



UNIVERSITÀ DEGLI STUDI DI TRENTO

Facoltà di Scienze Matematiche, Fisiche e Naturali

Corso di Laurea in Informatica

Tesi di Laurea

RICONOSCIMENTO ED ANALISI DEL TRAFFICO VOICE OVER IP

Relatore:
Prof. Mauro Brunato

Laureando:
Michele Dallachiesa

Anno Accademico 2005 - 2006

Indice

1	Introduzione	1
1.1	La rete telefonica tradizionale	2
1.2	Il VoIP	3
2	I protocolli di segnalazione	6
2.1	Il protocollo di segnalazione SIP	7
2.1.1	L'indirizzamento	7
2.1.2	I componenti dell'architettura	8
2.1.3	I Messaggi SIP	11
2.1.3.1	I messaggi di richiesta	11
2.1.3.2	I messaggi di risposta	15
2.1.3.3	I Campi Header dei messaggi	19
2.1.4	Il protocollo SDP	22
2.1.4.1	Il messaggio SDP	22
2.1.4.2	Utilizzo di SDP in SIP	25
2.1.5	I modelli di segnalazione	25
2.1.6	Alcuni possibili scenari	26
2.2	Il protocollo di segnalazione H.323	28
2.2.1	L'indirizzamento	28
2.2.2	I componenti dell'architettura	29
2.2.3	La raccomandazione H.225.0	32
2.2.3.1	I messaggi H.225.0	36
2.2.4	La raccomandazione H.245	36
2.2.5	I modelli di segnalazione	37
2.3	Il protocollo di segnalazione SCCP	42
2.3.1	L'indirizzamento	42
2.3.2	I componenti dell'architettura	42
2.3.3	I messaggi SCCP	43
2.3.4	I modelli di segnalazione	45
3	I protocolli RTP/RTCP	49
3.1	L'indirizzamento	49
3.2	I Componenti dell'architettura RTP/RTCP	49
3.3	Il protocollo RTP	50

3.3.1	Il pacchetto RTP	51
3.3.2	Il modello trasmissivo di RTP	52
3.4	Il protocollo RTCP	52
3.4.1	Il pacchetto RTCP	52
3.4.2	Il modello trasmissivo di RTCP	53
4	VoIP eavesdropping	54
4.1	Il progetto di stage	54
4.1.1	Requisiti	54
4.1.2	Implementazione	55
4.2	Il riconoscimento delle sessioni RTP e la decodifica dei flussi audio	62
4.2.1	L'applicazione rtpbreak	62
4.2.1.1	Applicazioni simili	62
4.2.1.2	Il riconoscimento delle sessioni RTP	62
4.2.1.3	I Requisiti	65
4.2.1.4	I Parametri	65
4.2.1.5	Esempio di utilizzo	66
4.2.2	L'applicazione sipcodec	66
4.2.2.1	Modifica dei messaggi SDP	67
4.2.2.2	L'attacco ARP Poisoning	68
4.2.2.3	L'attacco MITM via ARP Poisoning	68
4.2.2.4	IP Forwarding	70
4.2.2.5	I Requisiti	70
4.2.2.6	I Parametri	71
4.2.2.7	Esempio di utilizzo	71
5	Conclusioni	72
	Bibliografia	75

Capitolo 1

Introduzione

L'evoluzione delle reti di telecomunicazione ha introdotto alcune importanti innovazioni nel campo della telefonia, non più relegata nelle reti a commutazione di circuito ma integrata con altri servizi nelle reti a commutazione di pacchetto. L'insieme delle tecnologie che descrivono la trasmissione digitale della voce nelle reti a commutazione di pacchetto (solitamente di tipo IP) viene identificato con VoIP, acronimo di "Voice over IP". Come nella telefonia tradizionale, la trasmissione della voce è gestita attraverso i messaggi di segnalazione. Esistono differenti protocolli, ciascuno definisce un tipo diverso di rete VoIP con caratteristiche proprie che non è direttamente compatibile con le altre. Queste possono comunque essere interconnesse fra loro mediante sistemi Gateway multiprotocollo. I protocolli di segnalazione più diffusi sono SIP, H.323 ed SCCP mentre i protocolli comuni di trasmissione della voce sono RTP/RTCP. Il passaggio dalla telefonia tradizionale al VoIP non è ancora stato completato ma è in corso anche in Italia, dove i principali operatori telefonici si sono già mossi da tempo assieme a nuove società e realtà OpenSource che intendono offrire servizi simili. Queste raggiungono l'utenza in modo diretto attraverso la rete Internet. Con l'introduzione del VoIP, si rende inoltre possibile l'implementazione di nuovi servizi come la comunicazione video e la mobilità del proprio identificativo, un username di livello applicativo che può essere utilizzato da qualsiasi punto della rete dopo la procedura di autenticazione. Questi cambiamenti strutturali così importanti comportano anche una piccola rivoluzione nel campo delle intercettazioni telefoniche. Le Forze dell'ordine devono affiancare ai metodi tradizionali nuove applicazioni di supporto in grado di identificare e registrare le sessioni VoIP ritenute di interesse. La rete IP, per sua natura multiservizio, ha già richiesto l'implementazione di applicazioni dedicate all'analisi del traffico. Il supporto per il VoIP costituirà quindi solamente un nuovo modulo affiancato a quelli già esistenti. Le funzionalità principali richieste sono le seguenti:

1. Riconoscimento delle sessioni VoIP.
2. Riconoscimento degli username di livello applicativo VoIP.
3. Registrazione su file delle sessioni VoIP ritenute di interesse.
4. Decodifica dei dati trasportati¹.

Questa tesi descrive ed analizza i protocolli VoIP, proponendo un'implementazione completa dei primi tre punti. Vengono inoltre presentate due applicazioni OpenSource² sviluppate come progetto di tesi:

- *rtpbreak*

Riconosce, analizza e ricostruisce i flussi RTP in modo indipendente dai messaggi di segnalazione, proponendo una soluzione innovativa e più efficace rispetto ad applicazioni simili già esistenti come *voipong* e *vomit*. Utilizzando l'applicazione *sox*, viene inoltre dimostrato come sia possibile decodificare i flussi audio codificati con i codec *G.711 u-law*, *G.711 a-law* e *gsm*.

- *sipcodec*

Forza l'utilizzo del codec audio *G.711 u-law* nei flussi RTP associati al protocollo di segnalazione SIP, attuando un attacco di tipo MITM via ARP Poisoning.

La tesi si compone di Cinque capitoli. Il primo introduce il VoIP e ne descrive le differenze rispetto alla telefonia tradizionale, il secondo descrive i protocolli di segnalazione SIP, H.323 ed SCCP, il terzo descrive i protocolli RTP/RTCP, il quarto presenta le tre applicazioni sviluppate ed in fine il quinto presenta le conclusioni.

1.1 La rete telefonica tradizionale

Le reti telefoniche tradizionali sono a commutazione di circuito e sono state definite ed implementate per il trasporto della voce. Quando si intende avviare una comunicazione, gli switch interni della rete commutano per creare un circuito logico diretto tra sorgente e destinazione della chiamata. Per tutta la durata della comunicazione gli interlocutori dispongono di un canale dedicato non accessibile ad altri utenti, indipendentemente dal fatto che le parti siano coinvolte in una conversazione attiva oppure rimangano silenzio. Si ha così un'allocazione statica delle risorse di rete ed ogni circuito garantisce una larghezza di banda pari a 64 Kb/s in entrambi i sensi della comunicazione.

¹I dati trasportati possono essere audio, video oppure qualche altro tipo di informazione che si intende condividere fra le parti.

²Il codice sorgente di queste due applicazioni è liberamente disponibile sotto licenza GPL.

L'allocazione statica delle risorse ed il circuito diretto tra le due parti permettono al segnale vocale di avere una qualità garantita per tutta la durata della comunicazione. Tuttavia, questo corrisponde ad avere una bassa percentuale di utilizzazione della rete. Infatti, durante una comunicazione vocale gli interlocutori assumono a turno il ruolo di trasmettitore mentre gli altri (solitamente) rimangono in silenzio, nel ruolo di ricevitore. La banda riservata viene quindi utilizzata per meno della metà del periodo della conversazione, comportando un grande spreco di risorse. Inoltre questo tipo di reti non è in grado di fornire con la dovuta rapidità e facilità le nuove funzionalità richieste nel settore delle telecomunicazioni: basandosi su un'infrastruttura centralizzata e proprietaria che fa generalmente capo ad un unico produttore, non consente lo sviluppo di nuovi servizi in tempi rapidi poiché tali funzionalità possono essere implementate solamente dallo stesso produttore dei dispositivi il quale, avendo molto spesso più clienti, non potrà rispondere con la dovuta celerità alla domanda di tali applicativi. La trasmissione dati è consentita ma l'infrastruttura, non sviluppata originariamente per fornire questa funzionalità, non è adatta, traducendosi in un ulteriore spreco di risorse. Alla luce del fatto che ormai il traffico dati ha superato il traffico voce, si può considerare la rete telefonica tradizionale ulteriormente inadeguata. Questi motivi portano a concludere che è necessario operare dei cambiamenti profondi nell'organizzazione dell'infrastruttura telefonica, al fine di far convergere su un unico tipo di rete le trasmissioni voce, video e dati.

1.2 Il VoIP

Il VoIP identifica un insieme di tecnologie che hanno come scopo la trasmissione della voce e di altri tipi di informazioni multimediali attraverso le reti dati, solitamente di tipo IP. La rete IP è a commutazione di pacchetto: le informazioni vengono trasportate tramite unità minime di trasporto, i datagrammi IP. Questi sono in grado di essere instradati nella giusta direzione in base alle informazioni contenute negli stessi, senza richiedere la presenza di alcun canale logico preimpostato. Così facendo non si ha l'allocazione statica delle risorse, perciò su una linea di collegamento possono transitare contemporaneamente anche pacchetti appartenenti a flussi differenti, permettendo una gestione migliore delle risorse di rete. I pacchetti IP appartenenti ad uno stesso flusso possono essere instradati in modo differente fra sorgente e destinazione, non viene garantita la loro ricezione e possono giungere a destinazione in disordine. Inoltre, non viene fornito alcun tipo di QoS³. E' compito dei protocolli che utilizzano la rete di preoccuparsi di queste caratteristiche come meglio credono: limitando l'utilizzo della

³QoS, acronimo di "Quality Of Service".

rete in qualche modo per implementare il Servizio di QoS, riordinando i pacchetti IP ricevuti e richiedendo la trasmissione di quelli persi. Per quanto riguarda il VoIP, la mancanza di una preallocazione statica della banda favorisce l'utilizzo di nuove tecniche al fine di ridurre l'utilizzo. Ad esempio, si possono impiegare algoritmi di codifica audio che permettono un utilizzo della banda pari a 5.3 Kb/s (ITU G.723.1) oppure a 8 Kb/s (ITU G.729) con un degrado accettabile della qualità audio percepita dall'utente, oppure algoritmi di soppressione del silenzio che permettono di sospendere la trasmissione dati se l'utente non produce, rimanendo in silenzio. Questo comporta una riduzione dell'utilizzo della banda superiore al 50% rispetto alla rete telefonica tradizionale. Si noti che la rete IP non richiede la fase di CallSetup presente invece nelle reti telefoniche tradizionali, utilizzata per instaurare il canale logico fra sorgente e destinazione. Questo si traduce in una più veloce instaurazione della chiamata. La convergenza delle trasmissioni voce, audio e dati su un unico tipo di rete consentono inoltre una riduzione dei costi di gestione e manutenzione dell'intera infrastruttura. Il VoIP viene utilizzato in modo differente a seconda del tipo di utente, possiamo identificare tre classi:

1. "consumer"

Il VoIP viene inteso come tecnologia utilizzata in modo diretto tra gli utenti senza l'intervento degli operatori telefonici tradizionali. Terze parti possono fornire ulteriori servizi come localizzazione, autenticazione ed interconnessione con la rete telefonica tradizionale ma la loro presenza non è strettamente necessaria. I benefici ricadono direttamente sull'utente e si traducono in una riduzione dei costi, consentendo anche la presenza di servizi aggiuntivi in modo dinamico ed indipendente dagli operatori telefonici. In questo ambito vengono maggiormente utilizzati i protocolli di segnalazione H.323 e SIP.

2. "telecom"

Il VoIP viene inteso come tecnologia utilizzata in modo diretto dagli operatori telefonici ed il suo utilizzo è spesso del tutto trasparente per gli utenti. I benefici ricadono direttamente sull'operatore telefonico e si traducono in una riduzione dei costi di manutenzione e gestione dell'infrastruttura di rete. Il VoIP viene solitamente introdotto seguendo una di queste modalità:

- (a) Integrazione con l'infrastruttura telefonica già esistente

Il VoIP viene integrato con l'infrastruttura telefonica già esistente mediante l'utilizzo di Gateway che interconnettono i due tipi di rete. Il posizionamento di questi Gateway è critico: più sono vicini all'utente finale e più alto sarà il loro numero, con ricadute sui costi di gestione. La tendenza è

di posizionare questi Gateway in una posizione tale da utilizzare la tecnologia VoIP soltanto tra le centrali telefoniche di grandi dimensioni. Questa soluzione è completamente trasparente per l'utente, che continua ad utilizzare lo stesso telefono senza alcun servizio aggiuntivo. In questo ambito viene maggiormente utilizzato il protocollo di segnalazione H.323.

(b) Integrazione diretta con la rete dati IP

Il VoIP viene trasportato direttamente dalla linea ADSL oppure mediante collegamento in fibra ottica. Questa soluzione impone una sostituzione del telefono tradizionale con un telefono VoIP, consentendo però anche l'implementazione di nuovi servizi. In questo ambito vengono maggiormente utilizzati i protocolli di segnalazione H.323 e SIP.

3. "enterprise"

Il VoIP viene inteso come tecnologia utilizzata in modo diretto dalle aziende che lo utilizzano nelle comunicazioni interne, affidandosi solitamente ad un Gateway⁴ per raggiungere la rete telefonica tradizionale. I benefici ricadono direttamente sull'azienda e si traducono in una riduzione dei costi di gestione dell'infrastruttura di rete, prima separata per il trasporto di voce e dati. In questo ambito vengono utilizzati i protocolli di segnalazione H.323, SIP ed SCCP.

⁴Un Gateway VoIP multiprotocollo OpenSource molto utilizzato è *Asterisk*.

Capitolo 2

I protocolli di segnalazione

I protocolli di segnalazione utilizzati nel VoIP definiscono un insieme di procedure che forniscono almeno le seguenti funzionalità:

1. Localizzazione
Permette di accedere alle informazioni di localizzazione degli utenti registrati.
2. Registrazione
Permette di notificare la propria presenza, fornendo la propria posizione nella rete.
3. Segnalazione di chiamata
Permette di richiedere, accettare oppure modificare una comunicazione.
4. Scambio dei parametri di trasmissione dei flussi multimedia¹
Permette di concordare e rendere noti alle parti coinvolte nella comunicazione i parametri di trasmissione dei flussi multimedia.

I protocolli di segnalazione più diffusi sono i seguenti:

1. SIP
2. H.323
3. SCCP

¹Con multimedia si intende audio, video oppure un qualche altro tipo di informazione che viene scambiato fra le parti.

Non permettono esclusivamente la trasmissione della voce, consentendo anche altre forme di comunicazione come video e dati². Nelle sezioni che seguono vengono analizzati questi tre protocolli di segnalazione.

2.1 Il protocollo di segnalazione SIP

Il protocollo di segnalazione SIP[12], acronimo di “Session Initiation Protocol”, nasce in ambito IETF nel 1999 con la creazione del SIP Working Group in collaborazione con il MMUSIC Working Group. La proposta giunge dall’esperienza di Mbone dove si erano implementati alcuni applicativi audio/video, i quali utilizzavano delle semplici primitive di segnalazione che sono state poi sviluppate ed ampliate. SIP nasce come alternativa ad H.323, protocollo di segnalazione precedentemente definito e proposto da ITU-T, ponendo a suo vantaggio la semplicità. E’ un protocollo di segnalazione di livello applicativo che ha come obiettivo l’instaurazione, la modifica e la terminazione di una comunicazione audio, video oppure dati. Tra le caratteristiche peculiari di SIP vi è l’idea di distribuire l’intelligenza del network nei sistemi periferici quando possibile, ottenendo così un servizio scalabile. Viene utilizzato congiuntamente con altri protocolli sviluppati in ambito IETF come SDP per concordare i parametri di trasmissione dei flussi multimedia, RTP/RTCP come protocollo di trasporto dei flussi multimedia e RSVP per l’implementazione di una forma di QoS. Non specifica il tipo di rete sottostante ma viene solitamente trasportato da IP, utilizzando come protocolli di trasporto UDP e TCP. Esistono estensioni che permettono la cifratura del Signaling, le trasmissioni RTP/RTCP sono comunque non cifrate.

2.1.1 L’indirizzamento

SIP utilizza lo schema di indirizzamento URI, definendo due differenti tipi di indirizzi:

1. sip

Il più semplice ed il più diffuso nei network SIP, di default il protocollo di trasporto è UDP ma può essere utilizzato anche TCP. Sono possibili vari parametri attraverso i quali si possono fornire ulteriori informazioni sul contatto. La porta UDP e TCP di default è 5060 ma può essere cambiata, fornendone una differente nell’indirizzo. Possono essere presenti anche altri parametri, che descrivono ulteriormente le caratteristiche del contatto. Esempi:

- sip:marco@sip.science.unitn.it:5067

²Le comunicazioni dati possono implementare ad esempio l’IM (Instant Messaging) e sistemi di WB (White Board) condivisa.

- sip:Francesco@sip.science.unitn.it;transport=tcp
- sip:+390461012345@pstn_gateway.sip.science.unitn.it;user=phone

2. sips

Equivalentemente agli indirizzi sip, le informazioni vengono però trasportate con TLS[2], il Transport Security Layer. Esempi:

- sips:marco@sip.science.unitn.it

I componenti del network SIP possono supportare anche altri tipi di indirizzi definiti con lo schema URI, al fine di comunicare con utenti presenti in altri tipi di rete, raggiungibili attraverso un SIP Gateway Server. Ad esempio, tel permette di comunicare con la rete telefonica tradizionale mediante indirizzi E.164[7]. Esempi:

- tel:118
- tel:+0461821234

2.1.2 I componenti dell'architettura

I componenti del network SIP sono molteplici e ciascuno assolve ad un particolare compito:

1. SIP User Agent
2. SIP Back-to-Back User Agent
3. SIP Servers
 - (a) SIP Proxy Server
 - (b) SIP Redirect Server
 - (c) SIP Registration Server
 - (d) SIP Gateway Server

Segue la descrizione estesa di ciascuno.

