使用，代理服务器必须
尝试解析所有Proxy-Authorization头部字段的值到
确定其中之一是否拥有代理服务器认为的
有效的凭证。因为这可能是非常耗时的-
在大型网络中消费，代理服务器应使用
认证方案支持在Proxy-Authorization中的领域
表头字段。

如果请求被分叉（如第16.7节所述），各种代理
服务器和/或UA可能希望挑战UAC。在这种情况下，
分叉代理服务器负责汇总这些挑战
合并为一个响应。每个WWW-Authenticate和Proxy-Authenticate
响应分叉请求收到的值必须放入
单次响应，由分叉代理发送给UA；
这些头部字段值的排序是不重要的。

当代理服务器对请求做出响应时发起挑战，
它将在UAC重试之前不代理请求
请求带有有效凭证。一个分叉代理可能会转发一个
同时请求多个需要 {v*} 的代理服务器
身份验证，每个都不会转发请求
直到原始UAC在它们中认证了自己
相应领域。如果UAC不提供凭据，

每个挑战，发出挑战的代理服务器将
不将请求转发到可能包含目标用户的UA
位于，因此，分叉的优点在很大程度上丧失了。

当响应401（未授权）时重新提交其请求
407（需要代理身份验证）包含多个
挑战，一个UAC可能包括每个WWW-的授权值
验证每个Proxy的值和Proxy-Authorization值
验证UAC希望提供凭据的值。
如上所述，请求中应包含多个凭证
通过“ 领域 ”参数区分。

同一领域可能存在多个相关挑战
在相同的401（未授权）或407（代理认证）中显示
必需的)。这可能在例如，当多个代理在时发生。
同一管理域，使用一个共同的领域，可以访问
由一个分叉请求。当它重试请求时，因此一个UAC可能
在Authorization或Proxy-Authorization中提供多个凭证
标题字段具有相同的 "realm" 参数值。 同样
凭证应用于同一领域。

22.4 摘要认证方案

本节描述了所需的修改和澄清 {v*}
将HTTP摘要认证方案应用于SIP。 SIP
方案使用几乎与HTTP完全相同[17]。

由于RFC 2543基于RFC 2069 [39]中定义的HTTP摘要，
SIP服务器支持RFC 2617必须确保它们是向后兼容的
兼容RFC 2069。 此处的向后兼容程序
兼容性在RFC 2617中指定。然而，请注意，SIP
服务器不得接受或请求基本认证。

The rules for Digest authentication follow those defined in [17],
使用 "HTTP/1.1" 替换为 "SIP/2.0"，并附加以下内容
差异：

1. 挑战中包含的URI具有以下BNF：

URI = SIP-URI / SIPS-URI

2. RFC 2617中的BNF存在一个错误，即‘ uri ’参数
HTTP摘要认证头字段的关于

身份验证未用引号括起来。 （The
示例在RFC 2617的第3.5节中是正确的。) 对于SIP,
‘ uri ’必须用引号括起来。

3. digest-uri-value 的 BNF 如下：

摘要URI值 = 请求URI；如第25节所述

4. 基于Etag选择随机数的示例过程如下
不适用于SIP。

5. RFC 2617 [17] 中关于缓存操作的文本不
应用于SIP。

6. RFC 2617 [17] 要求服务器检查 URI 在
请求行和包含在授权头中的URI
字段指向同一资源。在SIP上下文中，这两个
统一资源标识符（URI）可能指代不同的用户，因为某些情况下的转发。
代理。因此，在SIP中，服务器可以检查该
请求URI在授权头字段值中
对应于服务器愿意接受的用户
转发或直接请求，但并不一定是
如果两个字段不相等，则为失败。

7. 作为对计算A2值的说明,
消息完整性保证在摘要认证中
方案，实现者应假设，当实体主体是
空（即当SIP消息没有主体时）的哈希
实体主体解析为空的MD5哈希值
字符串，或：

H(实体主体) = MD5("") =
d41d8cd98f00b204e9800998ecf8427e

8. RFC 2617 指出，cnonce 值绝不能在
授权（以及扩展的代理授权）头
字段如果没有发送qop指令。因此，任何
算法依赖于cnonce（包括
"MD5-Sess") 需要发送 qop 指令。使用
"qop" 参数在 RFC 2617 中是可选的，用于 {v*} 的目的
与RFC 2069的向后兼容性；自RFC 2543以来；
根据RFC 2069，"qop"参数不幸必须
客户端和服务器接收时保持可选。然而,
服务器必须始终在WWW-Authenticate中发送一个"qop"参数
并且Proxy-Authenticate头字段值。如果客户端
接收挑战头字段中的 "qop" 参数，它
必须在任何结果授权中发送 "qop" 参数
表头字段。

RFC 2543 没有允许使用 Authentication-Info 头字段
(it 有效地使用了 RFC 2069)。然而，我们现在允许使用此
表头字段，因为它对主体提供完整性检查
提供相互认证。RFC 2617 [17] 定义了机制，
使用请求中的qop属性实现向后兼容性。
这些机制必须由服务器使用以确定客户端
支持RFC 2617中未指定的新的机制
RFC 2069.

23 S/MIME

information体，并且MIME标准包括
确保完整性的MIME内容的安全机制
保密（包括‘ multipart/signed ’和
‘ application/pkcs7-mime ’ MIME 类型，参见 RFC 1847 [22]，RFC 2630 [23]
然而，实施者应注意，尽管如此（和RFC 2633 [24]）。
可能是一些罕见的网络中介（非典型代理服务器）
依赖于查看或修改SIP消息的主体（特别是
SDP)，并且安全MIME可能阻止这些中间代理
从功能。

这尤其适用于某些类型的防火墙。

PGP机制用于加密标题字段和正文的内容
SIP消息，在RFC 2543中描述的，已被弃用。

23.1 S/MIME 证书

The certificates that are used to identify an end-user for the 证书，用于识别最终用户，用于
S/MIME的目的与服务器所用的目的在一点上有所不同
尊重 - 而不是断言持有人身份
对应于特定的主机名，这些证书断言
持有者通过终端用户地址进行识别。此地址是
由 "userinfo" "@" 和 "domainname" 的连接组成
SIP或SIPS URI的部分（换句话说，一个电子邮件地址的
"bob@biloxi.com" 的格式)，通常对应于一个用户的
记录地址

这些证书还与用于的密钥相关联
签名或加密SIP消息的主体。主体使用{v*}进行签名。
发送者的私钥（其中可能包括他们的公钥）
消息适当），但正文使用公钥加密
的意图接收者。显然，发送者必须
对收件人公钥的预先了解以进行加密
消息体。公钥可以存储在UA上的虚拟
钥匙圈。

每个支持S/MIME的用户代理必须包含一个密钥环
专门用于最终用户证书。此密钥环应映射
在记录地址和相应证书之间。 以上
时间，用户在填充时应当使用相同的证书
源URI（From头字段）相同
记录地址

任何依赖于终端用户证书存在的机制
在以下方面严重受限，因为实际上几乎没有整合
今天提供终端用户应用程序证书的权威机构。
然而，用户应从已知的公共机构获取证书
证书颁发机构。作为替代，用户可以创建自签名证书。
已签名证书。自签名证书的含义
beì ji ǎ n zhí zh ō ng y ǔ n yìng zhèng zài Bù shù 26.4.2 zh ō ng. Shí sh ī huà y ě k ě néng sh ǐ yòng
预配置证书在部署中，其中包含先前的信任
所有SIP实体之间存在关系。

在获取最终用户证书的问题之上和之外，
存在少数知名的集中式目录进行分发
终端用户证书。然而，证书持有者应
发布他们的证书到任何适当的公共目录中。
同样，UACs 应该支持导入（手动或
自动发现的公共目录中的证书
对应于SIP请求的目标URI。

23.2 S/MIME 密钥交换

SIP 本身也可以用作分发公钥的手段
以下方式。

无论何时在SIP的S/MIME中使用CMS SignedData消息，它
必须包含包含所需公钥的证书
验证签名。

当UAC发送包含S/MIME正文的请求时，该请求启动
对話，或在对话上下文中发送非-INVITE请求。
对话，UAC 应将正文结构化为 S/MIME
’multipart/signed’ CMS 签名数据体。如果需要 CMS 服务
是 EnvelopedData（并且已知目标用户的公钥），
UAC 应该发送封装在其中的 EnvelopedData 消息
签名数据消息。

当UAS收到包含S/MIME CMS正文的请求时
包含证书，UAS 应首先验证
证书，如有可能，包括任何可用的根证书
证书颁发机构。UAS 应该还确定主题
证书（对于S/MIME，SubjectAltName将包含）
适当的身份)并将此值与“ From ”报头字段进行比较

的请求。如果证书无法验证，因为它是不正确的或已过期。
自签名，或由未知机构签名，或如果它是可验证的
但是其主题与“ From ”报头字段不对应
请求，UAS 必须通知其用户的状态
证书（包括证书的主题、其签署者、）
并且任何密钥指纹信息）并请求明确许可
在继续之前。如果证书已成功验证并
证书的主题对应于“ From ”报头字段
SIP请求的，或者如果用户（在通知后）明确
授权使用证书，UAS 应该添加此
证书到本地密钥环，按记录地址索引
证书持有者。

当UAS发送包含S/MIME正文的响应时
第一次对话请求或对非-INVITE请求的响应
在对话之外，UAS 应该将正文结构化为
一个S/MIME 'multipart/signed' CMS 签名数据体。如果所需的CMS
服务是EnvelopedData，UAS应发送EnvelopedData
消息封装在SignedData消息中。

当UAC收到包含S/MIME CMS正文的响应时
包含证书，UAC 应首先验证
证书，如果可能，带有任何适当的根证书。
UAC 应该还确定证书的主题并比较
此值到响应的“ 收件人 ”字段；尽管这两个可能非常
我们将不同，这并不一定表明
安全漏洞。如果证书无法验证，因为它是不变的。
自签名，或由未知机构签名，UAC 必须通知其
证书状态的用户（包括证书主题）
证书、其签署者以及任何密钥指纹信息）和
在继续之前请明确请求许可。如果证书
已成功验证，证书的主题
对应于响应中的 To 标头字段，或如果用户
(通知后)明确授权使用{v*}。
证书，UAC 应该将此证书添加到本地密钥环中，
按证书持有者的记录地址索引。
如果UAC没有在任何情况下将其自己的证书传输给UAS
上一笔交易，它应使用CMS SignedData体
下一个请求或响应。

在未来的场合，当UA收到请求或响应时，
包含一个与密钥环中某个值对应的 From 标头字段，
UA 应当将这些消息中提供的证书与
现有的证书在其密钥环中。如果存在差异，
UA 必须通知其用户证书变更
(最好用表明这是潜在安全性的术语来表达)
越权（）并在继续之前获取用户的许可

处理信号。如果用户授权此证书，则
应添加到密钥环中，与任何先前的值一起
此记录地址。

请注意，然而，这个密钥交换机制并不
确保在自签名证书时安全交换密钥，
或由一个不为人知的机构签发的证书被使用 - 它是
易受已知攻击。在作者看来，
然而，它提供的保障是众所周知地更好
什么都没有；实际上，它与广泛使用的SSH应用程序相当。
这些限制在26.4.2节中进行了更详细的探讨。

如果UA收到一个使用公钥加密的S/MIME正文
关键未知于收件人，它必须拒绝带有493的请求
(无法解读) 响应。此响应应包含一个有效的
证书针对被答辩人（如可能，对应于任何
记录地址在拒绝的“ 收件人 ”标题字段中给出
请求) 在具有“ certs-only ” "smime-type"的MIME体中
参数。

一个未解密的493（无法解读）未附带任何证书的发送表示
受访者无法或不愿使用S/MIME加密消息，
尽管他们可能仍然支持S/MIME签名。

请注意，接收包含S/MIME请求的用户代理
必选的body（带有Content-Disposition头）
"处理"参数的"必需"必须拒绝请求
415 不支持的媒体类型响应，如果MIME类型不是
理解了。当S/MIME收到此类响应时，一个用户代理
发送时应通知其用户远程设备不
支持S/MIME，并且它可能随后重新发送请求而不
S/MIME，如果适用；然而，此415响应可能构成
降级攻击。

如果用户代理在请求中发送了S/MIME正文，但收到
响应包含未加密的MIME主体的，UAC
应通知其用户会话无法被加密。
然而，如果一个支持S/MIME的用户代理收到一个带有 {v*}的请求
一个未加密的实体，它不应以加密的实体响应，但如果
它期望发件人使用S/MIME（例如，因为发件人的 {v*}
从标题字段值对应其密钥链上的一个身份标识),
UAS 应该通知其用户，会话无法被加密。

一些在前面文本中出现的条件需要
用户异常证书管理的通知
事件发生。用户可能会问在这种情况下他们应该做什么
情况。首先，一个意外的变化在
证书，或当预期安全时缺乏安全，都是

引起谨慎的原因，但不一定是攻击的迹象
进行中。用户可能会终止任何连接尝试或拒绝{v*}。
连接请求他们已接收；在电话术语中，他们
可以挂断并回拨。用户可能希望寻找替代方案
表示联系对方并确认他们的密钥的方式
合法更改。请注意，用户有时被迫
更改他们的证书，例如当他们怀疑时
私钥的安全性已受到损害。当他们的
私钥不再私密，用户必须合法生成
新密钥并与持有旧密钥的任何用户重新建立信任
键。

最后，如果在对话过程中UA收到一个证书
在一个与CMS SignedData消息不对应的CMS
证书在对话中先前已交换，UA必须通知
其变更的用户，最好是用表明这是
可能的安全漏洞。

23.3 保护 MIME 主体

有两种安全的MIME体类型，这些类型对{v*}感兴趣。
SIP：使用这些实体应遵循S/MIME规范[24]
带有一些变化。

"multipart/signed" 必须仅与 CMS 分离使用
签名。

这允许与非S/MIME-的向后兼容性
合规接收者。

o S/MIME 主体应包含 Content-Disposition 报头字段，
并且“ 处理 ”参数的值应该是“ 必需 ”。

如果UAC的密钥环上没有与其关联的证书，
记录地址，它想要发送请求的
无法发送加密的 "application/pkcs7-mime" MIME 消息。
UACs 可能发送一个初始请求，例如一个 OPTIONS 消息
使用CMS分离签名以征求
远程端证书（签名应覆盖在）
"消息/sip" 类型描述在第23.4节中所述的正文。

请注意，S/MIME的未来标准化工作可能会定义 {v*}
非证书密钥。

发送者应使用“ SMIMECapabilities ”
(参见[24]的第2.5.2节)属性来表示它们的
功能与偏好，用于进一步通讯。注意
特别地，发送者可以使用“ preferSignedData ”

能力鼓励接收者以CMS进行响应
签名数据消息（例如，在发送 OPTIONS 时）
请求如上所述）。

o S/MIME实现至少必须支持SHA1作为
数字签名算法，以及3DES作为加密
算法。所有其他签名和加密算法可能
支持。实现可以协商对这些功能的支持
具有 "SMIMECapabilities" 属性的算法。

每个S/MIME正文在SIP消息中应当仅使用{v*}进行签名
一份证书。如果UA收到包含多个
签名，最外层的签名应被视为
单一证书为此实体。并行签名应
不可使用。

以下是一个加密的S/MIME SDP体的示例
在SIP消息中：

邀请 sip:bob@biloxi.com SIP/2.0
通过：SIP/2.0/UDP pc33.atlanta.com;分支=z9hG4bKnashds8
收件人：Bob sip:bob@biloxi.com>
从：Alice <sip:alice@atlanta.com>;tag=1928301774
呼叫标识符：a84b4c76e66710
CSeq: 314159 邀请
最大转发数：70
联系：<sip:alice@pc33.atlanta.com>
内容类型：application/pkcs7-mime; smime-type=enveloped-data;
name=smime.p7m
内容处置：附件；文件名=smime.p7m
处理=所需

```
*******************************************************
        * 内容类型：application/sdp                    *
        *                                              *
        * v=0                                          *
        * o=alice 53655765 2353687637 IN IP4 pc33.atlanta.com *
        * s=-                                          *
        * t=0 0                                        *
        * c=IN IP4 pc33.atlanta.com                    *
        * m=音频 3456 RTP/AVP 0 1 3 99                 *
        * a=rtpmap:0 PCMU/8000                         *
*******************************************************
```

