

前 言

本标准以《IP 电话/传真业务总体技术要求》为基础,结合国内 IP 电话网关设备生产厂家的网关与网关之间以及网关与网守之间的实际互通测试制定的。它是 IP 电话网关设备研制、生产和测试的技术依据。

本标准的附录 A 为标准的附录。

本标准由信息产业部电信研究院提出并归口。

本标准起草单位:信息产业部电信传输研究所,深圳市华为技术有限公司,中国普天信息产业集团公司,上海贝尔有限公司,西安大唐电信有限公司,信息产业部数据通信技术研究,深圳市中兴通讯股份有限公司。

本标准主要起草人:蒋林涛、叶冠华、杨昆、陈忠、赵欣、金志勇、杨琪、赵庆轩、任红菊、范超、池毓东。

中华人民共和国通信行业标准

IP 电话网关设备互通技术规范

YD/T 1046—2000

Interoperability Specification for IP Telephony Gateway

1 范围

本标准规定了 IP 电话网关与网关之间以及网关与网守之间在通信协议上的互通要求。
本标准适用于国内 IP 电话网络上使用的 IP 电话网关设备。

2 引用标准

下列标准所包含的条文,通过在本标准中引用而构成为本标准的条文。本标准出版时,所示版本均为有效。所有标准都会被修订,使用本标准的各方应探讨使用下列标准最新版本的可能性。

ITU-T 建议 H. 323:1998 基于包交换的多媒体通信系统

ITU-T 建议 H. 225(v2):1998 基于包交换的多媒体通信系统的信令协议和打包方法

ITU-T 建议 H. 245:1998 多媒体通信的控制协议

YD/T 1044—2000 IP 电话/传真业务总体技术要求

3 缩略语

RRQ: Registration Request(注册请求)

RCF: Registration Confirm(注册确认)

RRJ: Registration Reject(拒绝注册)

URQ: Unregistration Request(注销请求)

UCF: Unregistration Confirm(注销确认)

URJ: Unregistration Reject(注销拒绝)

ARQ: Admission Request(接入请求)

ACF: Admission Confirm(接入确认)

ARJ: Admission Reject(接入拒绝)

BRQ: Bandwidth Request(带宽请求)

BCF: Bandwidth Confirm(带宽确认)

BRJ: Bandwidth Reject(带宽请求拒绝)

DCF: Disengage Confirm(退出确认)

DRJ: Disengage Reject(退出拒绝)

DRQ: Disengage Request(退出请求)

RIP: Request in Progress(进展请求)

IRQ: InfoRequest(信息查询)

IRR: InfoRequest Response(信息查询响应)

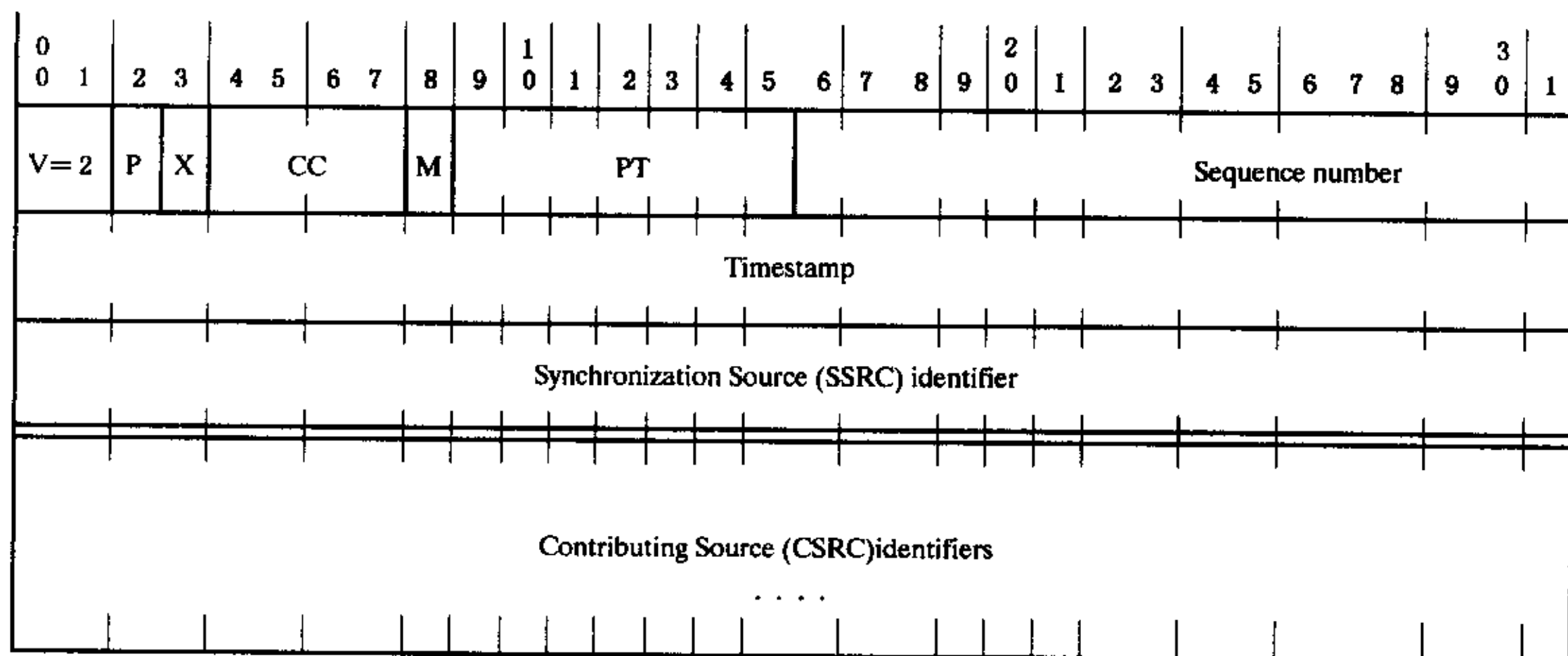
中华人民共和国信息产业部 2000-02-23 批准

2000-02-23 实施

IACK: InfoRequestAck(信息查询确认)
 INAK: InfoRequestNak(信息查询否认)
 RAI: ResourceAvailabilityIndication(资源可用性指示)
 RAC: ResourcesAvailabileConfirmation(资源可用性确认)
 TCS: Terminal Capacity Set(终端能力集)
 TCSA Terminal Capacity Set Acknowledge(终端能力集确认)
 TCSR: Terminal Capacity Set Reject(终端能力集拒绝)
 OLC: Open Logical Channel(打开逻辑通道)
 OLCA: Open Logical Channel Acknowledge(打开逻辑通道确认)
 OLCR: Open Logical Channel Reject(打开逻辑通道拒绝)
 CLC: Close Logical Channel(关闭逻辑通道)
 CLCA: Close Logical Channel Acknowledge(关闭逻辑通道确认)
 CLCR: Close Logical Channel Reject(关闭逻辑通道拒绝)
 ESC: End Session Command(结束会话命令)

4 RTP 协议

4.1 RTP 报头格式



T1527560-97

各 Field 值确定

V(版本): 2 bit 版本号置 2

P(填充): 1 bit 填充位置 0

X(扩展): 1 bit 扩展位置 0

CC(CSRC 数): 4bit CSRC 标识的数量, 此字段填充为 0, 本体制不使用 CSRC。

M(标志): 1 bit 标志位, 该标志在静音后的第一个语音包时置位。静音包仅发送一个, 不连续发送。

PT(净荷类型): 7 bit G. 723.1 4

G. 729 18

Sequence number(序列号): 16 bit 序列号, 初始值为一随机数, 此后以 1 递增, 收端以此判定包丢失及恢复包顺序。

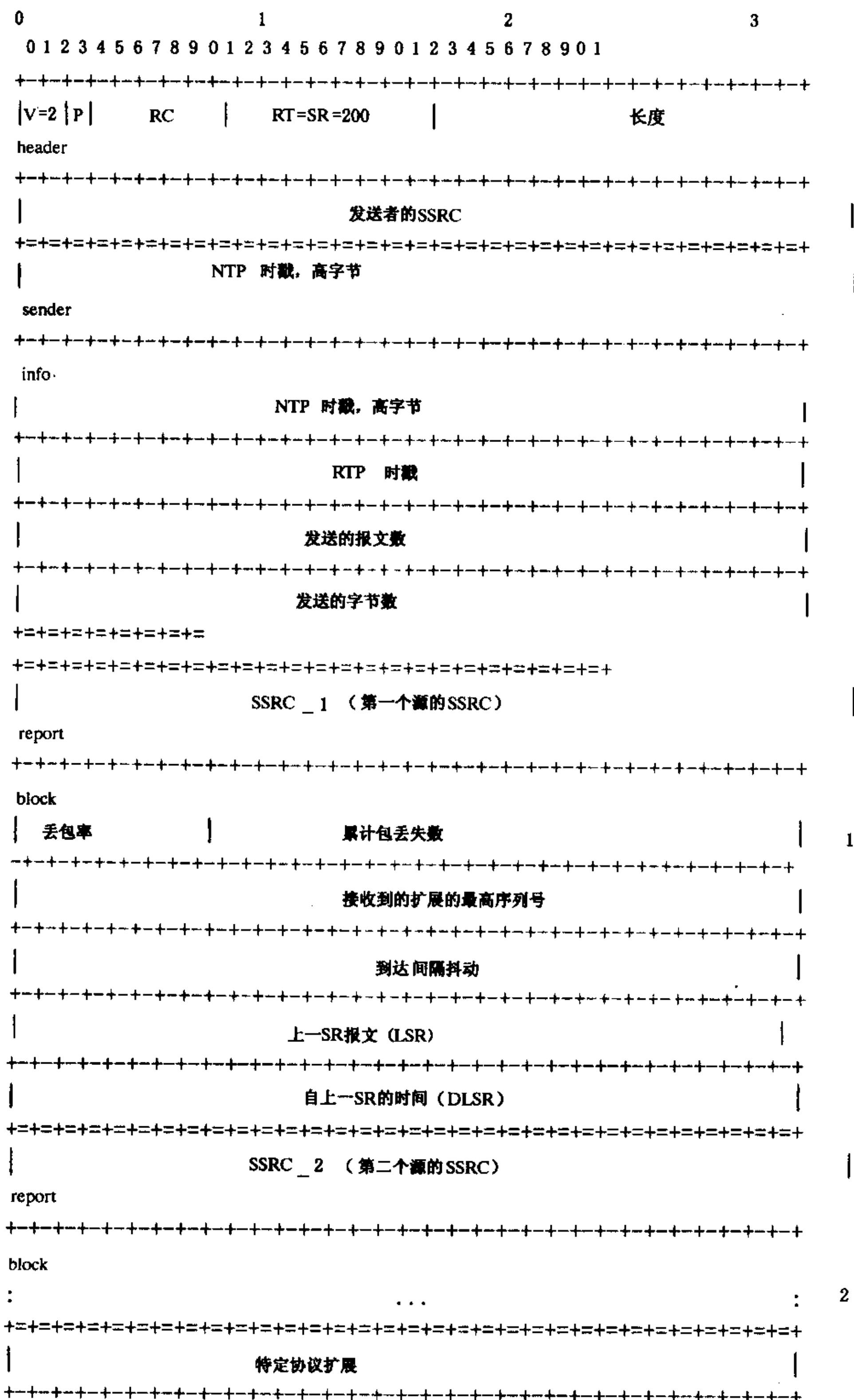
Time stamp(时戳): 32 bit 时戳。用于标识 RTP 数据包中第一个字节采样时的时刻, 其起始值为一随机值, 以 8 000 次/s 的速率递增。

Synchronization Source(SSRC)identifiers(同步源标志): 32 bit, 用来标识 RTP 包的数据源。

Contributing Source (CSRC) identifiers (贡献源标志): 每个 CSRC 32 bit, 0~15 个 CSRC 序列, 本体制不包含该字段。

4.2 RTCP 协议

RTCP 报文共有 5 类: RR SR SDES BYE APP。本标准只对 SR 和 RR 报文提出要求。



SR(发送报文)的格式如下:

其中的各项内容定义如下:

版本(V):2 bit 协议鉴别,在本体制中规定为 2

填充(P):1 bit 在本体制中规定为 0

接收报告数(RC):5 bit

在 SR 中包含的 RR 的数目,在本体制中规定不得大于 1。

净荷类型(PT):8 bit

报文类型,以二进制表示。其中十进制的 200 代表 SR。

长度(length):16 bit

报文长度,指在其后的报文长度,所以有可能为 0。

发送者的同步源标志(SSRC of sender):32 bit

源同步码,用以标识此次通话。

NTP 时戳(NTP timestamp):64 bit

绝对时戳。在测量环路时延时可在对方的 RR 报文中带回;如果发送方不具有绝对时钟的能力,则可以用通话开始时间作为时钟 0 点或将此域置 0。(在 NTP 格式中,64 位的前 32 位是从 1900 年 1 月 1 日 0 时开始到现在的以秒为单位的整数部分,后 32 位是此时间的小数部分)

RTP 时戳(RTP timestamp):32 bit

以 RTP 的时戳为基准。

发送的报文数(sender's packet count):32 bit

从通话开始后发送方总共发送的 RTP 报文的数目。

发送的字节数(sender's octet count):32 bit

从通话开始后发送方总共发送的有效载荷的数目(以字节记)。

随后描述的是一个或多个 RR 报文块,在本体制中规定在 SR 报文中最多只能有一个 RR 报文块。

源标志-*n*(SSRC-*n*):32 bit

源同步码,用以标识此 RR 块所从属的通话。

丢包率(fraction lost):8 bit

从上一个 SR 或 RR 报文发送后的丢包率,表现为接收方在此段时间内期待的 RTP 报文与所收到的 RTP 包数目的差值和它所期待的 RTP 报文的数目的比值,若为负值,置为 0。详见 RFC 1889。

累计的包丢失数(cumulative number of packets lost):24 bit

累计的包丢失数。

接收到的扩展的最高序列号(extended highest sequence number received):32 bit

其低 16 位是其收到的 RTP 包中的 sequence number 的最新值。其高 16 位标识其收到的 RTP 报文的 sequence number 的循环的次数。

到达间隔抖动(interarrival jitter):32 bit

时延抖动。每两个 RTP 包的抖动可以用其 RTP 包中的 RTP 时戳和接收的时刻进行计算,计算公式如下:设

R_j 代表第 j 个包的到达时刻, S_j 代表第 j 个包的 RTP 时戳值,则第 i 个 RTP 报文与第 j 个 RTP 报文间的抖动为 $D(i, j)$:

$$D(i, j) = (R_j - R_i) - (S_j - S_i) = (R_j - S_j) - (R_i - S_i)$$

在生成 RTCP 报文时,其应当传送的时延抖动的值可用如下公式进行递推计算:

$$J = J + (|D(I-1, I)| - J) / 16$$

其中, J 为要传送的时延抖动值。对后一项除以 16 是为了消除连带噪声。

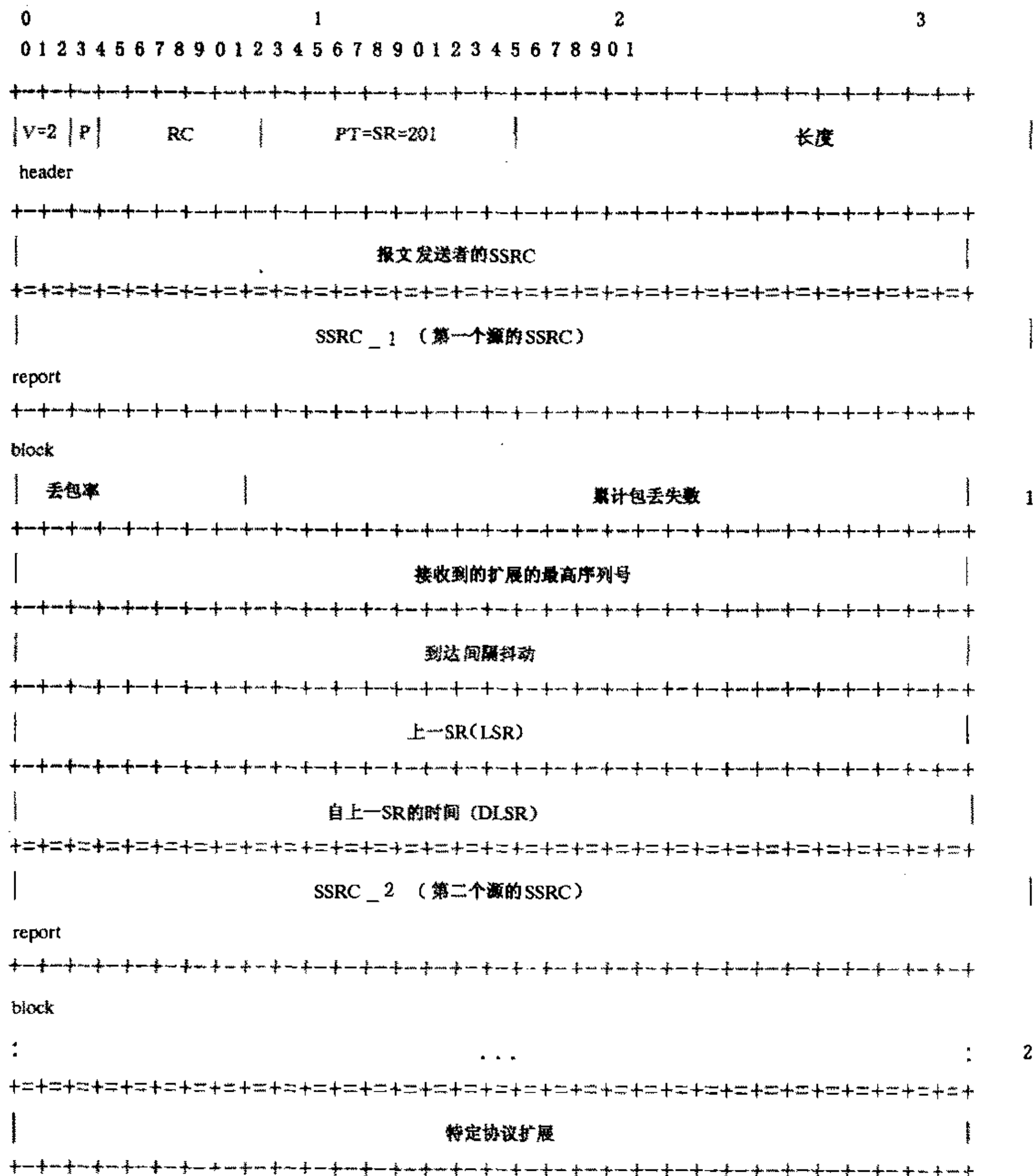
上一 SR 报文时戳(LSR):32 bit

收到的最近一个 SR 报文的 NTP 时戳的中间 32 位。

自上一 SR 的时间(DLSR):32 bit

在收到上一个 SR 报文与此次发送的报文之间的时间。以 1/65536 s 记。如果还没有收到任何 SR 报文,此值置 0。

RR 报文的格式如下:



其中各项的功能与形式如 SR 中的说明。若未收到任何 RTP 报文,则可发送一个空的 RR,即 RC=0。

RTCP 包发送机制:在两次 RTCP 报文之间,若端点没有发出任何 RTP 报文,则端点此次发送 RR (接收报文),否则,端点发送 SR(发送报文),RTCP 包每秒发送一次。

5 语音帧结构

5.1 G. 723.1

IP 电话可以选用 G. 723.1 编码。G. 723.1 的帧长有 3 种情况:24 字节(6.3 kbit/s),20 字节(5.3 kbit/s)和 4 字节。4 字节为 SID(插入静音描述帧)帧,它主要用在语音的静音段,用以发送比较舒服的噪声的参数描述。这 3 种帧可以用任意方式混合使用。第一个八位组的最低 2 个比特确定了帧的长度和编码类型。在 30 ms 的帧边界上,这两种速率可以进行任意切换,以获得最佳的音质。所有编码比特流都是从最低有效位开始传送,直至最高有效位。

G. 723.1 打包特征为:

a) 用 RTP 报头的标记位的置位方法来表示该报文是静音以后第一个包,其余包的标志位置零,发送了第一个静音帧以后,在静音期间不再发送 RTP 包,由收端网关根据静音帧产生舒适噪音。

b) 抽样频率为 8 000 Hz。

c) 帧长为 30 ms。

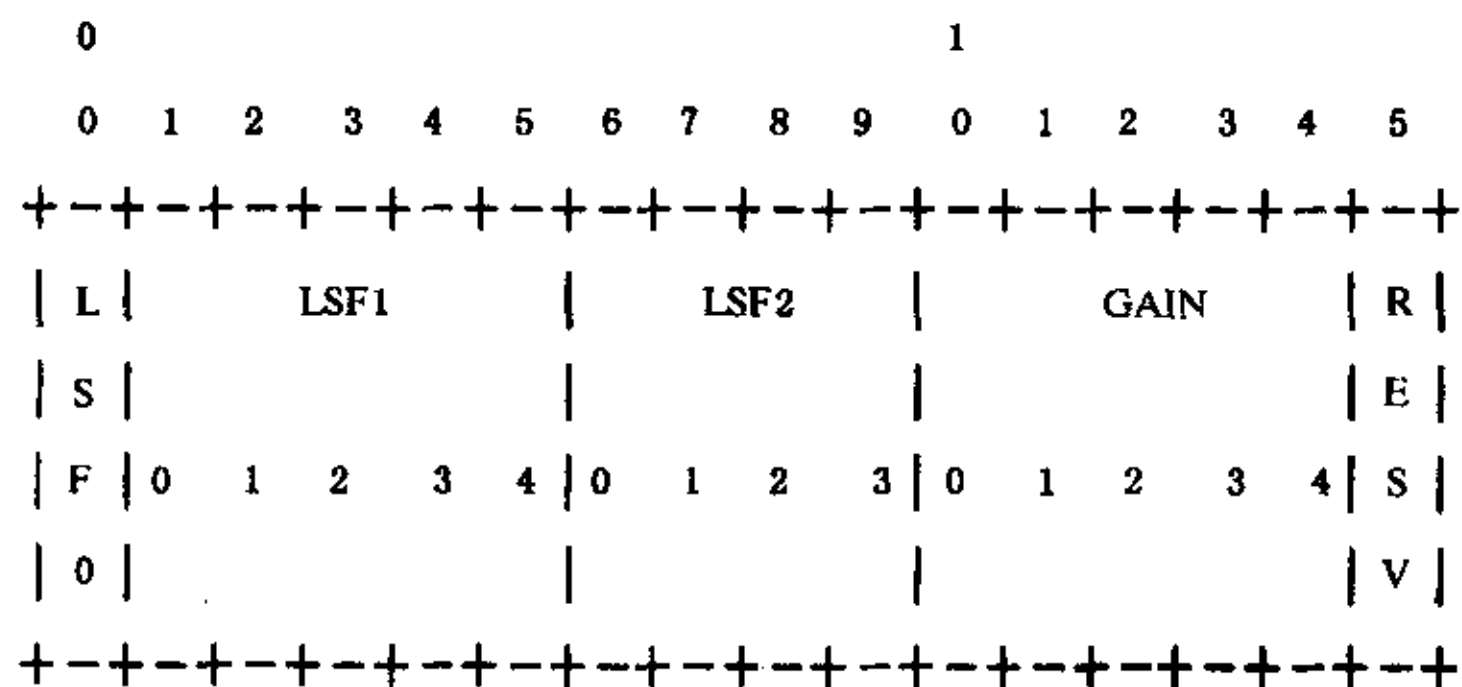
d) 在一个包中,编解码器可以编解码几个连续的帧。

e) 接收机必须要能连续接收 0~180 ms 的音频数据。

5.2 G. 729

这是一种 8 kbit/s 的编码算法,该种编码抗随机比特错误的能力与抗随机突发消失帧的能力相同。在噪声较大的环境下,它能有更好的语音质量。G. 729 附件 A 算法是 G. 729 算法降低了复杂度后的版本,两者能完全互操作,因而不必对这两种算法进行区分。

在 G. 729 附件 B 中,建议声音激活检测器(VAD)和舒适噪声发生器(CNG)用于数字模拟声音和数字应用,可以和 G. 729、G. 729 附件 A 结合使用。G. 729 帧长为 10 个八位组(字节),静音(附件 B)为 2 个八位组。舒适静音的格式为:



有声段帧格式为：

- 一帧为 10 ms。
- 帧长 10 个八位组。
- 一个 RT 包可以放 0 个、1 个或多个 G 729 或 G 729 附件 A 帧,后随 G 729 附件 B 的有效载荷。舒适噪声帧的存在可以减小 RTP 载荷的长度。
- 静音后的第一个有声包在 RTP 报头中标记位置位。
- 抽样率 8 000 Hz。
- 缺省打包时间段 20 ms。
- 编解码器可以进行单一包中连续 1~10 帧的编解码。
- 接收方必须能接收 0~200 ms 的用户语音数据。

5.3 其他语音编码算法

可根据实际情况增加其他语音编码算法。

6 流程

6.1 网关注册流程

网关设备在运行初期向网守登记注册,如图 1 所示。

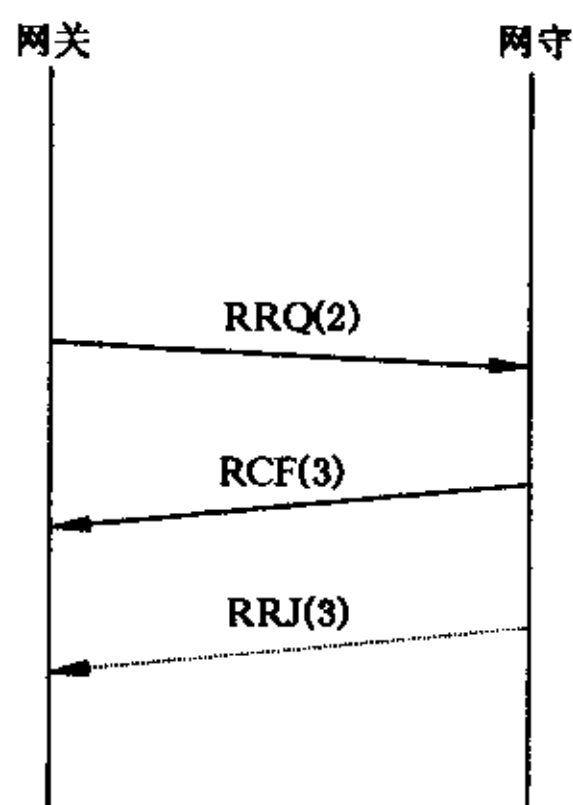


图 1 网关向网守的登录流程

- 1) 网关设备通过静态的初始配置获得网守的 IP 地址。
- 2) 网关向网守发送注册请求(RRQ)消息,传送网关信息。
- 3) 网守经过验证,将注册确认(RCF)或拒绝(RRJ)消息发送到网关。

注:网关在首次注册时应将 RRQ 消息中的 discoverycomplete 置 0,其余报告其存活的 RRQ 消息的 discoverycomplete 置 1。

6.2 呼叫流程

6.2.1 接入认证流程

如图 2 所示。

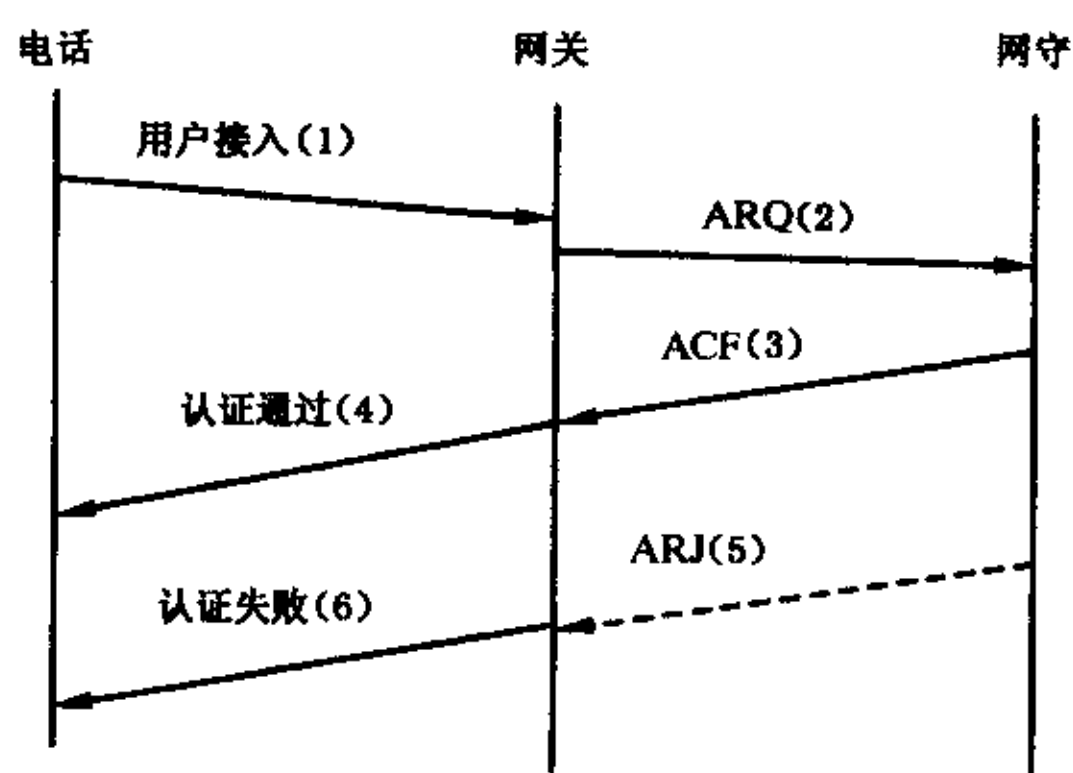


图 2 接入认证流程

- 1) 用户使用电话机拨接入码接入到网关。
- 2) 网关采集用户的接入信息(账号和密码),向网守发送请求用户接入认证(ARQ)消息(包含用户接入信息)。
- 3) 网守接收到来自网关的请求用户接入认证(ARQ)消息后,检查用户合法性,确定用户权限,如果用户为合法用户,则发送接入认证通过和授权(ACF)消息发送到网关;否则,发送拒绝(ARJ)消息到

网关。

4) 网关若收到 ACF 消息,确定用户权限合法,认证通过;若收到 ARJ 消息,确定用户为非法用户,拒绝用户接入。

6.2.2 更改密码流程

认证通过后,卡号用户通过网关向网守进行更改密码操作,如图 3 所示。

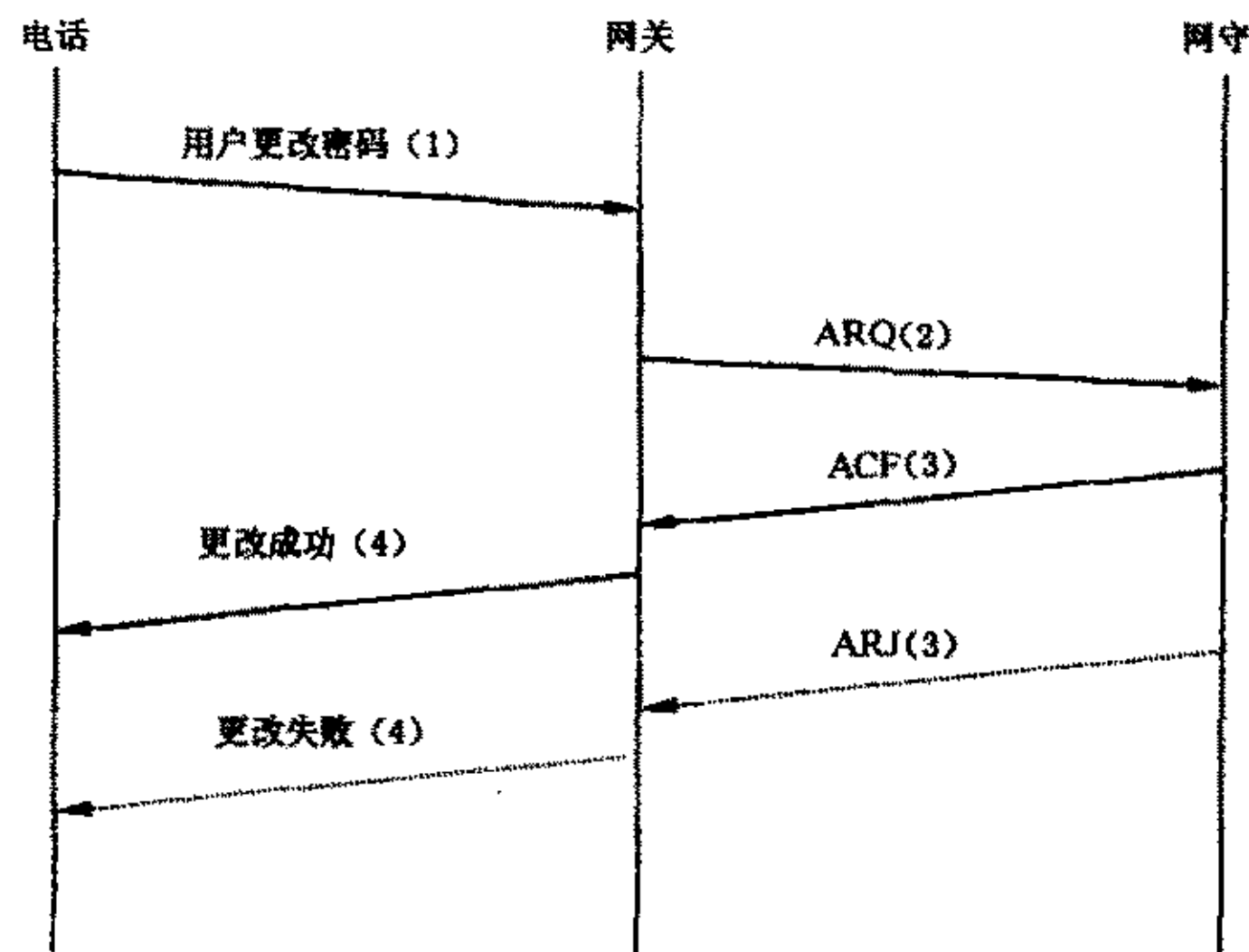


图 3 卡号用户更改密码流程

1) 用户接入通过后,更改密码。

2) 网关采集用户的更改密码信息(旧密码、新密码),向网守发送请求用户接入认证(ARQ)消息(包含用户更改密码信息)。

3) 网守接收到来自网关的请求用户接入认证(ARQ)消息后,检查密码的合法性,如果密码更改成功,则发送接入认证通过和授权(ACF)消息发送到网关;否则,发送拒绝(ARJ)消息到网关。