1. SIP User Agent

Indica qualsiasi device in grado di accettare oppure richiedere una comunicazione. Si compone di due entità logiche:

- (a) User Agent Client (UAC)
Inoltra le richieste verso il network SIP (richieste scatenate dall'utente).
- (b) User Agent Server (UAS)
Inoltra le risposte alle richieste provenienti dal network SIP.

Assieme permettono all'utente di accedere ai servizi forniti dal network SIP.

2. SIP Back-to-Back User Agent

Indica un particolare tipo di device che, quando riceve una richiesta SIP, la riformula e la ritrasmette come una nuova richiesta. Le risposte alla richiesta vengono quindi riformulate e ritrasmesse nella direzione opposta. Permette di implementare il servizio Anonymizer e Application Layer Gateway (ALG).

3. SIP Servers

Identificano un insieme di sistemi che accettano richieste SIP e sono in grado di rispondere. Un SIP Server non dovrebbe essere confuso con un UAS oppure con la natura Client/Server del protocollo, che descrive le operazioni in termini di Client (coloro che generano richieste) e Server (coloro che rispondono alle richieste). I SIP Server discussi in questa sezione sono entità logiche, le attuali implementazioni possono operare come un differente tipo di Server a seconda della situazione. Questi forniscono differenti servizi e funzionalità agli UAs, gestendo ciascuno soltanto un sottoinsieme di richieste. Devono supportare TCP, TLS ed UDP come protocollo di trasporto. Si noti che il protocollo utilizzato per interconnettere un Server con un servizio di localizzazione oppure un database non è definito in SIP. Le entità logiche definite sono:

(a) SIP Proxy Server

Riceve le richieste SIP, ritrasmettendole ad altri componenti del network. Quando riceve le risposte, si occupa di trasmetterle nella direzione opposta. Viene utilizzato per implementare il servizio di Call Forwarding e di Autenticazione. I messaggi generati sono percepiti dall'UA di destinazione come trasmessi dal Proxy stesso che viene visto come un UA, piuttosto che da qualche applicazione nascosta dietro di esso. Esistono due tipi di Proxy:

i. Stateless Proxy

Ciascuna richiesta SIP viene eseguita considerando unicamente il contenuto del singolo messaggio. Non vengono mantenute internamente informazioni sullo stato delle sessioni esistenti, richiedendo quindi meno risorse. Questo comporta un certo spreco delle risorse di rete e non permette l'implementazione di alcuni servizi come l'autenticazione.

ii. Stateful Proxy

Ciascuna richiesta SIP viene eseguita considerando le informazioni ottenute dalle richieste precedenti. Vengono

mantenute internamente informazioni sullo stato delle sessioni esistenti, richiedendo quindi più risorse ma consentendo alcuni servizi aggiuntivi:

- A. Permette la ritrasmissione delle richieste in mancanza di risposta, liberando l'UA da questo compito e riducendo l'utilizzo delle risorse di rete.
- B. Permette l'autenticazione dell'UA.

Un tipo particolare di Stateful Proxy è il Transaction Stateful Proxy. Questo si limita a tenere traccia dei messaggi fino a quando la singola transazione non viene completata. La transazione ha inizio con la ricezione della richiesta ed ha termine con la ricezione della risposta definitiva.

(b) SIP Redirect Server

Il suo compito è fornire informazioni circa la localizzazione di ciascun utente registrato. Risponde alle richieste ma non le ritrasmette, ritornando direttamente una risposta contenente l'indirizzo verso il quale ritrasmettere la richiesta. Si appoggia ad un Database oppure ad un Location Service per l'operazione di localizzazione dell'utente. L'informazione di localizzazione è trasmessa come messaggio di risposta di tipo 3xx (Redirect) che, dopo l'ulteriore ricezione del messaggio ACK, conclude la transazione.

(c) SIP Registration Server

Il suo compito è accettare oppure rifiutare l'ingresso di un UA nel network SIP. Accetta soltanto richieste SIP di tipo REGISTER, rispondendo altrimenti con il messaggio 501 not implemented. Le informazioni della registrazione vengono rese disponibili al Redirect Server. Nella richiesta di registrazione il campo dell'header To contiene il nome della risorsa che si sta registrando ed il campo dell'header Contact contiene una lista dei suoi indirizzi alternativi. Il Registration Server solitamente richiede all'UA di autenticarsi. Questo permette di avere un minimo di sicurezza riguardo l'identità dell'utente, garantendo l'identità del contatto. Una richiesta Register può essere utilizzata dall'UA per modificare le informazioni che lo riguardano, permettendogli di ricevere una lista di URI già registrate, quindi di eliminarne oppure aggiungerne.

(d) SIP Gateway Server

Si occupa di interconnettere il network SIP con altri tipi di network come la rete telefonica tradizionale. Svolge quindi le funzioni di adattamento per il trasporto dei flussi multimedia e del Signaling.

2.1.3 I Messaggi SIP

La segnalazione SIP viene trasportata dai messaggi SIP. Questi hanno la stessa struttura dei messaggi definiti dal protocollo HTTP/1.1, utilizzano la codifica UTF-8[14] e vengono definiti mediante la notazione ABNF[1]. Si dividono in richieste (Request) e risposte (Response), vengono così definiti:

```
SIP-message = Request / Response
Request = Request-Line *( message-header ) CRLF [ message-
body ]
Response = Status-Line *( message-header ) CRLF [ message-
body ]
```

I due tipi di messaggio sono molto simili e differiscono sintatticamente soltanto nella parte iniziale. A questa segue una sequenza di campi che compongono l'header. Infine, segue il corpo del messaggio che è opzionale e utilizzato principalmente soltanto durante l'instaurazione delle comunicazioni multimedia. Segue la discussione estesa di ciascun tipo di messaggio.

2.1.3.1 I messaggi di richiesta

Questi messaggi vengono trasmessi al fine di richiedere l'esecuzione di una certa operazione. Sono così definiti:

```
Request = Request-Line *( message-header ) CRLF [ message-
body ]
Request-Line = Method SP Request-URI SP SIP-Version
CRLF
```

Descrizione di Request-Line:

- Method
Indica il tipo di operazione richiesta.
- Request-URI
Indica il destinatario (cioè chi dovrebbe processare l'operazione).
- SIP-Version
Indica la versione del protocollo SIP.

Il significato dei campi header varia in funzione del metodo presente. Fino a quando non segue una risposta, l'UA è tenuto a ritrasmettere la richiesta ad intervalli regolari. I Metodi definiscono le azioni che possono essere richieste e sono i seguenti: INVITE, REGISTER, BYE, ACK, CANCEL, OPTIONS, REFER, SUBSCRIBE, NOTIFY, MESSAGE, UPDATE, INFO, PRACK. Vengono di seguito discussi separatamente.

1. INVITE

Richiede l'apertura di una sessione, chiamata anche Dialog. Nella richiesta INVITE è solitamente presente il corpo del messaggio, utilizzato per trasportare i messaggi di tipo SDP[3]. Questi contengono le informazioni necessarie per la ricezione di uno o più flussi multimedia. Il corpo del messaggio può però anche contenere informazioni di tipo differente che permettono di gestire ad esempio la crittografia. E' compito dell'UA che intende richiedere l'apertura di una sessione generare il campo header Call-Id ed il local-tag presente nel campo header From. Questi, assieme al remote-tag fornito dall'UA contattato nel campo header To, costituiscono l'identificativo univoco della sessione. Tutti i messaggi appartenenti a questa sessione dovranno avere questo identificativo. I campi header più importanti per questo tipo di richiesta sono: Call-ID, From, To, CSeq. Il valore del campo header CSeq in questo tipo di messaggio viene solitamente impostato a 1. Un INVITE trasmesso all'interno di una sessione già esistente è chiamato re-INVITE e permette di rinegoziare oppure modificare i parametri della sessione. Il CSeq viene incrementato per questo tipo di richiesta, che viene considerata un comando all'interno del Dialog. Se un re-INVITE viene rifiutato oppure fallisce in qualche modo, la sessione continua ad esistere con i parametri precedentemente concordati. Un re-INVITE non può essere trasmesso durante l'instaurazione della sessione. In questo caso, si deve utilizzare il metodo UPDATE.

2. REGISTER

Richiede la registrazione dell'UA presso un SIP Registration Server. Particolare importanza ha il campo header Contact, questo infatti contiene l'URI che si intende registrare e che verrà utilizzata per contattare l'UA. Il campo header Expires può essere utilizzato per imporre un periodo di validità della registrazione, che di default per le URI sip e sips è di 1 ora. Il CSeq viene incrementato per questo tipo di richiesta. I campi header più importanti per questa richiesta sono: Call-ID, From, To, CSeq, Expires. Il valore del campo header CSeq in questo tipo di messaggio non viene incrementato.

3. BYE

Richiede la terminazione di una sessione esistente. Una sessione è considerata esistente se ad un INVITE precedentemente trasmesso segue una risposta 2xx oppure quando, alla trasmissione del messaggio 200 OK, segue la ricezione di un messaggio ACK. Il valore del campo header CSeq in questo tipo di messaggio non viene incrementato.

4. ACK

Conferma l'avvenuta ricezione di una richiesta INVITE. Questo tipo di messaggi può contenere il corpo e viene solitamente utilizzato analogamente a quanto accade nelle richieste di tipo INVITE. I messaggi di tipo ACK non possono essere utilizzati per modificare i parametri della sessione che sono già stati trasmessi nel messaggio INVITE corrispondente, a tale scopo occorre utilizzare un messaggio di tipo re-INVITE come precedentemente discusso. Il valore del campo header CSeq in questo tipo di messaggio non viene incrementato. I campi header più importanti per questa richiesta sono: Call-ID, From, To, CSeq.

5. CANCEL

Richiede di cancellare l'esecuzione di una richiesta precedentemente trasmessa. Solitamente, viene utilizzato per interrompere una ricerca oppure un tentativo di chiamata. Può essere generato in seguito alla ricezione di una risposta lxx, questa indica che la richiesta è stata ricevuta ma non ancora eseguita completamente. Questo tipo di richiesta ha senso soltanto quando si intende cancellare l'esecuzione di una richiesta INVITE, l'unica che può richiedere un certo tempo per l'esecuzione. Un UA può accettare la cancellazione dell'esecuzione della richiesta, rispondendo con un messaggio 200 OK seguito da un messaggio 487 Request Terminated. Se la richiesta CANCEL è stata trasmessa troppo tardi e l'operazione è stata ormai eseguita, l'UA può terminare la sessione (che ormai esiste) con una richiesta BYE. Il valore del campo header CSeq in questo tipo di messaggio non viene incrementato. I campi header più importanti per questa richiesta sono: Call-ID, From, To, CSeq.

6. OPTIONS

Richiede le capacità di un UA. La risposta può contenere i campi header Allow, Accept, Accept-encoding, Accept-language, Supported. A questo tipo di richiesta si può rispondere con una risposta 200 OK nella quale tramite i campi header precedentemente citati verranno trasmesse le informazioni richieste. In caso di errore, l'UA è tenuto a rispondere con un messaggio 4xx oppure 6xx. Il valore del campo header CSeq in questo tipo di messaggio viene incrementato. I campi header più importanti per questa richiesta sono: Call-ID, From, To, CSeq.

7. REFER

Richiede ad un altro UA di accedere all'URI specificata nel campo header Refer-To. Se l'URI è di tipo sip oppure sips, questo tipo di richiesta implementa il servizio Call Transfer. Il valore del campo header CSeq in questo tipo di messaggio viene incrementato. I

campi header più importanti per questa richiesta sono: Call-ID, From, To, CSeq, Refer-To.

8. SUBSCRIBE

Richiede la sottoscrizione al fine di ricevere notifiche attraverso richieste di tipo NOTIFY circa un particolare evento. La sottoscrizione implica la creazione di una sessione, chiamata anche Dialog. Il messaggio contiene il campo header Expires, il quale indica il termine di validità della sottoscrizione. La sottoscrizione può essere rinnovata all'interno dello stesso Dialog prima della sua scadenza. Il valore del campo header CSeq in questo tipo di messaggio viene incrementato. I campi header più importanti per questa richiesta sono: Call-ID, From, To, CSeq, Expires.

9. NOTIFY

Fornisce informazioni riguardo l'esistenza di un certo evento. Questo tipo di richieste viene trasmessa all'interno di un Dialog, precedentemente creato con la richiesta SUBSCRIBE. Il valore del campo header CSeq in questo tipo di messaggio viene incrementato. I campi header più importanti per questa richiesta sono: Call-ID, From, To, CSeq.

10. MESSAGE

Permette una forma di Instant Messaging (IM) testuale e rudimentale tra gli UAs. Il testo viene trasmesso nel corpo del messaggio. Il valore del campo header CSeq in questo tipo di messaggio viene incrementato. I campi header più importanti per questa richiesta sono: Call-ID, From, To, CSeq.

11. INFO

Fornisce ulteriori informazioni riguardo il Call Signaling. Il valore del campo header CSeq in questo tipo di messaggio viene incrementato. I campi header più importanti per questa richiesta sono: Call-ID, From, To, CSeq.

12. PRACK

Conferma l'avvenuta ricezione di una risposta 1xx. Si noti che ACK può essere utilizzato soltanto per confermare la ricezione di risposte 2xx, 3xx, 4xx, 5xx e 6xx. Il valore del campo header CSeq in questo tipo di messaggio non viene incrementato. I campi header più importanti per questa richiesta sono: Call-ID, From, To, CSeq.

13. UPDATE

Richiede di modificare i parametri di una sessione in fase di creazione. Il valore del campo header CSeq in questo tipo di messaggio

non viene incrementato. I campi header più importanti per questa richiesta sono: Call-ID, From, To, CSeq.

2.1.3.2 I messaggi di risposta

Questi messaggi vengono trasmessi al fine di rispondere ad un messaggio di richiesta. Sono così definiti:

```
Response = Status-Line *( message-header ) CRLF [ message-  
body ]  
Status-Line = SIP-Version SP Status-Code SP Reason-Phrase  
CRLF
```

Descrizione di Status-Line:

- SIP-Version
Indica la versione del protocollo SIP.
- Status-Code
Identifica il tipo di risposta in modo univoco.
- Reason-Phrase
Descrive il tipo di risposta in modo testuale, fornendo eventualmente ulteriori informazioni utili per il debugging.

Il tipo della risposta è determinato dallo Status-Code, un codice di esattamente tre cifre. La prima cifra identifica la classe. Le classi sono 6, di queste le prime 5 derivano direttamente da HTTP/1.1. Le ultime due cifre specificano ulteriormente il tipo di risposta. Le classi definite sono le seguenti:

- 1xx Informational
Indica lo stato di esecuzione della richiesta.
- 2xx Success
Indica che la richiesta è stata eseguita con successo.
- 3xx Redirection
Indica che la richiesta non è stata eseguita e viene suggerito un differente indirizzo dove trasmettere la richiesta.
- 4xx Client error
Indica che la richiesta è fallita a causa di un errore da parte del client.

- 5xx Server failure
Indica che la richiesta è fallita a causa di un errore da parte del server.
- 6xx Global failure
Indica che la richiesta è fallita a causa di un errore da parte della rete SIP.

Segue la descrizione estesa di ciascuna classe.

1. Informational Class (1xx)

La classe di risposte 1xx viene utilizzata per indicare che la richiesta è stata ricevuta e si è ad un certo punto nella sua esecuzione. Un device può trasmettere un qualsiasi numero di risposte 1xx prima di trasmettere la risposta finale (2xx, 3xx, 4xx, 5xx e 6xx). Solitamente, la prima risposta 1xx ricevuta dall'UA conferma la ricezione della richiesta INVITE e quindi ne termina la ritrasmissione. Per questa ragione i Server che trasmettono la risposta 100 Trying riducono il numero di ritrasmissioni di richieste INVITE, riducendo così anche l'utilizzo delle risorse di rete.

(a) 100 Trying

Informa l'UA che la richiesta è stata ricevuta correttamente e che la si sta processando.

(b) 180 Ringing

Informa l'UA che la richiesta INVITE è stata ricevuta correttamente e che si sta allertando l'utente. Questo tipo di risposta è molto importante per l'interworking con la rete telefonica tradizionale.

(c) 181 Call Is Being Forwarded

Indica che la chiamata è stata trasferita ad un differente UA. Con questa viene implementato il servizio di Call Forwarding.

(d) 182 Call Queued

Indica che la chiamata è stata eseguita correttamente ed è stata inserita in una lista di attesa. La Reason-Phrase può essere utilizzata per fornire ulteriori informazioni, come il tempo di attesa stimato. Con questa viene implementato il servizio Call Center.

(e) 183 Session Progress

Indica lo stato della chiamata, stabilendo anche un Dialog. è particolarmente importante per l'interworking con la rete telefonica tradizionale, permettendo all'UA di ricevere l'audio di ringing, busy oppure un messaggio preregistrato. Questo

perché nella rete telefonica tradizionale lo stato della chiamata è solitamente comunicato all'utente tramite segnali sonori trasmessi all'interno del circuito già esistente.

2. Success

questa classe di risposte indica che l'operazione è stata eseguita con successo.

(a) 200 OK

Indica che l'operazione è stata eseguita senza errori. Quando viene trasmesso questo tipo di risposta per accettare una richiesta INVITE, il corpo del messaggio può contenere un messaggio SDP contenente i parametri di ricezione dei flussi multimedia.

(b) 202 Accepted

Indica che l'UA ha ricevuto e compreso la richiesta, ma che ancora non è stata eseguita per un qualche motivo.

3. Redirection

Questa classe di risposte è utilizzata per reindirizzare l'operazione verso una differente destinazione. Viene utilizzata per implementare i servizi di Call Forwarding. Particolare attenzione deve essere tenuta per evitare situazioni di loop. A tale fine viene impiegato il campo header Via.