23.4 使用S/MIME的SIP头部隐私和完整性：隧道SIP

作为一种提供一定程度端到端认证的手段,
完整性或机密性用于SIP头部字段，S/MIME可以
将整个SIP消息封装在类型为MIME的体中
"message/sip" 然后对这些体应用 MIME 安全性
与典型的SIP体相同。这些封装的SIP请求
并且响应不构成一个独立的对话或交易,
它们是用于验证的“ 外部 ”消息的副本
完整性或提供更多信息。

如果UAS收到包含隧道“ 消息/sip ”的请求
S/MIME 主体，它应包括一个隧道化的 "message/sip" 主体
响应与相同的smime-type。

任何传统的MIME正文（如SDP）应附加到
"内部"消息，以便他们也能从S/MIME安全中受益。
请注意，“ message/sip ”体可以作为MIME的一部分发送
"multipart/mixed" 体如果包含任何未加密的 MIME 类型也应
在请求中传输。

23.4.1 SIP头部的完整性和机密性属性

当使用S/MIME完整性或保密机制时,
可能存在“ 内部 ”消息中值之间的差异
并且“ 外部 ”消息中的值。处理此类值的规则为
所有在此文档中描述的标题字段的不同之处
本节中给出。

请注意，为了松散时间戳的目的，所有SIP消息
该隧道 "消息/sip" 应该同时包含一个日期头
"内部"和"外部"标题。

23.4.1.1 完整性

无论何时进行完整性检查，头部的完整性
字段应由匹配标题字段值来确定
在签名体中使用“ 外部 ”消息中的那个
SIP的比较规则，如20所述。

标题字段，可以被代理服务器合法修改的有：
请求URI、Via、Record-Route、Route、Max-Forwards和Proxy-
授权。如果这些头部字段在端到端传输过程中不完整,
实现不应将此视为安全违规。
对本文档中定义的任何其他标题字段所做的更改
构成完整性违规；用户**必须**被告知
差异。

23.4.1.2 保密性

当消息被加密时，头部字段可能包含在
加密体，这些体不在“ 外部 ”消息中。

一些标题字段必须始终有明文版本，因为它们
请求和响应中必需的头部字段 - 这些包括：

To，From，Call-ID，CSeq，Contact。虽然可能没有用
提供一个加密的替代方案用于 Call-ID、CSeq 或 Contact，
提供对“ 外部 ”的“ 到 ”或“ 从 ”信息的替代
允许。注意，加密体中的值不使用
为了识别交易或对话 - 它们是
仅作信息。如果加密正文中的“ From ”头字段
与“ 外部 ”消息中的值不同，{v*}内的值
加密体应向用户显示，但不得使用
在“ 外部 ”标题字段中的任何未来消息。

主要，用户代理将希望加密具有的头部字段
端到端语义，包括：主题，回复地址，组织，
接受, 接受编码, 接受语言, 警报信息, 错误信息,
认证信息，过期时间，回复至，要求，支持，
不支持、Retry-After、User-Agent、Server 和 Warning。如果其中任何一项
这些头部字段存在于加密体中，它们应该是
使用代替任何“ 外部 ”标题字段，无论这涉及什么
显示标题字段值给用户或设置内部
UA中的状态。然而，它们不应在“ 外部 ”中使用。
消息头部的任何未来消息。

如果存在，日期报头字段必须始终相同在
"内部"和"外部"标题。

由于MIME正文附加到“ 内部 ”消息中，
实现通常将MIME特定的头字段进行加密，
包括：MIME-Version, Content-Type, Content-Length, Content-
语言，内容编码和内容处置。 “ 外部 ”
消息将包含适用于S/MIME正文的正确MIME头字段。
这些头部字段（以及它们之前的内容）应该
作为正常的MIME头部字段处理，并在SIP中接收到的正文
消息。

以下头部字段加密并不特别有用：
最小过期时间、时间戳、授权、优先级和WWW-
验证。此类别还包括那些标题字段，
可以被代理服务器（在上一节中描述）更改。
UAs 应该永远不会在“ 内部 ”消息中包含这些，如果它们不是

包含在“ 外部 ”消息中。接收这些消息之一的用户代理（UAs）
加密体中的标题字段应忽略加密
值

请注意，SIP的扩展可能定义额外的头部字段；
作者应描述这些扩展的完整性
机密性属性此类报头字段。如果SIP UA
遇到一个未知的首部字段，存在完整性违规，它
必须忽略头部字段。

23.4.2 隧道完整性及认证

隧道化S/MIME正文内的SIP消息可以提供完整性保障
SIP头字段 如果发送方希望的头字段
安全的数据在带有CMS签名的“ 消息/sip ”MIME正文中进行复制
分离签名。

假定“ 消息/sip ”体至少包含以下内容
基本对话标识符（To，From，Call-ID，CSeq），然后是一个
签名MIME体可以提供有限的认证。在极
最少，如果用于签名主体的证书是未知的
接收者且无法验证，签名可以用来
确认在对话中较晚的请求是由以下方式传输的
与启动对话的同一位证书持有者。如果收件人
签名MIME体有更强的动机去信任
证书（他们能够验证它，他们从……获得了它）
可信的仓库，或者他们经常使用它）那么
签名可以被视为对身份的更强断言
证书主题。

为了消除关于加法或
整个标题字段的减法，发送者应复制所有
请求签名体内的头部字段。任何消息
需要完整性保护的实体必须连接到
"内部"消息。

如果消息中包含签名体且有日期头，则
接收者应将报头字段值与其自身内部值进行比较
时钟，如适用。如果检测到显著的时间差异
(一个或更多小时的数量级)，用户代理应提醒
用户到异常，并注意这是一个潜在的安全漏洞。

如果消息接收者检测到消息中的完整性违规，
消息可能会被拒绝，并返回403（禁止）响应，如果它是
一个请求，或任何现有对话可以终止。UAs 应该
通知用户此情况并请求明确指导
如何进行。

以下是一个使用隧道化的“ 消息/sip ”的示例
正文：

邀请 sip:bob@biloxi.com SIP/2.0
通过：SIP/2.0/UDP pc33.atlanta.com;分支=z9hG4bKnashds8
收件人：Bob sip:bob@biloxi.com>
从：Alice <sip:alice@atlanta.com>;tag=1928301774
呼叫标识符：a84b4c76e66710
CSeq: 314159 邀请
最大转发数：70
日期：周四，2002年2月21日 13:02:03 GMT
联系：<sip:alice@pc33.atlanta.com>
内容类型：multipart/signed；
协议="application/pkcs7-signature";
micalg=sha1; boundary=boundary42
内容长度：568

--boundary42
内容类型: sip 消息

邀请 sip:bob@biloxi.com SIP/2.0
通过：SIP/2.0/UDP pc33.atlanta.com;分支=z9hG4bKnashds8
收件人：Bob <bob@biloxi.com>
从：Alice <alice@atlanta.com>；标签=1928301774
呼叫标识符：a84b4c76e66710
CSeq: 314159 邀请
最大转发数：70
日期：周四，2002年2月21日 13:02:03 GMT
联系：<sip:alice@pc33.atlanta.com>
内容类型: application/sdp
内容长度：147

v=0
o=用户A 2890844526 2890844526 IN IP4 here.com
s=会话SDP
c=IN IP4 pc33.atlanta.com 翻译文本：c=IN IP4 pc33.atlanta.com
t=0 0
m=音频 49172 RTP/AVP 0
a=rtpmap:0 PCMU/8000 翻译文本：a=rtpmap:0 PCMU/8000

--boundary42
内容类型：application/pkcs7-signature; 名称=smime.p7s
内容传输编码：base64
内容处置：附件；文件名=smime.p7s；
处理=所需

ghyHhHUujhJhjH77n8HHGTrfvbnj756tbB9HG4VQpfyF467GhIGfHfYT6
4VQpfyF467GhIGfHfYT6jH77n8HHGghyHhHUujhJh756tbB9HGTrfvbnj
n8HHGTrfvhJhjH776tbB9HG4VQbnj7567GhIGfHfYT6ghyHhHUujpfyF4
7GhIGfHfYT64VQbnj756

--boundary42-

23.4.3 隧道加密

它也可能希望使用此机制来加密一个
"message/sip" MIME 主体在 CMS EnvelopedData 消息 S/MIME 中
主体，但在实践中，大多数头部字段至少有一些用途
网络；使用S/MIME进行加密的一般目的是为了确保安全
消息体如SDP而不是消息头。
信息性标题字段，例如主题或组织
可能值得端到端安全。由未来定义的标题
SIP应用程序可能还需要混淆。

另一个加密头部字段的可能应用是选择性
匿名。一个请求可以通过一个 From 头部字段来构造
包含没有个人信息（例如，
sip:anonymous@anonymizer.invalid)。然而，第二个From头
字段包含发起者的真实记录地址
可能在 "message/sip" MIME 体中加密，其中它将
仅对对话的端点可见。

请注意，如果使用此机制用于匿名，则From头
字段将不再由消息的接收者可用
索引到它们的证书密钥链以检索正确的
S/MIME 密钥与发件人关联。消息必须首先
解密，并且必须使用“ 内部 ”的From头字段。
索引。

为了提供端到端完整性，加密的“ 消息/sip ”
MIME 主体应由发送者签名。这会创建
multipart/signed MIME 主体，其中包含一个加密的正文和
签名，两者类型均为 "application/pkcs7-mime"。

在以下示例中，一个加密并签名的消息的
文本中用星号（*）括起来的内容是加密的：

邀请 sip:bob@biloxi.com SIP/2.0
通过：SIP/2.0/UDP pc33.atlanta.com;分支=z9hG4bKnashds8
收件人：Bob sip:bob@biloxi.com>
从：匿名 <sip:anonymous@atlanta.com>;tag=1928301774
呼叫标识符：a84b4c76e66710
CSeq: 314159 邀请
最大转发数：70
日期：周四，2002年2月21日 13:02:03 GMT
联系：<sip:pc33.atlanta.com>
内容类型：multipart/signed；
协议="application/pkcs7-signature";
micalg=sha1; boundary=boundary42
内容长度：568

--boundary42
内容类型：application/pkcs7-mime; smime-type=enveloped-data;
name=smime.p7m
内容传输编码：base64
内容处置：附件；文件名=smime.p7m
处理=所需
内容长度：231

```
**********************************************************
* 内容类型：message/sip                          *
*                                                *
* 邀请 sip:bob@biloxi.com SIP/2.0                *
* 通过：SIP/2.0/UDP pc33.atlanta.com;分支=z9hG4bKnashds8 *
* 收件人：Bob <bob@biloxi.com>                   *
* 来自：Alice <alice@atlanta.com>；标签=1928301774 *
* Call-ID: a84b4c76e66710                        *
* CSeq: 314159 INVITE                            *
* 最大转发次数：70                               *
* 日期：周四，2002年2月21日 13:02:03 GMT         *
* 联系方式：<sip:alice@pc33.atlanta.com>         *
*                                                *
* 内容类型：application/sdp                      *
*                                                *
* v=0                                            *
* o=alice 53655765 2353687637 IN IP4 pc33.atlanta.com *
* s=会话SDP                                      *
* t=0 0                                          *
* c=IN IP4 pc33.atlanta.com                      *
* m=音频 3456 RTP/AVP 0 1 3 99                   *
* a=rtpmap:0 PCMU/8000                           *
**********************************************************
```

--boundary42
内容类型：application/pkcs7-signature; 名称=smime.p7s
内容传输编码：base64
内容处置：附件；文件名=smime.p7s；
处理=所需

ghyHhHUujhJhjH77n8HHGTrfvbnj756tbB9HG4VQpfyF467GhIGfHfYT6
4VQpfyF467GhIGfHfYT6jH77n8HHGghyHhHUujhJh756tbB9HGTrfvbnj
n8HHGTrfvhJhjH776tbB9HG4VQbnj7567GhIGfHfYT6ghyHhHUujpfyF4
7GhIGfHfYT64VQbnj756

--boundary42-

24 示例

在以下示例中，我们通常省略消息体和
对应的内容长度和内容类型头部字段为
简洁。

24.1 注册

鲍勃在启动时注册。消息流程如图9所示。
请注意，通常用于注册的认证不是
为简单起见而展示。

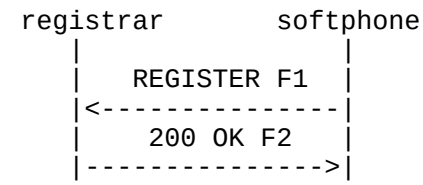

图 9：SIP 注册示例

F1 注册 Bob -> 注册员

注册 sip:registrar.biloxi.com SIP/2.0
通过：SIP/2.0/UDP bobspc.biloxi.com:5060;分支=z9hG4bKnashds7
最大转发数：70
收件人：Bob sip:bob@biloxi.com>
从：Bob <sip:bob@biloxi.com>;tag=456248
呼叫标识符：843817637684230@998sdasdh09
CSeq: 1826 注册
联系：<sip:bob@192.0.2.4>
过期时间：7200
内容长度：0

注册将在两小时后过期。注册员将做出回应
带有200 OK：

F2 200 OK 注册商 -> Bob

SIP/2.0 200 正确
通过：SIP/2.0/UDP bobspc.biloxi.com:5060;分支=z9hG4bKnashds7
;接收=192.0.2.4
收件人：Bob sip:bob@biloxi.com>;tag=2493k59kd
从：Bob <sip:bob@biloxi.com>;tag=456248
呼叫标识符：843817637684230@998sdasdh09
CSeq: 1826 注册
联系：<sip:bob@192.0.2.4>
过期时间：7200
内容长度：0

24.2 会话设置

此示例包含示例会话设置的完整细节
在4节中。消息流程如图1所示。注意,
这些流显示了所需的最小头字段集 - 一些
其他头部字段，如 Allow 和 Supported，通常会是
当前。

F1 邀请 Alice -> atlanta.com 代理

邀请 sip:bob@biloxi.com SIP/2.0 Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9
hG4bKnashds8 Max-Forwards: 70 To: Bob <sip:bob@biloxi.com> From: Alice <
sip:alice@atlanta.com>;tag=1928301774 Call-ID: a84b4c76e66710 CSeq: 31415
9 INVITE Contact: <sip:alice@pc33.atlanta.com> Content-Type: application/sdp
Content-Length: 142

(爱丽丝的SDP未显示)

F2 100 尝试 atlanta.com 代理 -> 爱丽丝

SIP/2.0 100 尝试 Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bKnashds8 ;
received=192.0.2.1 To: Bob <sip:bob@biloxi.com> From: Alice <sip:alice@atlan
ta.com>;tag=1928301774 Call-ID: a84b4c76e66710 CSeq: 314159 INVITE 内容
长度: 0

F3 邀请 atlanta.com 代理 -> biloxi.com p roxy

邀请 sip:bob@biloxi.com SIP/2.0 Via: SIP/2.0/UDP bigbox3.site3.atlanta.com;branch=z9hG4bK77ef4c23
12983.1 Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bKnashds8 ;received=192.0.2.1 Max-Forwards:
69 To: Bob <sip:bob@biloxi.com> From: Alice <sip:alice@atlanta.com>;tag=1928301774 Call-ID: a84b4
c76e66710 CSeq: 314159 INVITE Contact: <sip:alice@pc33.atlanta.com> Content-Type: application/sdp
Content-Length: 142