4) 网关若收到 ACF 消息,确定用户更改密码成功;若收到 ARJ 消息,则用户更改密码失败。

6.2.3 路由解析

对于一次拨号用户,用户的认证和地址解析同时完成;对于卡号用户,在认证通过后,进行地址解析,如图 4 所示。

1) 对于卡号用户,当认证通过后,输入对方电话号码;对于一次拨号用户,输入接入码及对方电话号码。

2) 网关采集用户信息及对方电话号码,向网守发送请求用户接入认证(ARQ)消息(包括用户账号、密码、对方电话号码、主叫电话号码)。

3) 网守接收到来自网关的请求用户接入认证(ARQ)消息后,检查信息合法性,进行地址翻译,如果成功,则将接入认证通过和授权(ACF)消息发送到网关;否则,发送拒绝(ARJ)消息到网关。

4) 网关若收到 ACF 消息,则地址解析成功,网关则进入呼叫流程;若收到 ARJ 消息,则为地址解析失败或为非法用户。

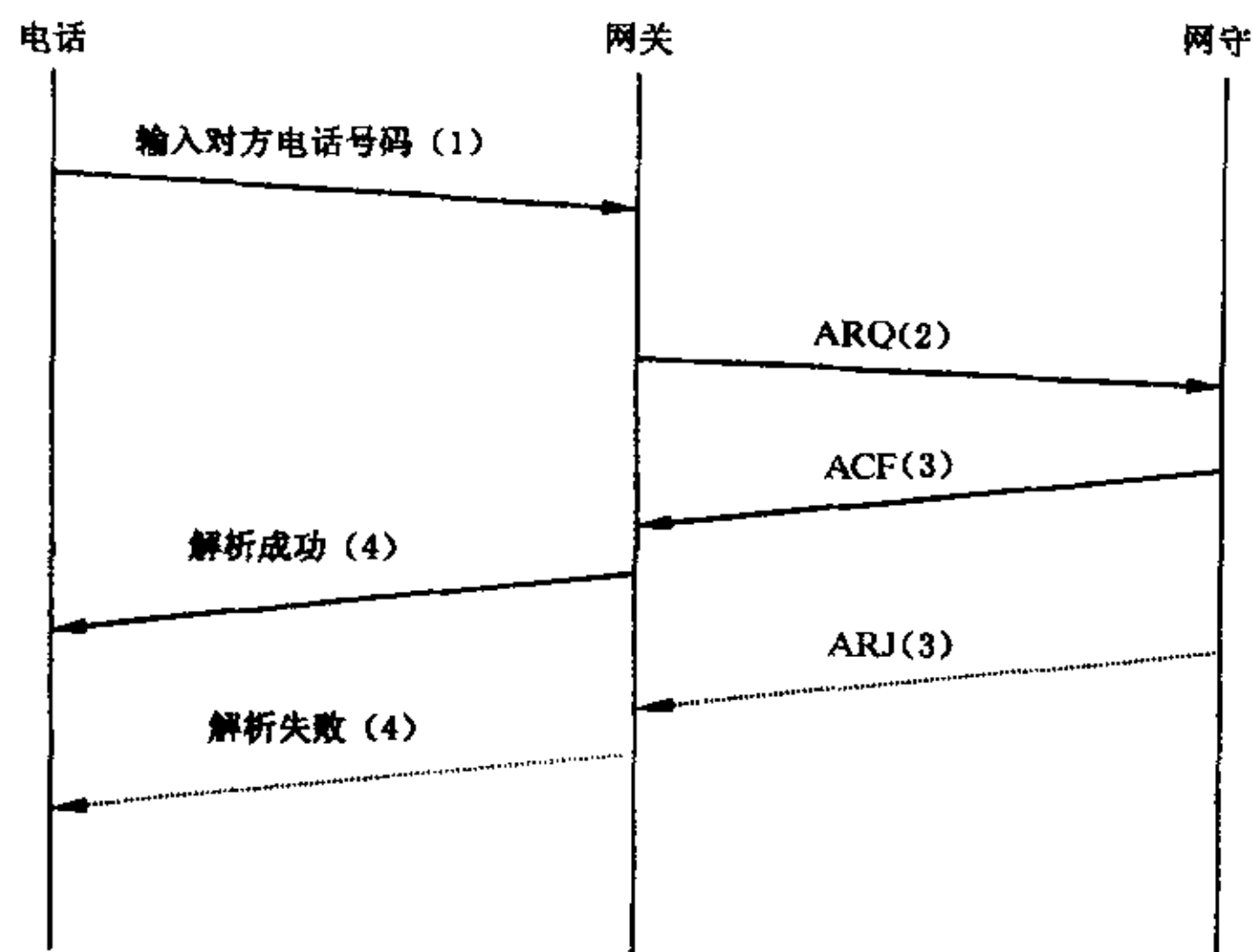


图 4 用户地址解析

注:地址解析完成后,网守应在 ACF 消息中返回通话时长。

6.2.4 快速呼叫流程

建议采用快速呼叫建立过程(fastStart 方式),如图 5 所示。在无法做到的情况下,也可以采用非快速建立方式。

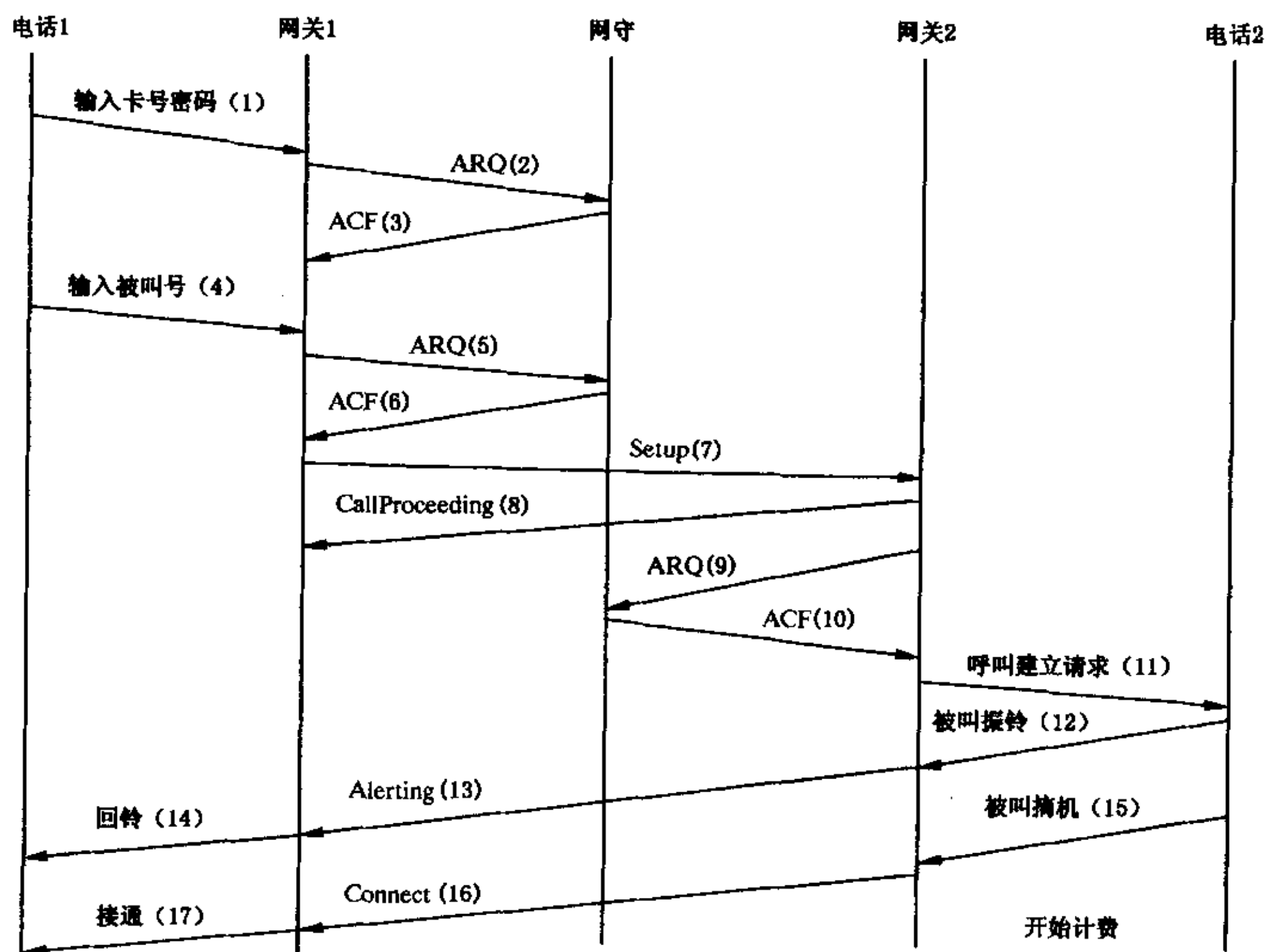


图 5 快速呼叫流程

- 1) 用户拨打 IP 电话,输入卡号密码。
 - 2) 网关 1 向网守发送 ARQ 消息,进行接入认证,其中应包含主叫号码或卡号(主叫号码采用 E.164 编码)。
 - 3) 网守回送 ACF,接入认证通过。
 - 4) 用户输入被叫号码。
 - 5) 网关 1 向网守发送 ARQ 进行地址解析。
 - 6) 地址解析通过后,网守发送 ACF。
 - 7) 网关 1 向被叫网关 2 发起呼叫建立请求“Setup”,里面包含有 H.245 的通道消息。
 - 8) 网关 2 向网关 1 发送“呼叫进展”(Call Proceeding)消息,里面可以包含有 H.245 的通道信息,也可以没有。
 - 9) 网关 2 同时向网守发送 ARQ 消息。
 - 10) 网守向网关 2 发送认证通过消息 ACF。
 - 11) 网关 2 向被叫发送呼叫建立请求消息。
 - 12) 被叫振铃。
 - 13) 网关 2 向网关 1 发送 Alerting 消息,该消息中,可以包含 H.245 的通道信息,也可以不包含,网关 1 需要识别这两种不同情况。
 - 14) 网关 1 收到 Alerting 消息后,产生回铃音。
 - 15) 被叫摘机。
 - 16) 网关 2 向网关 1 发送“连接”(Connect)消息,里面可以包含有 H.245 的通道信息,也可以不包含,网关 1 需要识别这两种不同情况。
 - 17) 网关 1 收到 Connect 消息,接通主叫。
- 注:在快速呼叫过程中,网关 1 只有在收到 Connect 消息后才能开始发送媒体信息。

6.2.5 非快速呼叫流程

如图 6 所示。

- 1) 用户拨打 IP 电话,输入卡号密码。
- 2) 网关 1 向网守发送 ARQ 消息,进行接入认证。
- 3) 网守回送 ACF,接入认证通过。
- 4) 用户输入被叫号码。
- 5) 网关 1 向网守发送 ARQ 进行地址解析。
- 6) 地址解析通过后,网守发送 ACF。
- 7) 网关 1 向被叫网关 2 发起呼叫建立请求“Setup”。
- 8) 网关 2 向网关 1 发送“呼叫进展”(Call Proceeding)消息,里面可以包含有 H.245 的通道信息,也可以没有。
- 9) 网关 2 同时向网守发送 ARQ 消息。
- 10) 网守向网关 2 发送认证通过消息 ACF。
- 11) 网关 2 向被叫发送呼叫建立请求消息。
- 12) 被叫振铃。
- 13) 网关 2 向网关 1 发送 Alerting 消息,该消息中可以包含 H.245 的通道信息,也可以不包含,网关 1 需要识别这两种不同情况。
- 14) 网关 1 收到 Alerting 消息后,产生回铃音。
- 15) 被叫摘机。
- 16) 网关 2 向网关 1 发送“连接”(Connect)消息,里面可以包含有 H.245 的通道信息,也可以不包含,网关 1 需要识别这两种不同情况,被叫开始计费(建议被叫网关在 Connect 消息中携带 H.245 地

址)。
17) 网关1收到 Connect 消息,接通主叫,主叫开始计费。

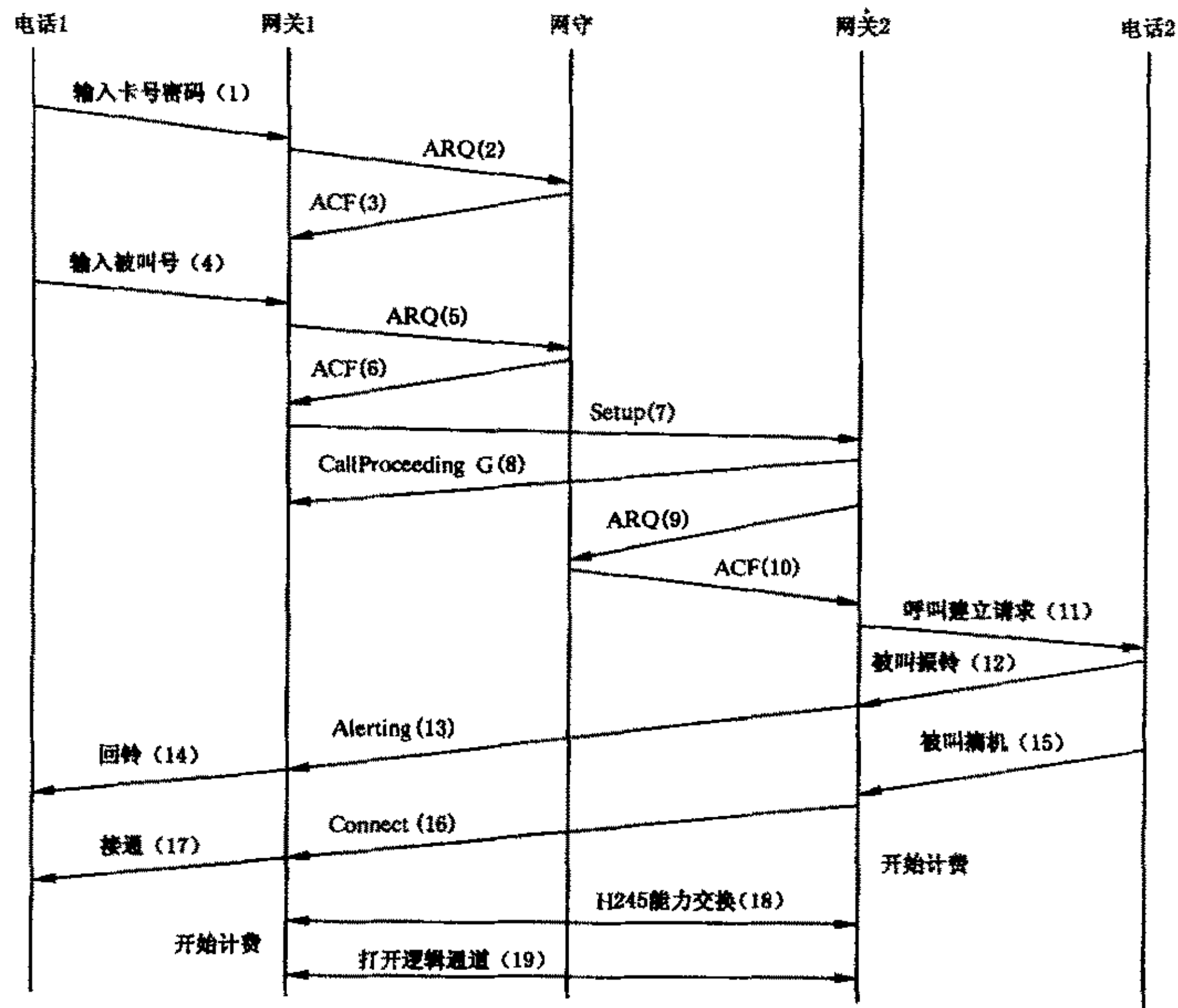


图 6 非快速呼叫流程

- 18) 网关之间进行能力交换。
- 19) 打开逻辑通道。

注: 在非快速连接中,不再作主从判决,约定主叫为主,被叫为从。若在 Connect 消息后,H. 245 能力交换或打开逻辑通道失败,网关可将 DRQ 消息的 terminalCause 置为 establishedFailed,网守可以根据它不对用户进行计费。

6.2.6 快速呼叫与非快速呼叫转换

如果在呼叫建立过程中,网关1直至 connect 消息仍没有得到对方网关对快速呼叫能力的支持的响应,则在 connect 消息之后转向非快速呼叫,进行 H. 245 的能力交换并开启 H. 245 通道,如图 7 所示。