(a) 300 Multiple Choices

Indica che la localizzazione dell'URI richiesta ha ritornato più indirizzi. Il messaggio contiene campi header Contact, il loro ordine è significativo e dovrebbero essere tentati in sequenza dal primo all'ultimo.

(b) 301 Moved Permanently

Indica che l'URI richiesta è disponibile altrove, suggerendo nel campo header Contact una nuova URI da utilizzare permanentemente per questo contatto.

(c) 302 Moved Temporarily

Indica che l'URI richiesta è al momento disponibile altrove, suggerendo nel campo header Contact una nuova URI da utilizzare temporaneamente per questo contatto. Può essere inoltre presente un campo header Expires che ne indica la durata di validità.

(d) 305 Use Proxy

Indica l'URI di un SIP Proxy Server che ha l'autorizzazione necessaria per contattare l'UA. Il chiamante dovrebbe ritrasmettere la richiesta al SIP Proxy Server, il quale la ritrasmetterà all'UA. Con questo sistema viene implementato il servizio di Call Screening (filtering delle chiamate).

(e) 380 Alternative Service

Indica che l'utente desidera essere contattato in modo differente, suggerendo come nel campo header Contact.

4. Client Error Class

Questa classe di risposte è utilizzata dai client per indicare che la richiesta non può essere soddisfatta a causa di una condizione di errore interna. Segue un elenco di possibili valori:

400	Bad Request
402	Payment Required
403	Forbidden
404	Not Found
405	Method Not Allowed
406	Not Acceptable
407	Proxy Authentication Required
408	Request Timeout
409	Conflict
410	Gone
411	Length Required
413	Request Entity Too Large
414	Request-URI Too Long
415	Unsupported Media Type
416	Unsupported URI Scheme
420	Bad Extension
421	Extension Required
422	Session Timer Interval Too Small
423	Interval Too Brief
428	Use Authentication Token
429	Provide Referrer Identity
480	Temporarily Unavailable
481	Dialog/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
489	Bad Event
491	Request Pending
493	Request Undecipherable

5. Server Error class

Questa classe di risposte è utilizzata dai Server per indicare che la richiesta non può essere soddisfatta a causa di una condizione di errore interna. Segue un elenco di possibili valori:

500	Server Internal Error
501	Not Implemented
502	Bad Gateway
503	Service Unavailable
504	Gateway Timeout
505	Version Not Supported
513	Message Too Large

6. Global Error

Questa classe di risposte è utilizzata dai server per indicare che la richiesta non può essere soddisfatta a causa di una condizione di errore nella rete. Segue un elenco di possibili valori:

600	Busy Everywhere
603	Decline
604	Does Not Exist Anywhere
606	Not Acceptable

2.1.3.3 I Campi Header dei messaggi

I campi Header seguono la prima linea Request-Line oppure Status-Line e sono così definiti:

Message-header = "name" HCOLON "value"

Descrizione:

- name
Indica il nome del campo header.
- value
Indica il valore del campo header.

Nei campi header più frequenti, il nome ha due rappresentazioni equivalenti: una estesa ed una compatta. Segue la descrizione dei campi header più importanti:

- Call-ID

è obbligatorio in tutti i messaggi SIP e viene utilizzato per identificare la sessione, assieme al `localtag` presente nel campo `From` ed al `remotetag` presente nel campo `To`. Il suo valore è una stringa case-sensitive. Il Call-ID è sempre creato dall'UA che inizia la procedura di chiamata e non viene mai modificato. La sua rappresentazione compatta è "i". Esempi:

- Call-ID: f81d4fae-7dec-11d0-a765-00a0c91e6bf6@biloxi.com
- Call-ID: 8d7f5d4a3f94mdhz53f0h8j
- i:f81d4fae-7dec-11d0-a765-00a0c91e6bf6@192.0.2.4

- Contact

è utilizzato per identificare chi ha generato la richiesta oppure la risposta, a seconda del tipo di messaggio. Per richieste e risposte future, si deve utilizzare questo contatto (rendendo in alcuni casi la comunicazione più diretta, con meno hop). Deve essere presente nelle richieste INVITE e nelle risposte 200 OK per questo motivo. La sua rappresentazione compatta è "m". Esempi:

- Contact: "Michele " <sip:Michele@sip.unitn.it>;q=0.7; expires=3600
- m: <sips:Michele@sip.unitn.it>;expires=60

- Content-Length

Indica la dimensione in byte del corpo del messaggio. La sua rappresentazione compatta è "l". Esempi:

- Content-length: 100
- l: 214

- Content-Type

Indica il tipo di contenuto del corpo del messaggio, è un campo obbligatorio in presenza del corpo del messaggio.

Esempi:

- Content-Type: application/sdp
- c: text/html; charset=ISO-8859-4

- CSeq
Indica il Command Sequence number ed il metodo al quale si riferisce il comando. Questo permette di rilevare una sequenza errata di comandi oppure di riconoscere il comando al quale si riferisce la risposta. Si noti che entrambi gli UA coinvolti nella comunicazione gestiscono un proprio CSeq, quindi in una comunicazione abbiamo sempre due CSeq crescenti (locale e remoto). Esempi:
 - CSeq: 4711 INVITE
- Max-Forwards
Indica il numero massimo di Proxy e Gateway attraverso i quali può essere inoltrato il messaggio. Viene impiegato per rilevare situazioni di loop e per implementare applicazioni simili a traceroute nelle reti IP. Esempi:
 - Max-Forwards: 6
- From
Indica l'URI di chi ha originato la richiesta. Può contenere anche il localtag, utilizzato per costruire l'identificatore del Dialog. La sua rappresentazione compatta è "f". Esempi:
 - From: "Bob" <sips:bob@biloxi.com> ;tag=a48s
 - From: sip:+12125551212@phone2net.com;tag=887s
 - From: Anonymous <sip:c8oqz84zk7z@privacy.org>;tag=hyh8
- Subject
Indica il soggetto della sessione. La sua rappresentazione compatta è "s". Esempi:
 - Subject: Need more boxes
 - s: Tech Support
- To
Indica l'URI alla quale è indirizzato il messaggio. Può contenere anche il remotetag, utilizzato per costruire l'identificatore del Dialog. La sua rappresentazione compatta è "t". Esempi:
 - To: The Operator <sip:Michele@sip.unitn.it>;tag=287447
 - t: sip:+12125551212@gateway.sip.unitn.it
- Via
Indica i nodi intermedi attraverso i quali è stato inoltrato il messaggio. Viene impiegato per procedure di debugging ed al fine di prevenire situazioni di loop. Esempi:

- Via: SIP / 2.0 / UDP first.example.com: 4000;ttl=16;maddr=224.2.0.1
;branch=z9hG4bKa7c6a8dlze.1

2.1.4 Il protocollo SDP

Il protocollo SDP[3], acronimo di “Session Description Protocol”, nasce in ambito IETF nel 1998 con il MMUSIC Working Group (Multiparty Multimedia Session Control) e descrive il formato dei messaggi SDP, questi hanno il compito di annunciare e quindi rendere disponibili i parametri di trasmissione dei flussi multimedia, solitamente trasportati utilizzando i protocolli RTP/RTCP. SDP è stato originariamente sviluppato per le reti multicast e soltanto in seguito è stato introdotto per annunciare le trasmissioni unicast. Le informazioni trasportate da SDP sono le seguenti:

- Nome della sessione e scopo (name and subject).
- Tempo di inizio e termine della trasmissione (timing).
- Il tipo di dati trasmesso (mediatype).
- Codifica dei dati utilizzata (encoding).
- Indirizzo di destinazione della trasmissione (solitamente, indirizzo IP e porta UDP).
- Informazioni che identificano la sorgente della trasmissione.

Si noti che anche se SDP è stato definito per annunciare le trasmissioni multimedia, viene anche utilizzato per richiederle verso un determinato indirizzo unicast, come nel protocollo di segnalazione SIP.

2.1.4.1 Il messaggio SDP

Il messaggio SDP utilizza la codifica UTF-8[14] ed è costituito da una sequenza di campi, ciascuno dei quali definito mediante la notazione ABNF[1]. La sintassi dei campi è la seguente:

$$t=v$$

Non sono ammessi spazi prima e dopo il carattere “=”, t indica il tipo di campo ed è composto da un solo carattere mentre v ne indica il valore che è terminato dalla sequenza di caratteri speciali “\n” oppure “\r\n”. L’ordine dei campi segue regole ben precise e predeterminate, questo semplifica la procedura di parsing e permette la rilevazione di possibili errori. Il messaggio si compone di tre sezioni, l’ultima di queste può ripetersi tante volte quante sono le trasmissioni che si intendono annunciare. Le sezioni sono presenti in questo ordine:

1. Session description

Vengono fornite informazioni relative all'intera sessione. Sono presenti i seguenti campi (* indica che il campo è opzionale), alcuni dei quali vengono descritti in modo esteso:

- (a) v= (protocol version)
indica la versione SDP, 0 (l'unica versione definita).
- (b) o= (owner/creator and session identifier)
- (c) s= (session name)
- (d) i=* (session information)
- (e) u=* (URI of description)
- (f) e=* (email address)
- (g) p=* (phone number)
- (h) c=* (connection information - not required if included in all media)
Indica l'indirizzo verso il quale avviene la trasmissione se non diversamente specificato nelle sezioni Media description, solitamente viene specificato un indirizzo IPv4. Non viene qui specificata la porta UDP, presente invece nelle sezioni Media description.
- (i) b=* (bandwidth information)
- (j) One or more time descriptions (see below)
- (k) z=* (time zone adjustments)
- (l) k=* (encryption key)
- (m) a=* (zero or more session attribute lines)
- (n) Zero or more media descriptions (see below)

2. Time description

Vengono fornite informazioni relative al tempo di inizio e fine della trasmissione. Sono presenti i seguenti campi (* indica che il campo è opzionale), alcuni dei quali vengono descritti in modo esteso:

- (a) t= (time the session is active)
Questo campo indica il tempo di inizio e fine della trasmissione.
- (b) r=* (zero or more repeat times)

3. Media description

Vengono fornite informazioni relative ai parametri di ciascuna trasmissione. Sono presenti i seguenti campi (* indica che il campo è opzionale), alcuni dei quali vengono descritti in modo esteso:

- (a) m= (media name and transport address)
Descrive il tipo di dati e la codifica utilizzata. Viene inoltre specificata la porta UDP verso la quale avviene la trasmissione.
- (b) i=* (media title)
- (c) c=* (connection information - optional if included at session-level)
Indica l'indirizzo verso il quale avviene la trasmissione, solitamente viene specificato un indirizzo IPv4. Questo indirizzo, se presente, sostituisce quello annunciato nella sezione Section Description.
- (d) b=* (bandwidth information)
- (e) k=* (encryption key)
- (f) a=* (zero or more media attribute lines)
Indicano ulteriori attributi associati alla trasmissione. In particolare, l'attributo di nome rtpmap permette di mappare un certo valore del campo payload type presente nell'header fisso dei messaggi RTP con un determinato tipo di dati.

Segue un esempio di messaggio SDP:

```
v=0
o=Michele 123456 654321 IN IP4 192.168.2.8
s=A conversation
c=IN IP4 192.168.2.8
t=0 0
m=audio 7078 RTP/AVP 0 111 110 3 8 101
a=rtpmap:0 PCMU/8000/1
a=rtpmap:111 speex/16000/1
a=rtpmap:110 speex/8000/1
a=rtpmap:3 GSM/8000/1
a=rtpmap:8 PCMA/8000/1
m=video 9078 RTP/AVP 97 98 99
a=rtpmap:97 theora/90000
a=rtpmap:98 H263-1998/90000
a=rtpmap:99 MP4V-ES/90000
```

In questo messaggio sono presenti una sezione Session description, seguita da una sezione Time description ed in fine due sezioni Media description. Vengono annunciate una trasmissione audio ed una trasmissione video.

2.1.4.2 Utilizzo di SDP in SIP

I messaggi SDP vengono trasportati nei messaggi SIP come corpo del messaggio, in questo caso il campo header Content-Type avrà come valore "application/sdp". Solitamente è presente nelle richieste di tipo INVITE, ACK ed UPDATE oppure nelle risposte 200 OK. Alcuni campi nel messaggio SDP non hanno alcun senso ma devono comunque essere presenti per rispettare le regole imposte dal protocollo. In particolare, si utilizzano questi valori per i campi "s" e "t":

```
s=-
t= 0 0
```

2.1.5 I modelli di segnalazione

Il network SIP di riferimento è descritto nella figura 2.3. Esistono due differenti modelli di segnalazione che differiscono in base al ruolo che il SIP Proxy Server ricopre nelle fasi che coinvolgono la comunicazione:

1. Direct

Gli UAs coinvolti gestiscono direttamente il Call Signaling senza l'intervento del SIP Proxy Server, come mostrato in figura 2.1.

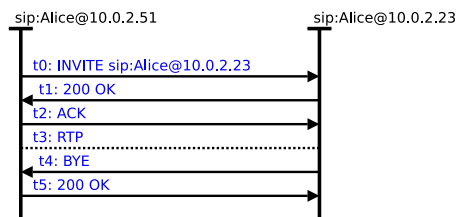


Figura 2.1: SIP Direct Signaling

2. Proxy Routed

Gli UAs coinvolti gestiscono il Call Signaling attraverso il SIP Proxy Server, come mostrato in figura 2.2.

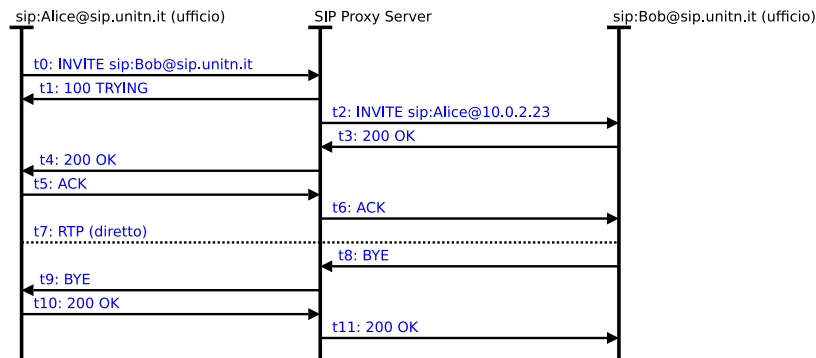


Figura 2.2: SIP Proxy Routed Signaling

Nel primo modello di segnalazione il SIP Proxy Server non viene coinvolto in alcun modo nella comunicazione e quindi non può attuare alcun tipo di controllo oppure limitazione sul tipo e sul numero di comunicazioni effettuate nel network SIP.

2.1.6 Alcuni possibili scenari

Il network SIP di riferimento è descritto nella figura 2.3. In questo network SIP si è deciso di differenziare il modello di Call Signaling a seconda del tipo di chiamata. Se la chiamata è completamente interna oppure esterna, viene utilizzato il Direct Signaling, eventualmente assieme al SIP Registration/Redirect Server. Se la chiamata proviene oppure è diretta verso Internet, viene utilizzato il Proxy Routed Signaling. In questo secondo caso, il SIP Server agisce come un SIP Registration/Proxy Server.

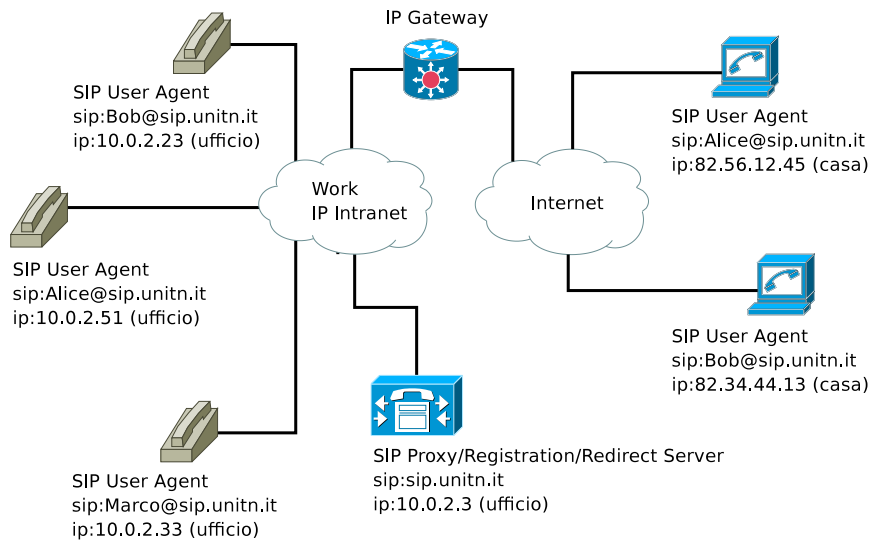


Figura 2.3: network SIP di riferimento

Seguono due esempi.

1. Alice è in ufficio mentre Bob è a casa. Entrambi si registrano sul SIP Registration Server dell'ufficio e quando Bob chiama Alice viene utilizzato il Proxy Routed Signaling, come mostrato in figura 2.4.

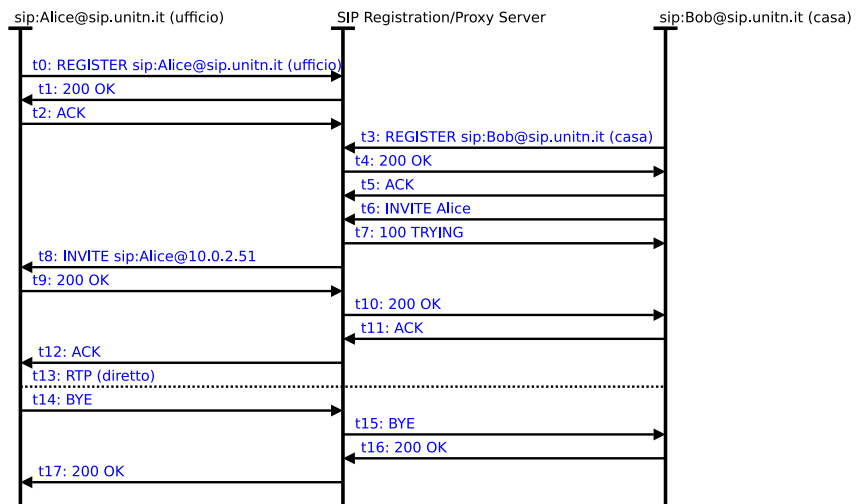


Figura 2.4: Scenario SIP 1

2. Alice e Bob sono a casa. Entrambi si registrano sul SIP Registration Server dell'ufficio e quando Bob chiama Alice viene utilizzato il Direct Signaling, come mostrato in figura 2.5.