(爱丽丝的SDP未显示)

F4 100 尝试 biloxi.com 代理 -> atlanta.com 代理

SIP/2.0 100 尝试 Via: SIP/2.0/UDP bigbox3.site3.atlanta.com;branch=z9hG4bK77ef4c2312983.1 ;receive
d=192.0.2.2 Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bKnashds8 ;received=192.0.2.1 To: Bob <s
ip:bob@biloxi.com> From: Alice <sip:alice@atlanta.com>;tag=1928301774 Call-ID: a84b4c76e66710 CS
eq: 314159 INVITE Content-Length: 0

F5 邀请 biloxi.com 代理 -> Bob

邀请 sip:bob@192.0.2.4 SIP/2.0 Via: SIP/2.0/UDP server10.biloxi.com;branch=z9hG4bK4b43c2ff8.1 Via: SIP/2.0/UDP bigbox3.site3.atlanta.com;branch=z9hG4bK77ef4c2312983.1 ;received=192.0.2.2 Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bKnashds8 ;received=192.0.2.1 Max-Forwards: 68 To: Bob <sip:bob@biloxi.com> From: Alice <sip:alice@atlanta.com>;tag=1928301774 Call-ID: a84b4c76e66710 CSeq: 314159 邀请 Contact: <sip:alice@pc33.atlanta.com> Content-Type: application/sdp Content-Length: 142

(爱丽丝的SDP未显示)

F6 180 铃声鲍勃 -> biloxi.com 代理

SIP/2.0 180 铃声 Via: SIP/2.0/UDP server10.biloxi.com;branch=z9hG4bK4b43c2ff8.1 ;received=192.0.2.3 Via: SIP/2.0/UDP bigbox3.site3.atlanta.com;branch=z9hG4bK77ef4c2312983.1 ;received=192.0.2.2 Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bKnashds8 ;received=192.0.2.1 To: Bob <sip:bob@biloxi.com>;tag=a6c85cf From: Alice <sip:alice@atlanta.com>;tag=1928301774 Call-ID: a84b4c76e66710 Contact: <sip:bob@192.0.2.4> CSeq: 314159 INVITE 内容长度: 0

F7 180 响铃 biloxi.com 代理 -> atlanta.com 代理

SIP/2.0 180 铃声 Via: SIP/2.0/UDP bigbox3.site3.atlanta.com;branch=z9hG4bK77ef4c2312983.1 ;received=192.0.2.2 Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bKnashds8 ;received=192.0.2.1 To: Bob <sip:bob@biloxi.com>;tag=a6c85cf From: Alice <sip:alice@atlanta.com>;tag=1928301774 Call-ID: a84b4c76e66710 Contact: <sip:bob@192.0.2.4> CSeq: 314159 INVITE 内容长度: 0

F8 180 铃声 atlanta.com 代理 -> 爱丽丝

SIP/2.0 180 铃声 Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bKnashds8 ;
received=192.0.2.1 To: Bob <sip:bob@biloxi.com>;tag=a6c85cf From: Alice <si
p:alice@atlanta.com>;tag=1928301774 Call-ID: a84b4c76e66710 Contact: <sip:b
ob@192.0.2.4> CSeq: 314159 INVITE 内容长度: 0

F9 200 OK Bob -> biloxi.com 代理

SIP/2.0 200 OK Via: SIP/2.0/UDP server10.biloxi.com;branch=z9hG4bK4b43c2ff8.1 ;received=192.0.2.3
Via: SIP/2.0/UDP bigbox3.site3.atlanta.com;branch=z9hG4bK77ef4c2312983.1 ;received=192.0.2.2 Via: S
IP/2.0/UDP pc33.atlanta.com;branch=z9hG4bKnashds8 ;received=192.0.2.1 To: Bob <sip:bob@biloxi.co
m>;tag=a6c85cf From: Alice <sip:alice@atlanta.com>;tag=1928301774 Call-ID: a84b4c76e66710 CSeq:
314159 INVITE Contact: <sip:bob@192.0.2.4> Content-Type: application/sdp Content-Length: 131

(鲍勃的SDP未显示)

F10 200 OK biloxi.com 代理 -> atlanta.com 代理

SIP/2.0 200 OK Via: SIP/2.0/UDP bigbox3.site3.atlanta.com;branch=z9hG4bK77ef4c2312983.1 ;received
=192.0.2.2 Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bKnashds8 ;received=192.0.2.1 To: Bob <si
p:bob@biloxi.com>;tag=a6c85cf From: Alice <sip:alice@atlanta.com>;tag=1928301774 Call-ID: a84b4c7
6e66710 CSeq: 314159 INVITE Contact: <sip:bob@192.0.2.4> Content-Type: application/sdp Content-Le
ngth: 131

(鲍勃的SDP未显示)

F11 200 OK atlanta.com 代理 -> 爱丽丝

SIP/2.0 200 OK Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bKnashds8 ;r
eceived=192.0.2.1 To: Bob <sip:bob@biloxi.com>;tag=a6c85cf From: Alice <sip
:alice@atlanta.com>;tag=1928301774 Call-ID: a84b4c76e66710 CSeq: 314159 I
NVITE Contact: <sip:bob@192.0.2.4> Content-Type: application/sdp Content-Le
ngth: 131

(鲍勃的SDP未显示)

F12 ACK 爱丽丝 -> 机器人

ACK sip:bob@192.0.2.4 SIP/2.0 Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9h
G4bKnashds9 Max-Forwards: 70 To: Bob <sip:bob@biloxi.com>;tag=a6c85cf Fr
om: Alice <sip:alice@atlanta.com>;tag=1928301774 Call-ID: a84b4c76e66710 C
Seq: 314159 ACK 内容长度: 0

媒体 s 会话在Alice和Bob之间现在已建立 已发布。

鲍勃先挂断。注意鲍勃的SIP电话维护其自己的CSeq
编号空格，在此示例中从231开始。由于Bob
正在发出请求时，To 和 From URI 以及标签已经被
交换了。

F13 BYE Bob -> Alice

再见 sip:alice@pc33.atlanta.com SIP/2.0 Via: SIP/2.0/UDP 192.0.2.4;br
anch=z9hG4bKnashds10 Max-Forwards: 70 From: Bob <sip:bob@biloxi
.com>;tag=a6c85cf To: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710 CSeq: 231 BYE Content-Length: 0

F14 200 OK Alice -> Bob

SIP/2.0 200 OK Via: SIP/2.0/UDP 192.0.2.4;branch=z9hG4bKnashds10
From: Bob <sip:bob@biloxi.com>;tag=a6c85cf To: Alice <sip:alice@atl
anta.com>;tag=1928301774 Call-ID: a84b4c76e66710 CSeq: 231 BYE
Content-Length: 0

SIP 通话流程文档 [40] 包含了更多关于 SIP 的示例
消息。

25 增强型巴科斯-诺尔范式（BNF）用于SIP协议

所有本文件中指定的机制均在以下描述中说明
两者都是散文和定义在RFC中的增强Backus-Naur形式（BNF）
2234 [10]。RFC 2234的第6.1节定义了一组核心规则,
此规范中已使用，此处不再重复。实现者
需要熟悉RFC 2234中的符号和内容
为了理解本规范。某些基本规则在
大写，例如 SP，LWS，HTAB，CRLF，DIGIT，ALPHA 等。 角度
括号在定义中使用以阐明规则的用法
名称。

方括号的使用在语法上是多余的。它被用作
语义提示，特定参数的使用是可选的。

25.1 基本规则

以下规则在本规范中始终使用。
描述基本解析构造。US-ASCII 编码字符集
由ANSI X3.4-1986定义。

字母数字 = 字母 / 数字

几个规则从RFC 2396 [5] 中纳入，但进行了更新。
使它们符合RFC 2234 [10]。这些包括：

```
reserved    =  ";" / "/" / "?" / ":" / "@" / "&" / "=" / "+"
               / "$" / ","
unreserved  =  alphanum / mark
mark        =  "-" / "_" / "." / "!" / "~" / "*" / "'"
               / "(" / ")"
escaped     =  "%" HEXDIG HEXDIG
```

SIP头字段值可以折叠到多行，如果
行续行以空格或水平制表符开始。所有线性
空格，包括折叠，与SP具有相同的语义。 A
接收者可以在单个SP之前替换任何线性空白
解释字段值或向下游转发消息。
这是旨在与RFC中描述的HTTP/1.1完全相同的行为
2616 [8]. SWS 结构用于线性空白时
可选，通常位于标记和分隔符之间。

```
LWS  = [*WSP CRLF] 1*WSP ; 线性空白
SWS  = [LWS] ; 分隔空格
```

为了将标题名称与值的其他部分分开，使用冒号，
哪个，根据上述规则，允许前面有空格，但不能有行
中断，以及之后的空白，包括换行符。 HCOLON
定义此结构。

```
HCOLON  =  *( SP / HTAB ) ":" SWS
```

TEXT-UTF8规则仅用于描述性字段内容
值，这些值不是旨在由消息解析器进行解释的。
单词 *TEXT-UTF8 包含来自 UTF-8 字符集（RFC）的字符
2279 [7]). TEXT-UTF8-TRIM 规则用于描述字段
内容为非引号字符串，其中包含前导和尾随空白字符
无意义。在这方面，SIP与使用{v*}的HTTP不同，
ISO 8859-1 字符集。

```
TEXT-UTF8-TRIM  = 1*TEXT-UTF8char *(*LWS TEXT-UTF8char)
文本-UTF8字符   = %x21-7E / UTF8-非ASCII
UTF8-NONASCII   = %xC0-DF 1UTF8-CONT
/  %xE0-EF 2UTF8-CONT
/  %xF0-F7 3UTF8-CONT
/  %xF8-Fb 4UTF8-CONT
/  %xFC-FD 5UTF8-CONT
UTF8-CONT     = %x80-BF
```

A CRLF 是允许在 TEXT-UTF8-TRIM 定义中作为部分出现的。
表头字段延续。 期望折叠空白字符（LWS）
将被替换为单个SP，在TEXT-解释之前
UTF8-TRIM 值。

十六进制数字字符用于多个协议元素。
某些元素（身份验证）强制十六进制字母必须小写。

```
LHEX  =  DIGIT / %x61-66 ;lowercase a-f
```

许多SIP头部字段值由LWS分隔的单词组成
特殊字符。除非另有说明，否则标记不区分大小写。
无反应。 这些特殊字符必须在引号字符串中才能
在参数值中使用。单词construct用于
调用-ID 以允许使用大多数分隔符。

```
token       =  1*(alphanum / "-" / "." / "!" / "%" / "*"
               / "_" / "+" / "`" / "'" / "~" )
separators  =  "(" / ")" / "<" / ">" / "@" /
               "," / ";" / ":" / "\" / DQUOTE /
               "/" / "[" / "]" / "?" / "=" /
               "{" / "}" / SP / HTAB
word        =  1*(alphanum / "-" / "." / "!" / "%" / "*" /
               "_" / "+" / "`" / "'" / "~" /
               "(" / ")" / "<" / ">" /
               ":" / "\" / DQUOTE /
               "/" / "[" / "]" / "?" /
               "{" / "}" )
```

当使用标记或元素之间使用分隔符时，
空格通常允许在这些字符前后：

STAR = SWS "*" SWS ; 星号
斜杠 = SWS "/" SWS ; 斜杠
相等 = SWS "=" SWS ; 相等
LPAREN = SWS "(" SWS ; 左括号
RPAREN = SWS ")" SWS ; 右括号
RAQUOT = ">" SWS ; 右引号
LAQUOT = SWS "<"; 左尖括号
逗号 = SWS "," SWS ; 逗号
SEMI = SWS ";" SWS ; 分号
冒号 = SWS ":" SWS ; 冒号
LDQUOT = SWS DQUOTE； 开双引号
RDQUOT = DQUOTE SWS ; 关闭双引号

注释可以通过将它们包围在空格中来包含在一些SIP头部字段中
注释文本带有括号。注释仅允许在字段中使用
包含作为其字段值定义一部分的 "comment"。在所有
其他字段，括号被视为字段值的一部分。

```
comment  =  LPAREN *(ctext / quoted-pair / comment) RPAREN
ctext    =  %x21-27 / %x2A-5B / %x5D-7E / UTF8-NONASCII
            / LWS
```

ctext 包含所有字符，除了左右括号和反斜杠。
一个文本字符串如果用引号引用，则被视为一个单词。
双引号。在引号字符串中，引号（") 和
反斜杠（\）需要转义。

```
quoted-string  =  SWS DQUOTE *(qdtext / quoted-pair ) DQUOTE
qdtext         =  LWS / %x21 / %x23-5B / %x5D-7E
                  / UTF8-NONASCII
```

反斜杠字符（"\")可以作为一个单字符使用
引用机制仅在引号字符串和注释结构中有效。
与HTTP/1.1不同，字符CR和LF不能通过这种方式转义
机制以避免与行折叠和标题分隔冲突。

引用对 = "\" (%x00-09 / %x0B-0C / %x0E-7F)

SIP-URI = "sip:" [userinfo] hostport uri-parameters [headers] SIPS-URI = "sips:
" [userinfo] hostport uri-parameters [headers] userinfo = (user / telephone-subscriber)
[":" password] "@" user = 1*(unreserved / escaped / user-unreserved) user-unreserved = "&" / "
=" / "+" / "$" / "," / ";" / "?" / "/" password = *(unreserved / escaped / "&" / "=" / "+" / "$
" / ",") hostport = host [":" port] host = hostname / IPv4address / IPv6reference hostname
= *(domainlabel ".") toplabel ["."] domainlabel = alphanum / alphanum *(alphanum / "-
") alphanum toplabel = ALPHA / ALPHA *(alphanum / "-") alphanum

```
IPv4address    =  1*3DIGIT "." 1*3DIGIT "." 1*3DIGIT "." 1*3DIGIT
IPv6reference  =  "[" IPv6address "]"
IPv6address    =  hexpart [ ":" IPv4address ]
hexpart        =  hexseq / hexseq "::" [ hexseq ] / "::" [ hexseq ]
hexseq         =  hex4 *( ":" hex4)
hex4           =  1*4HEXDIG
port           =  1*DIGIT
```

The BNF for telephone-subscriber can be found in RFC 2806 [9]. Note, however, that any characters allowed there that are not allowed in the user part of the SIP URI MUST be escaped.

```
uri-parameters    =  *( ";" uri-parameter)
uri-parameter     =  transport-param / user-param / method-param
                     / ttl-param / maddr-param / lr-param / other-param
transport-param   =  "transport="
                     ( "udp" / "tcp" / "sctp" / "tls"
                     / other-transport)
other-transport   =  token
user-param        =  "user=" ( "phone" / "ip" / other-user)
other-user        =  token
method-param      =  "method=" Method
ttl-param         =  "ttl=" ttl
maddr-param       =  "maddr=" host
lr-param          =  "lr"
other-param       =  pname [ "=" pvalue ]
pname             =  1*paramchar
pvalue            =  1*paramchar
paramchar         =  param-unreserved / unreserved / escaped
param-unreserved  =  "[" / "]" / "/" / ":" / "&" / "+" / "$"

headers         =  "?" header *( "&" header )
header          =  hname "=" hvalue
hname           =  1*( hnv-unreserved / unreserved / escaped )
hvalue          =  *( hnv-unreserved / unreserved / escaped )
hnv-unreserved  =  "[" / "]" / "/" / "?" / ":" / "+" / "$"

SIP-message    =  Request / Response
Request        =  Request-Line
                  *( message-header )
                  CRLF
                  [ message-body ]
Request-Line   =  Method SP Request-URI SP SIP-Version CRLF
Request-URI    =  SIP-URI / SIPS-URI / absoluteURI
absoluteURI    =  scheme ":" ( hier-part / opaque-part )
hier-part      =  ( net-path / abs-path ) [ "?" query ]
net-path       =  "//" authority [ abs-path ]
abs-path       =  "/" path-segments
```