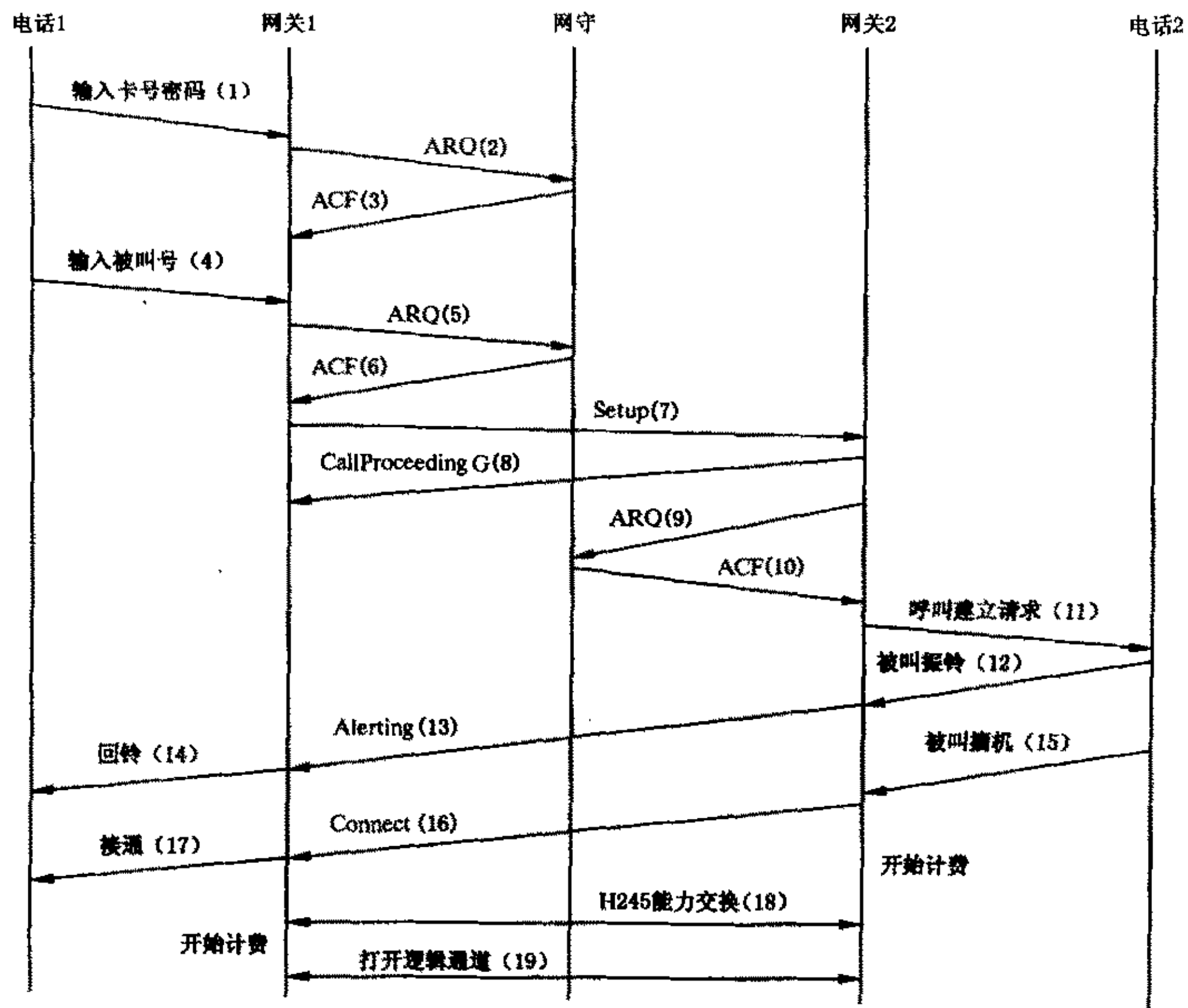


图 7 快速呼叫向非快速呼叫的转换流程

6.3 呼叫释放流程

呼叫释放应采用互不控方式,分 3 种情况:

- 1) 主叫用户先挂机;
- 2) 被叫用户先挂机;
- 3) 网络故障引起的释放。

当计费采集点收到以下 3 类消息的时候,开始进行停止计费处理。

- 1) 收到 PSTN 网侧来的 REL 消息;
- 2) 收到 IP 网侧发来的 Release Complete 消息;
- 3) 收到底侧发来的网络故障消息。

6.3.1 呼叫释放流程

以主叫用户先挂机为例。主叫用户挂机的呼叫释放流程如图 8 所示,其中电话 1 为主叫用户,电话 2 为被叫用户。

- 1) 主叫挂机。
- 2) 如果已打开 H. 245 通道,则网关之间要先关闭逻辑通道。
- 3) 如果已打开 H. 245 通道,则关闭逻辑通道后,网关间互送 End Session Command。
- 4) 网关 1 向网关 2 发送 Release complete 消息。
- 5) 网关 2 挂断被叫。
- 6) 网关 1 向网守发送 DRQ。
- 7) 网守向网关 1 发送 DCF。
- 8) 网关 2 向网守发送 DRQ。

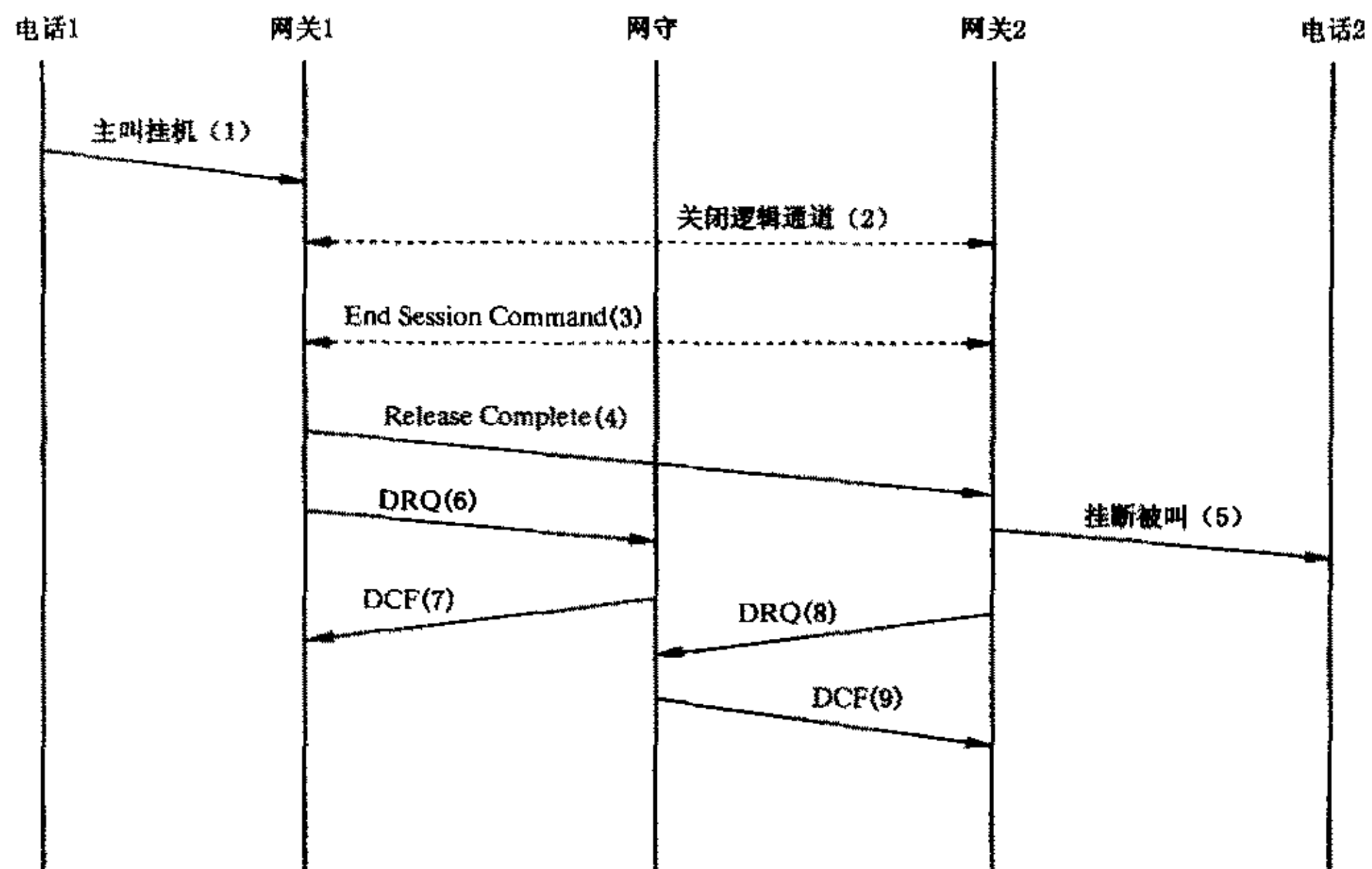


图 8 拆线流程

9) 网守向网关发送 DCF。

注: 6、7、8、9 的顺序无严格要求。

6.3.2 网络故障引起的呼叫释放流程

在通话过程中, 由于网络故障会导致至少其中的一个网关不能收到对方发来的 RTCP 包。如果网关检测到在 1 min 内仍未收到对方网关的 RTCP 包, 则作如图 9 所示的呼叫释放流程(拆线过程可由任一网关发起, 下面以网关 1 发起拆线为例)。

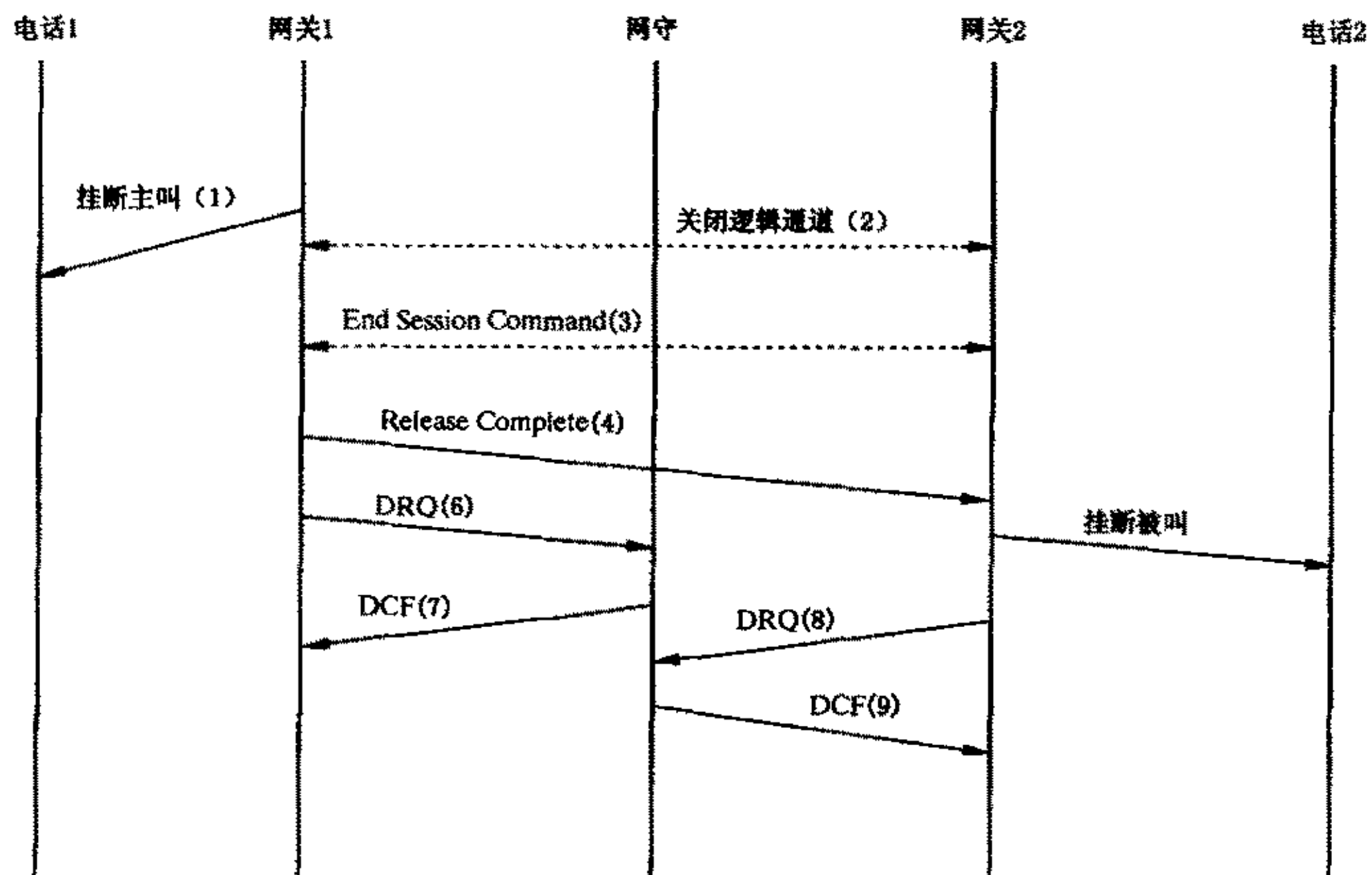


图 9 网络拆线流程

- 1) 网关 1 发拆线消息,挂掉主叫。
- 2) 如果已打开 H. 245 通道,则网关之间要先关闭逻辑通道。
- 3) 如果已打开 H. 245 通道,则关闭逻辑通道后,网关间互送 End Session Command。
- 4) 网关 1 向网关 2 发送 Release complete 消息。
- 5) 网关 2 挂断被叫。
- 6) 网关 1 向网守发送 DRQ。
- 7) 网守向网关 1 发送 DCF。
- 8) 网关 2 向网守发送 DRQ。
- 9) 网守向网关发送 DCF。

注:6、7、8、9 的顺序无严格要求。

7 RAS 消息内容及需要确定的具体参数

RAS 消息是网关和网守之间的注册(Registration)、接入认证(Admission)和状态查询(Status)协议。网关发送询问消息的 Request SeqNum 的最高位为 0,网守发送的询问消息的 Request SeqNum 的最高位为 1。

7.1 接入认证、地址解析和修改密码消息

7.1.1 ARQ(Admission Request)

ARQ 是网关向网守发送的用户接入认证/地址解析/修改密码请求消息。在非标准参数中我们添加了 2 项,第一为主叫控制方式,适用于主叫号码用户,第二为卡号控制方式,对应于它分为 3 步,step1 用于接入认证,step2 用于地址解析,step3 用于卡号用户在线修改密码。

表 1 ARQ 消息的主要内容

参 数	必备(M)/任选(O)
RequestSeqNum	M
CallType	M(0,pointTopoint)
CallModel	O
EndpointIdentifier	M(采用 RCF 消息内的 EndpointIdentifier)
DestinationInfo	O
DestCallSignalAddress	O
DestExtraCallInfo	O
SrcInfo	M(主叫认证时为主叫号码)
SrcCallSignalAddress	O
BandWidth	M(100 bit/s 的倍数)
CallReferenceValue	M
NonStandardData	M
CallServices	O
ConferenceID	M
ActiveMC	M(为 False)
AnswerCall	M(主叫为 False、被叫为 True)
CanMapAlias	O
CallIdentifier	O(全局唯一的标志,与 Q. 931 消息中 UUIE 的 CallIdentifier 一致)
SrcAlternatives	O

表 1(完)

参 数		必备(M)/任选(O)	
DestAlternatives		O	
GatekeeperIdentifier		O	
Token		O	
CrptoToken		O	
IntegrityCheckValue		O	
TranspotQQS		O	
WillSupplyUUIEs		O	
ARQ“非标准数据”(NonStandardData)域中增加相应的参数			
主叫控制方式			
参数	M/O	类型	描述
CallerControl	M	SEQUENCE	
卡号控制方式—step1			
CardNumber	M	IA5STRING (SIZE (1..128))(FROM ("0123456789# ", ")))	卡号
Password	M	OCTETSTRING (SIZE(1..32))	密码,采用 MD5 加密
UserName	O	OCTETSTRING (SIZE(1..32))	用户名
卡号控制方式—step2(无任何参数)			
参数	M/O	类型	描述
Step2	M	SEQUENCE	
卡号控制方式—step3(修改密码)			
参数	M/O	类型	描述
Newpassword	M	OCTETSTRING (SIZE(1..32))	新密码值,采用 MD5 算法加密

7.1.2 ACF(Admission Confirm)

网守对 ARQ 的确认回答,对于卡号用户,还需要给出用户余额或最长通话时长。

表 2 ACF 消息的主要内容

参 数	必备(M)/任选(O)
RequestSeqNum	M
BandWidth	M(100 bit/s 的倍数)
CallModel	O
DestCallSignalAddress	O
IrrFrequency	O(可用于指明 IRR 的发送间隔,应<10 s)

表 2(完)