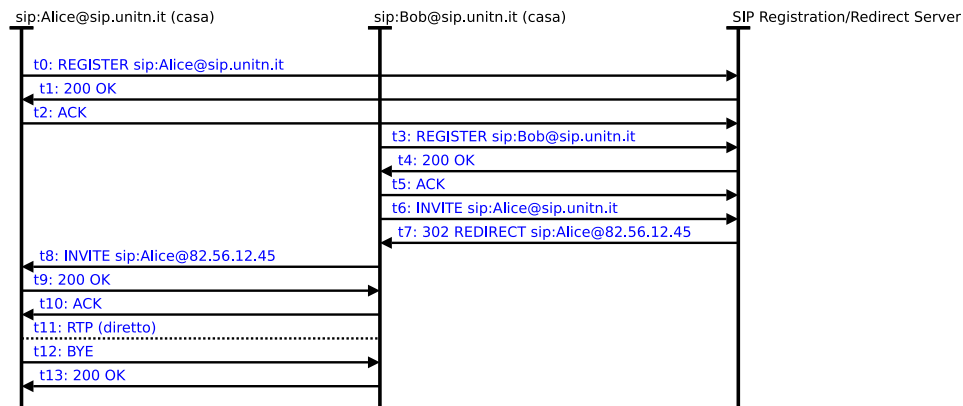


Figura 2.5: Scenario SIP 2

2.2 Il protocollo di segnalazione H.323

La raccomandazione H.323[9] nasce in ambito ITU-T nel 1996 con la creazione dell'ITU-T Study Group 16 (SG16). Non definisce alcun protocollo ma descrive invece come un certo insieme di altre raccomandazioni e protocolli debbano collaborare per costituire un network H.323. Questi protocolli di segnalazione sono di livello applicativo ed hanno come obiettivo l'instaurazione, la modifica e la terminazione di una comunicazione multimedia. Le raccomandazioni più importanti sono:

1. H.225.0[5]
Definisce le procedure di segnalazione per il controllo del terminale e della chiamata.
2. H.245[8]
Definisce come avviene la comunicazione dei parametri di trasmissione dei flussi multimedia.

Il trasporto dei flussi multimedia è affidato ai protocolli RTP/RTCP. Esistono estensioni che permettono la cifratura del Signaling, le trasmissioni RTP/RTCP sono comunque non cifrate.

2.2.1 L'indirizzamento

H.323 utilizza due differenti schemi di indirizzamento:

1. H.323 URI

Il più semplice ed il più diffuso nei network H.323. Esempi:

- h323:marco@science.unitn.it:1067
- h323:+390461012345@pstn_gateway.science.unitn.it

2. H.323 Alias

Permette di associare un Alias al terminale, che deve essere univoco all'interno della zona (quindi verrà risolto in modo univoco dal Gatekeeper di riferimento per la zona). Esempi:

- marco
- segreteria
- 200

3. Alcuni componenti del network H.323 possono supportare anche altri schemi di indirizzamento che possono essere utilizzati per comunicare con utenti presenti in altri tipi di rete, tramite un H.323 Gateway Server. Il più importante è lo schema definito in E.164[7], utilizzato nelle reti telefoniche tradizionali. Esempi:

- 118
- +0461821234

2.2.2 I componenti dell'architettura

I componenti del network H.323 sono molteplici e ciascuno assolve ad un particolare compito:

1. H.323 Terminal
2. H.323 Servers
 - (a) H.323 Gatekeeper
 - (b) H.323 Multipoint Control Unit
 - (c) H.323 Gateway Server
3. H.323 Zones

Vengono di seguito descritti in modo esteso:

1. Terminal

Indica qualsiasi device in grado di accettare oppure richiedere una comunicazione. E' il nodo terminale di qualsiasi comunicazione e

rappresenta l'interfaccia fra l'utente ed il network. Deve supportare almeno il codec audio definito nella raccomandazione *G.711*. In maniera opzionale, un terminale può supportare altri Codec audio definiti in altre raccomandazioni come *G.723.1* e *G.729*. Il codec video è un componente opzionale in un terminale *H.323*. Se presente, deve essere in grado di codificare e decodificare un segnale video implementando almeno la raccomandazione *H.261*. Il canale dati è un elemento opzionale in un terminale *H.323*. Se presente, deve necessariamente trasmettere le informazioni seguendo le procedure definite nella raccomandazione *T.120*.

2. H.323 Servers

Sono sistemi che accettano richieste *H.323* e sono in grado di rispondere, fornendo differenti servizi e funzionalità ai terminali. Ciascun Server è in grado di gestire soltanto un sottoinsieme di funzionalità:

(a) Gatekeeper

Quando presente, regola e limita l'accesso alle risorse del network *H.323*. Il controllo esercitato dal Gatekeeper consiste nel fornire oppure negare l'autorizzazione ai terminali controllati che intendono eseguire un qualche tipo di operazione nel network *H.323*. Le sue principali funzionalità sono le seguenti:

i. Address Translation

Permette la risoluzione degli indirizzi Alias *H.323* ed *E.164*. Questa funzionalità è implementata mantenendo le informazioni riguardanti le registrazioni precedentemente effettuate in un qualche modo non definito da *H.323*. Un Gatekeeper, anche se generalmente opzionale, è invece assolutamente necessario se sono presenti dei Gateway Server che ad esempio interconnettono il network *H.323* con reti ad esempio SIP.

ii. Admission Control

Permette di limitare l'accesso al network *H.323*. Questa funzionalità viene implementata dal Gatekeeper con i messaggi *ARQ*, *ACF* e *ARJ*.

iii. Bandwidth Control

Permette di controllare l'utilizzo della capacità trasmissiva del network *H.323* potendo così cercare di garantire la qualità del servizio. Questa funzionalità viene implementata dal Gatekeeper con i messaggi *BRQ*, *BCF* e *BRJ*. Per aumentare il controllo del Gatekeeper sull'utilizzo delle risorse utilizzate, può eventualmente gestire anche il Call Signaling. In questo modo potrà ottenere anche informazioni riguardo ciascun flusso multimedia.

iv. Gestione delle Zone

Permette di gestire l'interconnessione tra più zone H.323.

La sua presenza è opzionale.

(b) H.323 Multipoint Control Unit (MCU)

Si occupa di gestire le conferenze multiutente. Il Multipoint Control Unit (MCU) è costituito da due componenti: Il Multipoint Controller (MC) ed il Multipoint Processor (MP). Mentre la presenza del primo è obbligatoria, il secondo è opzionale. Il Multipoint Controller (MC) si occupa di gestire i canali di controllo H.245, ricevendo e ritrasmettendo i flussi multimedia dei partecipanti. Il Multipoint Processor (MP) si occupa invece di elaborare i flussi multimedia, permettendone la ricodifica in un differente formato oppure unendoli. Questa funzionalità permette di adattare la trasmissione alle risorse di rete di ciascun terminale. Le tipologie di conferenza supportate da un sistema MCU sono quattro:

i. Centralizzata:

Il MCU riceve dai partecipanti il Signaling H.245 e i flussi multimedia, ritrasmettendoli.

ii. Decentralizzata

Il MCU riceve dai partecipanti soltanto il Signaling H.245 mentre i flussi multimedia vengono trasmessi e ricevuti in modalità multicast direttamente tra i partecipanti.

iii. Ibrida

I partecipanti comunicano sia in modo centralizzato che decentralizzato. Ad esempio, in una singola conferenza si può avere la trasmissione dei flussi audio centralizzata e la trasmissione dei flussi video decentralizzata.

iv. Mista

Alcuni partecipanti partecipano in modo centralizzato mentre altri in modo decentralizzato.

La sua presenza è opzionale.

(c) H.323 Gateway Server

Si occupa di interconnettere il network H.323 con altri tipi di network come la rete telefonica tradizionale. Svolge quindi le funzioni di adattamento per il trasporto dei flussi multimedia e del Signaling. Si compone di due parti logiche:

i. Signaling Gateway

Permette l'adattamento del Signaling.

ii. media gateway

Permette l'adattamento dei flussi multimedia.

La sua presenza è opzionale.

3. Zones

Nei network H.323 si possono individuare delle aree amministrative, chiamate zone, definite come insieme di componenti H.323 (Gateway, Terminali, MCU) gestite da un singolo Gatekeeper. I limiti di una zona possono essere basati su limiti amministrativi, struttura di naming, confini geografici, ecc. Le chiamate all'interno di una zona sono gestite da un unico Gatekeeper, mentre le chiamate che coinvolgono differenti zone sono gestite da più Gatekeeper.

2.2.3 La raccomandazione H.225.0

La raccomandazione H.225.0 definisce le procedure di segnalazione che i componenti del network H.323 devono utilizzare. Queste si dividono in:

1. Registration, Admission and Status (RAS) Signaling
2. Call Signaling

Vengono di seguito descritte in modo esteso:

1. Registration, Admission and Status (RAS) Signaling

Questo insieme di procedure permette al Gatekeeper di coordinare e gestire le comunicazioni tra i componenti nel network H.323, fornendo le seguenti funzionalità:

(a) Registration

Descrive come i terminali si debbano registrare al Gatekeeper, al fine di annunciare la loro presenza. Durante la registrazione viene indicato il canale logico H.225.0 che deve essere utilizzato per il Call Signaling e l'indirizzo alias H.323 col quale si desidera essere riconosciuti. Questa funzionalità viene implementata con i seguenti messaggi:

- RRQ (Registration ReQuest)
Trasmesso dal terminale al Gatekeeper, richiede la registrazione.
- RCF (Registration ConFirmation)
Trasmesso dal Gatekeeper in risposta ad un messaggio RRQ, conferma l'avvenuta registrazione.
- RRJ (Registration Reject)
Trasmesso dal Gatekeeper in risposta ad un messaggio RRQ, rifiuta la registrazione.

Il terminale può anche deregistrarsi, questa funzionalità viene implementata con i seguenti messaggi:

- URQ (Unregistration ReQuest)
Trasmesso dal terminale al Gatekeeper, richiede la deregistrazione.
- UCF (Unregistration Confirmation)
Trasmesso dal Gatekeeper in risposta ad un messaggio URQ, accetta la deregistrazione.
- URJ (Unregistration Reject)
Trasmesso dal Gatekeeper in risposta ad un messaggio URQ, rifiuta la deregistrazione. Ad esempio, se il terminale è al momento impegnato in una conversazione, deve prima terminare questa per deregistrarsi.

Il Gatekeeper può essere indicato esplicitamente, altrimenti si ricorre alla procedura di Gatekeeper discovery (che utilizza il protocollo di trasporto UDP, porta 1718). Questa funzionalità viene implementata con i seguenti messaggi:

- GRQ (Gatekeeper ReQuest)
Trasmesso dal terminale all'indirizzo multicast 224.0.1.41 porta UDP 1718, viene utilizzato nella procedura di Gatekeeper discovery per richiedere quali Gatekeeper siano disponibili.
- GRJ (Gatekeeper ReJect)
Trasmesso dal Gatekeeper come risposta ad un messaggio GRQ, indica che il Gatekeeper è presente ma non disponibile per la registrazione del terminale. Può fornire però una lista di altri Gatekeeper.
- GCF (Gatekeeper ConFirmation)
Trasmesso dal Gatekeeper come risposta ad un messaggio GRQ, indica che il Gatekeeper è presente e disponibile per la registrazione del terminale.

(b) Admission

Descrive come i terminali debbano contattare il Gatekeeper per richiedere oppure accettare una comunicazione. In generale, i terminali sono identificati all'interno del network H.323 tramite un indirizzo alias H.323. Occorre quindi una procedura per la loro localizzazione, intesa a risolvere questo tipo di indirizzi. In presenza di un Gatekeeper, è quest'ultimo a occuparsi di tale operazione. Oltre al chiamante, anche il chiamato deve ottenere il permesso dal Gatekeeper per poter iniziare la comunicazione. Qualsiasi comunicazione, in presenza di un Gatekeeper, richiede quindi la sua esplicita autorizzazione in entrambe le direzioni. Per terminare la comunicazione, nuovamente occorre ottenere il permesso dal Gatekeeper. La notifica di terminazione di una comunicazione è molto importante per il Bandwidth Control ed in genera-

le migliora l'utilizzo delle risorse di rete. Questa funzionalità viene implementata con i seguenti messaggi:

- **ARQ (Admission ReQuest)**
Trasmesso dal terminale (inizio procedura di chiamata) oppure dal Gatekeeper (inizio procedura di accettazione chiamata in ingresso), richiede di partecipare ad una comunicazione.
- **ACF (Admission ConFirm)**
Trasmesso dal terminale oppure dal Gatekeeper, indica che la richiesta è stata accettata, specificando inoltre il canale logico H.255.0 che deve essere utilizzato per il Call Signaling.
- **ARJ (Admission ReJect)**
Trasmesso dal terminale oppure dal Gatekeeper, indica che la richiesta è stata rifiutata

Se il Gatekeeper non possiede le informazioni necessarie, è suo compito cercarle. Questa funzionalità (che utilizza il protocollo di trasporto UDP, porta 1718) viene implementata con i seguenti messaggi:

- **LRQ (Location ReQuest)**
Trasmesso dal Gatekeeper verso altri Gatekeeper oppure verso l'indirizzo multicast 224.0.1.41:1718, richiede informazioni riguardo la risoluzione di un alias H.323.
- **LCF (Location ConFirmation)**
Trasmesso dal Gatekeeper come risposta ad un messaggio LRQ, contiene le informazioni richieste.

Per terminare la comunicazione, nuovamente occorre ottenere il permesso dal Gatekeeper. La notifica di terminazione di una comunicazione è molto importante per il Bandwidth Control ed in generale migliora l'utilizzo delle risorse di rete. Questa funzionalità viene implementata con i seguenti messaggi:

- **Bandwidth ReQuest (BRQ)**
- **Bandwidth ConFirmed (BCF)**
- **Bandwidth Rejected (BRJ)**

(c) Status

Questa procedura descrive come i terminali informano il Gatekeeper circa il loro stato. Questa funzionalità viene implementata con i seguenti messaggi:

- **IRQ (Information ReQuest)**
Trasmesso dal terminale oppure dal Gatekeeper, richiede un qualche tipo di informazione.

- IRR (InfoRmation Response)

Trasmesso in risposta ad un messaggio IRQ, contiene le informazioni richieste.

I messaggi definiti in questa raccomandazione costituiscono il canale logico del RAS Signaling e vengono trasmessi utilizzando il protocollo TCP, porta 1719. Questa raccomandazione è utile unicamente in presenza di almeno un Gatekeeper, altrimenti diventa completamente inapplicata. La comunicazione diretta fra terminali infatti richiede soltanto le procedure di Call Signaling.

2. Call Signaling

Questo insieme di procedure permette ai terminali di richiedere, accettare e terminare una comunicazione. Queste funzionalità vengono implementate con i seguenti messaggi:

(a) CallProceeding

Trasmesso come risposta ad messaggio Setup, indica che la richiesta è stata ricevuta e la si sta processando.

(b) Setup

Trasmesso per iniziare la procedura di chiamata. In presenza di un Gatekeeper, segue la trasmissione e ricezione dei messaggi ARQ e ACF.

(c) Alerting

Trasmesso per indicare al chiamante che si sta allertando il chiamato in qualche modo.

(d) ReleaseComplete

Trasmesso per terminare la comunicazione, non prevede alcuna risposta.

(e) Connect

Trasmesso come risposta di un messaggio Setup, indica che la richiesta è stata accettata e quindi che l'utente ha accettato la chiamata.

(f) Progress

Trasmesso dal Gateway per indicare che il sistema sta processando la richiesta.

(g) Facility

Trasmesso per trasferire la richiesta. Viene utilizzato per implementare il servizio Call Forwarding.

I messaggi definiti in questa raccomandazione costituiscono il canale logico del Call Signaling e vengono trasmessi utilizzando il protocollo di trasporto TCP, porta 1720. Per velocizzare la fase di

instaurazione della comunicazione, nelle ultime versioni della raccomandazione è definita la possibilità di includere in parte i messaggi H.245 mediante il campo FastStart nei messaggi H.225.0 che riguardano il Call Signaling.

2.2.3.1 I messaggi H.225.0

I messaggi della raccomandazione H.225.0 sono definiti utilizzando la notazione ASN.1 e sono trasmessi utilizzando il protocollo di trasporto TCP, incapsulati in pacchetti Q.931[4], questi a loro volta incapsulati in pacchetti TPKT[11]. Il protocollo TPKT permette di riconoscere nel flusso l'inizio e la fine dei singoli pacchetti Q.931. Il formato dell'header dei pacchetti TPKT è il seguente, definito come struttura C:

```
struct h323_tpkt_hdr
{
    u_int8_t vrsn; // indica la versione TPKT. L'attuale è 3
    u_int8_t reserved; // riservato per scopi futuri
    u_int16_t length; // lunghezza del messaggio che segue
                    immediatamente
};
// segue il messaggio
```

L'header TPKT è immediatamente seguito dal pacchetto Q.931. Questo ulteriore Layer è previsto per semplificare l'interworking con le reti ISDN. I messaggi sono infine codificati in ASN.1 Packet Encoding Rules (PER)[6].

2.2.4 La raccomandazione H.245

La raccomandazione H.245 definisce come devono essere concordati i parametri di trasmissione dei flussi multimedia, fornendo queste funzionalità:

1. Determinazione Master/slave

Permette di assegnare il ruolo di Master ad uno dei terminali partecipanti alla chiamata. Gli altri terminali assumono il ruolo di Slave. l'importanza dell'assegnazione di questi ruoli si manifesta nella risoluzione di possibili conflitti quando più terminali inizializzano contemporaneamente la procedura per uno stesso evento.

2. Scambio delle capacità dei terminali

Permette di assicurare che il ricevente sia in grado di decodificare il flusso multimedia che riceve.