不透明部分 = 尿素无斜杠 *尿素 尿素 = 保留的 / 未保留的 / 转义尿素无斜杠 = 未保留的 / 转义的 / ";" / "?" / ":" / "@" / "&" / "=" / "+" / "$" / "," 路径段 = 段 *("/" 段) 段 = *pchar *(";" 参数) 参数 = *pchar pchar = 未保留的 / 转义的 / ":" / "@" / "&" / "=" / "+" / "$" / "," 方案 = ALPHA *(ALPHA / DIGIT / "+" / "-" / ".") 权限 = srvr / reg-name srvr = [[用户信息 "@"] 主机端口] reg-name = 1*(未保留的 / 转义的 / "$" / "," / ";" / ":" / "@" / "&" / "=" / "+") 查询 = *uric SIP版本 = "SIP" "/" 1*DIGIT "." 1*DIGIT 消息头 = (接受 / 接受编码 / 接受语言 / 警报信息 / 允许 / 认证信息 / 授权 / 通话ID / 通话信息 / 联系 / 内容处置 / 内容编码 / 内容语言 / 内容长度 / 内容类型 / 顺序 / 日期 / 错误信息 / 过期 / 来自 / 回复至 / 最大转发次数 / MIME版本 / 最小过期时间 / 组织 / 优先级 / 代理认证 / 代理授权 / 代理要求 / 记录路由 / 回复至

/ 需要的 / Retry-After / 路由 / 服务器 / 主题 / 支持的 / 时间戳 / 发送到 / 不支持的 / 用户代理 / 通过 / 警告 / WWW-Authenticate / 扩展头) 回车换行 INVITEm = %x49.4E.56.49.54.45 ; INVITE 大写 ACKm = %x41.43.4B ; ACK 大写 OPTIONSm = %x4F.50.54.49.4F.4E.53 ; OPTIONS 大写 BYEm = %x42.59.45 ; BYE 大写 CANCELm = %x43.41.4E.43.45.4C ; CANCEL 大写 REGISTERm = %x52.45.47.49.53.54.45.52 ; REGISTER 大写 方法 = INVITEm / ACKm / OPTIONSm / BYEm / CANCELm / REGISTERm / 扩展方法 扩展方法 = 令牌 响应 = 状态行 *(消息头) 回车换行 [消息体] 状态行 = SIP版本 SP 状态码 SP 原因短语 回车换行 状态码 = 信息性 / 重定向 / 成功 / 客户端错误 / 服务器错误 / 全局失败 / 扩展码 扩展码 = 3个数字 原因短语 = *(保留 / 未保留 / 转义 / UTF8-NONASCII / UTF8-CONT / 空格 / 水平制表符) 信息性 = "100" ; 尝试 / "180" ; 铃声 / "181" ; 呼叫正在转发 / "182" ; 排队 / "183" ; 会话进行

```
Success  =  "200"  ;  OK

Redirection  =  "300"  ;  Multiple Choices
             /   "301"  ;  Moved Permanently
             /   "302"  ;  Moved Temporarily
             /   "305"  ;  Use Proxy
             /   "380"  ;  Alternative Service

Client-Error  =  "400"  ;  Bad Request
              /   "401"  ;  Unauthorized
              /   "402"  ;  Payment Required
              /   "403"  ;  Forbidden
              /   "404"  ;  Not Found
              /   "405"  ;  Method Not Allowed
              /   "406"  ;  Not Acceptable
              /   "407"  ;  Proxy Authentication Required
              /   "408"  ;  Request Timeout
              /   "410"  ;  Gone
              /   "413"  ;  Request Entity Too Large
              /   "414"  ;  Request-URI Too Large
              /   "415"  ;  Unsupported Media Type
              /   "416"  ;  Unsupported URI Scheme
              /   "420"  ;  Bad Extension
              /   "421"  ;  Extension Required
              /   "423"  ;  Interval Too Brief
              /   "480"  ;  Temporarily not available
              /   "481"  ;  Call Leg/Transaction Does Not Exist
              /   "482"  ;  Loop Detected
              /   "483"  ;  Too Many Hops
              /   "484"  ;  Address Incomplete
              /   "485"  ;  Ambiguous
              /   "486"  ;  Busy Here
              /   "487"  ;  Request Terminated
              /   "488"  ;  Not Acceptable Here
              /   "491"  ;  Request Pending
              /   "493"  ;  Undecipherable

Server-Error  =  "500"  ;  Internal Server Error
              /   "501"  ;  Not Implemented
              /   "502"  ;  Bad Gateway
              /   "503"  ;  Service Unavailable
              /   "504"  ;  Server Time-out
              /   "505"  ;  SIP Version not supported
              /   "513"  ;  Message Too Large
```

全局失败 = "600" ; 任何地方忙碌 / "603" ; 拒绝 / "604" ; 任何地方不存在
/ "606" ; 不可接受

接受 = "接受" HCOLON [接受范围 *(COMMA 接受范围)] 接受范围 = 媒体范围 *(SEMI 接受参数) 媒体范围 = ("*/*" / (m-type SLASH "*") / (m-type SLASH m-subtype)) *(SEMI m-参数) 接受参数 = ("q" 等于 qvalue) / 通用参数 qvalue = ("0" ["." 0*3DIGIT]) / ("1" ["." 0*3("0")]) 通用参数 = 标识符 [等于 通用值] 通用值 = 标识符 / 主机 / 引号字符串 接受编码 = "接受编码" HCOLON [编码 *(COMMA 编码)] 编码 = 编码 *(SEMI 接受参数) 编码 = 内容编码 / "*" 内容编码 = 标识符 接受语言 = "接受语言" HCOLON [语言 *(COMMA 语言)] 语言 = 语言范围 *(SEMI 接受参数) 语言范围 = ((1*8ALPHA *("-" 1*8ALPHA)) / "*")

警报信息 = "警报信息" HCOLON 警报参数 *(逗号 警报参数) 警报参数 = LAQUOT 绝对URI RAQUOT *(分号 通用参数)

允许 = "允许" HCOLON [方法 *(COMMA 方法)] 授权 = "授权" HCOLON 凭证 凭证 = ("摘要" LWS 摘要响应) / 其他响应 摘要响应 = dig-resp *(COMMA dig-resp) dig-resp = 用户名 / 区域 / 随机数 / 摘要URI / dresponse / 算法 / cnonce / 不透明 / 消息-qop / 随机数计数 / 认证参数 用户名 = "用户名" EQUAL 用户名值 用户名值 = 引号字符串 摘要URI = "uri" EQUAL LDQUOT 摘要URI值 RDQUOT 摘要URI值 = rquest-uri ; 等于HTTP/1.1消息中指定的request-uri message-qop = "qop" EQUAL qop值

```
cnonce            =  "cnonce" EQUAL cnonce-value
cnonce-value      =  nonce-value
nonce-count       =  "nc" EQUAL nc-value
nc-value          =  8LHEX
dresponse         =  "response" EQUAL request-digest
request-digest    =  LDQUOT 32LHEX RDQUOT
auth-param        =  auth-param-name EQUAL
                     ( token / quoted-string )
auth-param-name   =  token
other-response    =  auth-scheme LWS auth-param
                     *(COMMA auth-param)
auth-scheme       =  token

Authentication-Info  =  "Authentication-Info" HCOLON ainfo
                        *(COMMA ainfo)
ainfo                =  nextnonce / message-qop
                         / response-auth / cnonce
                         / nonce-count
nextnonce            =  "nextnonce" EQUAL nonce-value
response-auth        =  "rspauth" EQUAL response-digest
response-digest      =  LDQUOT *LHEX RDQUOT

Call-ID  =  ( "Call-ID" / "i" ) HCOLON callid
callid   =  word [ "@" word ]

Call-Info   =  "Call-Info" HCOLON info *(COMMA info)
info        =  LAQUOT absoluteURI RAQUOT *( SEMI info-param)
info-param  =  ( "purpose" EQUAL ( "icon" / "info"
               / "card" / token ) ) / generic-param

Contact        =  ("Contact" / "m" ) HCOLON
                  ( STAR / (contact-param *(COMMA contact-param)))
contact-param  =  (name-addr / addr-spec) *(SEMI contact-params)
name-addr      =  [ display-name ] LAQUOT addr-spec RAQUOT
addr-spec      =  SIP-URI / SIPS-URI / absoluteURI
display-name   =  *(token LWS)/ quoted-string

contact-params     =  c-p-q / c-p-expires
                      / contact-extension
c-p-q              =  "q" EQUAL qvalue
c-p-expires        =  "expires" EQUAL delta-seconds
contact-extension  =  generic-param
delta-seconds      =  1*DIGIT

Content-Disposition   =  "Content-Disposition" HCOLON
                         disp-type *( SEMI disp-param )
disp-type             =  "render" / "session" / "icon" / "alert"
                         / disp-extension-token
```

```
disp-param            =  handling-param / generic-param
handling-param        =  "handling" EQUAL
                         ( "optional" / "required"
                         / other-handling )
other-handling        =  token
disp-extension-token  =  token

Content-Encoding  =  ( "Content-Encoding" / "e" ) HCOLON
                     content-coding *(COMMA content-coding)

Content-Language  =  "Content-Language" HCOLON
                     language-tag *(COMMA language-tag)
language-tag      =  primary-tag *( "-" subtag )
primary-tag       =  1*8ALPHA
subtag            =  1*8ALPHA

Content-Length  =  ( "Content-Length" / "l" ) HCOLON 1*DIGIT
Content-Type     =  ( "Content-Type" / "c" ) HCOLON media-type
media-type       =  m-type SLASH m-subtype *(SEMI m-parameter)
m-type           =  discrete-type / composite-type
discrete-type    =  "text" / "image" / "audio" / "video"
                    / "application" / extension-token
composite-type   =  "message" / "multipart" / extension-token
extension-token  =  ietf-token / x-token
ietf-token       =  token
x-token          =  "x-" token
m-subtype        =  extension-token / iana-token
iana-token       =  token
m-parameter      =  m-attribute EQUAL m-value
m-attribute      =  token
m-value          =  token / quoted-string

CSeq  =  "CSeq" HCOLON 1*DIGIT LWS Method

Date          =  "Date" HCOLON SIP-date
SIP-date      =  rfc1123-date
rfc1123-date  =  wkday "," SP date1 SP time SP "GMT"
date1         =  2DIGIT SP month SP 4DIGIT
                 ; day month year (e.g., 02 Jun 1982)
time          =  2DIGIT ":" 2DIGIT ":" 2DIGIT
                 ; 00:00:00 - 23:59:59
wkday         =  "Mon" / "Tue" / "Wed"
                 / "Thu" / "Fri" / "Sat" / "Sun"
month         =  "Jan" / "Feb" / "Mar" / "Apr"
                 / "May" / "Jun" / "Jul" / "Aug"
                 / "Sep" / "Oct" / "Nov" / "Dec"

Error-Info  =  "Error-Info" HCOLON error-uri *(COMMA error-uri)
```

```
error-uri   =  LAQUOT absoluteURI RAQUOT *( SEMI generic-param )

Expires     =  "Expires" HCOLON delta-seconds
From        =  ( "From" / "f" ) HCOLON from-spec
from-spec   =  ( name-addr / addr-spec )
               *( SEMI from-param )
from-param  =  tag-param / generic-param
tag-param   =  "tag" EQUAL token

In-Reply-To  =  "In-Reply-To" HCOLON callid *(COMMA callid)

Max-Forwards  =  "Max-Forwards" HCOLON 1*DIGIT

MIME-Version  =  "MIME-Version" HCOLON 1*DIGIT "." 1*DIGIT

Min-Expires  =  "Min-Expires" HCOLON delta-seconds

Organization  =  "Organization" HCOLON [TEXT-UTF8-TRIM]

Priority        =  "Priority" HCOLON priority-value
priority-value  =  "emergency" / "urgent" / "normal"
                  / "non-urgent" / other-priority
other-priority  =  token

Proxy-Authenticate  =  "Proxy-Authenticate" HCOLON challenge
challenge           =  ("Digest" LWS digest-cln *(COMMA digest-cln))
                       / other-challenge
other-challenge     =  auth-scheme LWS auth-param
                       *(COMMA auth-param)
digest-cln          =  realm / domain / nonce
                        / opaque / stale / algorithm
                        / qop-options / auth-param
realm               =  "realm" EQUAL realm-value
realm-value         =  quoted-string
domain              =  "domain" EQUAL LDQUOT URI
                       *( 1*SP URI ) RDQUOT
URI                 =  absoluteURI / abs-path
nonce               =  "nonce" EQUAL nonce-value
nonce-value         =  quoted-string
opaque              =  "opaque" EQUAL quoted-string
stale               =  "stale" EQUAL ( "true" / "false" )
algorithm           =  "algorithm" EQUAL ( "MD5" / "MD5-sess"
                       / token )
qop-options         =  "qop" EQUAL LDQUOT qop-value
                       *("," qop-value) RDQUOT
qop-value           =  "auth" / "auth-int" / token

Proxy-Authorization  =  "Proxy-Authorization" HCOLON credentials
```

```
Proxy-Require  =  "Proxy-Require" HCOLON option-tag
                  *(COMMA option-tag)
option-tag     =  token

Record-Route  =  "Record-Route" HCOLON rec-route *(COMMA rec-route)
rec-route     =  name-addr *( SEMI rr-param )
rr-param      =  generic-param

Reply-To      =  "Reply-To" HCOLON rplyto-spec
rplyto-spec   =  ( name-addr / addr-spec )
                 *( SEMI rplyto-param )
rplyto-param  =  generic-param
Require       =  "Require" HCOLON option-tag *(COMMA option-tag)

Retry-After  =  "Retry-After" HCOLON delta-seconds
                [ comment ] *( SEMI retry-param )

retry-param  =  ("duration" EQUAL delta-seconds)
                / generic-param

Route        =  "Route" HCOLON route-param *(COMMA route-param)
route-param  =  name-addr *( SEMI rr-param )

Server           =  "Server" HCOLON server-val *(LWS server-val)
server-val       =  product / comment
product          =  token [SLASH product-version]
product-version  =  token

Subject  =  ( "Subject" / "s" ) HCOLON [TEXT-UTF8-TRIM]

Supported  =  ( "Supported" / "k" ) HCOLON
              [option-tag *(COMMA option-tag)]

Timestamp  =  "Timestamp" HCOLON 1*(DIGIT)
               [ "." *(DIGIT) ] [ LWS delay ]
delay      =  *(DIGIT) [ "." *(DIGIT) ]

To        =  ( "To" / "t" ) HCOLON ( name-addr
             / addr-spec ) *( SEMI to-param )
to-param  =  tag-param / generic-param

Unsupported  =  "Unsupported" HCOLON option-tag *(COMMA option-tag)
User-Agent   =  "User-Agent" HCOLON server-val *(LWS server-val)
```

```
Via               =  ( "Via" / "v" ) HCOLON via-parm *(COMMA via-parm)
via-parm          =  sent-protocol LWS sent-by *( SEMI via-params )
via-params        =  via-ttl / via-maddr
                     / via-received / via-branch
                     / via-extension
via-ttl           =  "ttl" EQUAL ttl
via-maddr         =  "maddr" EQUAL host
via-received      =  "received" EQUAL (IPv4address / IPv6address)
via-branch        =  "branch" EQUAL token
via-extension     =  generic-param
sent-protocol     =  protocol-name SLASH protocol-version
                     SLASH transport
protocol-name     =  "SIP" / token
protocol-version  =  token
transport         =  "UDP" / "TCP" / "TLS" / "SCTP"
                     / other-transport
sent-by           =  host [ COLON port ]
ttl               =  1*3DIGIT ; 0 to 255

Warning        =  "Warning" HCOLON warning-value *(COMMA warning-value)
warning-value  =  warn-code SP warn-agent SP warn-text
warn-code      =  3DIGIT
warn-agent     =  hostport / pseudonym
                  ;  the name or pseudonym of the server adding
                  ;  the Warning header, for use in debugging
warn-text      =  quoted-string
pseudonym      =  token

WWW-Authenticate  =  "WWW-Authenticate" HCOLON challenge

extension-header  =  header-name HCOLON header-value
header-name       =  token
header-value      =  *(TEXT-UTF8char / UTF8-CONT / LWS)
message-body  =  *OCTET
```