参 数		必备(M)/任选(O)	
NonStandardData		O	
DestinationInfo		O	
DestExtraCallInfo		O	
DestinationType		O	
RemoteExtensionAddress		O	
AlternateEndpoint		O	
Token		O	
CrptoToken		O	
IntegrityCheckValue		O	
TranspotQoS		O	
WillRepondToIRR		O(若置 TRUE,表明网守可对 IRR 消息做确认或拒绝回答,若置为 FALSE,则不对 IRR 消息作回答,即使 IRR 的 needResponse 置为 TRUE)	
UuiesRequested		O(当 RRQ 消息中的 WillSupplyUuies 为 False 时该项必为 False,如果 RRQ 消息中的 WillSupplyUuies 为 True 时该项任意)	
ACF“非标准数据”(NonStandardData)域中增加相应的参数			
分两种情况:一为主叫控制:一为卡号控制。在卡号控制方式下分三步,第一步为接入认证用,返回余额信息;第二步,返回通话时长;第三步是对密码修改的确认。			
主叫控制			
参数	M/O	类型	描述
CallerControl	O	SEQUENCE	
卡号控制方式——step1			
参数	M/O	类型	描述
RemainMoney	O	INTEGER(0..4294967295)	余额,单位:分
卡号控制方式——step2			
参数	M/O	类型	描述
RemainTime	O	INTEGER(0..4294967295)	最长通话时长,单位:s
卡号控制方式——step3			
参数	M/O	类型	描述
PasswordChang	O	SEQUENCE	

7.1.3 ARJ(Admission Reject)

网守对 ARQ 的拒绝回答,并给出拒绝原因。

表 3 ARJ 消息的主要内容

参 数		必备(M)/任选(O)	
RequestSeqNum		M	
RejectReason		M	
NonStandardData		O	
AlternateGatekeeper		O	
AltGKispermanent		O	
Token		O	
CrptoToken		O	
IntegrityCheckValue		O	
ARJ“非标准数据”(NonStandardData)域中增加相应的参数			
参数	M/O	类型	描述
ErrorReason	O	ARJsErrorReason	Radius 拒绝原因： (0) 用户信息不合法(卡号长度不够等) (1) 用户不存在 (2) 用户密码错误 (3) 系统错误 (4) 卡号用户已在线 (5) 主叫用户已被暂停 (6) 备用 (7) 非本地用户不能修改密码 (8) 用户余额不足

7.2 网关注册消息

7.2.1 RRQ(Registration Request)

网关向网守的注册登记请求消息,网关必须定期(小于 RCF 的 timetolive 确定的时间)向网守发送 RRQ 消息,以表明网关仍然存活,具体的超时和重发次数要求见 RAS 消息的定时器及重发次数。消息内容见表 4。

网关在首次注册时应将 RRQ 消息中的 discoverycomplete 置 0,其余报告其存活的 RRQ 消息的 discoverycomplete 置 1。

表 4 RRQ 消息内容

参 数	必备(M)/任选(O)
RequestSeqNum	M
ProtocolIdentifier	M
NonStandardData	O
DiscoveryComplete	M(第一次为 False 以后各次为 True)
CallSignalAddress	M
RasAddress	M
TerminalType	M

表 4(完)

参 数	必备(M)/任选(O)
TerminalAlias	O
GatekeeperIdentifier	O
EndpointVendor	M
AlternateEndpoints	O
Time ToLive	O(不大于为 30 s)
Tokens	O
CryptoTokens	O
IntegrityCheckValue	O
KeepAlive	O
EndpointIdentifier	O
WillSupplyUUIEs	O(该项为 False 时:IRQ 与 ACF 中的 UuiesRequested 必为 False; 该项为 True 时:IRQ 与 ACF 中的 UuiesRequested 任意)

7.2.2 RCF (Registration Confirm)

网守对 RRQ 消息的确认回答。消息内容见表 5。

表 5 RCF 消息内容

参 数	必备(M)/任选(O)
RequestSeqNum	M
ProtocolIdentifier	M
NonStandardData	O
CallSignalAddress	M
TerminalAlias	O
GatekeeperIdentifier	O
EndpointIdentifier	M
AlternateGatekeeper	O
TimeToLive	O(小于等于 RRQ 消息中的 TimeToLive 值)
Tokens	O
CryptoTokens	O
IntegrityCheckValue	O
WillRespondToIRR	O(该值为 False 时网关收到 IRQ 时回的 IRR 消息中的 NeedResponse 必为 False; 该值为 True 时网关收到 IRQ 时回的 IRR 消息中的 NeedResponse 任意)
PreGrantedARQ	O

7.2.3 RRJ(Registration Reject)

网守 RRQ 消息的拒绝回答,见表 6。

表 6 网守发送的 RRJ 消息内容

参 数	必备(M)/任选(O)
RequestSeqNum	M
ProtocolIdentifier	M
NonStandardData	O
RejectReason	M
GatekeeperIdentifier	O
AlternateGatekeeper	O
AltGKispermanent	O
TimeToLive	O
Tokens	O
CryptoTokens	O
IntegrityCheckValue	O

7.3 呼叫脱离消息

7.3.1 DRQ(DisengageRequest)

网关与网守之间的呼叫脱离请求命令。当该命令由网关发起时,本消息应同时传递计费信息。计费信息放在“非标准数据”(NonStandardData)中。若是网守向网关发送 DRQ 消息,该消息不包含计费信息,并且网关必须回应 DCF。

表 7 DRQ 消息的主要内容

参 数		必备(M)/任选(O)	
RequestSeqNum		M	
EndpointIdentifier		M(采用 RCF 消息的 EndpointIdentifier)	
ConferenceID		M(与主叫网关的 MAC 地址和时间信息有关的随机数)	
CallReferenceValue		M	
DisengageReason		M	
NonStandardData		O	
CallIdentifier		O	
GatekeeperIdentifier		O	
Token		O	
CrptoToken		O	
IntegrityCheckValue		O	
AnsweredCall		O(主叫为 False、被叫为 True)	
DRQ“非标准数据”(NonStandardData)域中增加相应的参数			
参数	M/O	类型	描述
SrcCallSignalAddress	M	TransportAddress	主叫网关地址
DestCallSignalAddress	M	TransportAddress	被叫网关地址
AcctStart Time	M	INTEGER (0..4294967295)	计费开始时间,单位:绝对秒,以 1970 年 1 月 1 日 0 点 0 分 0 秒开始

表 7(完)

参数	M/O	类型	描述
AcctSessionTime	M	INTEGER (0..4294967295)	通话时长,单位为 s
ServiceType	M	DRQsServiceType	呼叫方式: (0) 电话 (1) IP 传真
InBytes	O	INTEGER (0..4294967295)	入字节数
OutBytes	O	INTEGER (0..4294967295)	出字节数
EnCodecType	O	AudioCapability	编码方式
DeCodecType	O	AudioCapability	解码方式
TerminationCause	M	DRQsTerminationCause	呼叫终止原因 (0) 主叫挂断 (1) 被叫挂断 (2) 网络异常中断 (3) 呼叫建立失败

7.3.2 DCF(DisengageConfirm)

对 DRQ 的确认回答。

表 8 DCF 消息的主要内容

参 数	必备(M)/任选(O)
RequestSeqNum	M
NonStandardData	O
Token	O
CrptoToken	O
IntegrityCheckValue	O

7.3.3 DRJ(Disengage Reject)

网守对 DRQ 的拒绝回答,并给出拒绝原因。

表 9 DRJ 消息的主要内容

参 数	必备(M)/任选(O)
RequestSeqNum	M
RejectReason	M
NonStandardData	O
AlternateGatekeeper	O
AltGKispermanent	O
token	O
crptoToken	O
IntegrityCheckValue	O

7.4 状态消息

7.4.1 IRQ

网守向网关发的询问某一话路或所有话路的状态请求消息。若 callReferenceValue 为 0, 则网关需要在同一条 IRR 消息中报告所有呼叫的状态信息。CallReferenceValue 为 0 的 IRQ 的发送间隔应大于 10 s。

表 10 IRQ 消息的主要内容

参 数	必备(M)/任选(O)
RequestSeqNum	M
CallReferenceValue	M
NonStandardData	O
ReplyAddress	O
CallIdentifier	O
Tokens	O
CrptoTokens	O
IntegrityCheckValue	O
UuiesRequested	O(网守可以要求网关报告需要的 Q.931 消息,若 ARQ 消息的 will-SupplyUUIEs 置为 TRUE,则网关必须报告,若为 FALSE,在网关可以不报告)

7.4.2 IRR

网关根据 ACF 命令设定的间隔或 IRQ 请求向网守发送的状态消息。

表 11 IRR 消息的主要内容

参 数	必备(M)/任选(O)
nonStandardData	O
RequestSeqNum	M
endpointType	M
endpointIdentifier	M
rasAddress	M
callSignalAddress	M
EndpointAlias	O
PerCallInfo	M
Tokens	O
CrptoTokens	O
IntegrityCheckValue	O
NeedResponse	O(该项为 False 时:不用应答; 该项为 True 时:需应答)

7.4.3 IACK

对 IRR 的证实消息。

表 12 IACK 消息的主要内容

参 数	必备(M)/任选(O)
RequestSeqNum	M
NonStandardData	O
Tokens	O
CrptoTokens	O
IntegrityCheckValue	O

7.4.4 INAK

对 IRR 的拒绝消息。

表 13 INAK 消息的主要内容

参 数	必备(M)/任选(O)
RequestSeqNum	M
nonStandardData	O
nakReason	M
alternateGatekeeper	O
altGkisPermanent	M
tokens	O
crptoTokens	O
integrityCheckValue	O

7.5 网关资源报告消息

7.5.1 RAI

网关向网守发送的资源可利用性报告。

表 14 RAI 消息的主要内容

参 数	必备(M)/任选(O)
RequestSeqNum	M
ProtocolIdentifier	M
NonStandardData	O
EndpointIdentifier	M
Protocols	M
AlmostOutOfResources	M
Token	O
CrptoToken	O
IntegrityCheckValue	O

7.5.2 RAC

网守对 RAC 的确认消息。

表 15 RAC 消息的主要内容

参 数	必备(M)/任选(O)
RequestSeqNum	M
ProtocolIdentifier	M
NonStandardData	O
Token	O
CrptoToken00000	O
IntegrityCheckValue	O

7.6 RAS 的请求进展消息(RIP)

RIP 消息是当终端收到一个请求消息后,如果判断在相应的超时(timeout)时间内不能及时返回回答消息,则该终端可通过发送 RIP 消息以延长对方等待时间,这个等待时间由 RIP 消息的 delay 域决定。对端在 timeout 加 delay 的时间内若没有收到回应则作超时处理。

表 16 RIP 消息的主要内容

参 数	必备(M)/任选(O)
RequestSeqNum	M
NonStandardData	O
token	O
CrptoToken	O
IntegrityCheckValue	O
delay	M(以 ms 为单位)

7.7 带宽管理消息

7.7.1 BRQ

网关与网守之间的带宽改变的请求的消息,当网守具备带宽管理能力时,则带宽管理消息是有实用意义的。由于 ARQ 消息的 bandwidth 所取的值总是大于每一话路实际占用的带宽,因此为了能使网守掌握网关的带宽利用情况,网关应根据实际带宽利用情况利用 BRQ 消息改变带宽,以便释放多余的带宽或请求增加带宽。若是利用 BRQ 消息增加带宽,则网关必须等待网守的确认。

表 17 BRQ 消息的主要内容

参 数	必备(M)/任选(O)
RequestSeqNum	M
EndpointIdentifier	M
ConferenceID	M
CallreferenceValue	M
CallType	O
BandWidth	M
NonStandardData	O
CallIdentifier	O(全局唯一的标志,与 Q.931 消息中 UIIE 的 CallIdentifier 一致)
GatekeeperIdentifier	O
Token	O
CrptoToken	O
IntegrityCheckValue	O
AnsweredCall	O

7.7.2 BCF

网关或 PC 与网守之间的带宽改变的确认的消息。

表 18 BCF 消息的主要内容

参 数	必备(M)/任选(O)
RequestSeqNum	M
BandWidth	M
NonStandardData	O
token	O
CrptoToken	O
IntegrityCheckValue	O

7.7.3 BRJ

网关或 PC 与网守之间的带宽改变的拒绝的消息。

表 19 BRJ 消息的主要内容

参 数	必备(M)/任选(O)
RequestSeqNum	M
RejectReason	M
AllowedBandWidth	M
NonStandardData	O
AlternateGatekeeper	O
AltGKisPermanent	O
Token	O
CrptoToken	O
IntegrityCheckValue	O

7.8 注销消息

7.8.1 URQ(Unregistration Request)

URQ 消息用于网关和网守之间登记的注销。

表 20 URQ 消息内容

参 数	必备(M)/任选(O)
RequestSeqNum	M
NonStandardData	O
EndpointAlias	O
EndpointIdentifier	O
AlternateEndpoints	O
GatekeeperIdentifier	O
Tokens	O
CryptoTokens	O
IntegrityCheckValue	O
Reason	O

7.8.2 UCF(UnregistrationConfirm)

网关或网守对 URQ 消息的确认回答

表 21 UCF 消息内容

参 数	必备(M)/任选(O)
RequestSeqNum	M
NonStandardData	O
Tokens	O
CryptoTokens	O
IntegrityCheckValue	O

7.8.3 URJ(UnregistrationReject)

网关或网守对 URQ 消息的拒绝回答。

表 22 URJ 消息内容

参 数	必备(M)/任选(O)
RequestSeqNum	M
RejectReason	M
NonStandardData	O
AlternateGatekeeper	O
AltGKispermanent	O
Tokens	O
CryptoTokens	O
IntegrityCheckValue	O

7.9 对 ARQ 中 Password 的加密描述

采用算法 MD5 算法。

MD5 算法中需要两个输入值,一串无须定长的字符串,一个对称密钥,计算后得出一串 16 位长的字符串。公式如下:

$\text{Result} = \text{MD5}(\text{string1}, \text{key})$

具体应用中, string1 取值 ARQ 中的卡号,加密方和解密方的 key 必须一样。

加密过程

用原密码 P 同 MD5 的结果 Result 按位异或,得出加密后的密码 P',公式如下:

$P'[0] = P[0] \text{ xor } \text{Result}[0];$

$P'[1] = P[1] \text{ xor } \text{Result}[1];$

.....

$P'[\text{strlen}(P)-1] = P[\text{strlen}(P)-1] \text{ xor } \text{Result}[\text{strlen}(P)-1]$

解密过程

用加密后的密码 P' 同 MD5 的结果 Result 按位异或,得出密码 P'',公式如下:

$P''[0] = P'[0] \text{ xor } \text{Result}[0];$

$P''[1] = P'[1] \text{ xor } \text{Result}[1];$

.....

$P''[\text{strlen}(P)-n] = P'[\text{strlen}(P)-n] \text{ xor } \text{Result}[\text{strlen}(P)-n]$

直到 P''[n] 为结束符为止。

得到的密码 P" 即为原密码 P。

7.10 Q.931 消息的 UUIE 部分

Q.931 消息中交换的电话号码信息,以 Q.931 消息的域内消息为主。

7.10.1 Setup-UUIE

信息单元	M/O	描述
ProtocolIdentifier	M	必填
h245Address	O	可选
SourceAddress	O	填写主叫号码或卡号,对于卡号用户还需要填写卡号(可选)
SourceInfo	M	主叫信息
DestinationAddress	O	被叫号码(可选)
DestCallSignalAddress	O	被叫网关 IP 地址(可选)
ActiveMC	M	False
ConferenceID	M	与 MAC 和时间信息有关的随机数
ConferenceGoal	M	Creat
FastStart	O	对于快速呼叫,必填

7.10.2 CallProceeding-UUIE

信息单元	M/O	描述
ProtocolIdentifier	M	必填
destinationInfo	M	必填
h245Address	O	对于快速呼叫不填,对于非快速呼叫可选
CallIdentifier	O	如选用,则与 RAS 消息中的 callIdentifier 一致
FastStart	O	对于非快速呼叫不填,对于快速呼叫可选

7.10.3 Alerting-UUIE

信息单元	M/O	描述
ProtocolIdentifier	M	必填
destinationInfo	M	必填
h245Address	O	对于快速呼叫不填,对于非快速呼叫可选
callIdentifier	O	如选用,则与 RAS 消息中的 callIdentifier 一致
FastStart	O	对于非快速呼叫不填,对于快速呼叫可选

7.10.4 Connect-UUIE

信息单元	M/O	描述
ProtocolIdentifier	M	必填
DestinationInfo	M	必填
h245Address	O	快速呼叫不填,非快速呼叫可选。
conferenceID	M	对应 Setup 消息中的 ConferenceID
callIdentifier	O	如选用,则与 RAS 消息中的 callIdentifier 一致
FastStart	O	非快速呼叫不填,快速呼叫可选。

7.10.5 ReleaseComplete-UUIE

信息单元	M/O	描 述	测试情况
ProtocolIdentifier	M	必填	
CallIdentifier	O	如选用,则与 RAS 消息中的 callIdentifier 一致	

8 RAS 消息的定时器和重发次数

表 23 规定了各 RAS 消息对应的定时器值和重发次数。

表 23 RAS 消息对应的定时器值和重发次数

RAS 消息	定时器值(s)	重发次数
RRQ ¹	3	2
URQ	3	1
ARQ	3	1
IRQ	3	1
IRR ²	5	2
DRQ	3	6
RAI	3	2

注

1 RRQ 消息发送周期应小于 RCF 消息中的存活时间(time-to-live-10)。

2 此处的定时器值规定的是网关首先向网守发 IRR 消息情况下(其发送周期根据 ACF 消息的 irrFrequency 确定,若网守在规定的时间内没收到 IRR 包,则发 IRQ 包查询,若网关没回应,网守拆线),网守应在此定时器值时间范围内回以 IACK 证实消息或 INAK 拒绝消息,否则出现超时。