3. Segnalazione dei Canali Logici

Permette l'apertura e la terminazione di un canale logico. A ciascun canale logico è associato un flusso multimedia.

Queste funzionalità vengono implementate con i seguenti messaggi:

1. Open Logical Channel

Trasmesso per richiedere l'apertura di un canale logico.

2. Open Logical Channel Acknowledge

Trasmesso in seguito alla ricezione di un messaggio Open Logical Channel, indica che la richiesta è stata accettata. Nel caso di una comunicazione bidirezionale, il messaggio contiene anche i parametri della trasmissione dei flussi multimedia nella direzione opposta.

3. Open Logical Channel Reject

Trasmesso in seguito alla ricezione di un messaggio Open Logical Channel, indica che la richiesta è stata rifiutata.

4. Close Logical Channel

Trasmesso per richiedere la terminazione di un canale logico.

5. Close Logical Channel Acknowledge

Trasmesso in seguito alla ricezione di un messaggio Close Logical Channel, rifiuta la richiesta.

I messaggi H.245 sono definiti utilizzando la notazione ASN.1 e codificati in ASN.1 Packet Encoding Rules (PER)[6]. La raccomandazione non specifica come questi messaggi debbano essere scambiati, solitamente vengono trasportati nel canale logico utilizzato per trasportare i messaggi H.225.0 in modo diretto oppure mediante il campo FastStart, presente in alcuni messaggi H.225.0.

2.2.5 I modelli di segnalazione

Il network H.323 di riferimento è descritto nella figura 2.6.

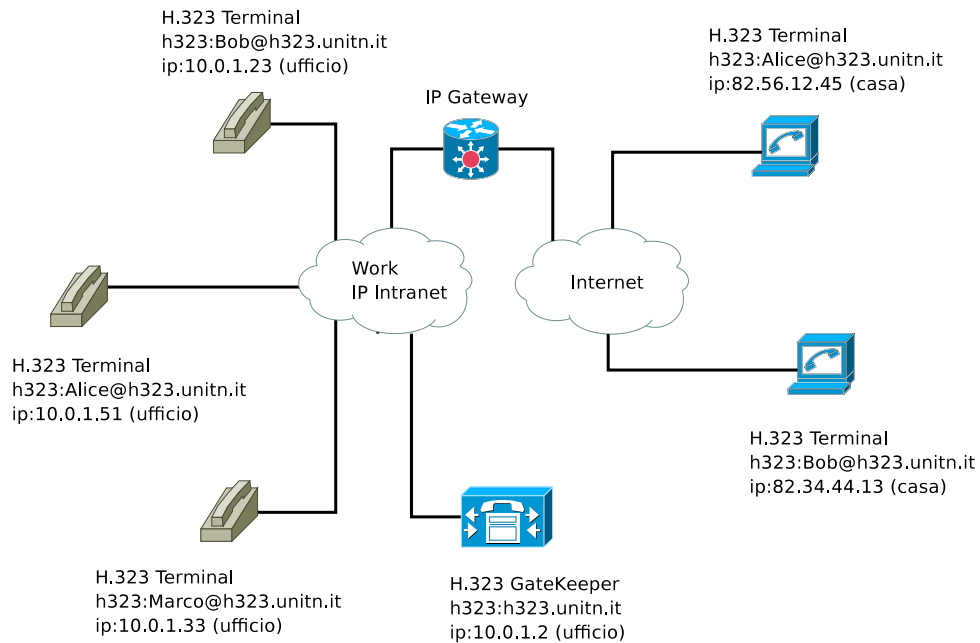


Figura 2.6: network H.323 di riferimento

Esistono differenti modelli di segnalazione che differiscono in base al ruolo che il Gatekeeper ricopre nelle fasi che coinvolgono la comunicazione:

1. Direct

I terminali coinvolti gestiscono direttamente il Call Signaling ed il Signaling H.245, senza l'intervento del Gatekeeper (che si occupa soltanto del RAS Signaling), come mostrato in figura 2.7. Il RAS Signaling non è comunque obbligatorio.

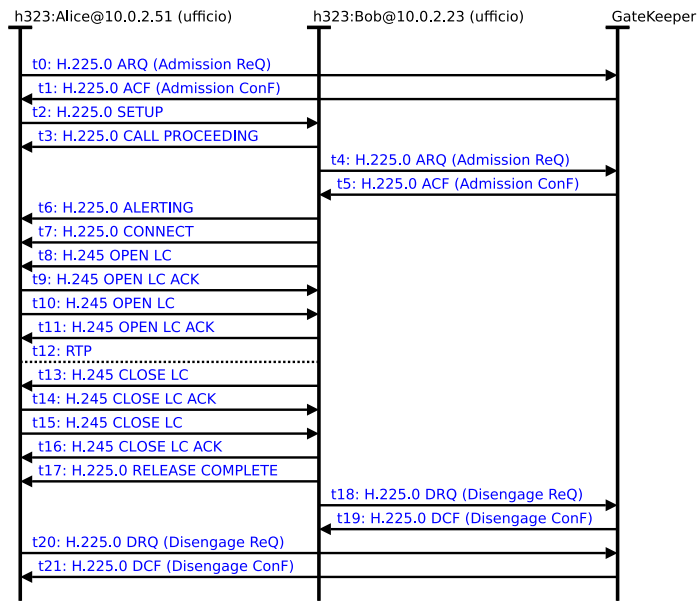


Figura 2.7: H.323 Direct Signaling

2. Gatekeeper Routed (H.225.0)

I terminali coinvolti gestiscono direttamente il Signaling H.245 lasciando gestire il Call Signaling al Gatekeeper, come mostrato in figura 2.8.

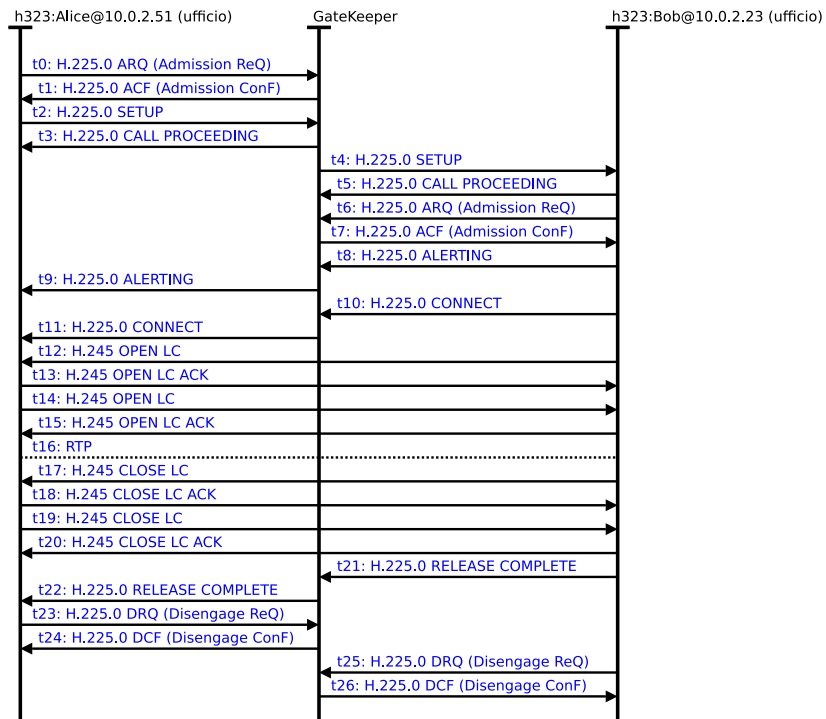


Figura 2.8: H.323 Routed H.225.0 Signaling

3. Gatekeeper Routed (H.225.0/H.245)

Il Gatekeeper si occupa sia del Call Signaling che del Signaling H.245, come mostrato in figura 2.9.

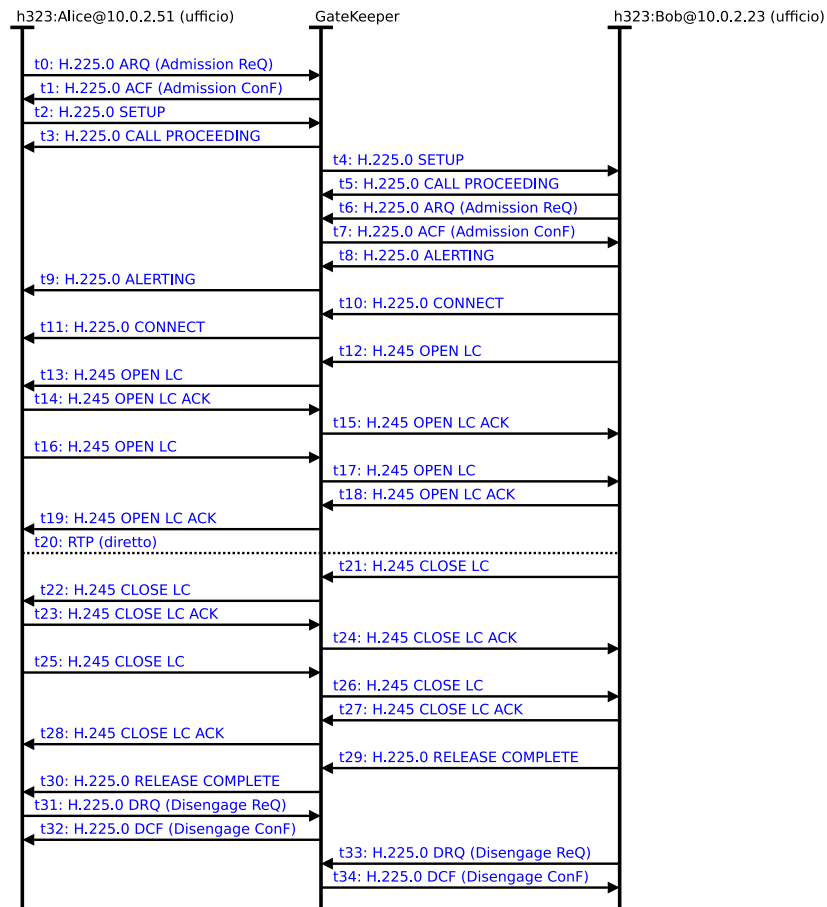


Figura 2.9: H.323 Routed H.225.0/H.245 Signaling

Confrontando i tre modelli di segnalazione, possiamo notare che la differenza consiste da una parte nelle risorse richieste dal Gatekeeper per la gestione delle comunicazioni, dall'altra nelle informazioni che il Gatekeeper detiene sulle comunicazioni attive. Queste ultime possono risultare utili per la gestione del servizio, in maniera tale da garantire che le comunicazioni sperimentino una qualità del servizio adeguata. In particolare, nel modello Direct, il Gatekeeper non gestisce alcuna informazione relativa alla chiamata, essendo a conoscenza solo delle informazioni scambiate sul canale logico per il RAS Signaling. In questo caso, le risorse impegnate dal Gatekeeper sono ovviamente minime. Nei casi Gatekeeper Routed (H.225.0) e Gatekeeper Routed (H.225.0/H.245) le risorse impegnate sono maggiori e, in particolare nel secondo caso, il Gatekeeper oltre a gestire il Call Signaling si occupa anche del Signaling H.245. La gestione diretta di queste informazioni

permette al Gatekeeper di conoscere non solo le chiamate attive (informazioni che può ricavare dalla gestione del Call Signaling) ma anche il numero ed il tipo di flussi multimedia attivi.

2.3 Il protocollo di segnalazione SCCP

Il protocollo SCCP, acronimo di “Skinny Client Control Protocol” e proprietario Cisco, è stato sviluppato originariamente dalla Selsius Corporation e poi acquisito da Cisco negli anni '90. Come ricordo del suo passato, il nome di default degli IP Phone Cisco è SEP (Selsius Ethernet Phone) seguito dal MAC address. Il punto di forza di SCCP sono i requisiti richiesti dai client, molto inferiori rispetto ai terminali H.323 ed dagli UAs SIP. In SCCP si è deciso di concentrare la complessità e quindi l'intelligenza in un server, il Call Manager. In questo modo è stato possibile per Cisco sviluppare dei client funzionali ma semplici e quindi meno costosi e più competitivi rispetto alle soluzioni H.323 e SIP. Il Call Manager integra un Gateway H.323, questo permette l'interconnessione della rete SCCP con una rete H.323. Viene qui analizzato in modo parziale perché il documento di specifica del protocollo non è pubblico e non è stato possibile recuperarlo in alcun modo. Le informazioni sono state ottenute in modo diretto attraverso l'analisi di tracciati di rete oppure studiando il codice sorgente dell'applicazione OpenSource Wireshark. Non esistono estensioni che permettono la cifratura del Signaling e delle trasmissioni RTP/RTCP.

2.3.1 L'indirizzamento

SCCP utilizza lo schema di indirizzamento definito in E.164[7].

2.3.2 I componenti dell'architettura

I componenti del network SCCP sono due:

1. Skinny Client

Indica l'unico componente client del network, questo device è in grado soltanto di informare il Call Manager circa il suo stato e di eseguire le operazioni richieste sempre dal Call Manager, dal quale dipende completamente.

2. Skinny Call Manager

Indica l'unico componente server del network. Sincronizza e coordina le comunicazioni tra i client, implementando anche la funzionalità di Gateway SCCP-H.323, come presentato in figura 2.10.

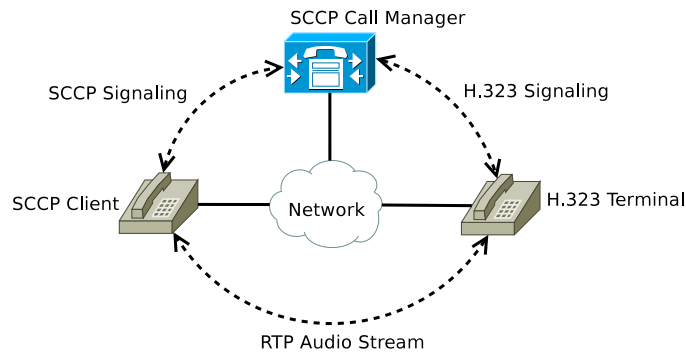


Figura 2.10: SCCP Call Manager Gateway

2.3.3 I messaggi SCCP

I componenti dell'architettura comunicano scambiandosi messaggi definiti dal protocollo SCCP. I messaggi sono trasmessi utilizzando il protocollo di trasporto TCP oppure UDP, porta 2000. I messaggi SCCP sono "codificati" in strutture definite nel linguaggio C, il Byte Order è Little-Endian. Sono un'eccezione i campi che rappresentano indirizzi IPv4, codificati in Network Byte Order. Il parsing, utilizzando il linguaggio C, si riduce quindi alla sola inizializzazione di un puntatore se l'architettura del device utilizza il Byte Order Little-Endian, come l'architettura Intel. Il formato dei messaggi viene definito mediante strutture C. I messaggi SCCP sono composti da un header che svolge la stessa funzione del protocollo TPKT utilizzato in H.323, così definito:

```
struct SCCPPrefix {
    UINT32 length; //Length of the SCCP message to follow
    UINT32 type; //Unused, must be set to zero
};
```

Questo header permette di riconoscere l'inizio e la fine di ciascun messaggio all'interno del flusso TCP. Il tipo di messaggio che segue (che identifica la struttura C da considerare) è indicato nei 32 bit che seguono l'header e che rappresentano anche il primo campo della struttura C da considerare. Questo campo è sempre presente. Stranamente, non viene utilizzato il campo type di SCCPPrefix che è stato probabilmente definito a tale scopo. Attraverso i messaggi SCCP, vengono implementate le seguenti funzionalità:

1. Registrazione

Questa funzionalità viene implementata con i seguenti tipi di messaggio:

- **StationRegister**
Trasmesso dal client, richiede la registrazione al Call Manager.
- **StationUnregister**
Trasmesso dal client, richiede la deregistrazione al Call Manager.
- **StationRegisterAcknowledge**
Trasmesso dal Call Manager, conferma l'avvenuta registrazione.
- **StationRegisterReject**
Trasmesso dal Call Manager, rifiuta la registrazione.
- **StationUnregisterAck**
Trasmesso dal Call Manager, conferma l'avvenuta deregistrazione.

2. Call Signaling

- **Station Key Pad**
Trasmesso dal client, informa il CallManager che una cifra è stata premuta sul keypad.
- **4.2.1.3 Station Stimulus**
Trasmesso dal client, informa il CallManager che un qualche tasto è stato premuto.
- **4.2.2.8 Station Call Information**
Trasmesso dal Call Manager, informa il client circa l'identità dei client coinvolti in una comunicazione.
- **4.2.2.13 StationCallState**
Trasmesso dal Call Manager, informa il client circa lo stato della comunicazione.

3. Comunicazione dei parametri di ricezione del flusso multimedia

- **StationOpenReceiveChannel**
Trasmesso dal Call Manager, informa il client che un canale di ricezione deve essere aperto, richiedendo verso quale porta UDP attivare la trasmissione.
- **StationOpenReceiveChannelAck**
Trasmesso dal client, informa il Call Manager circa lo stato di apertura del canale di ricezione ed indica la porta UDP verso la quale attivare la trasmissione.
- **StationCloseReceiveChannel**
Trasmesso dal Call Manager, indica al client che deve chiudere il canale.

- **StationStartMediaTransmission**
Trasmesso dal Call Manager, indica al client che deve iniziare la trasmissione del flusso multimedia, indicando l'indirizzo IPv4 e la porta UDP di destinazione.
- **StationStopMediaTransmission**
Trasmesso dal Call Manager, indica al client di terminare la trasmissione.

2.3.4 I modelli di segnalazione

Il network SCCP di riferimento è descritto nella figura 2.11.