26 安全考虑：威胁模型和安全使用 建议

SIP不是一种容易保障安全的协议。它使用中介，
其多方面的信任关系，其预期使用之间
元素完全不信任，以及其用户间操作使
安全性远非平凡。需要的是安全解决方案，它们是
今日可部署，无需广泛协调，适用于广泛领域
环境与用法。为了满足这些多样化的需求，
几个适用于不同方面的不同机制
SIP的使用将是必需的。

请注意，SIP信令本身的安全性对此没有影响。
协议安全，如与SIP一起使用的RTP或与
任何特定实体的SIP可能带来的安全影响
尽管MIME安全在确保SIP安全方面发挥着重要作用。
任何与会议相关的媒体都可以进行端到端加密
独立于任何相关的SIP信令。媒体加密是
此文档范围之外。

以下考虑首先考察一组经典威胁
模型广泛识别SIP的安全需求。其集合为
安全服务需要解决这些威胁的细节随后详细说明，
随后是关于几种安全机制的说明
用于提供这些服务。接下来，对{v*}的要求为
SIP实现者被列举，以及示例部署
在哪些安全机制可用于改进的
SIP的安全性。一些关于隐私的注释结束本节。

26.1 攻击和威胁模型

本节详细介绍了大多数情况下都应存在的某些威胁
SIP的部署。这些威胁是特意选择的，以
说明SIP所需的所有安全服务。

以下示例绝对不能提供详尽的列表。
威胁针对SIP；相反，这些是“经典”威胁
展示特定安全服务的必要性，这些服务能够
可能预防整个威胁类别的出现。

这些攻击假设一个攻击者可以的环境
可能读取网络上的任何数据包 - 预计
SIP 将经常在公共互联网上使用。攻击者将
网络可能能够修改数据包（可能在某些受损的
中间人)。攻击者可能希望窃取服务，窃听{v*}。
通信，或中断会话。

26.1.1 注册劫持

SIP注册机制允许用户代理识别自身
至登记机构作为设备，其中用户（由地址指定）
记录）的位置。一个注册员评估所声明的身份。
从REGISTER消息的From头部字段确定是否
请求可以修改与以下相关的联系地址
记录地址在“收件人”标题字段中。虽然这两个字段是
经常相同，有许多有效的部署，其中包含一个 {v*}
第三方可能代表用户注册联系信息。

SIP请求的“ From ”报头字段，然而，可以被修改
任意由UA的所有者决定，这为
恶意注册。一个成功冒充的攻击者
一个授权更改与地址关联的联系人的党派
记录可以，例如，注销所有现有的联系人
URI 然后将其自己的设备注册为适当的联系人
地址，从而将所有受影响用户的请求定向到
攻击者的设备。

这个威胁属于一个依赖于缺失的威胁家族
加密保证请求发起者的来源。任何SIP UAS
表示一个有价值的服务（一个与SIP互操作的网关）
请求使用传统电话呼叫（例如）可能希望
通过验证请求来控制对其资源的访问
接收。即使最终用户代理，例如SIP电话，也有
对确定请求发起者身份的兴趣。

此威胁证明了需要能够启用安全服务的必要性
SIP实体用于验证请求的发起者。

26.1.2 伪装服务器

请求目标域通常在以下位置指定：
请求URI。用户代理通常在这个域中联系服务器
直接以发送请求。然而，始终存在
攻击者冒充远程服务器的可能性，以及
该UA的请求可能被其他方截获。

例如，考虑一个情况，其中一个重定向服务器位于一个
域名，chicago.com，冒充另一个地址的重定向服务器
域名，biloxi.com。一个用户代理向biloxi.com发送请求，但
芝加哥.com的重定向服务器以伪造的响应回答
具有适当SIP头部字段的响应
biloxi.com. 重定向响应中伪造的联系方式地址
可能导致发起UA访问不适当或不安全的内容
资源，或者简单地阻止对biloxi.com的请求成功。

这个威胁家族拥有庞大的成员，其中许多是
关键。 作为注册劫持威胁的反面，
考虑这种情况，向biloxi.com发送的注册信息
被chicago.com截获，并回复了被截获的
注册伪造的301（永久移动）响应。
响应可能看起来来自biloxi.com，但实际上指定为chicago.com
作为适当的注册机构。所有未来的注册请求来自
源自UA的则会前往chicago.com。

预防此威胁需要一种方法，通过这种方法 UAs 可以
验证他们发送请求的服务器。

26.1.3 消息体篡改

当然，SIP UAs 通过受信任的代理路由请求
服务器。无论该信任如何建立（身份验证
代理在此节的其他地方进行了讨论），一个UA可能会信任一个
代理服务器用于路由请求，但不检查或可能
修改该请求中包含的实体。

考虑一个使用SIP消息体进行会话通信的UA
加密密钥用于媒体会话。尽管它信任代理
服务器是它联系以正确传递信令的域名服务器，
它可能不想让该域的管理员能够
解密任何后续媒体会话。更糟糕的是，如果代理
服务器是积极恶意的，它可以修改会话密钥，
或者作为中间人，或者可能改变
请求的源UA的安全特性。

这个威胁家族不仅适用于会话密钥，还适用于大多数
可承载于SIP中的内容形式。这些可能
包含应呈现给用户的 MIME 主体，SDP，或
封装的电信信号，以及其他。攻击者可能会
尝试修改SDP主体，例如，以便指向RTP
媒体流到一个窃听设备上，以便进行窃听
后续语音通讯。

也请注意，SIP 中的一些头部字段具有有意义的端到端意义，
例如，主题。用户代理可能对这些头部字段持保护态度
以及主体（一个恶意的中介更改了主题）
表头字段可能会使一个重要的请求看起来像是垃圾邮件，对于
示例）。然而，由于许多头部字段是合法的
被代理服务器在请求路由过程中检查或修改，并非所有
标题字段应全程加密。

因此，UA可能希望确保SIP消息体安全，
在某些有限情况下，头部字段，端到端。安全性
服务对实体所需包括保密性、完整性和
身份验证。这些端到端服务应独立于
确保与中介机构等互动所使用的手段包括
代理服务器。

26.1.4 会话终止

一旦通过初始消息建立了对话，随后的
请求可以发送以修改对话的状态和/或
会话。确保会话中的负责人可以确信
此类请求不是由攻击者伪造的。

考虑一个第三方攻击者捕获一些初始
消息是两个当事人为了学习而共享的对话中的
会话参数（标记为、来自标记等）然后
将 BYE 请求插入到会话中。攻击者可以选择
锻造请求，使其看起来似乎来自以下任一
参与者。一旦BYE被其目标接收，会话
将提前拆除。

类似的半场威胁包括伪造的{v*}传输
邀请改变会话（可能为了降低会话安全性）
或作为窃听攻击的一部分重定向媒体流）。

此威胁的最有效对策是
身份验证 BYE 的发送者。在此情况下，
接收方只需要知道 BYE 是来自同一方的
与对应对话建立的人（与{v*}相对）
确定发送者的绝对身份）。此外，如果
攻击者无法学习会话的参数，因为
保密性，将无法伪造 BYE。然而，
一些中间代理（如代理服务器）将需要检查这些
参数在会话建立时。

26.1.5 服务拒绝和放大

拒绝服务攻击专注于使特定网络不可用
元素不可用，通常是通过指向过量的
网络接口处的网络流量。 分布式拒绝服务
攻击允许一个网络用户导致多个网络主机
用大量网络流量淹没目标主机。

在许多架构中，SIP代理服务器面向公共互联网
为了接受来自全球IP端点的请求。SIP创建了一个
潜在分布式拒绝服务攻击的机会数量
攻击必须由实施者识别和解决
SIP系统运营商。

攻击者可以创建包含伪造源头的虚假请求
IP地址和相应的Via头部字段，用于标识
目标主机作为请求的发起者，然后发送此
请求大量SIP网络元素，从而使用
不幸的SIP UAs或代理生成拒绝服务流量
针对目标。

同样，攻击者可能会在{v*}中使用伪造的路由报头字段值。
一个请求以识别目标主机然后发送此类消息
为放大发送到目标的消息的分叉代理。

当攻击者时，Record-Route可以产生类似的效果
确定由请求发起的SIP对话将导致
众多来自反向方向的交易。

如果注册请求被拒绝，将打开一系列拒绝服务攻击
未经注册机构正确认证和授权。
攻击者可能会注销部分或所有用户在管理
域，从而防止这些用户被邀请到新的
会话。攻击者还可以注册大量联系人
指定给定的记录地址的同一主机
使用注册机构和任何相关代理服务器作为放大器在
拒绝服务攻击。攻击者还可能试图耗尽
可用的内存和磁盘资源通过注册来获取注册机构
大量绑定。

SIP请求使用多播进行传输可以大大增加
拒绝服务攻击的潜在可能性。

这些问题表明了定义架构的一般需求
最小化拒绝服务风险和需要
谨慎推荐此类安全机制的建议
攻击。

26.2 安全机制

从上述威胁中，我们了解到基本
SIP协议所需的安全服务包括：保留{v*}
保密性和消息完整性，防止重放攻击
或消息欺骗，提供认证和隐私保障
会话中的参与者，以及防止拒绝服务
攻击。SIP消息中的主体分别需要安全
保密、完整性和认证服务。

而不是定义针对SIP的特定安全机制，SIP
尽可能重用从以下方面派生的现有安全模型：{v*}
HTTP 和 SMTP 空间。

消息的完全加密是保护信息最佳的手段
信号机密性 - 它也可以保证消息
未被任何恶意中间代理修改。然而，SIP
请求和响应不能天真地端到端加密
他们全部，因为消息字段如请求URI、路由
并且 Via 需要在大多数网络架构中对代理可见
确保SIP请求正确路由。注意代理服务器
需要修改消息的一些功能（例如添加 Via
标题字段值）以便SIP能够运行。代理服务器
必须因此在一定程度上信任SIP UAs。
目的，建议采用低层安全机制用于SIP

加密整个SIP请求或响应在链路上的跳转
通过跳转基，并允许端点验证身份
代理服务器，他们向其发送请求。

SIP实体也需要在安全的环境中相互识别
时尚。当SIP端点向其用户声明其身份时，
对等UA或代理服务器，该身份应以某种方式被
可验证的。在{v*}中提供了一个密码认证机制。
SIP 以满足这一需求。

一个独立的SIP消息体安全机制提供
替代端到端相互认证的方法，以及
提供对用户代理必须信任程度的限制
中介方。

26.2.1 传输和网络层安全

传输或网络层安全加密信令流量，
保证消息的机密性和完整性。

通常，证书用于建立底层
安全，并且这些证书也可以用来提供一种手段
许多架构中的认证。

两种在运输和提供安全性的流行替代方案
网络层分别是TLS [25]和IPSec [26]。

IPSec 是一组网络层协议工具，共同可以
用作传统IP（互联网协议）的安全替代品
协议）。IPSec最常用于哪种架构中，其中{v*}
主机集或管理域已存在信任
彼此之间的关系。IPSec通常在{v*}处实现。
操作系统级别在主机上，或在安全网关上
为所有接收到的流量提供保密性和完整性
从特定的接口（如在VPN架构中）。IPSec可以
也可以按跳转方式使用。

在许多架构中，IPSec 不需要与SIP集成
应用；IPSec可能最适合部署在以下场景中
将安全性直接添加到SIP主机将是一项艰巨的任务。UAs
与他们的第一跳代理有预共享密钥关系
服务器也是使用IPSec的好候选者。任何部署
IPSec for SIP 需要一个描述协议的 IPSec 配置文件
工具，用于确保SIP的安全性。没有提供此类配置文件
在此文档中。

TLS在面向连接的连接上提供传输层安全
协议（本文件中指TCP）；"tls"（表示）
TLS over TCP) 可以指定为所需的传输协议
在 Via 头字段值或 SIP-URI 中。TLS 最适合
架构中需要在主机之间实现跳到跳的安全
没有预先存在的信任关联。例如，Alice信任
她的本地代理服务器，在证书交换后决定
信任鲍勃的本地代理服务器，鲍勃信任它，因此鲍勃和爱丽丝
可以安全通信。

TLS必须与SIP应用程序紧密耦合。 注意到
SIP中按跳转逐跳指定传输机制，因此
一个通过TLS向代理服务器发送请求的UA没有保证
TLS将端到端使用。

TLS_RSA_WITH_AES_128_CBC_SHA 加密套件 [6] 必须支持
在SIP应用程序中使用TLS时，实现者应实现的最低要求。
向后兼容的目的、代理服务器、重定向服务器、
并且注册机构应支持 TLS_RSA_WITH_3DES_EDE_CBC_SHA。
实现者也可以支持任何其他密钥套件。

26.2.2 SIPS URI 方案

SIPS URI方案遵循SIP URI（描述如下）的语法
在19)中，尽管方案字符串是"sips"而不是"sip"。
SIPS的语义与SIP URI非常不同。
允许资源指定它们应通过安全方式访问。

SIPS URI可以用作特定用户的记录地址
- 用户在业务中被规范识别的URI（在其业务上）
卡片，在其请求的“ From ”头字段中，在“ To ”头字段中
字段为REGISTER请求）。当用作请求URI时
请求，SIPS方案表示每个跳转都
请求被转发，直到请求达到SIP实体
负责 Request-URI 的域名部分，必须是
使用TLS加密；一旦到达相关域名，它就是
根据当地安全和路由策略处理，相当
可能使用TLS连接到UAS的任何最后一跳。当由
请求发起者（如果他们雇佣了SIPS）
URI作为目标记录的地址），SIPS规定
整个请求路径到目标域应如此安全。

SIPS方案适用于许多其他SIP的方式
统一资源标识符（URI）在SIP中今天除了请求URI外，还包括
在记录地址中，联系地址（联系内容）
表头，包括那些注册方法（REGISTER methods）的表头，以及路由表头。
每个实例，SIPS URI方案允许这些现有字段

指定安全资源。SIPS URI 的方式是
在以下任何上下文中解引用都具有其自身的安全属性
这些在[4]中有详细说明。

SIPS的特别使用意味着相互TLS认证
应当被采用，正如密钥套件应当被采用
TLS_RSA_WITH_AES_128_CBC_SHA. 收到的证书
身份验证过程应使用根证书进行验证
由客户端持有；未能验证证书应导致
在请求失败的情况下。

请注意，在SIPS URI方案中，传输与TLS独立，
并且因此 "sips:alice@atlanta.com;transport=tcp" 和
sips:alice@atlanta.com;transport=sctp 都是有效的（尽管
请注意，UDP不是SIPS的有效传输方式）。使用
"transport=tls" 因此已被弃用，部分原因是
它特定于请求的单跳。这是一个更改
自RFC 2543以来。

用户将SIPS URI作为记录地址分发时，可以选择
操作拒绝通过不安全传输请求的设备。

26.2.3 HTTP 认证

SIP提供基于HTTP认证的挑战功能，
依赖于401和407响应代码以及头部
字段用于承载挑战和凭证。 没有显著
修改，在{v*}中重新使用HTTP摘要认证方案
SIP允许回放保护和单向认证。

SIP中摘要认证的使用在第22节中详细说明。

26.2.4 S/MIME

如上所述，端到端加密整个SIP消息
保密的目的不适用于网络
中介（如代理服务器）需要查看某些头部
字段以正确路由消息，并且如果这些
中介被排除在安全关联之外，然后是SIP
消息将基本上是不可路由的。