9 Q.931 消息的定时器和重发次数

表 24 规定了各 Q.931 消息对应的定时器值和重发次数。

表 24 Q.931 消息对应的定时器值和重发次数

定时器名称	定时器值(s)
Setup 定时器 1	4
Establishment 定时器 2	≤60(市话),90(长话),120(国际)

注

1 Setup 定时器值是指主叫网关送出 Setup 消息后等待被叫网关回送 ALERTING 或 CALL PROCEEDING 或 CONNECT 或 RELEASE COMPLETE 或其他消息时间。

2 Establishment 定时器值指主叫网关收到 ALERTING 消息等待被叫网关回送 CONNECT 消息的时间或主叫网关中断此次呼叫而送出 RELEASE COMPLETE 消息的时间。

10 互通中需注意的其他问题

- 1) 可以将 ACF 里的被叫号码作为 Q.931 消息里的真正的被叫号码。
- 2) 每个 RRQ 消息都应有 timeToLive 这一项。

- 3) 在非快速呼叫里,应在收到 OLC ACK 后打开 RTP 通道。
- 4) 网守发送 IRQ 项网关讯问某一话路状态时,若网关已无该话路的呼叫参考值,则网关回空的 IRR。
- 5) 发送端的语音增益
有待于进一步研究。
- 6) 接收端的语音增益
有待于进一步研究。

附录 A

(标准的附录)

H. 323 消息的 ASN.1 语言描述

H323-MESSAGES DEFINITIONS AUTOMATIC TAGS ::=
 BEGIN

IMPORTS

SIGNED {},
 ENCRYPTED {},
 HASHED {},
 ChallengeString,
 timestamp,
 Random Val,
 Password,
 EncodedPwdCertToken,
 ClearToken,
 CryptoToken,
 AuthenticationMechanism
 FROM H235-SECURITY-MESSAGES
 Capability,
 MultiplexCapability,
 OpenLogicalChannel

FROM MULTIMEDIA-SYSTEM-CONTROL;

H323-UserInformation ::= SEQUENCE

```
{
  h323-uu-pdu H323-UU-PDU,
  user-data SEQUENCE
  {
    protocol-discriminator INTEGER (0..255),
    user-information OCTET STRING (SIZE(1..131)),
    ...
  } OPTIONAL,
  ...
}
```

H323-UU-PDU ::= SEQUENCE

```
{
  h323-message-body CHOICE
  {
    setup Setup-UUIE,
```

```

    callProceeding    CallProceeding-UUIE,
    connect           Connect-UUIE,
    alerting          Alerting-UUIE,
    userInformation   UI-UUIE,
    releaseComplete   ReleaseComplete-UUIE,
    facility          Facility-UUIE,
    ...,
    progress          Progress-UUIE,
    empty             NULL
},
nonStandardData      NonStandardParameter OPTIONAL,
...,
h4501supplementaryService SEQUENCE OF OCTET STRING OPTIONAL
}

```

Alerting-UUIE ::= SEQUENCE

```

{
    protocolIdentifier    ProtocolIdentifier,
    destinationInfo       EndpointType
    h245Address           TransportAddress OPTIONAL,
    ...,
    callIdentifier        CallIdentifier,
    h245SecurityMode      H245Security OPTIONAL,
    tokens                SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens          SEQUENCE OF CryptoH323Token OPTIONAL,
    fastStart             SEQUENCE OF OpenLogicalChannel OPTIONAL,
    fastCap               FastCap OPTIONAL
}

```

CallProceeding-UUIE ::= SEQUENCE

```

{
    protocolIdentifier    ProtocolIdentifier,
    destinationInfo       EndpointType,
    h245Address           TransportAddress OPTIONAL,
    ...,
    callIdentifier        CallIdentifier,
    h245SecurityMode      H245Security OPTIONAL,
    tokens                SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens          SEQUENCE OF CryptoH323Token OPTIONAL,
    fastStart             SEQUENCE OF OpenLogicalChannel OPTIONAL,
    fastCap               FastCap OPTIONAL
}

```

Connect-UUIE ::=SEQUENCE

```
{
    protocolIdentifier    ProtocolIdentifier,
    h245Address           TransportAddress OPTIONAL,
    destinationInfo       EndpointType,
    conferenceID          ConferenceIdentifier,
    ...,
    callIdentifier        CallIdentifier,
    h245SecurityMode      H245Security OPTIONAL,
    tokens                SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens          SEQUENCE OF CryptoH323Token OPTIONAL,
    fastStart             SEQUENCE OF OpenLogicalChannel OPTIONAL,
    fastCap               FastCap OPTIONAL
}
```

UI-UUIE ::=SEQUENCE

```
{
    protocolIdentifier    ProtocolIdentifier,
    ...
}
```

ReleaseComplete-UUIE ::=SEQUENCE

```
{
    protocolIdentifier    ProtocolIdentifier,
    reason               ReleaseCompleteReason OPTIONAL,
    ...,
    callIdentifier        CallIdentifier
}
```

ReleaseCompleteReason ::=CHOICE

```
{
    noBandwidth           NULL,
    gatekeeperResources   NULL,
    unreachableDestination NULL,
    destinationRejection  NULL,
    invalidRevision       NULL,
    noPermission          NULL,
    unreachableGatekeeper NULL,
    gatewayResources      NULL,
    badFormatAddress      NULL,
    adaptiveBusy          NULL,
    inConf                NULL,
    undefinedReason       NULL,
}
```

```

...,
facilityCallDeflection      NULL,
securityDenied              NULL,
calledPartyNotRegistered    NULL,
callerNotRegistered         NULL
}

```

Setup-UUIE ::= SEQUENCE

```

{
    protocolIdentifier      ProtocolIdentifier,
    h245Address              TransportAddress OPTIONAL,
    sourceAddress            SEQUENCE OF AliasAddress OPTIONAL,--(必备)
    sourceInfo               EndpointType,
    destinationAddress      SEQUENCE OF AliasAddress OPTIONAL,--(必备)
    destCallSignalAddressTransportAddress OPTIONAL,
    destExtraCallInfo        SEQUENCE OF AliasAddress OPTIONAL,
    destExtraCRV             SEQUENCE OF CallReference Value OPTIONAL,
    activeMC                 BOOLEAN,
    conferenceID             ConferenceIdentifier,
    conferenceGoal           CHOICE
    {
        create              NULL,
        join                 NULL,
        invite              NULL,
        ...,
        capability-negotiation      NULL,
    },
    callIndependensSupplementaryService      NULL
},
callServices QseriesOptions OPTIONAL,
callType   CallType,
...,
sourceCallSignalAddressTransportAddress OPTIONAL,
remoteExtensionAddressAliasAddress OPTIONAL,
callIdentifier      CallIdentifier,
h245SecurityCapability      SEQUENCE OF H245Security OPTIONAL,
tokens                   SEQUENCE OF ClearToken OPTIONAL,
cryptoTokens            SEQUENCE OF CryptoH323Token OPTIONAL,
fastStart               SEQUENCE OF OpenLogicalChannel OPTIONAL,
fastCap                 FastCap OPTIONAL
canOverlapSend          BOOLEAN
}

```

Facility-UUIE ::= SEQUENCE

```

{
    protocolIdentifier      ProtocolIdentifier,
    alternativeAddress      TransportAddress OPTIONAL,
    alternativeAliasAddress SEQUENCE OF AliasAddress OPTIONAL,
    conferenceID            ConferenceIdentifier OPTIONAL,
    reason                  FacilityReason,
    ...,
    callIdentifier          CallIdentifier,
    destExtraCallInfo       SEQUENCE OF AliasAddress OPTIONAL,
    remoteExtensionAddressAliasAddress OPTIONAL,
    tokens                  SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens            SEQUENCE OF CryptoH323Token OPTIONAL,
    conferences             SEQUENCE OF ConferenceList OPTIONAL
}

```

ConferenceList ::= SEQUENCE

```

{
    conferenceID      ConferenceIdentifier OPTIONAL,
    conferenceAlias    AliasAddress OPTIONAL,
    nonStandardData    NonStandardParameter OPTIONAL,
    ...
}

```

FacilityReason ::= CHOICE

```

{
    routeCallToGatekeeper NULL, --call must use gatekeeper model
                                --gatekeeper is alternativeAddress
    callForwarded          NULL,
    routeCallToMC          NULL,
    undefinedReason        NULL,
    ...,
    conferenceListChoice   NULL
}

```

Progress-UUIE ::= SEQUENCE

```

{
    protocolIdentifier      ProtocolIdentifier,
    destinationInfo        EndpointType,
    h245Address            TransportAddress OPTIONAL,
    callIdentifier          CallIdentifier,
    h245SecurityMode        H245Security OPTIONAL,
    ...,
    tokens                  SEQUENCE OF ClearToken OPTIONAL,
}

```

```

cryptoTokens    SEQUENCE OF CryptoH323Token OPTIONAL,
fastStart       SEQUENCE OF OpenLogicalChannel OPTIONAL,
fastCap         FastCap OPTIONAL
}

```

TransportAddress ::= CHOICE

```

{
    ipAddress    SEQUENCE
    {
        ip        OCTET STRING(SIZE(4)),
        port      INTEGER(0..65535)
    },
    inSourceRoute SEQUENCE
    {
        ip        OCTET STRING(SIZE(4)),
        port      INTEGER(0..65535)
        route      SEQUENCE OF OCTET STRING(SIZE(4)),
        routing    CHOICE
        {
            strictNULL,
            loose    NULL,
            ...
        },
        ...
    },
    ipxAddress    SEQUENCE
    {
        node      OCTET STRING(SIZE(6)),
        netnum     OCTET STRING(SIZE(4)),
        port      OCTET STRING(SIZE(2))
    },
    ip6Address    SEQUENCE
    {
        ip        OCTET STRING(SIZE(16)),
        port      INTEGER(0..65535),
        ...
    },
    netBios       OCTET STRING(SIZE(16)),
    nsap          OCTET STRING(SIZE(1..20)),
    nonStandardAddress NonStandardParameter,
    ...
}

```

EndpointType ::= SEQUENCE

```
{
    nonStandardData    NonStandardParameter OPTIONAL,
    vendor              VendorIdentifier OPTIONAL,
    gatekeeper          GatekeeperInfo OPTIONAL,
    gateway             GatewayInfo OPTIONAL,
    mcu                 McuInfo OPTIONAL, --mc must be set as well
    terminal            TerminalInfo OPTIONAL,
    mc                  BOOLEAN,      --shall not be set by itself
    undefinedNode       BOOLEAN,
    ...
}
```

GatewayInfo ::= SEQUENCE

```
{
    protocol            SEQUENCE OF SupportedProtocols OPTIONAL,
    nonStandardData     NonStandardParameter OPTIONAL,
    ...
}
```

SupportedProtocols ::= CHOICE

```
{
    nonStandardData     NonStandardParameter,
    H310                H310Caps,
    H320                H320Caps,
    H321                H321Caps,
    H322                H322Caps,
    H323                H323Caps,
    H324                H324Caps,
    voice               VoiceCaps,
    t120-only           T120OnlyCaps,
    ...,
    nonStandardProtocol NonStandardProtocol
}
```

H310Caps ::= SEQUENCE

```
{
    nonStandardData     NonStandardParameter OPTIONAL,
    ...,
    dataRatesSupported SEQUENCE OF DataRate OPTIONAL,
    supportedPrefixes   SEQUENCE OF SupportedPrefix
}
```


H320Caps ::= SEQUENCE

```
{
    nonStandardData    NonStandardParameter OPTIONAL,
    ...,
    dataRatesSupportedSEQUENCE OF DataRate OPTIONAL,
    supportedPrefixes   SEQUENCE OF SupportedPrefix
}
```

H321Caps ::= SEQUENCE

```
{
    nonStandardData    NonStandardParameter OPTIONAL,
    ...,
    dataRatesSupportedSEQUENCE OF DataRate OPTIONAL,
    supportedPrefixes   SEQUENCE OF SupportedPrefix
}
```

H322Caps ::= SEQUENCE

```
{
    nonStandardData    NonStandardParameter OPTIONAL,
    ...,
    dataRatesSupportedSEQUENCE OF DataRate OPTIONAL,
    supportedPrefixes   SEQUENCE OF SupportedPrefix
}
```

H323Caps ::= SEQUENCE

```
{
    nonStandardData    NonStandardParameter OPTIONAL,
    ...,
    dataRatesSupportedSEQUENCE OF DataRate OPTIONAL,
    supportedPrefixes   SEQUENCE OF SupportedPrefix
}
```

H324Caps ::= SEQUENCE

```
{
    nonStandardData    NonStandardParameter OPTIONAL,
    ...,
    dataRatesSupportedSEQUENCE OF DataRate OPTIONAL,
    supportedPrefixes   SEQUENCE OF SupportedPrefix
}
```

VoiceCaps ::= SEQUENCE

```
{
    nonStandardData    NonStandardParameter OPTIONAL,
```

```

    ...,
    dataRatesSupportedSEQUENCE OF DataRate OPTIONAL,
    supportedPrefixes    SEQUENCE OF SupportedPrefix
}

```

T120OnlyCaps ::=SEQUENCE

```

{
    nonStandardData    NonStandardParameter OPTIONAL,
    ...,
    dataRatesSupportedSEQUENCE OF DataRate OPTIONAL,
    supportedPrefixes    SEQUENCE OF SupportedPrefix
}

```

NonStandardProtocol ::=SEQUENCE

```

{
    nonStandardData    NonStandardParameter OPTIONAL,
    dataRatesSupportedSEQUENCE OF DataRate OPTIONAL,
    supportedPrefixes    SEQUENCE OF SupportedPrefix,
    ...
}

```

MeuInfo ::=SEQUENCE

```

{
    nonStandardData    NonStandardParameter OPTIONAL,
    ...
}

```

TerminalInfo ::=SEQUENCE

```

{
    nonStandardData    NonStandardParameter OPTIONAL,
    ...
}

```

GatekeeperInfo ::=SEQUENCE

```

{
    nonStandardData    NonStandardParameter OPTIONAL,
    ...
}

```

VendorIdentifier ::=SEQUENCE

```

{
    vendor            H221NonStandard,
    productId          OCTET STRING (SIZE(1..256))OPTIONAL,--per vendor
}

```

```

    versionId OCTET STRING (SIZE(1..256))OPTIONAL,--per product
    ...
}

```

H221NonStandard ::=SEQUENCE

```

{  t35CountryCode      INTEGER(0..255),    --country,as per T.35
   t35Extension        INTEGER(0..255),    --assigned nationally
   manufacturerCode    INTEGER(0..65535),  --assigned nationally
   ...
}

```

NonStandardParameter ::=SEQUENCE

```

{
    nonStandardIdentifier NonStandardIdentifier,
    data                  OCTET STRING
}

```

NonStandardIdentifier ::=CHOICE

```

{
    object          OBJECT IDENTIFIER,
    h221NonStandard H221NonStandard,
    ...
}

```

AliasAddress ::=CHOICE

```

{
    e164          IA5String (SIZE(1..128))(FROM("0123456789#* ;")),
    h323-ID       BMPString(SIZE(1..256)),
    ...,
    url-ID        IA5String (SIZE(1..512)),
    transportID   TransportAddress,
    email-ID      IA5String (SIZE(1..512)),
    partyNumber   PartyNumber
}

```

PartyNumber ::=CHOICE

```

{
    publicNumber      PublicPartyNumber,
                      --the numbering plan is according to
                      --Recommendations E.163 and E.164.
    dataPartyNumber   NumberDigits,
                      --not used,value reserved.
}

```

```

    telexPartyNumber    NumberDigits,
                        --not used,value reserved.
    privateNumber        PrivatePartyNumber,
    nationalStandardPartyNumber NumberDigits,
                        --not used,value reserved.
    ...
}

PublicPartyNumber ::=SEQUENCE
{
    publicTypeOfNumber    PublicTypeOfNumber,
    publicNumberDigits    NumberDigits
}

PrivatePartyNumber ::=SEQUENCE
{
    privateTypeOfNumber    PrivateTypeOfNumber,
    privateNumberDigits    NumberDigits
}

NumberDigits ::=IA5String (SIZE(1..128))(FROM("0123456789#*,"))

PublicTypeOfNumber ::=CHOICE
{
    unknown                NULL,
                        --if used number digits carry prefix indicating type
                        --of number according to national recommendations.
    internationalNumber    NULL,
    nationalNumber         NULL,
    networkSpecificNumber  NULL,
                        --not used,value reserved
    subscriberNumber       NULL,
    abbreviatedNumber NULL,
                        --valid only for called party number at the outgoing
                        --access,network substitutes appropriate number.
    ...
}

PrivateTypeOfNumber ::=CHOICE
{
    unknown                NULL,
    level2RegionalNumber  NULL,
    level1RegionalNumber  NULL,

```

```

    pISNSpecificNumber    NULL,
    localNumber            NULL,
    abbreviatedNumber      NULL,
    ...
}

```

Endpoint ::= SEQUENCE

```

{
    nonStandardData        NonStandardParameter OPTIONAL,
    aliasAddress            SEQUENCE OF AliasAddress OPTIONAL,
    callSignalAddress      SEQUENCE OF TransportAddress OPTIONAL,
    rasAddress              SEQUENCE OF TransportAddress OPTIONAL,
    endpointType           EndpointType OPTIONAL,
    tokens                  SEQUENCE OF ClearToken OPTIONAL,
    priority                INTEGER(0..127) OPTIONAL,
    destinationInfo        SEQUENCE OF AliasAddress OPTIONAL,
    destExtraCallInfo      SEQUENCE OF AliasAddress OPTIONAL,
    ...
}

```

AlternateGK ::= SEQUENCE

```

{
    rasAddress              TransportAddress,
    gatekeeperIdentifierGatekeeperIdentifier OPTIONAL,
    needToRegister          BOOLEAN,
    priority                INTEGER(0..127),
    ...
}

```

SecurityServiceMode ::= CHOICE

```

{
    nonStandard            NonStandardParameter,
    none                   NULL,
    default                NULL,
    ...                    --can be extended with other specific modes
}

```

SecurityCapabilities ::= SEQUENCE

```

{
    nonStandard            NonStandardParameter OPTIONAL,
    encryption             SecurityServiceMode,
    authentication         SecurityServiceMode,
    integrity              SecurityServiceMode,
}

```

...