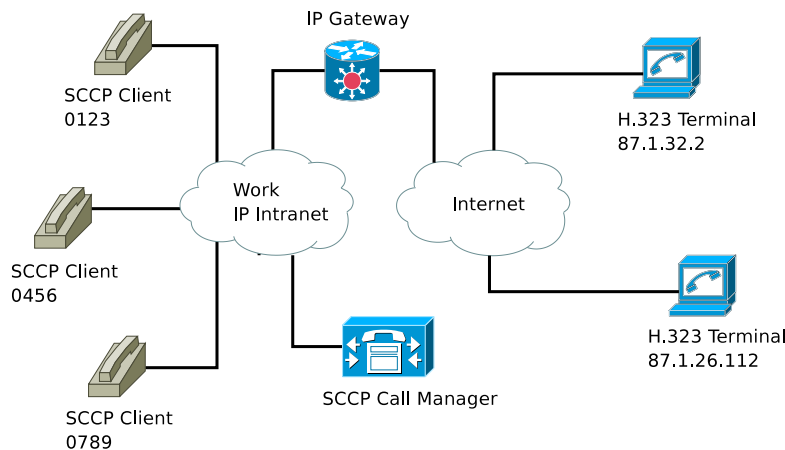


Figura 2.11: network SCCP di riferimento

Esiste un unico modello di segnalazione, i Client dipendono completamente dal Call Manager. Questo detiene qualsiasi tipo di informazione sullo stato di ciascun componente, permettendo il controllo completo delle comunicazioni e del network SCCP. Nella figura 2.12 viene mostrato il Call Signaling di una conversazione tra i Client 0123 e 045.

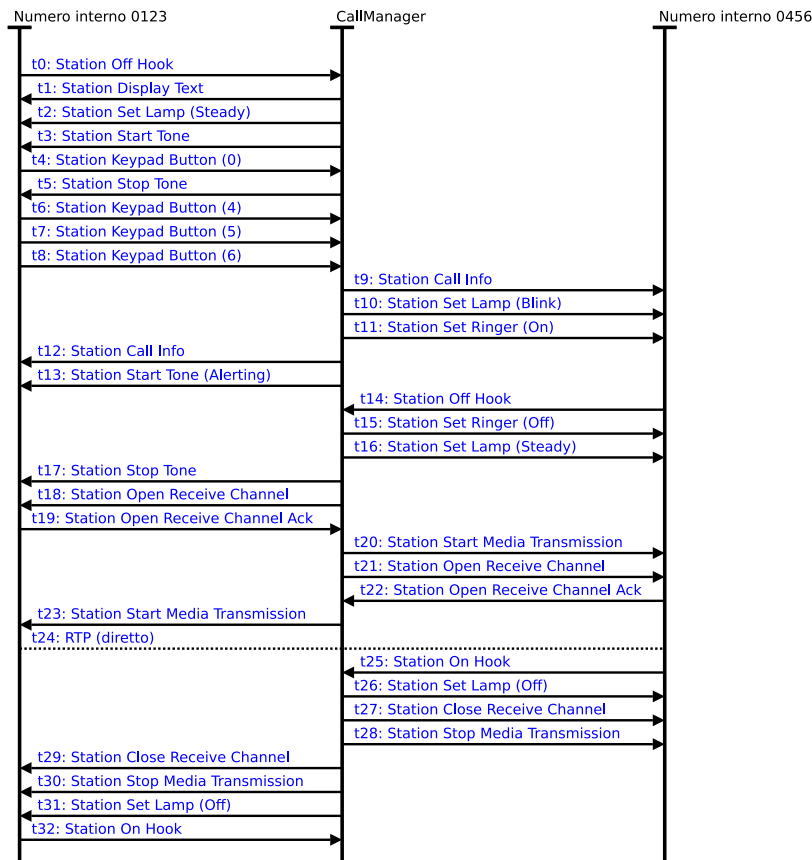


Figura 2.12: SCCP Routed Signaling

Viene di seguito descritto ogni istante, dove CL1 è il Client che ha come identificativo 0123, CL2 è il Client che ha come identificativo 0456 e CM è il Call Manager:

- t0: il CL1 informa il CM che “è stata alzata la cornetta”
- t1: il CM richiede al CL1 la visualizzazione sullo schermo di un messaggio testuale
- t2: il CM richiede al CL1 di attivare la retroilluminazione
- t3: il CM richiede al CL1 di attivare il tono “steady” (libero)
- t4: il CL1 informa il CM che è stato premuto il tasto “0”
- t5: il CM richiede al CL1 di disattivare il tono “steady”
- t6: il CL1 informa il CM che è stato premuto il tasto “4”

- t7: il CL1 informa il CM che è stato premuto il tasto “5”
- t8: il CL1 informa il CM che è stato premuto il tasto “6”
- t9: il CM informa il CL2 circa l'identità di CL1
- t10: il CM richiede al CL2 di attivare la retroilluminazione intermittente
- t11: il CM richiede al CL2 di attivare la suoneria
- t12: il CM informa il CL1 circa l'identità di CL2
- t13: il CM richiede al CL1 di attivare il tono “alerting” (si sta allertando l'utente)
- t14: il CL2 informa il CM che “è stata alzata la cornetta”
- t15: il CM richiede al CL2 di disattivare il tono “ringing”
- t16: il CM richiede al CL2 di attivare la retroilluminazione
- t17: il CM richiede al CL1 di disattivare il tono “alerting”
- t18: il CM richiede al CL1 di aprire un canale di ricezione (RTP)
- t19: il CL1 risponde al CM, informandolo sui parametri di ricezione
- t20: il CM richiede al CL2 di iniziare la trasmissione RTP verso CL1, fornendogli i parametri
- t21: il CM richiede al CL2 di aprire un canale di ricezione (RTP)
- t22: il CL2 risponde al CM, informandolo sui parametri di ricezione
- t23: il CM richiede al CL1 di iniziare la trasmissione RTP verso CL2, fornendogli i parametri
- t24: trasmissione RTP...
- t25: il CL1 informa il CM che “la cornetta è stata abbassata”
- t26: il CM richiede al CL2 di disattivare la retroilluminazione
- t27: il CM richiede al CL2 di chiudere il canale di ricezione
- t28: il CM richiede al CL2 di chiudere il canale di trasmissione
- t29: il CM richiede al CL1 di chiudere il canale di ricezione
- t30: il CM richiede al CL1 di chiudere il canale di trasmissione

- t31: il CM richiede al CL1 di disattivare la retroilluminazione
- t32: il CL1 informa il CM che “la cornetta è stata abbassata”

Questa descrizione mette in evidenza quanto i Client dipendano dal Call Manager, per qualsiasi funzionalità.

Capitolo 3

I protocolli RTP/RTCP

I protocolli RTP/RTCP[13], nati in ambito IETF nel 1996, descrivono un insieme di funzionalità che permettono il trasporto di dati Real Time senza occuparsi in alcun modo del QoS. Si noti che RTP non richiede necessariamente RTCP ma la sua presenza può essere determinante per ottenere migliori prestazioni, solitamente quindi si utilizzano assieme. RTP/RTCP non fa parte del livello di trasporto ma dell'applicazione, lo sviluppatore deve quindi implementare i protocolli RTP/RTCP integrandoli nell'applicazione stessa. Vengono anche utilizzati assieme ai protocolli di segnalazione VoIP come SIP, H.323 ed SCCP per il trasporto dei flussi multimedia. Le informazioni trasportate possono essere cifrate ma di fatto non succede mai. E' stato definito un nuovo protocollo nel 2004, SRTP, acronimo di "Secure RTP". Questo dovrebbe sostituire il protocollo RTP e richiede la cifratura del traffico, non è però ancora molto applicato.

3.1 L'indirizzamento

I protocolli RTP/RTCP sono designati per essere indipendenti dal tipo di rete sottostante e dal protocollo di trasporto, vengono comunque utilizzati solitamente in reti IP utilizzando come protocollo di trasporto UDP. L'indirizzo IP che può essere unicast oppure multicast e la porta UDP costituiscono quindi l'indirizzo delle trasmissioni RTP/RTCP.

3.2 I Componenti dell'architettura RTP/RTCP

Sono definiti quattro componenti:

1. Mixer
2. Translator

3. Sender
4. Receiver

vengono di seguito descritti separatamente.

1. Mixer

La presenza di questo sistema è trasparente e permette di cambiare la codifica dei dati trasmessi per adeguarsi alle risorse di rete dei partecipanti alla comunicazione. E' così possibile diversificare la qualità delle informazioni trasmesse a seconda della banda disponibile per la ricezione del flusso multimedia cambiando il codec utilizzato oppure riducendo la quantità di dati trasportati.

2. Translator

La presenza di questo sistema è trasparente e permette, in presenza di ostacoli fra il Sender ed il Receiver, la trasmissione e la ricezione delle trasmissioni RTP creando e gestendo un tunnel tra i due punti. Gli ostacoli possono essere apparati di rete come Firewall e Gateway IP.

3. Sender

Indica il device sorgente che trasmette la sequenza di pacchetti RTP, in modalità unicast oppure multicast.

4. Receiver

Indica il device destinazione che riceve la sequenza di pacchetti RTP, in modalità unicast oppure multicast.

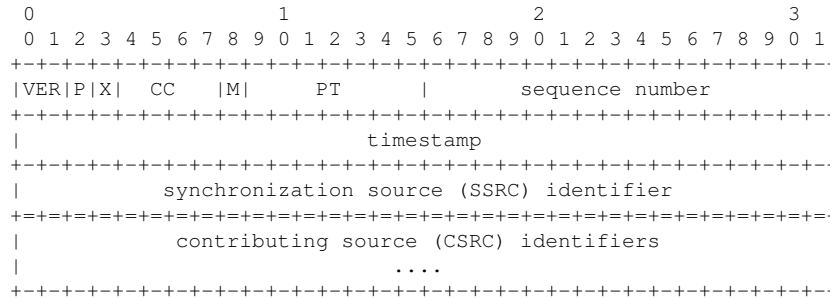
3.3 Il protocollo RTP

Il protocollo RTP (Real Time Protocol) si occupa della trasmissione dei dati Real Time. Le sue caratteristiche sono le seguenti:

- Il trasporto dei pacchetti RTP è affidato al protocollo di trasporto UDP.
- Non garantisce la consegna dei pacchetti.
- Non garantisce l'ordine di consegna dei pacchetti.
- Permette di identificare il tipo e la codifica dei dati trasportati.
- Permette di identificare la sorgente e la destinazione della trasmissione.
- Permette di riordinare i pacchetti (operazione che deve implementare l'applicazione).
- Permette di sincronizzare diversi flussi (voce, video, ...).

3.3.1 Il pacchetto RTP

Il pacchetto RTP consiste in un header fisso RTP, seguito da un eventuale lista di CSRC (Contributing SouRCes) inserita dai Mixer. Seguono i dati trasportati. Il formato dell'header fisso RTP è strutturato come segue:



I campi hanno il seguente significato:

- version (V): 2 bit, indica la versione del protocollo RTP.
- padding (P): 1 bit, indica che nella parte dati esistono dei byte di padding.
- extension (X): 1 bit, se vale 1 l'header è seguito da un'estensione con formato definito dall'applicazione.
- CSRC count (CC): 4 bit, contiene il numero dei CSRC (Contributing SouRces) che seguono l'header.
- payload type (PT): 7 bits, indica il formato dei dati trasportati.
- sequence number: 16 bit, identifica ogni pacchetto RTP spedito in modo sequenziale. Permette di riordinare i pacchetti e di rilevare i pacchetti persi.
- timestamp: 32 bit, indica l'istante di campionamento del primo byte trasportato nella parte dati.
- synchronization source (SSRC): 32 bit, identifica in modo univoco la sorgente dello stream RTP trasportato. Non dipende dall'indirizzo di rete. Tutti i pacchetti con lo stesso synchronization source devono avere lo stesso clock e lo stesso generatore di sequence number, in modo che il ricevente possa raggruppare correttamente i pacchetti. Se un partecipante genera più stream RTP, ciascuno dovrà essere indicato da un differente SSRC contributing source (CSRC).
- length: 32 bit, indica la lunghezza dell'estensione dell'header espressa in word da 4 byte.

3.3.2 Il modello trasmissivo di RTP

Il modello di trasmissione prevede un certo numero di Sender, ciascuno dei quali può trasmettere più flussi RTP verso un insieme di Receiver. Ciascun flusso RTP identifica una sessione RTP. La trasmissione può avvenire in modalità unicast oppure multicast. Per ciascun flusso di dati Real Time viene creata una sessione RTP per il trasporto. Quindi, se il Sender intende trasmettere più flussi di dati Real Time, dovrà creare più sessioni RTP. Questo vincolo è motivato dal fatto che ogni singolo Receiver può essere interessato soltanto alla ricezione di alcuni flussi RTP. Per ciascuna sessione RTP viene inoltre creata una sessione RTCP di controllo.

3.4 Il protocollo RTCP

Il protocollo RTCP (Real Time Transport Protocol) si affianca al protocollo RTP e si occupa di trasmettere informazioni aggiuntive riguardo la sessione RTP, come:

- Il numero di pacchetti persi, RTT (Round Time Trip) e Jitter. Queste informazioni possono essere utilizzate dall'applicazione per adeguarsi alle condizioni della rete.
- L'intenzione di terminare la sessione e quindi la ricezione/trasmissione del flusso RTP (BYE packet).
- Informazioni circa l'identità dei partecipanti alla sessione RTP.

3.4.1 Il pacchetto RTCP

Sono definiti cinque differenti tipi di pacchetti RTCP, ciascuno contenente un differente tipo di informazioni. Sono definiti i seguenti pacchetti:

- SR (sender report)
Fornisce le statistiche di spedizione effettuate dai sender RTP.
- RR (receiver report)
Fornisce le statistiche di spedizione effettuate dai receiver RTP.
- SDES (source descriptor)
Fornisce le informazioni circa l'identità dei Sender e Receiver.
- BYE
Indica che il Receiver intende terminare la ricezione del flusso RTP.

- APP

indica un differente tipo di informazioni definito in un qualche modo. Viene utilizzato per implementare le estensioni.

3.4.2 Il modello trasmissivo di RTCP

La sequenza di dati Real Time viene trasmessa come sequenza di pacchetti RTP ed i Receiver sono tenuti a trasmettere periodicamente i pacchetti RTCP. Questi, contenenti anche statistiche sulla qualità della ricezione, permettono al Sender di monitorare la qualità del servizio. Al crescere del numero di Receiver di una sessione RTP, cresce il numero di pacchetti RTCP che sono tenuti a trasmettere verso il Sender. Questo implica un aumento dell'overhead, ponendo il problema di come gestirne la scalabilità. RTCP dovrebbe dedicare per i propri pacchetti non più del 5% della banda utilizzata per i pacchetti RTP, aggiustando dinamicamente il tempo che intercorre tra la trasmissione dei pacchetti RTCP. In questo modo si riduce la frequenza di trasmissione dei pacchetti RTCP, risolvendo il problema.

Capitolo 4

VoIP eavesdropping

Con il passaggio dalla rete telefonica tradizionale alla tecnologia VoIP, le tecniche solitamente utilizzate per analizzare e controllare il traffico telefonico non sono più adeguate. Si sono quindi rese necessarie nuove applicazioni, dedicate all'analisi ed al controllo del traffico VoIP.

4.1 Il progetto di stage

Il progetto di stage ha richiesto l'implementazione di un'applicazione in grado di monitorare e quindi registrare il traffico VoIP proveniente oppure diretto verso un insieme ben definito di utenti, tenendo in considerazione i protocolli di segnalazione più diffusi.

4.1.1 Requisiti

I requisiti considerati sono i seguenti:

1. Funzionali
 - (a) L'applicazione deve supportare le reti di tipo IP ed i protocolli di trasporto TCP ed UDP.
 - (b) L'applicazione deve supportare i protocolli di segnalazione SIP, H.323 ed SCCP.
 - (c) L'applicazione deve permettere la registrazione su file del traffico VoIP proveniente oppure diretto verso un insieme ben definito di utenti.
2. Non funzionali
 - (a) L'applicazione deve essere scritta utilizzando il linguaggio di programmazione C.

- (b) L'applicazione deve essere facilmente estendibile ed integrabile in altre applicazioni.

Si noti che non viene richiesto alcun tipo di decodifica dei flussi multimedia. Questa operazione verrà svolta da una differente applicazione in un secondo momento, analizzando i pacchetti IP registrati.

4.1.2 Implementazione

Il traffico di ciascuna sessione VoIP si compone dei messaggi di segnalazione e dei flussi multimedia ad essa associati. Soltanto le sessioni che coinvolgono un certo insieme di utenti devono essere considerate. Gli username di livello applicativo sono presenti nei primi messaggi di Call Signaling scambiati, questi determinano se la sessione deve essere considerata oppure ignorata. Nei messaggi seguenti, vengono scambiati i parametri di trasmissione dei flussi multimedia. Segue una descrizione dei messaggi di segnalazione che contengono queste informazioni:

1. H.323

- (a) Identificazione degli username

Gli username di livello applicativo sono presenti nel messaggio di Setup che avvia la procedura di chiamata. Questo messaggio è definito nel file asn di specifica della raccomandazione H.225.0 come tipo Setup-UUIE, definito come segue:

```
Setup-UUIE ::= SEQUENCE {
    protocolIdentifier ProtocolIdentifier,
    h245Address TransportAddress OPTIONAL,
    sourceAddress SEQUENCE OF AliasAddress OPTIONAL,
    sourceInfo EndpointType,
    destinationAddress SEQUENCE OF AliasAddress
OPTIONAL,
    destCallSignalAddress TransportAddress OPTIONAL,
    destExtraCallInfo SEQUENCE OF AliasAddress
OPTIONAL,
    destExtraCRV SEQUENCE OF CallReferenceValue
OPTIONAL,
    activeMC BOOLEAN,
    conferenceID ConferenceIdentifier,
    conferenceGoal
CHOICE {create NULL,
    join NULL,
    invite NULL, ...,
    capability-negotiation NULL,
    callIndependentSupplementaryService NULL},
```

```

    callServices QseriesOptions OPTIONAL,
    callType CallType, ...,
    sourceCallSignalAddress TransportAddress OPTIONAL,
    remoteExtensionAddress AliasAddress OPTIONAL,
    callIdentifier CallIdentifier,
    h245SecurityCapability SEQUENCE OF H245Security
OPTIONAL,
    tokens SEQUENCE OF ClearToken OPTIONAL,
    cryptoTokens SEQUENCE OF CryptoH323Token OPTIONAL,
    fastStart SEQUENCE OF OCTET STRING OPTIONAL,
    mediaWaitForConnect BOOLEAN,
    canOverlapSend BOOLEAN,
    endpointIdentifier EndpointIdentifier OPTIONAL,
    multipleCalls BOOLEAN,
    maintainConnection BOOLEAN,
    connectionParameters
    SEQUENCE-- additional gateway parameters--
{connectionType
    ScnConnectionType,
    numberOfScnConnections
    INTEGER(0..65535),
    connectionAggregation
    ScnConnectionAggregation, ...} OPTIONAL,
    language SEQUENCE OF IA5String(SIZE (1..32))
OPTIONAL,
    presentationIndicator PresentationIndicator
OPTIONAL,
    screeningIndicator ScreeningIndicator OPTIONAL,
    serviceControl SEQUENCE OF ServiceControlSession
OPTIONAL,
    symmetricOperationRequired NULL OPTIONAL,
    capacity CallCapacity OPTIONAL,
    circuitInfo CircuitInfo OPTIONAL,
    desiredProtocols SEQUENCE OF SupportedProtocols
OPTIONAL,
    neededFeatures SEQUENCE OF FeatureDescriptor
OPTIONAL,
    desiredFeatures SEQUENCE OF FeatureDescriptor
OPTIONAL,
    supportedFeatures SEQUENCE OF FeatureDescriptor
OPTIONAL,
    parallelH245Control SEQUENCE OF OCTET STRING
OPTIONAL,
    additionalSourceAddresses SEQUENCE OF ExtendedAliasAddress
OPTIONAL,
    hopCount INTEGER(1..31) OPTIONAL

```

```
}

```

I campi che occorre considerare sono `sourceAddress` e `destinationAddress`. Questi identificano in modo univoco gli utenti della comunicazione e sono di tipo `AliasAddress`, definito come segue:

```
AliasAddress ::= CHOICE {
  dialedDigits IA5String(SIZE (1..128)) (FROM
("0123456789#*, ")),
  h323-ID BMPString(SIZE (1..256)), ...,
  url-ID IA5String(SIZE (1..512)), -- URL style
address
  transportID TransportAddress,
  email-ID IA5String(SIZE (1..512)),
  partyNumber PartyNumber,
  mobileUIM MobileUIM,
  isupNumber IsupNumber
}
```

Il tipo `AliasAddress` può rappresentare molti tipi di indirizzi. Fra questi consideriamo unicamente `h323-ID`, una stringa codificata in UTF-16. Questa contiene l'username H.323.