然而，S/MIME 允许 SIP 用户代理在 SIP 中加密 MIME 主体，其中 {v*} 保持不变。
确保这些实体端到端安全，不影响消息头。
S/MIME可以为以下提供端到端机密性和完整性：
消息体，以及相互认证。它也是
可能使用S/MIME提供一种完整性形式
通过SIP消息隧道对SIP头部字段进行保密性处理。

S/MIME在SIP中的使用在第23节中详细说明。

26.3 实施安全机制

26.3.1 SIP实施者的要求

代理服务器、重定向服务器和注册机构必须实现TLS，
必须支持双向和单向认证。
强烈建议UA（用户代理）能够启动TLS；UA也可以
能够作为TLS服务器运行。代理服务器，重定向
服务器和注册机构应拥有一个网站证书，其
主题对应它们的规范主机名。UAs 可能有
证书用于TLS相互认证，但没有
本文件中规定了其使用条款。所有SIP
元素支持TLS必须有一个验证机制
证书在TLS协商期间收到；这涉及到拥有
由证书颁发机构签发的一个或多个根证书
(首选知名网站证书的分销商，可比拟
针对那些为网络浏览器颁发根证书的机构）。

所有支持TLS的SIP元素也必须支持SIPS URI
方案。

代理服务器、重定向服务器、注册商和用户代理（UAs）也可能
实现IPSec或其他底层安全协议。

当UA尝试联系代理服务器、重定向服务器时，
注册机构，UAC 应该通过它来初始化 TLS 连接
将发送SIP消息。在某些架构中，UASs可能会接收
请求此类TLS连接也是如此。

代理服务器、重定向服务器、注册商和用户代理（UAs）必须实现
摘要授权，包括22中所需的所有方面。
代理服务器、重定向服务器和注册机构应配置
至少包含一个Digest域，以及至少一个"realm"字符串
由给定服务器支持的 {v*} 应当对应于服务器的
主机名或域名。

UAs 可支持 MIME 体的签名和加密，并且
证书的S/MIME传输，如第23节所述。
如果UA持有一个或多个证书的根证书
为了验证TLS或IPSec的证书，当局需要这样做，它
应能够重新使用这些来验证S/MIME证书，如
适当。一个UA可以专门持有根证书，用于
验证S/MIME证书。

请注意，预计未来的安全扩展可能会
升级与S/MIME相关的规范性强度，作为S/MIME
实现出现，问题空间变得更好
理解了。

26.3.2 安全解决方案

这些安全机制协同运作可以遵循
现有的网络和电子邮件安全模型在一定程度上。在高度
等级，用户代理（UAs）向服务器（代理服务器）进行身份验证
重定向服务器和注册商）使用Digest用户名和
密码；服务器向一个跳步之外的UAs或向{v*}进行身份验证。
另一个跳数外的服务器（反之亦然），带有站点证书
通过TLS交付。

在点对点层面上，UAs信任网络以验证一个
另一个通常；然而，S/MIME也可以用来提供
直接认证，当网络不可用或网络不可用时
自身不受信任。

以下是一个说明性示例，其中包含这些安全
机制被各种UA和服务器用来防止各种类型
26.1节中描述的威胁。虽然实施者和网络
管理员可以遵循在以下规范指南中给出的
本节剩余部分，这些仅作为示例提供
实现。

26.3.2.1 注册

当UA上线并与其本地管理进行注册时
域名，它应与其注册商建立TLS连接
(第10节描述了UA如何到达其注册机构)。
注册机构应向UA提供证书，并站点
由证书标识的必须与域名相对应
UA打算注册的内容；例如，如果UA打算
注册记录地址 'alice@atlanta.com'，该站点
证书必须识别atlanta.com域名内的一个主机（例如
当它接收到TLS证书消息时，as sip.atlanta.com）。
UA 应该验证证书并检查所标识的网站
根据证书。如果证书无效、已撤销或如果它
未识别适当的当事人，UA不得发送
注册消息并继续进行注册。

当注册机构提供有效的证书时，
UA知道注册商不是一个可能重定向的攻击者
UA，窃取密码，或尝试任何类似攻击。

UA 然后创建一个应被发送到某个地址的 REGISTER 请求
请求URI对应于从网站证书接收到的
注册员。当UA通过现有{v*}发送REGISTER请求时
TLS连接，注册机构应使用401挑战请求
(代理身份验证所需) 响应。 "realm" 参数
响应的Proxy-Authenticate头字段中应当
对应于网站证书之前给出的域。
当UAC收到挑战时，它应该提示用户
为凭证或从密钥环中选取适当的凭证
对应挑战中的“ realm ”参数。
用户名应与“ userinfo ”对应
URI中To头字段中REGISTER请求的部分。
一旦将摘要凭据插入到适当的
代理授权报头字段，注册应重新提交
寄给登记员。

由于注册员要求用户代理进行验证
它本身，攻击者伪造 REGISTER 将会困难
请求用户的记录地址。同时请注意，由于
注册信息通过一个保密的TLS连接发送，攻击者
无法拦截以记录凭证的REGISTER
对于任何可能的回放攻击。

一旦注册被注册员接受，UA
应保持此TLS连接开启，前提是注册机构
也充当代理服务器，将请求发送给用户
此管理域。现有的TLS连接将
重新用于将传入请求发送到刚刚完成操作的UA
注册。

因为UA已经对另一边的服务器进行了认证
TLS连接的侧，所有通过此侧发送的请求
连接已知已通过代理服务器 -
攻击者无法创建看似已发送的伪造请求
通过那个代理服务器发送。

26.3.2.2 跨域请求

现在假设Alice的UA想与一个会话进行初始化。
用户在远程管理域中，即 "bob@biloxi.com"。我们
也将说明本地管理域（atlanta.com）有
本地出站代理。

代理服务器处理针对管理员的入站请求
域也可能充当本地出站代理；为了简便起见
我们将假设这是针对atlanta.com的情况（否则用户
代理将启动一个新的TLS连接到另一个服务器
此点）。假设客户端已完成注册

在上一节中描述的过程，它应该重用TLS
连接到本地代理服务器当它发送一个INVITE请求时
给另一个用户。 UA 应该重用缓存的凭证。
邀请以避免不必要地提示用户。

当地出站代理服务器验证了凭证时
由UA在INVITE中提出的，它应该检查Request-URI
确定消息应如何路由（见[4]）。
请求URI的"domainname"部分曾对应于本地
域名（atlanta.com）而不是biloxi.com，然后是代理服务器
将已咨询其位置服务以确定如何最佳地
达到请求的用户。

如果 "alice@atlanta.com" 正在尝试联系，比如说，
"alex@atlanta.com"，本地代理会代理到
请求与Alex建立的TLS连接
注册时。由于Alex会收到这个
请求通过他的认证通道，他将确信
爱丽丝的请求已被代理服务器授权
本地行政域。

然而，在这个例子中，Request-URI指定了一个远程域名。
atlanta.com 的本地出站代理服务器因此应该
建立与远程代理服务器上的TLS连接
biloxi.com. 由于此TLS连接中的两位参与者
是具有站点证书、相互TLS认证的服务器
应发生。连接的每一侧应验证和检查
其他证书，注明出现的域名
SIP头部字段的比较证书
消息。例如，atlanta.com代理服务器应验证
在此阶段，从远程端收到的证书
对应于biloxi.com域名。一旦完成，并且TLS
谈判已完成，结果在双方之间建立了一个安全通道
两个代理，atlanta.com代理可以将INVITE请求转发到
biloxi.com.

The proxy server at biloxi.com 应该检查 {v*} 的证书
代理服务器在atlanta.com上依次并比较声明的域名
通过具有“ 域名 ”部分的“ 发件人 ”头部的证书
字段在INVITE请求中。Biloxi代理可能具有严格的
安全策略要求拒绝不匹配的请求
他们被代理的行政域。

此类安全策略可以设立以防止SIP
SMTP '开放中继'的等效，这些中继经常被利用
生成垃圾邮件。

此政策，然而，仅保证请求来自
域它赋予自己；它不允许biloxi.com
确认atlanta.com如何验证了Alice。只有当biloxi.com
有其他方式了解atlanta.com的认证策略
可能确定爱丽丝是如何证明她的身份的。
biloxi.com 可能会实施更加严格的政策，禁止
请求来自行政上未知域名的请求
与biloxi.com共享一个共同的认证策略。

一旦INVITE已被biloxi代理批准，代理
服务器应识别任何现有的TLS通道，如果有的话
与该请求针对的用户（在这种情况下
"bob@biloxi.com")。 The INVITE 应该通过此通道代理
给Bob。 由于请求是通过TLS连接接收的，该连接具有
之前已验证为比洛克斯代理，鲍勃知道
从标题字段未被篡改，并且 atlanta.com 有
已验证Alice，尽管不一定是否信任
爱丽丝的身份。

在它们转发请求之前，两个代理服务器都应该添加
记录-路由报头字段到请求中，以便所有未来的请求
在这个对话框将通过代理服务器。 代理
服务器因此可以继续为以下提供安全服务
此对话的寿命。如果代理服务器没有将自己添加
到记录路由，未来的消息将直接端到端传输
Alice和Bob之间没有任何安全服务（除非两者
各方同意一些独立端到端的安全措施，例如
S/MIME）。在这方面，SIP 梯形模型可以提供一个很好的
结构，其中站点代理之间的协议惯例可以
提供Alice和Bob之间一个相对安全的通道。

一个针对此架构的攻击者，例如，将会
无法伪造 BYE 请求并将其插入到信令中
流在Bob和Alice之间，因为攻击者没有方法
确定会话参数以及因为 {v*}
完整性机制传递保护之间的流量
爱丽丝和鲍勃。

26.3.2.3 对等请求

或者，考虑一个断言身份的UA
"carol@chicago.com" 没有本地出站代理。 当 Carol
希望向 "bob@biloxi.com" 发送一个 INVITE，她的 UA 应该发起
一个使用biloxi代理直接（使用机制）的TLS连接
在[4]中描述了如何最佳地达到给定的
请求URI）。当她的UA从biloxi接收证书时
代理，它应该在通过她的INVITE之前正常验证
通过TLS连接。然而，Carol没有证明的方法

她的身份信息传给了比洛克斯代理，但她确实有一个CMS分离的
签名在INVITE中的“ 消息/sip ”体上。这不太可能
这个实例表明Carol在biloxi.com有任何凭证
领域，因为她与biloxi.com没有正式的联系。
比洛克斯代理可能也有一个严格的政策，禁止其
甚至麻烦挑战那些没有biloxi.com的请求
"From"报头字段中的 "domainname" 部分 - 它将这些
用户作为未认证的。

The biloxi代理为Bob有一个策略，即所有未认证的
请求应重定向到相应的联系地址
已注册于‘ bob@biloxi.com ’，即{s3}sip:bob@192.0.2.4{s4}。
Carol 通过 TLS 连接接收重定向响应
与biloxi代理建立，因此她相信其真实性
联系地址。

Carol 应该随后与指定的建立 TCP 连接
地址并发送一个新的INVITE请求，其中包含请求URI：{v*}
接收到的联系地址（在正文中重新计算签名作为
请求已准备）。Bob在一个不安全的连接上收到这个INVITE
接口，但他的UA检查并，在此实例中，识别出
从请求的报头字段中获取并随后匹配本地
缓存的证书与签名中呈现的证书一致
正文部分为INVITE。他以类似的方式回复，进行身份验证
他自己向Carol，并开始了一个安全的对话。

有时，管理域中的防火墙或NAT可能会
阻止建立到UA的直接TCP连接。
在这些情况下，代理服务器也可能潜在地中继请求
对UAs的方式，这种方式没有信任含义（例如，
放弃现有的TLS连接并转发请求
根据本地策略进行。

26.3.2.4 DoS 防护

为了最小化针对 {v*} 的拒绝服务攻击风险
架构使用这些安全解决方案的，实施者应
请注意以下指南。

当SIP代理服务器所在的主机是可路由的
从公共互联网，它应该部署在管理
域具有防御性操作策略（阻止源路由
交通，最好是过滤ping流量）。TLS和IPSec都可以
也利用行政域边缘的堡垒主机
参与安全关联以聚合安全的
隧道和插座。这些堡垒主机也可以承受大部分压力
拒绝服务攻击，确保SIP主机在
管理域不受多余消息的拖累。

无论部署何种安全解决方案，都会出现大量消息
针对代理服务器可能导致代理服务器资源锁定
阻止期望的交通到达目的地。 有一个
与处理SIP交易相关的计算开销
一个代理服务器，并且对于有状态的代理，这项费用更高
服务器比无状态代理服务器多。因此，有状态
代理比无状态代理更容易受到洪水攻击
服务器。

UAs 和代理服务器应向可疑请求进行挑战
仅一个401（未授权）或407（代理认证）
必需的），放弃正常响应重传算法，并且
因此对未认证的请求无状态地行为。

重新传输401（未授权）或407（代理认证）
(必需)状态响应放大了攻击者的问题
使用伪造的报头字段值（如Via）来引导
流量到第三方。

总结来说，通过代理服务器之间的相互认证
机制，如TLS，显著降低了恶意行为的发生可能性
中介引入伪造的请求或响应，这些可以
拒绝服务。这使得攻击者更难
将无辜的SIP节点变成放大器的代理。

26.4 局限性 {v*}

尽管这些安全机制，当谨慎应用时
方式，可以挫败许多威胁，但在范围上存在局限性
的机制必须被实施者和网络理解
操作符。

26.4.1 HTTP摘要

HTTP摘要在SIP中使用的一个主要限制是
摘要中的完整性机制对SIP不起作用。
具体来说，它们提供对Request-URI和方法的保护
消息中，但不适用于UAs会使用的任何标题字段
最有可能希望确保。

现有的RFC 2617中描述的重放保护机制也
SIP存在一些限制。下一个随机数机制，对于
示例，不支持管道请求。nonce-count
机制应用于回放保护。

HTTP摘要的另一个限制是域的范围。Digest是
当用户想要验证自己的身份时很有价值
与它们有预先存在的关联，例如一项服务

供应商的用户是客户（这相当常见）
场景因此Digest提供了一个极其有用的功能）。
与对比方式不同，TLS的作用范围是域间或多域，因为
证书通常具有全球可验证性，因此UA可以
验证服务器，无需预先存在的关联。

26.4.2 S/MIME

S/MIME机制最大的未解决缺陷是缺乏
一个普遍的面向最终用户的公钥基础设施。 如果自-
已签名的证书（或无法由一方验证的证书）
参与者（在对话中）使用时，基于SIP的关键交换
机制如第23.2节所述，易受中间人攻击
中间攻击，攻击者可以潜在地检查和
修改S/MIME正文。攻击者需要拦截第一个
双方在对话中交换密钥，删除
现有请求和响应中的CMS分离签名，以及
插入包含证书的不同CMS分离签名
由攻击者提供（但似乎是一张证书）
正确的记录地址）。每一方都会认为他们已经交换了
键与另一个键，实际上每个都拥有公钥
攻击者。

需要注意的是，攻击者只能利用这一点
漏洞出现在双方第一次交换密钥时 - 在
随后的场合，关键字的变更将明显
对UAs。它也难以让攻击者保持
所有未来双方对话随时间变化的路径（如
可能需要数天、数周或数年）。

SSH容易受到第一次中间人攻击的影响
密钥交换；然而，人们普遍认为，尽管SSH
不是完美的，但它确实提高了连接的安全性。 的使用
关键指纹可能为SIP提供一些帮助，就像它
对SSH执行。例如，如果双方使用SIP来建立
语音通信会话，每个都可以读取{v*}的指纹
他们从其他人那里收到的关键，可以与之进行比较
原文。 对于这个人来说，这肯定会更困难。
中间部分以模拟参与者的声音为主
信号（一种与基于Clipper芯片的实践相结合的做法）
安全电话）。