}

H245Security ::= CHOICE

{

nonStandard NonStandardParameter,

noSecurity NULL,

tls SecurityCapabilities,

ipsec SecurityCapabilities,

...

}

FastCap ::= SEQUENCE

{

caps SEQUENCE OF Capability OPTIONAL,

--Capability structure from H. 245

multiplexCap MultiplexCapability OPTIONAL,

--MultiplexCapability structure from H. 245

...

}

QseriesOptions ::= SEQUENCE

{

q932Full BOOLEAN, --if true, indicates full support for Q. 932

q951Full BOOLEAN, --if true, indicates full support for Q. 951

q952Full BOOLEAN, --if true, indicates full support for Q. 952

q953Full BOOLEAN, --if true, indicates full support for Q. 953

q955Full BOOLEAN, --if true, indicates full support for Q. 955

q956Full BOOLEAN, --if true, indicates full support for Q. 956

q957Full BOOLEAN, --if true, indicates full support for Q. 957

q954Info Q954Details,

...

}

Q954Details ::= SEQUENCE

{

conferenceCalling BOOLEAN,

threePartyService BOOLEAN,

...

}

GloballyUniqueID ::= OCTET STRING (SIZE(16))

ConferenceIdentifier ::= GloballyUniqueID

```

RequestSeqNum      ::= INTEGER (1..65535)
GatekeeperIdentifier ::= BMPString (SIZE(1..128))
BandWidth          ::= INTEGER (0.4294967295)--in 100s of bits
CallReference Value ::= INTEGER(0.65535)
EndpointIdentifier  ::= BMPString (SIZE(1..128))
ProtocolIdentifier  ::= OBJECT IDENTIFIER
                    --shall be set to
                    --{itu-t(0) recommendation (0) h (8) 2250 version (0) 2}
TimeToLive          ::= INTEGER (1..4294967296    --in seconds

CallIdentifier ::=SEQUENCE
{
    guid      GloballyUniqueID,
    ...
}

EncryptIntAlg ::=CHOICE
{
    nonStandard      NonStandardParameter,
    isoAlgorithm      OBJECT IDENTIFIER,--defined in ISO/IEC 9979
    ...
}

NonIsoIntegrityMechanism ::=CHOICE
{
    hMAC-MD5          NULL,
    hMAC-iso10118-2-s EncryptIntAlg,--according to ISO/IEC 10118-2 using
                        --EncryptIntAlg as core block encryption algorithm
                        --(short MAC)
    hMAC-iso10118-2-1 EncryptIntAlg,--according to ISO/IEC 10118-2 using
                        --EncryptIntAlg as core block encryption algorithm
                        --(long MAC)
    hMAC-iso10118-3    OBJECT IDENTIFIER,--according to ISO/IEC 10118-3 using
                        --OID as hash function (OID is SHA-1,RIPE-MD160,
                        --RIPE-MD128)
    ...
}

IntegrityMechanism ::=CHOICE
{
    nonStandard      NonStandardParameter,
    digSig           NULL,--indicates to apply a digital signature
    iso9797          OBJECT IDENTIFIER,--according to ISO/IEC 9797 using OID as
                        --core encryption algorithm (X-CBC MAC)

```

```

nonIsoIM      NonIsoIntegrityMechanism,
...
}

ICV ::=SEQUENCE
{
    algorithmOID  OBJECT IDENTIFIER,
    icv           BIT STRING
                --or signature
}

FastStartToken ::=ClearToken(WITH COMPONENTS{...,timestamp,dhkey,generalID})
EncodedFastStartToken ::=TYPE-IDENTIFIER.&Type (FastStartToken)
CryptoH323Token ::=CHOICE
{
    cryptoEPPwdHash SEQUENCE
    {
        alias      AliasAddress,--alias of entity generating hash
        timestamp   timestamp,--timestamp used in hash
        token HASHED{EncodedPwdCertToken 机-generalID set to 'alias' --}
    },
    cryptoGKPwdHash SEQUENCE
    {
        gatekeeperId GatekeeperIdentifier,
        timestamp     timestamp,--timestamp used in hash
        token          HASHED {EncodedPwdCertToken --}
    },
    cryptoEPPwdEncr   ENCRYPTED
    {EncodedPwdCertToken-generalID set to Gatekeeperid--},
    cryptoGKPwdEncr   ENCRYPTED
    {EncodedPwdCertToken-generalID set to Gatekeeperid--},
    cryptoEPCert      SIGNED {EncodedPwdCertToken},
    cryptoGKCert       SIGNED {EncodedPwdCertToken},
    cryptoFastStart    SIGNED {EncodedFastStartToken},
    nestedcryptoToken CryptoH323Token,
    ...
}

DataRate ::=SEQUENCE
{
    nonStandardData  NonStandardParameter OPTIONAL,
    channelRate       BandWidth,
    channelMultiplier INTEGER(1..256) OPTIONAL,

```



```

...
}

SupportedPrefix ::= SEQUENCE
{
    nonStandardData      NonStandardParameter OPTIONAL,
    prefix                AliasAddress,
    ...
}

```

```

RasMessage ::= CHOICE
{
    gatekeeperRequest      GatekeeperRequest,
    gatekeeperConfirm      GatekeeperConfirm,
    gatekeeperReject       GatekeeperReject,
    registrationRequest     RegistrationRequest,
    registrationConfirm     RegistrationConfirm,
    registrationReject      RegistrationReject,
    unregistrationRequest   UnregistrationRequest,
    unregistrationConfirm   UnregistrationConfirm,
    unregistrationReject    UnregistrationReject,
    admissionRequest        AdmissionRequest,
    admissionConfirm        AdmissionConfirm,
    admissionReject         AdmissionReject,
    bandwidthRequest        BandwidthRequest,
    bandwidthConfirm        BandwidthConfirm,
    bandwidthReject         BandwidthReject,
    disengageRequest        DisengageRequest,
    disengageConfirm        DisengageConfirm,
    disengageReject         DisengageReject,
    locationRequest         LocationRequest,
    locationConfirm         LocationConfirm,
    locationReject          LocationReject,
    infoRequest             InfoRequest,
    infoRequestResponse     InfoRequestResponse,
    nonStandardMessage      NonStandardMessage,
    unknownMessageResponse  UnknownMessageResponse,
    ...,
    requestInProgress       RequestInProgress,
    resourcesAvailableIndicate ResourcesAvailableIndicate,
    resourcesAvailableConfirm ResourcesAvailableConfirm,
    infoRequestAck          InfoRequestAck,
    infoRequestNak          InfoRequestNak
}

```

}

GatekeeperRequest ::= SEQUENCE--(GRQ)

{

requestSeqNum	RequestSeqNum,
protocolIdentifier	ProtocolIdentifier,
nonStandardDate	NonStandardParameter OPTIONAL,
rasAddress	TransportAddress,
endpointType	EndpointType,
gatekeeperIdentifier	GatekeeperIdentifier OPTIONAL,
callServices	QseriesOptions OPTIONAL,
endpointAlias	SEQUENCE OF AliasAddress OPTIONAL,
...	
alternateEndpoints	SEQUENCE OF Endpoint OPTIONAL,
tokens	SEQUENCE OF ClearToken OPTIONAL,
cryptoTokens	SEQUENCE OF CryptoH323Token OPTIONAL,
authenticationCapability	SEQUENCE OF AuthenticationMechanism OPTIONAL,
algorithmOIDs	SEQUENCE OF OBJECT IDENTIFIER OPTIONAL,
integrity	SEQUENCE OF IntegrityMechanism OPTIONAL,
integrityCheckValue	ICV OPTIONAL

}

GatekeeperConfirm ::= SEQUENCE--(GCF)

{

requestSeqNum	RequestSeqNum,
protocolIdentifier	ProtocolIdentifier,
nonStandardDate	NonStandardParameter OPTIONAL,
gatekeeperIdentifier	GatekeeperIdentifier OPTIONAL,
rasAddress	TransportAddress,
...	
alternateGatekeeper	SEQUENCE OF AlternateGK OPTIONAL,
authenticationMode	AuthenticationMechanism OPTIONAL,
tokens	SEQUENCE OF ClearToken OPTIONAL,
cryptoTokens	SEQUENCE OF CryptoH323Token OPTIONAL,
algorithmOID	OBJECT IDENTIFIER OPTIONAL,
integrity	SEQUENCE OF IntegrityMechanism OPTIONAL,
integrityCheckValue	ICV OPTIONAL

}

GatekeeperReject ::= SEQUENCE--(GRJ)

{

requestSeqNum	RequestSeqNum,
protocolIdentifier	ProtocolIdentifier,
nonStandardData	NonStandardParameter OPTIONAL,
gatekeeperIdentifier	GatekeeperIdentifier OPTIONAL,
rejectReason	GatekeeperRejectReason,
...	
alternateGatekeeper	SEQUENCE OF AlternateGK OPTIONAL,
altGKisPermanent	BOOLEAN OPTIONAL,
tokens	SEQUENCE OF ClearToken OPTIONAL,
cryptoTokens	SEQUENCE OF CryptoH323Token OPTIONAL,
integrityCheckValue	ICV OPTIONAL

}

GatekeeperRejectReason ::= CHOICE

{		
resourceUnavailable	NULL,	
terminalExcluded	NULL,	--permission failure, not a resource failure
invalidRevision	NULL,	
undefinedReason	NULL,	
...		
securityDenial	NULL	
}		

RegistrationRequest ::= SEQUENCE--(RRQ)

{	
requestSeqNum	RequestSeqNum,
protocolIdentifier	ProtocolIdentifier,
nonStandardData	NonStandardParameter OPTIONAL,
discoveryComplete	BOOLEAN,
callSignalAddress	SEQUENCE OF TransportAddress,
rasAddress	SEQUENCE OF TransportAddress,
terminalType	EndpointType,
terminalAlias	SEQUENCE OF AliasAddress OPTIONAL,
gatekeeperIdentifier	GatekeeperIdentifier OPTIONAL,
endpointVendor	VendorIdentifier,
...	
alternateEndpoints	SEQUENCE OF Endpoint OPTIONAL,
timeToLive	TimeToLive OPTIONAL,
tokens	SEQUENCE OF ClearToken OPTIONAL,
cryptoTokens	SEQUENCE OF CryptoH323Token OPTIONAL,
integrityCheckValue	ICV OPTIONAL,
}	

keepAlive	BOOLEAN,
endpointIdentifier	EndpointIdentifier OPTIONAL,
willSupplyUIEs	BOOLEAN

}

RegistrationConfirm ::= SEQUENCE--(RCF)

{

requestSeqNum	RequestSeqNum,
protocolIdentifier	ProtocolIdentifier,
nonStandardData	NonStandardParameter OPTIONAL,
callSignalAddress	SEQUENCE OF TransportAddress,
terminalAlias	SEQUENCE OF AliasAddress OPTIONAL,
gatekeeperIdentifier	GatekeeperIdentifier OPTIONAL,
endpointIdentifier	EndpointIdentifier,
...	
alternateGatekeeper	SEQUENCE OF AlternateGK OPTIONAL,
timeToLive	TimeToLive OPTIONAL,
tokens	SEQUENCE OF ClearToken OPTIONAL,
cryptoTokens	SEQUENCE OF CryptoH323Token OPTIONAL,
integrityCheckValue	ICV OPTIONAL,
willRespondToIRR	BOOLEAN,
preGrantedARQ	SEQUENCE

{

makeCall	BOOLEAN,
useGKCallSignalAddressToMakeCall	BOOLEAN,
answerCall	BOOLEAN,
useGKCallSignalAddressToAnswer	BOOLEAN,
...	

}OPTIONAL

}

RegistrationReject ::= SEQUENCE--(RRJ)

{

requestSeqNum	RequestSeqNum,
protocolIdentifier	ProtocolIdentifier,
nonStandardData	NonStandardParameter OPTIONAL,
rejectReason	RegistrationRejectReason,
gatekeeperIdentifier	GatekeeperIdentifier OPTIONAL,
...	
alternateGatekeeper	SEQUENCE OF AlternateGK OPTIONAL,
altGKisPermanent	BOOLEAN OPTIONAL,

```

tokens          SEQUENCE OF ClearToken OPTIONAL,
cryptoTokens    SEQUENCE OF CryptoH323Token OPTIONAL,
integrityCheckValue  ICV OPTIONAL
}

```

RegistrationRejectReason ::= CHOICE

```

{
    discoveryRequired    NULL,--registration permission has aged
    invalidRevision      NULL,
    invalidCallSignalAddress  NULL,
    invalidRASAddress    NULL,--supplied address is invalid
    duplicateAlias       SEQUENCE OF AliasAddress,
                        --alias registered to another endpoint
    invalidTerminalType  NULL,
    undefinedReason      NULL,
    transportNotSupported  NULL,--one or more of the transports
    ...,
    transportQOSNotSupported  NULL,--endpoint QoS not supported
    resourceUnavailable    NULL,--gatekeeper resources exhausted
    invalidAlias           NULL,--alias not consistent with gatekeeper rules
    securityDenial         NULL
}

```

UnregistrationRequest ::= SEQUENCE--(URQ)

```

{
    requestSeqNum        RequestSeqNum,
    callSignalAddress    SEQUENCE OF TransportAddress,
    endpointAlias        SEQUENCE OF AliasAddress OPTIONAL,
    nonStandardData      NonStandardParameter OPTIONAL,
    endpointIdentifier    EndpointIdentifier OPTIONAL,
    ...,
    alternateEndpoints   SEQUENCE OF Endpoint OPTIONAL,
    gatekeeperIdentifier  GatekeeperIdentifier OPTIONAL,
    tokens               SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens          SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue   ICV OPTIONAL,
    reason               UnregRequestReason OPTIONAL
}

```

UnregRequestReason ::= CHOICE

```

{
    reregistrationRequired  NULL,

```

```

    ttlExpired          NULL,
    securityDenial       NULL,
    undefinedReason      NULL,
    ...
}

```

UnregistrationConfirm ::= SEQUENCE--(UCF)

```

{
    requestSeqNum        RequestSeqNum,
    nonStandardData       NonStandardParameter OPTIONAL,
    ...,
    tokens                SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens          SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue   ICV OPTIONAL
}

```

UnregistrationReject ::= SEQUENCE--(URJ)

```

{
    requestSeqNum        RequestSeqNum,
    rejectReason          UnregRejectReason,
    nonStandardData       NonStandardParameter OPTIONAL,
    ...,
    alternateGatekeeper   SEQUENCE OF AlternateGK OPTIONAL,
    altGKisPermanent      BOOLEAN OPTIONAL,
    tokens                SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens          SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue   ICV OPTIONAL
}

```

UnregRejectReason ::= CHOICE

```

{
    notCurrentlyRegistered NULL,
    callInProgress         NULL,
    undefinedReason        NULL,
    ...,
    permissionDenied        NULL, --requesting user not allowed to unregister
                                --specified user
    securityDenial          NULL
}

```

AdmissionRequest ::= SEQUENCE--(ARQ)

```

{
    requestSeqNum      RequestSeqNum,
    callType           CallType,
    callModel          CallModel OPTIONAL,
    endpointIdentifier  EndpointIdentifier,
    destinationInfo    SEQUENCE OF AliasAddress OPTIONAL,
    destCallSignalAddress TransportAddress OPTIONAL,
    destExtraCallInfo  SEQUENCE OF AliasAddress OPTIONAL,
    srcInfo            SEQUENCE OF AliasAddress,
    srcCallSignalAddress TransportAddress OPTIONAL,
    bandWidth          BandWidth,
    callReferenceValue  CallReferenceValue,
    nonStandardData     ARQsNonStandardParameter OPTIONAL,
    callServices        QseriesOptions OPTIONAL,
    conferenceID        ConferenceIdentifier,
    activeMC            BOOLEAN,
    answerCall          BOOLEAN,
    ...,
    canMapAlias         BOOLEAN,
    callIdentifier      ,
    srcAlternatives     SEQUENCE OF Endpoint OPTIONAL,
    desAlternatives     SEQUENCE OF Endpoint OPTIONAL,
    gatekeeperIdentifier GatekeeperIdentifier OPTIONAL,
    tokens             SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens        SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue ICV OPTIONAL,
    transportQOS        TransportQOS OPTIONAL,
    willSupplyUUIEs     BOOLEAN
}