(b) Identificazione dei parametri di trasmissione dei flussi multimedia

I parametri di trasmissione dei flussi multimedia sono presenti nel messaggio `OpenLogicalChannel`. Questo messaggio è definito nel file `asn` di specifica della raccomandazione H.245 come tipo `OpenLogicalChannel`, definito come segue:

```
OpenLogicalChannel ::= SEQUENCE {
  forwardLogicalChannelNumber LogicalChannelNumber,
  forwardLogicalChannelParameters
  SEQUENCE {portNumber INTEGER(0..65535) OPTIONAL,
  dataType DataType,
  multiplexParameters
  CHOICE {h222LogicalChannelParameters
  H222LogicalChannelParameters,
  h223LogicalChannelParameters
  H223LogicalChannelParameters,
  v76LogicalChannelParameters
  V76LogicalChannelParameters, ...,
  h2250LogicalChannelParameters
  H2250LogicalChannelParameters,
  none NULL}, ...,
  forwardLogicalChannelDependency LogicalChannelNumber
OPTIONAL,
  replacementFor LogicalChannelNumber OPTIONAL
```

```

    },
    reverseLogicalChannelParameters
    SEQUENCE {dataType DataType,
    multiplexParameters
    CHOICE {
    h223LogicalChannelParameters
    H223LogicalChannelParameters,
    v76LogicalChannelParameters
    V76LogicalChannelParameters,
    . . . ,
    h2250LogicalChannelParameters
    H2250LogicalChannelParameters} OPTIONAL,
    . . . ,
    reverseLogicalChannelDependency LogicalChannelNumber
OPTIONAL,
    replacementFor LogicalChannelNumber OPTIONAL
    } OPTIONAL,
    . . . ,
    separateStack NetworkAccessParameters OPTIONAL,
    encryptionSync EncryptionSync OPTIONAL,
    genericInformation SEQUENCE OF GenericInformation
OPTIONAL
    }

```

Il campo che occorre considerare è `forwardLogicalChannelParameters`. Questo può rappresentare molti tipi di indirizzi, fra questi consideriamo unicamente il tipo `h2250LogicalChannelParameters`, definito come segue:

```

H2250LogicalChannelParameters ::= SEQUENCE
{
    nonStandard SEQUENCE OF NonStandardParameter
OPTIONAL,
    sessionID INTEGER(0..255),
    associatedSessionID INTEGER(1..255) OPTIONAL,
    mediaChannel TransportAddress OPTIONAL,
    mediaGuaranteedDelivery BOOLEAN OPTIONAL,
    mediaControlChannel TransportAddress OPTIONAL,
    mediaControlGuaranteedDelivery BOOLEAN OPTIONAL,
    silenceSuppression BOOLEAN OPTIONAL,
    destination TerminalLabel OPTIONAL,
    dynamicRTPPayloadType INTEGER(96..127) OPTIONAL,
    mediaPacketization
    CHOICE {h261aVideoPacketization NULL, . . . ,
    rtpPayloadType RTPPayloadType} OPTIONAL, . . . ,
    transportCapability TransportCapability OPTIONAL,
    redundancyEncoding RedundancyEncoding OPTIONAL,

```

```

source TerminalLabel OPTIONAL
}

```

Il campo che occorre considerare è `mediaChannel`, di tipo `TransportAddress`, definito come segue:

```

TransportAddress ::= CHOICE {
    unicastAddress UnicastAddress,
    multicastAddress MulticastAddress,
    ...
}

```

Il Tipo `TransportAddress` può contenere un indirizzo `unicast` oppure un indirizzo `multicast`. Viene considerato il campo `unicastAddress` di tipo `UnicastAddress`, definito come segue:

```

UnicastAddress ::= CHOICE {
    ipAddress
    SEQUENCE {network OCTET STRING(SIZE (4)),
    tsapIdentifier INTEGER(0..65535), ...},
    ipXAddress
    SEQUENCE {node OCTET STRING(SIZE (6)),
    netnum OCTET STRING(SIZE (4)),
    tsapIdentifier OCTET STRING(SIZE (2)),
    ...},
    ip6Address
    SEQUENCE {network OCTET STRING(SIZE (16)),
    tsapIdentifier INTEGER(0..65535), ...},
    netBios OCTET STRING(SIZE (16)),
    ipSourceRouteAddress
    SEQUENCE {routing CHOICE {strict NULL,
    loose NULL},
    network OCTET STRING(SIZE (4)),
    tsapIdentifier INTEGER(0..65535),
    route SEQUENCE OF OCTET STRING(SIZE (4)),
    ...}, ...,
    nsap OCTET STRING(SIZE (1..20)),
    nonStandardAddress NonStandardParameter
}

```

I campi `ipAddress` e `tsapIdentifier` contengono rispettivamente l'indirizzo IP e la porta UDP, la trasmissione del flusso multimedia avrà come destinazione questo indirizzo.

2. SIP

(a) Identificazione degli username

Gli username di livello applicativo sono presenti in tutti i messaggi SIP nei campi header `From` e `To`, precedentemente discussi.

(b) Identificazione dei parametri di trasmissione dei flussi multimedia

I parametri di trasmissione dei flussi multimedia sono presenti nei messaggi SDP, trasportati come corpo dei messaggi SIP.

3. SCCP

(a) Identificazione degli username

Gli username di livello applicativo sono presenti nel messaggio di tipo `StationCallInfoMessage`, definito come segue:

```
struct StationCallInfoMessage
{
    StationMessageID 0x008F;
    char callingPartyName[StationMaxNameSize];
    char callingParty[StationMaxDirnumSize];
    char calledPartyName[StationMaxNameSize];
    char calledParty[StationMaxDirnumSize];
    UINT32 lineInstance;
    UINT32 callReference;
    StationCallType callType;
    char originalCalledPartyName[StationMaxNameSize];
    char originalCalledParty[StationMaxDirnumSize];
};
```

I campi che occorre considerare sono i seguenti:

- `callingPartyName`, `calledPartyName`
Sono stringhe di caratteri, con una lunghezza massima di 40 byte, contenenti il nome testuale della parte chiamata e chiamante.
- `callingParty`, `calledParty`
Sono stringhe di caratteri, con una lunghezza massima di 24 byte, contenenti la rappresentazione testuale degli indirizzi E.164. della parte chiamata e chiamante.

(b) Identificazione dei parametri di trasmissione dei flussi multimedia

I parametri di trasmissione dei flussi multimedia sono presenti in due tipi di messaggio:

- Il messaggio `StationStartMediaTransmission`, definito come segue:

```
struct StationStartMediaTransmission
{
    StationMessageID 0x008A;
    UINT32 conferenceID;
```

```

UNIT32 passThruPartyID;
UNIT32 remoteIpAddress;
UINT32 remotePortNumber;
UINT32 millisecondPacketSize;
Media_PayloadType compressionType;
Media_QualifierOutgoing qualifierOut;
};

```

Questo messaggio viene trasmesso dal Call Manager e viene utilizzato per richiedere al Client di iniziare la trasmissione di un flusso audio, indicando l'indirizzo IP e la porta UDP di destinazione. I campi che occorre considerare sono `remoteIpAddress` e `remotePortNumber`, questi indicano rispettivamente l'indirizzo IP e la porta UDP di destinazione dei pacchetti RTP.

- Il messaggio `OpenReceiveChannelAck`, definito come segue:

```

struct StationOpenReceiveChannelAckMessage
{
    StationMessageID 0x22;
    OpenReceiveChanStatus
    UINT32 ipAddr;
    UINT32 portNumber;
    UINT32 passThruPartyID;
};

```

Questo messaggio viene trasmesso dal Client al Call Manager come risposta ad un messaggio di tipo `StationOpenReceiveChannelMessage` e viene utilizzato per indicare al Call Manager l'indirizzo IP e la porta UDP dove si intende ricevere il flusso multimedia. I campi che occorre considerare sono `ipAddr` e `PortNumber`, questi indicano rispettivamente l'indirizzo IP e la porta UDP di ricezione dei pacchetti RTP.

Considerando questi ed altri messaggi dei protocolli di segnalazione SIP, H.323 ed SCCP, l'applicazione mantiene internamente una lista di sessioni attive (gestendone inoltre lo stato) ed una lista di indirizzi IP e porte UDP di destinazione del traffico RTP. Queste informazioni consentono di monitorare il traffico VoIP, permettendone la registrazione su file quando richiesto.

4.2 Il riconoscimento delle sessioni RTP e la decodifica dei flussi audio

Vengono in questa sezione descritte due applicazioni sviluppate come progetto di tesi che consentono la ricostruzione dei flussi RTP ed in alcuni casi permettono anche la decodifica dei flussi audio trasportati.

4.2.1 L'applicazione rtpbreak

Questa applicazione permette di ricostruire le sessioni RTP riconosciute nel traffico di rete, occupandosi di riordinare i pacchetti e di estrarne il contenuto. E' indipendente dal protocollo di segnalazione, funziona quindi con SIP, H.323 ed SCCP. Se i dati trasportati sono flussi audio codificati con i codec *G.711 u-law*, *G.711 a-law* e *gsm* allora possono essere decodificati in un secondo momento utilizzando l'applicazione OpenSource *sox*.

4.2.1.1 Applicazioni simili

Esistono alcune applicazioni simili già esistenti, ciascuna con alcuni difetti:

- *voipong*

Permette di riconoscere e decodificare in alcuni casi le sessioni RTP. Ha vinto il premio "Best project of 2004" di IBM Turkiye, lo sviluppo è fermo dal 2005 ed è ormai considerato obsoleto. Il riconoscimento delle sessioni RTP avviene tramite alcune considerazioni sui pacchetti di controllo RTCP, non sempre trasmessi dalle recenti applicazioni VoIP.

- *vomit*

Permette di ricostruire e decodificare in alcuni casi le sessioni RTP che utilizzano come protocollo di segnalazione SCCP. Lo sviluppo è fermo dal 2004. Soltanto le conversazioni effettuate con telefoni IP Cisco vengono riconosciute.

rtpbreak si pone come sostituto di entrambi, permettendo di riconoscere e ricostruire qualsiasi sessione RTP.

4.2.1.2 Il riconoscimento delle sessioni RTP

Le sessioni RTP sono costituite da una sequenza ordinata di pacchetti RTP. Questi trasportano le informazioni Real Time utilizzando il protocollo di trasporto UDP, come precedentemente discusso. I pacchetti RTP devono rispettare alcune regole ben definite per essere considerati

validi, questa caratteristica permette di definire un pattern sul singolo pacchetto che viene utilizzato al fine di poter discriminare il traffico di rete catturato fra pacchetti che potrebbero essere di tipo RTP e pacchetti che sicuramente non lo sono. I controlli che vengono effettuati sono i seguenti:

1. Porta UDP di destinazione

La porta UDP di destinazione deve essere pari, come indicato in [13]. Inoltre deve essere maggiore di 1024, questo perché nei protocolli di trasporto UDP e TCP le porte inferiori oppure uguali a 1024 sono considerate privilegiate e non utilizzabili dalle applicazioni utente, come ad esempio i client VoIP.

2. Dimensione minima del pacchetto

La dimensione del payload del pacchetto UDP deve essere superiore a 12 byte, dimensione dell'header fisso sempre presente nei pacchetti RTP.

3. versione RTP

La versione del protocollo RTP attualmente in uso è la 2. Viene quindi verificato che il campo V dell'header fisso RTP abbia come valore 2.

4. Padding bit

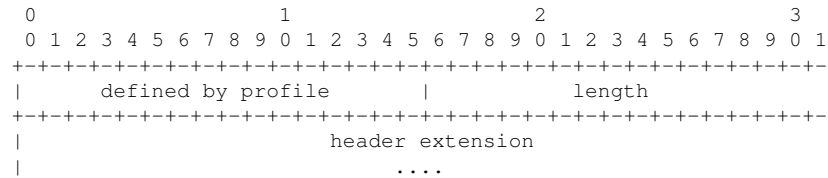
RTP consente di inserire alcuni byte di riempimento alla fine del pacchetto, che devono essere ignorati. Il numero di byte di riempimento è indicato esattamente nell'ultimo byte del pacchetto. Il campo P dell'header fisso RTP indica quando questa funzionalità è attiva. Se attiva, la dimensione del payload RTP viene aggiustata di conseguenza, controllando nuovamente che sia superiore a 0.

5. lista CSRC

RTP consente ai Mixer RTP di inserire una lista di sorgenti che hanno contribuito a creare i dati trasportati. Questa lista, se presente, segue immediatamente l'header fisso RTP e si compone di indirizzi di 32 bit ciascuno, il loro numero è indicato dal valore del campo CC dell'header fisso RTP. Se presenti, la dimensione del payload RTP viene aggiustata di conseguenza, controllando nuovamente che sia superiore a 0.

6. Extension bit

RTP consente di estendere l'header fisso RTP. Se presente, questa estensione segue l'header fisso RTP e la lista opzionale CSRC. Il suo formato è il seguente:



Il campo length indica la dimensione dell'estensione, header dell'estensione escluso. La sua presenza è indicata dal valore del campo X. Se attivo, la dimensione del payload RTP viene aggiustata di conseguenza, controllando nuovamente che sia superiore a 0.

I pacchetti UDP che superano questi controlli sono considerati dei possibili pacchetti RTP. Si noti che il checksum dei pacchetti IP ed UDP non viene controllato perché molto spesso vengono erroneamente calcolati dai client VoIP, come verificato durante i test di laboratorio. I pacchetti UDP che superano questi controlli vengono confrontati con le sessioni RTP precedentemente riconosciute. Il confronto avviene considerando le seguenti informazioni:

1. SSRC

Il valore del campo SSRC dell'header fisso RTP indica l'identificativo univoco del Sender della sessione. Il suo valore è costante in tutti i pacchetti RTP appartenenti alla sessione.
2. indirizzi IP e porte UDP

Gli indirizzi IP e le porte UDP di Sender e Receiver sono costanti in tutti i pacchetti RTP appartenenti alla sessione.
3. Sequence number

Il campo seq nell'header fisso RTP indica il sequence number del pacchetto, un valore che non necessariamente viene inizializzato a 1 ma che è strettamente crescente nei pacchetti appartenenti alla sessione. Viene considerata una finestra di valori accettabili per ciascuna sessione, che varia dinamicamente nel tempo. Questo permette di considerare l'eventualità che dei pacchetti RTP vengano perduti.
4. Timestamp

Il campo ts dell'header fisso RTP indica il timestamp del campionamento riferito al primo byte del payload RTP, un valore strettamente crescente nei pacchetti appartenenti alla sessione. Viene anche in questo caso considerata una finestra di valori accettabili

per ciascuna sessione, che varia dinamicamente nel tempo. Questo permette di considerare l'eventualità che dei pacchetti RTP vengano perduti.

Se viene identificata una sessione corrispondente, il pacchetto UDP viene inserito nel buffer dei pacchetti appartenenti a quella sessione. Se non viene identificata alcuna sessione corrispondente, viene creata una nuova sessione. Quando ad una sessione vengono riconosciuti un numero minimo predefinito di pacchetti UDP, questa viene considerata valida e qualsiasi pacchetto UDP nel buffer associato a tale sessione viene considerato definitivamente RTP. Questo deve verificarsi entro un certo tempo, oltre il quale la sessione viene considerata un falso positivo e quindi distrutta.

4.2.1.3 I Requisiti

L'applicazione richiede il sistema operativo Linux e le librerie di sistema libpcap e libnet1. La decodifica dei flussi audio codificati con i codec *G.711 u-law*, *G.711 a-law* e *gsm* richiede l'applicazione *sox*.