S/MIME机制允许UAs发送加密请求而不
序言 如果他们拥有目的地地址的证书-
在他们的钥匙环上的记录。然而，任何
特定设备注册的记录地址将不会保留
证书已被设备之前使用的
当前用户，因此将无法处理

加密请求正确，可能导致一些可避免的错误
信号。这尤其可能发生在加密请求时
分支

S/MIME 相关联的密钥在与其关联时最有用
特定用户（记录地址）而不是设备（用户代理）。
当用户在不同设备之间切换时，可能难以传输
私钥在UAs之间安全传输；这些密钥可能如何被获取
设备不在本文件的范围内。

另一个与S/MIME机制相关的更平凡的困难是
可能导致非常长的消息，尤其是在SIP隧道传输时
机制在23.4节中描述的是使用的。因此，它是
建议在以下情况下使用TCP作为传输协议：
S/MIME 隧道传输被采用。

26.4.3 TLS

关于TLS最常被提出的问题是其无法在{v*}上运行。
UDP；TLS需要基于连接的底层传输
协议，在本文件中指TCP。

它可能对于本地出口代理服务器和/或来说也很困难
注册商维护许多同时存在的长期TLS连接
具有众多UAs。这引发了一些有效的可扩展性问题，
特别适用于密集的加密套件。 维护冗余的
长期存在的TLS连接，尤其是当UA仅
负责其建立，也可能很繁琐。

TLS仅允许SIP实体对它们进行身份验证的服务器
相邻；TLS提供严格的跳-跳安全。TLS，
未指定本文件中的任何其他机制允许客户端
验证无法直接建立TCP连接的代理服务器
连接。

26.4.4 SIPS URI

实际上，在请求路径的每个段上使用TLS意味着
终止的UAS必须通过TLS可访问（可能需要注册
使用SIPS URI作为联系地址）。这是首选用法
SIPS. 许多有效的架构，然而，使用TLS来保护部分
请求路径，但依赖于某些其他机制进行最终跳转
到UAS，例如。因此SIPS不能保证TLS的使用
将真正实现端到端。请注意，由于许多用户代理（UAs）将不接受
进入的TLS连接，即使是那些支持TLS的UAs也可能
需要维持如所述的持久TLS连接
TLS限制部分上方以接收TLS上的请求
作为一个UAS。

位置服务无需提供 {v*} 绑定。
SIPS 请求 URI。尽管位置服务通常被填充
通过用户注册（如第10.2.1节所述），各种其他
协议和接口可能提供联系地址
对于AOR，并且这些工具可以自由地将SIPS URI映射到SIP URI
适当的。当查询绑定时，位置服务返回
其联系地址，不考虑是否接收了
请求带有SIPS请求URI。如果重定向服务器正在访问
位置服务，它取决于处理该实体的
联系重定向的头部字段以确定其适当性
联系地址。

确保所有请求段都将使用TLS，直到{v*}
目标域有些复杂。它可能存在以下情况：
加密认证的代理服务器沿途
不合规或受损的可以选择忽略转发
与SIPS相关的规则（以及一般转发规则）
第16.6节）。例如，这样的恶意中介可能可以，
重定向一个来自SIPS URI的请求到SIP URI，以尝试
降级安全。

或者，一个中介可能合法地重新定位一个请求
从SIP到SIPS URI。请求的请求URI的接收者
使用SIPS URI方案，因此不能基于此假设
仅请求URI，SIPS用于整个请求路径
(从客户端开始)。

为了解决这些担忧，建议接收者
请求其Request-URI包含SIP或SIPS URI的检查To
表头字段值以查看它是否包含一个SIPS URI（尽管请注意
如果此URI相同，则不构成安全违规
方案但与To头字段中的URI不等价）。
尽管客户端可以选择填充Request-URI和To头
请求字段不同，当使用SIPS时，这种差异
可能被解释为一种安全违规行为，并且
请求因此可能被其接收者拒绝。接收者
也可能检查Via头部链以进行双重确认
是否在整个请求路径中使用了TLS
本地行政域已到达。S/MIME也可能被用于
原始UAC以帮助确保原始表单的
报头字段是端到端传输的。

如果UAS有理由相信Request-URI的方案
已不正确地在传输过程中修改，UA 应通知其
潜在安全漏洞的用户。

作为进一步防止降级攻击的措施，实体需要
仅接受SIPS请求也可能拒绝不安全的连接
端口。

最终用户无疑会区分SIPS和
SIP URI，并且它们可以手动编辑它们以响应刺激。
这可能会提高或降低安全性。例如，如果一个
攻击者篡改DNS缓存，插入一个伪造的记录集，其中
有效地删除代理服务器的所有SIPS记录，然后任何
SIPS请求穿越此代理服务器可能会失败。当用户，
然而，看到对SIPS AOR的重复调用失败，他们
在某些设备上可以手动将方案从SIPS转换为SIP
当然，对此有一些安全措施（如果{v*}发生）。
目标 UA 实际上非常偏执，可能会拒绝所有非 SIPS
requests)，但这是一个值得注意的限制。在另一方面，
用户也可能推断出，即使在他们的情况下，'SIPS' 也是有效的
仅以SIP URI呈现。

26.5 隐私

SIP消息通常包含有关其{v*}的敏感信息
发送者 - 不仅在于他们要说什么，还在于他们与谁交流
沟通，当它们沟通以及沟通了多久，以及从哪里
他们参与会议。许多应用程序及其用户
要求此类个人信息对任何人都隐藏
各方无需了解的内容。

请注意，也存在一些不那么直接的方法，其中包含私有
信息可能被泄露。如果用户或服务选择
可从该人的姓名中猜出的地址中获取
组织归属（描述了大多数地址-）
记录），通过使用{v*}等传统方法来确保隐私
未列出的“ 电话号码 ”已泄露。用户位置服务可以
侵犯会话邀请接收者的隐私
泄露他们的具体位置给来电者；一个实现
因此 应该能够根据每个用户来限制
某种位置和可用信息被提供给某些
调用者类别。这是一个整体的问题类别，是
预期将在持续进行的SIP工作中进一步研究。

在某些情况下，用户可能希望在以下内容中隐藏个人信息：
标题字段，用于传达身份。这不仅可以应用于
从和表示请求发起者的相关头信息，
但也要注意，向最终用户传达 - 可能不合适
目标地址一个快速拨号昵称，或一个未展开的标识符
一组目标，其中任何一个都将从
请求URI作为请求路由，但在“ 收件人 ”中未改变

表头字段如果最初是相同的。因此它可能
出于隐私原因，希望创建一个To头字段
与请求URI不同。

27 IANA 考虑事项

所有方法名称、头部字段名称、状态码和选项标签
用于SIP应用程序的{v*}通过IANA进行注册
指示位于RFC中的IANA考虑部分。

规格指示IANA创建四个新的子-
注册表位于 http://www.iana.org/assignments/sip-parameters:
选项标签，警告代码（warn-codes），方法及响应代码，
已添加到已存在的头字段子注册表中
那里。

27.1 选项标签

本规范在以下位置建立了选项标签子注册表：
http://www.iana.org/assignments/sip-parameters.

选项标签用于如 Require、Supported 等头部字段中。
代理要求，不支持以支持SIP兼容性
机制扩展（第19.2节）。选项标签本身是
字符串与特定SIP选项相关（即，一个
扩展）。它标识了SIP端点的选项。

选项标签在它们发布时由IANA进行注册。
标准跟踪RFC。RFC的IANA考虑部分
必须包含以下信息，这些信息出现在IANA中
注册信息以及该出版物对应的RFC编号。

- 选项标签的名称。名称的长度可以任意，但
应不超过二十个字符长。名称必须
仅由字母数字（第25节）字符组成。

- 描述扩展的描述性文本。

27.2 警告代码

本规范在以下位置建立了 Warn-codes 子注册表：
http://www.iana.org/assignments/sip-parameters 并启动其
人口具有第20.43节中列出的警告代码。附加
警告代码由RFC出版物注册。

描述性文本为 warn-codes 表格是：

警告代码提供有关状态代码补充信息。
SIP响应消息，当事务失败时
从会话描述协议（SDP）（RFC 2327 [1]）问题。

"warn-code" 由三位数字组成。第一位数字为 "3"
指示特定于SIP的警告。 除非未来规范
描述了除3xx以外的警告码的使用，只有3xx警告码可以使用
已注册。

警告 300 至 329 保留用于指示问题
会话描述中的关键词，330至339为警告
与会议中请求的基本网络服务相关
描述，370至379是关于定量QoS的警告
会话描述中请求的参数，以及390至399
是杂项警告，不属于上述任何一种
类别

27.3 报头字段名称

此废弃了关于头部子注册表的IANA指令
在 http://www.iana.org/assignments/sip-parameters. 下

以下信息需要在RFC出版物中提供
为了注册一个新的头部字段名称：

- o 头部注册的RFC编号；
- o 头部字段注册名称 ered;
- o 该报头字段的紧凑形式版本，如果有的话
  定义；

一些常见且广泛使用的报头字段可能被分配一个字母
紧凑形式（第7.3.3节）。紧凑形式只能分配
SIP工作组审查后，随后发布RFC。

27.4 方法与响应代码

本规范确立了方法和响应码子-
注册信息位于 http://www.iana.org/assignments/sip-parameters 和
启动其种群如下。初始方法表如下：

邀请 [RFC3261] 确认 [RFC3261] 再见 [RFC3261] 取消 [RFC
3261] 注册 [RFC3261] 选项 [RFC3261] 信息 [RFC2976]

响应代码表最初从第21节填充。
信息性、成功、重定向、客户端错误部分
服务器错误，和全局故障。该表如下
格式： Translated Text: 格式：

类型（例如，信息性）
数字 默认原因短语 [RFC3261]

以下信息需要在RFC出版物中提供
为了注册一个新的响应代码或方法：

o RFC编号中包含该方法或响应代码
已注册；

o 响应代码的编号或方法的名称
已注册；

o 该响应代码的默认原因短语，如果
适用；

27.5 "消息/sip" MIME 类型。

此文档注册了“ 消息/sip ”MIME媒体类型，以便
允许SIP消息作为SIP体内的隧道，主要用于
端到端安全目的。此媒体类型由以下定义：
以下信息：

媒体类型名称：消息
媒体子类型名称：sip
所需参数：无

可选参数：版本
版本：所包含消息的SIP版本号（例如，{v*}）。
"2.0)。如果不存在，版本默认为“ 2.0 ”。"
编码方案：SIP消息由一个8位头部组成
可选地跟一个二进制 MIME 数据对象。因此，SIP
消息必须被视为二进制。在正常情况下
SIP消息通过二进制传输方式传输，无
特殊编码是必需的。

安全注意事项：见下文
动机和此用法作为安全机制的示例
与S/MIME一起给出的内容见23.4。

27.6 新的 Content-Disposition 参数注册

此文档还注册了四个新的Content-Disposition头
"disposition-types": 警报、图标、会话和渲染。 作者
请求将这些值记录在IANA注册表中
内容处置。

描述这些 "处置类型"，包括动机和
示例在第20.11节中给出。

简短 d 描述适合IANA注册的 y 是：

警告 主体是一个自定义铃声以提醒用户
图标 主体以图标形式显示给用户
渲染 主体应显示给用户
会话 主体描述了一个通信会话，对于
示例，如RFC 2327 SDP正文

28 RFC 2543 中的 28 个变更

本RFC修订了RFC 2543。它与旧版本基本兼容。
RFC 2543. 这里描述的更改修复了许多在 {v*}中发现的错误。
RFC 2543 并提供RFC中未详细说明的场景信息
2543. 协议已被以更清晰分层的方式呈现 {v*}
模型在此。

我们根据功能行为将差异分解，该行为是
重大变化来自RFC 2543，这对{v*}有影响
互操作性或在某些情况下的正确操作，以及功能
行为与RFC 2543不同，但不是潜在来源
互操作性问题的。已经发生了无数
澄清事项，此处未记录。

28.1 主要功能变更

当UAC希望在接听之前终止通话时，
它发送 CANCEL。如果原始 INVITE 仍然返回 2xx，则
UAC 然后发送 BYE。BYE 只能在现有的通话链路上发送
(现在在本RFC中称为对话)，而它可以在任何
时间在RFC 2543中。

SIP BNF 已转换为符合 RFC 2234 规范。

SIP URL BNF 被扩展得更通用，允许更大的集合
用户部分的字符。此外，比较规则是
简化为主要是大小写不敏感，并详细处理
在存在参数的情况下，比较的方法被描述了。
大多数实质性变化是，带有参数的URI发生了变化
默认值与没有该参数的URI不匹配。

通过隐藏移除。它存在严重的信任问题，因为它依赖于
在下一跳处执行混淆过程。相反，通过
隐藏可以作为在有状态中的本地实现选择进行。
代理，因此不再有文档记录。

在RFC 2543中，CANCEL和INVITE事务交织在一起。
他们现在分开了。当用户发送一个INVITE然后一个
取消，INVITE 事务仍然正常终止。一个 UAS
需要以487响应原始的INVITE请求
响应。

同样，CANCEL 和 BYE 交易交织在一起；RFC 2543
允许UAS在收到BYE时不对INVITE发送响应
接收。这是不允许的。原始的INVITE需要{v*}。
响应。

在RFC 2543中，UAs只需要支持UDP。在这份RFC中，UAs
需要支持UDP和TCP。

在RFC 2543中，一个分叉代理只向上传递了一个挑战从
下游元素在多个挑战的情况下。在此
RFC，代理应收集所有挑战并将它们放置
进入转发响应中。

在摘要凭据中，URI需要加引号；这不够明确
从RFC 2617和RFC 2069，这两个都对它不一致。

SDP处理已被拆分为单独的规范
[13]，并更详细地指定为正式的出价/答复交换
过程通过SIP有效隧道。SDP允许
在INVITE/200或200/ACK对于基线SIP实现；RFC
2543 暗示了在 INVITE、200 和 ACK 中使用它的能力
单笔交易，但这并未明确说明。更复杂的
SDP使用在扩展中是被允许的。

o 在URI和Via头部字段中完全支持IPv6。
支持 Via 中的 IPv6 需要 其报头字段
参数允许方括号和冒号字符。这些
字符之前不允许。在理论上，这可以
导致与旧实现兼容性问题。然而，我们
已观察到大多数实现接受任何非控制
ASCII字符在这些参数中。

DNS SRV 程序现在已在单独的规范中进行了文档化
[4]. 此过程同时使用SRV和NAPTR资源记录
不再将如描述中所述的SRV记录中的数据合并
RFC 2543.