```

ARQsNonStandardParameter ::= CHOICE

```

{
    callerControl SEQUENCE
    {
        ...
    },
    cardControl CHOICE
    {
        step1 SEQUENCE
        {
            cardNumber IA5STRING(SIZE(1..128))(FROM("0123456789#*,")),
            password OCTET STRING(SIZE(1..32)),

```

```

        userName    OCTET STRING(SIZE(1..32))OPTIONAL,
        ...
    },
    step2    SEQUENCE
    {
        ...
    },
    step3    SEQUENCE
    {
        newPassword    OCTET STRING(SIZE(1..32)),
        ...
    },
    ...
},
...
}

```

CallType ::= CHOICE

```

{
    pointToPoint    NULL,
    oneToN          NULL,
    nToOne          NULL,
    nToN            NULL,
    ...
}

```

CallModel ::= CHOICE

```

{
    direct          NULL,
    gatekeeperRouted    NULL,
    ...
}

```

TransportQOS ::= CHOICE

```

{
    endpointControlled    NULL,
    gatekeeperControlled    NULL,
    noControl              NULL,
    ...
}

```

AdmissionConfirm ::= SEQUENCE--(ACF)

```

{

```

requestSeqNum	RequestSeqNum,
bandWidth	BandWidth,
callModel	CallModel,
destCallSignalAddress	TransportAddress,
irrFrequency	INTEGER(1..65535)OPTIONAL,
nonStandardData	ACFsNonStandardParameter OPTIONAL,
...	
destinationInfo	SEQUENCE OF AliasAddress OPTIONAL,
destExtraCallInfo	SEQUENCE OF AliasAddress OPTIONAL,
destinationType	EndpointType OPTIONAL,
remoteExtensionAddress	SEQUENCE OF AliasAddress OPTIONAL,
alternateEndpoints	SEQUENCE OF Endpoint OPTIONAL,
tokens	SEQUENCE OF ClearToken OPTIONAL,
cryptoTokens	SEQUENCE OF CryptoH323Token OPTIONAL,
integrityCheckValue	ICV OPTIONAL,
transportQOS	TransportQOS OPTIONAL,
willRespondToIRR	BOOLEAN,
uuiesRequested	UUIEsRequested

}

ACFsNonStandardParameter ::= CHOICE

```
{
  callerControl SEQUENCE
  {
    ...
  },
  cardControl CHOICE
  {
    step1 SEQUENCE
    {
      remainMoney INTEGER(0..4294967295) OPTIONAL,
      ...
    },
    step2 SEQUENCE
    {
      remainTime INTEGER(0..4294967295) OPTIONAL,
      ...
    },
    step3 SEQUENCE
```

```

    {
        ...
    },
    ...
},
...
}

```

UIEsRequested ::= SEQUENCE

```

{
    setup                BOOLEAN,
    callProceeding       BOOLEAN,
    connect              BOOLEAN,
    alerting             BOOLEAN,
    userInformation      BOOLEAN,
    releaseComplete      BOOLEAN,
    facility             BOOLEAN,
    progress            BOOLEAN,
    empty               BOOLEAN,
    ...
}

```

AdmissionReject ::= SEQUENCE--(ARJ)

```

{
    requestSeqNum        RequestSeqNum,
    rejectReason         AdmissionRejectReason,
    nonStandardData      ARJsNonStandardParameter OPTIONAL,
    ...,
    alternateGatekeeper  SEQUENCE OF AlternateGK OPTIONAL,
    altGKisPermanent    BOOLEAN,
    tokens               SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens         SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue  ICV OPTIONAL
}

```

ARJsNonStandardParameter ::= SEQUENCE

```

{
    errorReason    ARJsErrorReason OPTIONAL,
    ...
}

```

ARJsErrorReason ::= CHOICE

```

{

```

```

invalidUser          NULL,
noexistUser          NULL,
passwordError        NULL,
systemError          NULL,
cardOnline           NULL,
userForbidden        NULL,
reserved             NULL,
passwordChangeInvalid NULL,
outOfMoney           NULL,
...
}
AdmissionRejectReason ::= CHOICE
{
    calledPartyNotRegistered    NULL, --can't translate address
    invalidPermission           NULL, --permission has expired
    requestDenied               NULL, --no bandwidth available
    undefinedReason             NULL,
    callerNotRegistered         NULL,
    routeCallToGatekeeper       NULL,
    invalidEndpointIdentifier    NULL,
    resourceUnavailable          NULL,
    ...,
    securityDenial              NULL,
    qosControlNotSupported       NULL,
    incompleteAddress            NULL
}

BandwidthRequest ::= SEQUENCE--(BRQ)
{
    requestSeqNum      RequestSeqNum,
    endpointIdentifier  EndpointIdentifier,
    conferenceID        ConferenceIdentifier,
    callReferenceValue  CallReferenceValue,
    callType            CallType OPTIONAL,
    bandWidth           BandWidth,
    nonStandardData     NonStandardParameter OPTIONAL,
    ...,
    callIdentifier      CallIdentifier,
    gatekeeperIdentifier GatekeeperIdentifier,
    tokens              SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens        SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue ICV OPTIONAL,
    answeredCall        BOOLEAN
}

```

}

BandwidthConfirm ::=SEQUENCE--(BCF)

{

requestSeqNum	RequestSeqNum,
bandWidth	BandWidth,
nonStandardData	NonStandardParameter OPTIONAL,
...	
tokens	SEQUENCE OF ClearToken OPTIONAL,
cryptoTokens	SEQUENCE OF CryptoH323Token OPTIONAL,
integrityCheckValue	ICV OPTIONAL

}

BandwidthReject ::=SEQUENCE--(BRJ)

{

requestSeqNum	RequestSeqNum,
rejectReason	BandRejectReason,
allowedBandWidth	BandWidth,
nonStandardData	NonStandardParameter OPTIONAL,
...	
alternateGatekeeper	SEQUENCE OF AlternateGK OPTIONAL,
altGKisPermanent	BOOLEAN,
tokens	SEQUENCE OF ClearToken OPTIONAL,
cryptoTokens	SEQUENCE OF CryptoH323Token OPTIONAL,
integrityCheckValue	ICV OPTIONAL

}

BandRejectReason ::=CHOICE

{

notBound	NULL,--discovery permission has aged
invalidConferenceID	NULL,--possible revision
invalidPermission	NULL,--true permission violation
insufficientResources	NULL,
invalidRevision	NULL,
undefinedReason	NULL,
...	
securityDenial	NULL

}

LocationRequest ::=SEQUENCE--(LRQ)

{

requestSeqNum	RequestSeqNum,
endpointIdentifier	EndpointIdentifier OPTIONAL,

destinationInfo	SEQUENCE OF AliasAddress,
nonStandardData	NonStandardParameter OPTIONAL,
replyAddress	TransportAddress,
...	
sourceInfo	SEQUENCE OF AliasAddress OPTIONAL,
canMapAlias	BOOLEAN,--can handle alias address
gatekeeperIdentifier	GatekeeperIdentifier OPTIONAL,
tokens	SEQUENCE OF ClearToken OPTIONAL,
cryptoTokens	SEQUENCE OF CryptoH323Token OPTIONAL,
integrityCheckValue	ICV OPTIONAL

LocationConfirm ::=SEQUENCE--(LCF)

requestSeqNum	RequestSeqNum,
callSignalAddress	TransportAddress,
rasAddress	TransportAddress,
nonStandardData	NonStandardParameter OPTIONAL,
...	
destinationInfo	SEQUENCE OF AliasAddress OPTIONAL,
destExtraCallInfo	SEQUENCE OF AliasAddress OPTIONAL,
destinationType	EndpointType OPTIONAL,
remoteExtensionAddress	SEQUENCE OF AliasAddress OPTIONAL,
alternateEndpoints	SEQUENCE OF Endpoint OPTIONAL,
tokens	SEQUENCE OF ClearToken OPTIONAL,
cryptoTokens	SEQUENCE OF CryptoH323Token OPTIONAL,
integrityCheckValue	ICV OPTIONAL

LocationReject ::=SEQUENCE--(LRJ)

requestSeqNum	RequestSeqNum,
rejectReason	LocationRejectReason,
nonStandardData	NonStandardParameter OPTIONAL,
...	
alternateGatekeeper	SEQUENCE OF AlternateGK OPTIONAL,
altGKisPermanent	BOOLEAN,
tokens	SEQUENCE OF ClearToken OPTIONAL,
cryptoTokens	SEQUENCE OF CryptoH323Token OPTIONAL,
integrityCheckValue	ICV OPTIONAL

LocationRejectReason ::=CHOICE

```

{
    notRegistered          NULL,
    invalidPermission      NULL,--exclusion by administrator or feature
    requestDenied          NULL,--can't find location
    undefinedReason        NULL,
    ...,
    securityDenial          NULL
}

```

DisengageRequest ::= SEQUENCE--(DRQ)

```

{
    requestSeqNum           RequestSeqNum,
    endpointIdentifier       EndpointIdentifier,
    conferenceID            ConferenceIdentifier,
    callReferenceValue      CallReference Value,
    disengageReason         DisengageReason,
    nonStandardData         DRQs NonStandardParameter OPTIONAL,
    ...,
    callIdentifier           CallIdentifier,
    gatekeeperIdentifier     GatekeeperIdentifier,
    tokens                  SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens             SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue      ICV OPTIONAL,
    answeredCall             BOOLEAN
}

```

DRQsNonStandardParameter ::= SEQUENCE

```

{
    srcCallSignalAddress     TransportAddress,
    destCallSignalAddress    TransportAddress,
    acctStartTime            INTEGER(0..4294967295),--in seconds
    acctSessionTime          INTEGER(0..4294967295),--in seconds
    serviceType              DRQsServiceType,
    inBytes                   INTEGER(0..4294967295)OPTIONAL,--in bytes
    outBytes                  INTEGER(0..4294967295)OPTIONAL,--in bytes
    encodecType              AudioCapability OPTIONAL,
    decodecType              AudioCapability OPTIONAL,
    terminalCause             DRQsTerminalCause,
    ...
}

```

DRQsServiceType ::= CHOICE

```

{

```

```

    phone          NULL,
    ipFax          NULL,
    ...
}

```

DRQsTerminalCause ::= CHOICE

```

{
    callerHookOn    NULL,
    callerHookOn    NULL,
    networkError    NULL,
    establishedFailed NULL,
    ...
}

```

DisengageReason ::= CHOICE

```

{
    forcedDrop      NULL, --gatekeeper is forcing the drop
    normalDrop      NULL, --associated with normal drop
    undefinedReason NULL,
    ...
}

```

DisengageConfirm ::= SEQUENCE--(DCF)

```

{
    requestSeqNum      RequestSeqNum,
    nonStandardData     NonStandardParameter OPTIONAL,
    ...,
    tokens              SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens        SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue ICV OPTIONAL
}

```

DisengageReject ::= SEQUENCE--(DRJ)

```

{
    requestSeqNum      RequestSeqNum,
    rejectReason       DisengageRejectReason,
    nonStandardData     NonStandardParameter OPTIONAL,
    ...,
    alternateGatekeeper SEQUENCE OF AlternateGK OPTIONAL,
    altGKisPermanent   BOOLEAN,
    tokens              SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens        SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue ICV OPTIONAL
}

```

}

DisengageRejectReason ::= CHOICE

```
{
    notRegistered          NULL,    --not registered with gatekeeper
    requestToDropOther     NULL,    --
    ...,
    securityDenial         NULL
}
```

InfoRequest ::= SEQUENCE--(IRQ)

```
{
    requestSeqNum          RequestSeqNum,
    callReferenceValue     CallReference Value,
    nonStandardData        NonStandardParameter OPTIONAL,
    replyAddress           TransportAddress OPTIONAL,
    ...,
    callIdentifier         CallIdentifier,
    tokens                 SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens           SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue    ICV OPTIONAL,
    uuiesRequested         UUIEsRequested OPTIONAL
}
```

InfoRequestResponse ::= SEQUENCE--(IRR)

```
{
    nonStandardData        NonStandardParameter OPTIONAL,
    requestSeqNum          RequestSeqNum,
    endpointType           EndpointType,
    endpointIdentifier      EndpointIdentifier,
    rasAddress             TransportAddress,
    callSignalAddress       SEQUENCE OF TransportAddress,
    endpointAlias           SEQUENCE OF AliasAddress OPTIONAL,
    perCallInfo            SEQUENCE OF SEQUENCE
    {
        nonStandardData    NonStandardParameter OPTIONAL,
        callReferenceValue  CallReference Value,
        conferenceID        ConferenceIdentifier,
        originator          BOOLEAN,
        audio               SEQUENCE OF RTPSession OPTIONAL,
        video               SEQUENCE OF RTPSession OPTIONAL,
        data                SEQUENCE OF TransportChannelInfo OPTIONAL,
        h245                TransportChannelInfo,
    }
}
```


callSignaling	TransportChannelInfo,
callType	CallType,
bandWidth	BandWidth,
callModel	CallModel,
...	
callIdentifier	CallIdentifier,
tokens	SEQUENCE OF ClearToken OPTIONAL,
cryptoTokens	SEQUENCE OF CryptoH323Token OPTIONAL,
substituteConfIDs	SEQUENCE OF ConferenceIdentifier,
pdu	SEQUENCE OF SEQUENCE
{	
h323pdu	H323-UU-PDU,
sent	BOOLEAN --TRUE is sent,FALSE is received
}OPTIONAL	
}OPTIONAL,	
...	
tokens	SEQUENCE OF ClearToken OPTIONAL,
cryptoTokens	SEQUENCE OF CryptoH323Token OPTIONAL,
integrityCheckValue	ICV OPTIONAL,
needResponse	BOOLEAN
}	

TransportChannelInfo ::=SEQUENCE

```
{
    sendAddress      TransportAddress OPTIONAL,
    recvAddress      TransportAddress OPTIONAL,
    ...
}
```

RTPSession ::=SEQUENCE

```
{
    rtpAddress        TransportChannelInfo,
    rtcpAddress        TransportChannelInfo,
    cname              PrintableString,
    ssrc              INTEGER(1..4294967295),
    sessionId          INTEGER(1..255),
    associatedSessionIds SEQUENCE OF INTEGER(1..255),
    ...
}
```

InfoRequestAck ::=SEQUENCE--(IACK)

```
{
    requestSeqNum      RequestSeqNum,
```

```

nonStandardData      NonStandardParameter OPTIONAL,
tokens                SEQUENCE OF ClearToken OPTIONAL,
cryptoTokens          SEQUENCE OF CryptoH323Token OPTIONAL,
integrityCheckValue  ICV OPTIONAL,
...
}

```

InfoRequestNak ::= SEQUENCE--(INAK)

```

{
    requestSeqNum      RequestSeqNum,
    nonStandardData     NonStandardParameter OPTIONAL,
    nakReason           InfoRequestNakReason,
    alternateGatekeeper SEQUENCE OF AlternateGK OPTIONAL,
    altGKisPermanent   BOOLEAN,
    tokens              SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens         SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue ICV OPTIONAL,
    ...
}

```

InfoRequestNakReason ::= CHOICE

```

{
    notRegistered      NULL,
    securityDenial      NULL,
    undefinedReason     NULL,
    ...
}

```

NonStandardMessage ::= SEQUENCE

```

{
    requestSeqNum      RequestSeqNum,
    nonStandardData     NonStandardParameter,
    ...,
    tokens              SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens         SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue ICV OPTIONAL,
}

```

UnknownMessageResponse ::= SEQUENCE--(XBS)

```

{
    requestSeqNum      RequestSeqNum,
    ...,
    tokens              SEQUENCE OF ClearToken OPTIONAL,
}

```

```

    cryptoTokens      SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue ICV OPTIONAL,
    ...
}

```

RequestInProgress ::= SEQUENCE--(RIP)

```

{
    requestSeqNum      RequestSeqNum,
    nonStandardData    NonStandardParameter OPTIONAL,
    tokens             SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens       SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue ICV OPTIONAL,
    delay              INTEGER(1..65535),
    ...
}

```

ResourcesAvailableIndicate ::= SEQUENCE--(RAI)

```

{
    requestSeqNum      RequestSeqNum,
    protocolIdentifier ProtocolIdentifier,
    nonStandardData    NonStandardParameter OPTIONAL,
    endpointIdentifier EndpointIdentifier,
    protocols          SEQUENCE OF SupportedProtocols,
    almostOutOfResources BOOLEAN,
    tokens             SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens       SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue ICV OPTIONAL,
    ...
}

```

ResourcesAvailableConfirm ::= SEQUENCE--(RAC)

```

{
    requestSeqNum      RequestSeqNum,
    protocolIdentifier ProtocolIdentifier,
    nonStandardData    NonStandardParameter OPTIONAL,
    tokens             SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens       SEQUENCE OF CryptoH323Token OPTIONAL,
    integrityCheckValue ICV OPTIONAL,
    ...
}

```

END --of ASN.1

消息编码算法参见 X. 680 和 X. 691