循环检测已变为可选，被强制要求取代
使用Max-Forwards。RFC 2543中的循环检测过程
存在一个严重的错误，会将“ 螺旋 ”错误地报告为错误
条件当它不是时。可选的循环检测过程
此处有更全面和准确的说明。

o 标签的使用现在是强制性的（在RFC 2543中是可选的），
它们现在是对话的基本构建块
识别。

o 添加了支持的头部字段，允许客户端指示
支持的服务器扩展有哪些，哪些可以应用
扩展响应，并使用一个来指示它们的用法
在响应中需要。

o BNF中缺少了几个头部的扩展参数
字段，并且它们已经被添加。

处理路由和记录路由构建非常
未在RFC 2543中明确说明，也不是正确的方法。
在此规范中已大幅修改（并对{v*}进行了修改）
大大简化），这可以说是最大的变化。
向后兼容性仍然适用于那些仍然使用旧部署方式的部署。
不使用 "预加载路线"，其中初始请求有一个集合
在某种方式之外获得的路由报头字段值
记录路由。在这些情况下，新的机制不是
互操作。

在RFC 2543中，消息中的行可以用CR、LF终止
或CRLF。本规范仅允许CRLF。

o RFC 2543 中对 CANCEL 和 ACK 中 Route 的使用定义不明确。
它现在已很好地定义；如果一个请求有一个路由头字段,
其取消或确认对非2xx响应的请求需要
携带相同的路由报头字段值。2xx响应的ACK
使用从2xx的Record-Route中学习的Route值
响应。

o RFC 2543 允许在单个 UDP 数据包中包含多个请求。
使用已被移除。

绝对时间在Expires头字段和参数中的使用
已被删除。它导致元素中存在互操作性问题时
那些未进行时间同步的，这是一个常见现象。相对
时间被使用代替。

o Via头字段值的分支参数现在
所有元素必须使用。它现在扮演着...的角色。
唯一的交易标识符。这避免了复杂和bug-
从RFC 2543加载交易识别规则。一个魔法
cookie用于参数值中，以确定是否为之前的
跳跃已使参数全局唯一，比较落在
返回旧规则，当其不存在时。因此,
互操作性得到保证。

在RFC 2543中，TCP连接的关闭被等同于
取消。这几乎不可能实现（而且是不正确的）对于
TCP 代理之间的连接。这已被消除，因此
没有TCP连接状态和SIP之间的耦合
处理。

o RFC 2543 没有说明 UA 是否可以发起一个新的
在另一个正在进行时向对等方进行交易。现在是这样
在此指定。对于非-INVITE请求允许，不允许
对于INVITE。

o PGP 已移除。它未充分说明，并且不
兼容更完整的PGP MIME。它被替换为
S/MIME

o 添加了 "sips" URI 方案以支持端到端 TLS。 此方案是
与RFC 2543不兼容。现有元素
接收带有SIPS URI方案的Request-URI请求
可能拒绝请求。这实际上是一个特性；它确保
仅当所有路径跳数都可以成功传递时，才会将SIPS URI的调用传递过去
确保安全。

o 使用TLS添加了额外的安全功能，这些功能是
描述在一个更大且更完整的网络安全考虑中
章节。

在RFC 2543中，代理不需要转发临时
响应来自101到199的上游。这被改为MUST。
这很重要，因为许多后续功能都依赖于 {v*}
所有临时响应从101到199的交付。

关于RFC 2543中503响应代码的讨论很少。它有
自发现大量用于指示故障或过载
代理条件。这需要某种特殊处理。
具体来说，收到503应该触发尝试
联系DNS SRV查找结果中的下一个元素。此外，
503 响应仅在特定情况下由代理向上游转发
条件。

o RFC 2543 定义了，但未充分指定，一种机制
服务器UA身份验证。已被移除。相反，
RFC 2617的相互认证过程是允许的。

o 一个UA在收到对呼叫的ACK之前不能发送BYE
初始的INVITE。这在RFC 2543中被允许，但会导致
潜在竞态条件。

o 一个UA或代理在收到一个 {v*}之前不能为一个事务发送CANCEL
临时响应请求。这在RFC中是允许的。
2543 但可能导致潜在的竞态条件。

o 注册中的动作参数已被弃用。它曾是
不足以提供任何有用的服务，并在发生冲突时
应用处理被应用于代理中。

RFC 2543 对多播有多个特殊情况。对于
示例，某些响应被抑制，计时器被调整，
等等。多播现在扮演着更有限的角色，并且
协议操作不受多播使用的影响
向单播。因此产生的限制已记录在案。

o 基本身份验证已被完全移除，其使用
禁止。

o 代理在接收到6xx后不再立即转发。
相反，他们立即取消挂起的分支。这避免了
潜在的竞态条件，可能导致UAC获得6xx错误
随后是一个2xx。在所有除这种竞争条件之外的情况下，
结果将相同 - 6xx将被向上游转发。

RFC 2543未解决请求合并的问题。
当请求在代理处分叉并在稍后重新连接到时发生
元素。合并处理仅在UA处进行，程序
仅定义用于拒绝除第一个请求之外的所有请求。

28.2 小型功能更改

o 添加了Alert-Info、Error-Info和Call-Info头部字段，用于
可选内容向用户展示。

o 添加了Content-Language、Content-Disposition和MIME-Version
表头字段。

o 添加了“ 眩光处理 ”机制以处理{v*}的情况
双方同时向对方发送重邀请。它使用
新的491（请求待处理）错误代码。

o 添加了 In-Reply-To 和 Reply-To 头部字段以支持
未接来电或短信的后续回拨。

o 添加了TLS和SCTP作为有效的SIP传输方式。

o 描述了多种处理故障的机制
在任何通话期间；这些现在一般已统一。再见
被发送以终止。

o RFC 2543 规定了在 TCP 上重新传输 INVITE 响应，但
注意它实际上只需要用于2xx。那是一个遗留问题。
协议分层不足。具有更一致的交易
层在此定义，不再需要。仅2xx响应
到INVITEs在TCP上重新传输。

客户端和服务器交易机现在基于 {v*} 驱动
超时而不是重传计数。这允许状态
机器应正确指定TCP和UDP。

o 日期报头字段用于在REGISTER响应中提供{v*}
简单的用户代理日期自动配置方法。

o 允许注册商拒绝那些到期为 {v*} 的注册
持续时间太短。定义了423响应代码和
此目的的最小过期时间。

29 规范性引用

[1] Handley, M. 和 V. Jacobson, "SDP: Session Description Protocol"
协议", RFC 2327, 1998年4月。

[2] Bradner, S., "在RFC中使用的关键词以指示需求"
层级", BCP 14, RFC 2119, 1997年3月。

[3] Resnick, P., 《互联网消息格式》, RFC 2822, 2001年4月。

[4] Rosenberg, J. 和 H. Schulzrinne, "SIP: 定位 SIP 服务器"
RFC 3263, 2002年6月。

[5] Berners-Lee, T., Fielding, R. 和 L. Masinter, "统一资源标识符 {v*}
标识符（URI）：通用语法", RFC 2396, 1998年8月。

[6] Chown, P., "高级加密标准（AES）加密套件"
传输层安全性（TLS）, RFC 3268, 2002年6月。

[7] Yergeau, F., "UTF-8, ISO 10646的转换格式", RFC
2279, 1998年1月。

[8] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., {v*}
Leach, P. 和 T. Berners-Lee, "超文本传输协议 --"
HTTP/1.1", RFC 2616, 1999年6月。

[9] Vaha-Sipila, A., "电话通话的URL", RFC 2806, 四月
2000.

[10] Crocker, D. 和 P. Overell, "增强BNF用于语法"
规格：ABNF", RFC 2234, 1997年11月。

[11] Freed, F. 和 N. Borenstein, "多用途互联网邮件 {v*}
扩展（MIME）第二部分：媒体类型", RFC 2046, 11月
1996.

[12] Eastlake, D., Crocker, S. 和 J. Schiller, "随机性
安全建议", RFC 1750, 1994年12月。

[13] Rosenberg, J. 和 H. Schulzrinne, "带有 {v*} 的 Offer/Answer 模型"
SDP", RFC 3264, 2002年6月。

[14] Postel, J., "用户数据报协议", STD 6, RFC 768, 八月
1980.

[15] Postel, J., "DoD标准传输控制协议", RFC
761, 1980年1月。

[16] Stewart, R., Xie, Q., Morneault, K., Sharp, C., Schwarzbauer H., Taylor, T., Rytina, I., Kalla, M., 张, L. 和 V. Paxson, "流控制传输协议", RFC 2960, 2000年10月。

[17] Franks, J., Hallam-Baker, P., Hostetler, J., Lawrence, S. Leach, P., Luotonen, A. 和 L. Stewart, "HTTP 认证： 基本和摘要访问认证", RFC 2617, 1999年6月。

[18] Troost, R., Dorner, S. 和 K. Moore, "Communicating Presentation {v*}" 互联网消息中的信息：内容处置头 字段", RFC 2183, 1997年8月。

[19] Zimmerer, E., Peterson, J., Vemuri, A., Ong, L., Audet, F. 沃森, M. 和 Zonoun, M., "ISUP 和 QSIG 的 MIME 媒体类型 {v*} 对象", RFC 3204, 2001年12月。

[20] Braden, R., "互联网主机需求 - 应用和 {v*} 支持", STD 3, RFC 1123, 1989年10月。

[21] Alvestrand, H., "IETF关于字符集和语言的策略", BCP 18, RFC 2277, 1998年1月。

[22] 加尔文, J., 墨菲, S., 克罗克, S. 和 N. 弗里德, 《安全 MIME的多部分：Multipart/Signed 和 Multipart/Encrypted" RFC 1847, 1995年10月。

[23] Housley, R., "加密消息语法", RFC 2630, 六月 1999.

[24] Ramsdell B., "S/MIME 版本 3 消息规范", RFC 2633, 1999年6月。

[25] Dierks, T. 和 C. Allen, "TLS协议版本1.0", RFC 2246, 1999年1月。

[26] Kent, S. 和 R. Atkinson, "Security Architecture for the {v*}" 互联网协议", RFC 2401, 1998年11月。

30 informative references

[27] R. Pandya, "新兴的移动和个人通信系统," IEEE 通信杂志, 第 33 卷, 第 44--52 页, 1995 年 6 月。

[28] Schulzrinne, H., Casner, S., Frederick, R. 和 V. Jacobson, "RTP: 实时应用的传输协议", RFC 1889年, 1996年1月。

[29] Schulzrinne, H., Rao, R. 和 R. Lanphier, "实时流媒体传输协议（RTSP）", 协议（RTSP）, RFC 2326, 1998年4月。

[30] Cuervo, F., Greene, N., Rayhan, A., Huitema, C., Rosen, B. 和 J. Segers, "Megaco协议版本1.0", RFC 3015, 11月 2000.

[31] Handley, M., Schulzrinne, H., Schooler, E. 和 J. Rosenberg, "SIP: 会话初始化协议", RFC 2543, 1999年3月。

[32] Hoffman, P., Masinter, L. 和 J. Zawinski, "The mailto URL 方案", RFC 2368, 1998年7月。

[33] E. M. Schooler, "用于{v*}的多播用户目录服务" 同步会合, “ 硕士学位论文 CS-TR-96-18, 系部 ” 计算机科学系, 加州理工学院, 帕萨迪纳, 加利福尼亚, 1996年8月。

[34] Donovan, S., "SIP INFO 方法", RFC 2976, 2000年10月。

[35] Rivest, R., "MD5消息摘要算法", RFC 1321, 四月 1992.

[36] Dawson, F. 和 T. Howes, "vCard MIME 目录配置文件", RFC 2426, 1998年9月。

[37] Good, G., "LDAP数据交换格式（LDIF）- 技术 规范", RFC 2849, 2000年6月。

[38] Palme, J., "常见的互联网消息头", RFC 2076, 二月份1997年。

[39] Franks, J., Hallam-Baker, P., Hostetler, J., Leach, P. Luotonen, A., Sink, E. 和 L. Stewart, 《对HTTP的扩展： 摘要访问认证", RFC 2069, 1997年1月。

[40] Johnston, A., Donovan, S., Sparks, R., Cunningham, C., Willis, D., Rosenberg, J., Summers, K.和H. Schulzrinne, 《SIP呼叫 流示例", 进行中。

[41] E. M. Schooler, "案例研究：多媒体会议控制中的{v*}" 分组交换电话会议系统," 期刊 互联网：研究与实践, 第4卷, 第99--120页, 六月 1993。ISI 重印系列 ISI/RS-93-359。

[42] H. Schulzrinne, "多媒体服务中的个人移动性"
互联网,"在欧洲交互式分布式
多媒体系统与服务（IDMS），（德国柏林），三月
1996.

[43] Floyd, S., "拥塞控制原则", RFC 2914, 九月
2000.

一个定时器值表

表4总结了各种定时器的含义和默认值
由本规范使用。

Timer	Value	Section	Meaning
T1	500ms default	Section 17.1.1.1	RTT Estimate
T2	4s	Section 17.1.2.2	The maximum retransmit interval for non-INVITE requests and INVITE responses
T4	5s	Section 17.1.2.2	Maximum duration a message will remain in the network
Timer A	initially T1	Section 17.1.1.2	INVITE request retransmit interval, for UDP only
Timer B	64*T1	Section 17.1.1.2	INVITE transaction timeout timer
Timer C	> 3min	Section 16.6 bullet 11	proxy INVITE transaction timeout
Timer D	> 32s for UDP 0s for TCP/SCTP	Section 17.1.1.2	Wait time for response retransmits
Timer E	initially T1	Section 17.1.2.2	non-INVITE request retransmit interval, UDP only
Timer F	64*T1	Section 17.1.2.2	non-INVITE transaction timeout timer
Timer G	initially T1	Section 17.2.1	INVITE response retransmit interval
Timer H	64*T1	Section 17.2.1	Wait time for ACK receipt
Timer I	T4 for UDP 0s for TCP/SCTP	Section 17.2.1	Wait time for ACK retransmits
Timer J	64*T1 for UDP 0s for TCP/SCTP	Section 17.2.2	Wait time for non-INVITE request retransmits
Timer K	T4 for UDP 0s for TCP/SCTP	Section 17.1.2.2	Wait time for response retransmits

表 4：计时器摘要

致谢

我们希望感谢IETF MMUSIC和SIP工作组成员们为他们的 {v*}
评论和建议。详细的评论由Ofir提供
Arkin, Brian Bidulock, Jim Buller, Neil Deason, Dave Devanathan,
基思·德拉格，比尔·费纳，塞德里克·弗卢基格，亚龙·戈兰，约翰
热情，伯尼·霍伊尼森，乔·霍恩斯比，菲尔·霍弗，克里斯蒂安·胡伊特玛，
希沙姆·哈塔比尔，让·杰维斯，加迪·卡米，彼得·耶勒斯特德，安德斯
克里斯蒂安森，乔纳森·莱诺克斯，盖思·利德尔，艾莉森·曼金，威廉
Marshall, Rohan Mahy, Keith Moore, Vern Paxson, Bob Penfield, Moshe
J. Sambol, Chip Sharp, Igor Slepchin, Eric Tremblay 和 Rick
工人。

布ライ恩·罗森提供了编译后的 BNF。

Jean Mahoney提供了技术写作协助。

这项工作基于[41,42]，等等。]

作者地址

作者地址按字母顺序列出供编辑参考，
作者，然后是RFC 2543的原始作者。所有列出的
作者积极为此文档贡献了大量文本。

乔纳森·罗森伯格
dynamicsoft
72 鹰岩大道
东方汉诺威，新泽西州 07936
美国

电子邮件: jdrosen@dynamicsoft.com

施洛兹林内亨宁
计算机科学系
哥伦比亚大學
1214 阿姆斯特丹大道
纽约，NY 10027
美国

电子邮件: schulzrinne@cs.columbia.edu

冈萨洛·卡马里洛
爱立信
高级信号研究实验室。
FIN-02420 约瓦斯
芬兰

电子邮件：Gonzalo.Camarillo@ericsson.com

艾伦·约翰斯顿
世界通信
100 南四街
圣路易斯，密苏里州 63102
美国

电子邮件: alan.johnston@wcom.com

乔恩·彼得森
NeuStar, Inc 译文字符串：NeuStar， 公司
1800 Sutter Street, Suite 570
康科德，加利福尼亚州 94520
美国

电子邮件: jon.peterson@neustar.com

Robert Sparks Translated Text: 罗伯特·斯帕克斯
dynamicsoft, Inc.
5100 Tennyson Parkway
1200室
普莱诺，德克萨斯州 75024
美国

电子邮件: rsparks@dynamicsoft.com

Handley 马克
国际计算机科学研究所
1947 Center St, Suite 600 翻译文本：1947 Center街，600号室
伯克利，加利福尼亚州 94704
美国

电子邮件: mjh@icir.org

伊夫·舒勒
AT&T 实验室-研究
75 Willow Road 75 Willow路
门洛帕克，CA 94025
美国

邮箱：schooler@research.att.com

认可

资金目前由提供RFC编辑功能的组织提供。
互联网社会。