4.2.1.4 I Parametri

I parametri accettati dalla linea di comando sono i seguenti:

- -i str
Non opzionale, indica l'interfaccia di rete che si intende utilizzare come sorgente dei pacchetti IP. Ad esempio, "-i eth0". Questa opzione è alternativa a -r.
- -r str
Non opzionale, indica il file in formato pcap che si intende utilizzare come sorgente dei pacchetti IP. Ad esempio, "-r trace0.pcap". Questa opzione è alternativa a -i.
- -p str
Opzionale, indica un ulteriore prefisso che deve essere utilizzato nel nel pathname dei file di output generati.
- -v
Opzionale, permette di abilitare la visualizzazione dei messaggi di debug.
- -l str
Opzionale, permette di specificare il file dove scrivere i messaggi, di default mostrati sullo standard output.
- -h
Opzionale, visualizza una descrizione dei parametri accettati.

4.2.1.5 Esempio di utilizzo

L'applicazione viene eseguita da linea di comando specificando l'interfaccia di rete eth0:

```
./rtpbreak -i eth0
```

Analizza il traffico di rete e quando riconosce una sessione RTP, genera alcuni file:

1. *rtp_session.0.pcap*
Contiene i pacchetti RTP riordinati ed appartenenti alla sessione RTP
2. *rtp_session.0.raw*
Contiene i dati raw trasportati nella sessione RTP
3. *rtp_session.0.txt*
Contiene alcune informazioni testuali come il tipo di dati trasportato

Se il tipo di dati trasportato è pari a 0, 3 oppure 8 allora si tratta di un flusso audio codificato rispettivamente con il codec *G.711 u-law*, *G.711 a-law* e *gsm*. I dati raw possono essere quindi decodificati utilizzando l'applicazione *sox*:

1. Con il codec *g.711 u-law*:

```
sox -t ul rtp_session.0.raw -t wav 0.wav
```

2. Con il codec *gsm*:

```
sox -t gsm rtp_session.0.raw -t wav 0.wav
```

3. Con il codec *G.711 a-law*:

```
sox -t al rtp_session.0.raw -t wav 0.wav
```

E' possibile inoltre unire due flussi audio di una conversazione Full-Duplex utilizzando l'applicazione *soxmix*:

```
soxmix -t wav 0.wav -t wav 1.wav -t wav call01.wav
```

4.2.2 L'applicazione sipcodec

Questa applicazione permette di forzare l'utilizzo del codec *G.711 u-law* nelle comunicazioni VoIP che utilizzano SIP come protocollo di segnalazione, modificando i messaggi SDP trasportati dai messaggi SIP in modo opportuno. Per porsi nel mezzo della comunicazione, deve essere effettuato un attacco MITM via ARP Poisoning utilizzando l'applicazione *arpspoof*, presente nella suite *dsniff* sviluppata da Dug Song.

4.2.2.1 Modifica dei messaggi SDP

I messaggi SDP, come precedentemente discusso, trasportano le informazioni relative ai parametri di ricezione delle sessioni RTP. Questi parametri consistono in indirizzo IP, porta UDP ed una lista di codec supportati dal Receiver. Il Sender trasmetterà verso l'indirizzo IP e la porta UDP indicati i pacchetti RTP, codificando il flusso audio/video con un codec supportato da entrambe le parti. Le informazioni relative ai codec supportati sono codificate nei campi "m" ed "a". Questa applicazione permette di modificare questi campi, al fine di annunciare il supporto unicamente per il codec *G.711 u-law*. Segue un esempio di tale modifica:

- Messaggio SDP originale

```
v=0
o=Michele 123456 654321 IN IP4 192.168.2.8
s=A conversation
c=IN IP4 192.168.2.8
t=0 0
m=audio 7078 RTP/AVP 0 111 110 3 8 101
a=rtpmap:0 PCMU/8000/1
a=rtpmap:111 speex/16000/1
a=rtpmap:110 speex/8000/1
a=rtpmap:3 GSM/8000/1
a=rtpmap:8 PCMA/8000/1
m=video 9078 RTP/AVP 97 98 99
a=rtpmap:97 theora/90000
a=rtpmap:98 H263-1998/90000
a=rtpmap:99 MP4V-ES/90000
```

- Messaggio SDP modificato

```
v=0
o=Michele 123456 654321 IN IP4 192.168.2.8
s=A conversation
c=IN IP4 192.168.2.8
t=0 0
m=audio 7078 RTP/AVP 0
a=X:AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
m=video 9078 RTP/AVP 97 98 99
a=rtpmap:97 theora/90000
a=rtpmap:98 H263-1998/90000
a=rtpmap:99 MP4V-ES/90000
```

Si noti che la sequenza di A viene utilizzata per sovrascrivere esattamente soltanto le informazioni circa i codec supportati che si intendono nascondere. Questo perché potrebbero seguire altre sezioni Media description nel messaggio SDP, come precedentemente già discusso.

4.2.2.2 L'attacco ARP Poisoning

Questa tecnica permette di falsare la risoluzione degli indirizzi IP in indirizzi Ethernet, al fine di impersonare altri nodi della rete oltre il proprio. Il protocollo ARP[10], acronimo di "Address Resolution Protocol", definisce come deve avvenire la risoluzione degli indirizzi di livello network in indirizzi Ethernet di livello datalink, mediante lo scambio di messaggi ARP. I messaggi ARP utilizzati per richiedere la risoluzione di un indirizzo di livello network (come gli indirizzi IP) vengono trasmessi in modalità Broadcast sulla rete Ethernet mentre i messaggi ARP di risposta vengono trasmessi nella direzione opposta e contengono le informazioni richieste. Il protocollo ARP non definisce alcun tipo di autenticazione e sessione, i messaggi vengono quindi considerati singolarmente. Per ridurre al minimo il numero di richieste ARP trasmesse, i sistemi operativi mantengono una cache dei messaggi di risposta ARP precedentemente ricevuti. Questa ottimizzazione consente ad un attaker di costruire e trasmettere messaggi di risposta ARP, al fine di manipolare la cache ARP del nodo "vittima". L'applicazione *arpspoof* implementa questo tipo di attacco. Ci si deve preoccupare inoltre dell'IP Forwarding. Se non si intendono modificare i pacchetti l'IP Forwarding verso il legittimo destinatario può essere gestito direttamente dal sistema operativo, abilitando l'opzione "IP Forwarding". In Linux questo consiste nell'eseguire il seguente comando:

```
echo 1 >proc/sys/net/ipv4/ip_forward
```

Se invece si intendono modificare i pacchetti IP, come nell'applicazione *sipcodec*, ci si dovrà preoccupare direttamente di questo. Segue un esempio di utilizzo dell'applicazione *arpspoof*, da linea di comando:

```
arpspoof -t 192.168.1.4 192.168.1.2
```

Quando in esecuzione, l'applicazione trasmette all'indirizzo Ethernet corrispondente all'indirizzo IP 192.168.1.4 una sequenza di risposte ARP, al fine di impersonare il nodo che ha come indirizzo IP 192.168.1.2.

4.2.2.3 L'attacco MITM via ARP Poisoning

Questa tecnica, conosciuta come attacco MITM¹ via ARP Poisoning, permette di introdursi nelle comunicazioni tra due nodi della rete, consentendone la ricostruzione e la modifica. L'attacco consiste nel modificare la cache ARP di entrambi i nodi coinvolti nella comunicazione

¹MITM, acronimo di "Man In The Middle".

mediante l'attacco ARP Poisoning. La situazione prima dell'attacco è mostrata in figura 4.1.

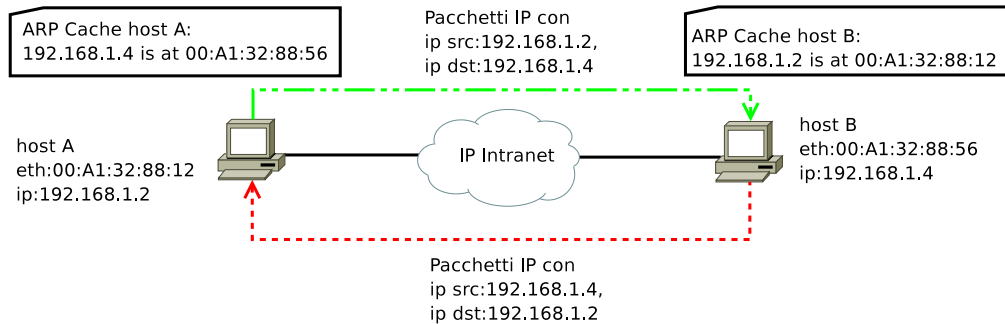


Figura 4.1: MITM via ARP Poisoning: situazione prima dell'attacco

Segue un esempio di utilizzo dell'applicazione *arp spoof*, da linea di comando:

```
arp spoof -t 192.168.1.4 192.168.1.2
arp spoof -t 192.168.1.2 192.168.1.4
```

Quando in esecuzione, questi due comandi permettono di impersonare i nodi con indirizzo IP 192.168.1.2 e 192.168.1.4 nelle loro rispettive cache ARP, come mostrato in figura 4.2.

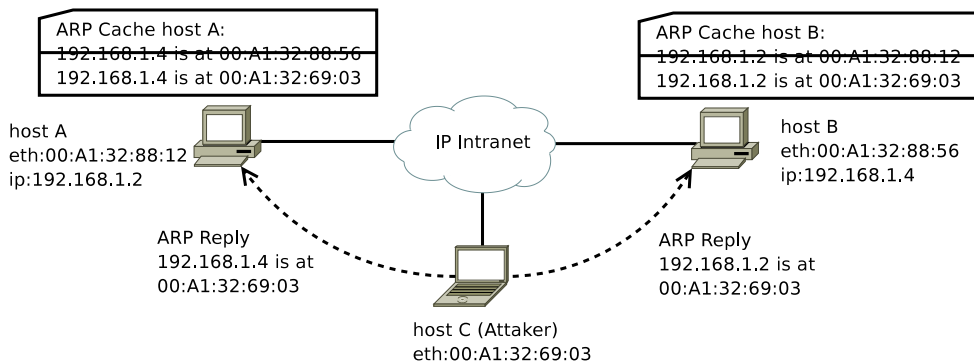


Figura 4.2: MITM via ARP Poisoning: esecuzione dell'attacco

Le comunicazioni fra questi due nodi saranno quindi trasmesse verso il nodo dell'attacker che può monitorarle e modificarle, come mostrato in figura 4.3.

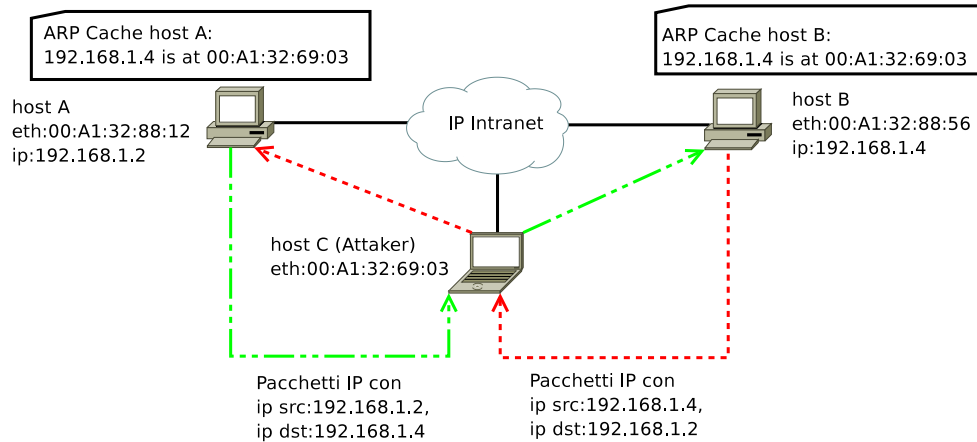


Figura 4.3: MITM via ARP Poisoning: situazione dopo l'attacco

4.2.2.4 IP Forwarding

Siano A e B i nodi che si intendono controllare. L'IP Forwarding viene implementato nell'applicazione *sipcodec* utilizzando il seguente algoritmo:

1. Per ciascun frame Ethernet catturato:
 - (a) Se l'indirizzo Ethernet sorgente è pari all'indirizzo Ethernet dell'interfaccia di rete, torna a 1
 - (b) Se l'indirizzo IP di destinazione non è pari a nessuno dei due indirizzi IP dei nodi A e B, torna a 1
 - (c) Se l'indirizzo IP di destinazione è pari all'indirizzo IP del nodo A, sostituisci l'indirizzo Ethernet di destinazione con l'indirizzo Ethernet del nodo A.
 - (d) Se l'indirizzo IP di destinazione è pari all'indirizzo IP del nodo B, sostituisci l'indirizzo Ethernet di destinazione con l'indirizzo Ethernet del nodo B.
 - (e) Ritrasmetti il frame Ethernet

Gli indirizzi Ethernet dei nodi A e B vengono risolti mediante la trasmissione e la ricezione di pacchetti ARP, questo è gestito internamente dall'applicazione *sipcodec*.

4.2.2.5 I Requisiti

L'applicazione richiede il sistema operativo Linux, le librerie di sistema *libpcap*, *libnet1* e l'applicazione *arp spoof*.

4.2.2.6 I Parametri

I parametri accettati dalla linea di comando sono i seguenti:

- -i str
Non opzionale, indica l'interfaccia di rete Ethernet che si intende utilizzare che si intende utilizzare come sorgente dei pacchetti IP. Ad esempio, "-i eth0".
- -a str
Non opzionale, indica l'indirizzo IP della "vittima" A. Ad esempio, "-a 192.168.1.2".
- -b str
Non opzionale, indica l'indirizzo IP della "vittima" B. Ad esempio, "-b 192.168.1.4".
- -v
Opzionale, permette di abilitare la visualizzazione dei messaggi di debug.
- -l str
Opzionale, permette di specificare il file dove visualizzare i messaggi, di default mostrati sullo standard output.
- -h
Opzionale, visualizza una descrizione dei parametri accettati.

4.2.2.7 Esempio di utilizzo

Siano 192.168.1.2 e 192.168.1.4 gli indirizzi IP dei nodi A e B coinvolti in comunicazioni VoIP che utilizzano il protocollo di segnalazione SIP. Il codec utilizzato non è direttamente decodificabile mediante l'applicazione *sox*, si vuole quindi forzare l'utilizzo del codec *G.711 u-law*. Occorre attivare l'attacco MITM via ARP Poisoning, vengono eseguiti i seguenti comandi:

```
arpspoof -t 192.168.1.4 192.168.1.2
arpspoof -t 192.168.1.2 192.168.1.4
```

Viene avviata l'applicazione sipcodec:

```
./sipcodec -i eth0 -a 192.168.1.2 -b 192.168.1.4
```

L'applicazione analizza il traffico di rete e si occupa di modificare opportunamente i messaggi SIP scambiati tra A e B, forzando l'utilizzo del codec *G.711 u-law*. La trasmissione RTP può essere ricostruita utilizzando l'applicazione *rtplib* ed il flusso audio può essere decodificato utilizzando l'applicazione *sox*.

Capitolo 5

Conclusioni

Il VoIP è destinato a sostituire completamente la rete telefonica tradizionale, probabilmente entro i prossimi dieci anni. Grazie alle nuove possibilità, gli utenti tendono ad utilizzare solo quando necessario le reti VoIP degli operatori telefonici, in favore di reti VoIP costituite da nodi interconnessi attraverso Internet. Questa soluzione permette di annullare completamente oppure ridurre considerevolmente il costo delle comunicazioni. Gli operatori telefonici vedono in questa fuga di utenti grandi perdite economiche e cercano di ostacolare questo modello di comunicazione, arrivando anche a filtrare in alcuni casi le porte TCP/UDP utilizzate dai protocolli VoIP nei collegamenti ad Internet. Ma come è già stato dimostrato più volte dai fatti¹, l'informazione in Internet è difficilmente controllabile su larga scala e non c'è motivo di credere che non lo sia anche per il VoIP. Questa tendenza, unita al fatto che nelle reti IP la crittografia è facilmente implementabile in qualsiasi tipo di servizio, sta già creando qualche problema alle Forze dell'ordine che si ritrovano nell'impossibilità di decodificare il traffico VoIP della rete Skype². Probabilmente in futuro interverrà il Legislatore a porre dei paletti nelle reti VoIP, risolvendo così questa situazione anomala. Se non viene utilizzata la crittografia le comunicazioni sono facilmente intercettabili, come ha dimostrato l'applicazione sviluppata come progetto di stage (che è al momento operativa ed in esecuzione su alcuni server in produzione). Chi intende comunicare in modo sicuro potrà comunque utilizzare altri servizi multiprotocollo OpenSource offerti attraverso la rete Internet, come ad esempio Tor³. Le comunicazioni VoIP possono essere instradate attraverso la rete Tor, garantendo così l'anonimato di entrambe le parti coinvolte. Un layer crittografico come ad esempio TLS può garantire anche la riservatezza della trasmissione,

¹Si pensi alle reti P2P, che hanno messo in crisi il concetto di Copyright.

²la rete di Skype è una rete P2P cifrata e gestita da una società che non intende collaborare con le Forze dell'ordine se non c'è un obbligo legislativo, come accade ora.

³Tor, anonimato in rete. Indirizzo URL <http://tor.eff.org>

che Tor non permette completamente. Le applicazioni *rtpbreak* e *sipcodec*⁴ hanno dimostrato inoltre che le intercettazioni VoIP nelle reti locali sono facilmente attuabili se non si utilizza alcun tipo di protezione. In particolare, si noti che:

- L'applicazione *rtpbreak* ha riconosciuto e ricostruito correttamente una conversazione fra due telefoni SIP wireless modello Zyxel P200W, questo modello non trasmette i pacchetti RTCP ed è quindi non supportato dall'applicazione *voipong*.
- L'applicazione *sipcodec* è stata utilizzata con successo per forzare il codec *G.711 u-law* nella rete *Alice VoIP* di Telecom Italia, che di default utilizza il codec *G.729a* (non royalty free).

Il protocollo SRTP dovrebbe risolvere questo problema di riservatezza nelle reti locali ma è ancora praticamente inapplicato. In definitiva, il VoIP rende le intercettazioni telefoniche più semplici se non ci si cura del problema e più complesse se si prendono alcuni accorgimenti.

⁴Le applicazioni *rtpbreak* e *sipcodec*, sotto licenza GPL, sono disponibili all'indirizzo URL <http://xenion.antifork.org>.

Elenco delle figure

2.1 SIP Direct Signaling	25
2.2 SIP Proxy Routed Signaling	26
2.3 network SIP di riferimento	27
2.4 Scenario SIP 1	27
2.5 Scenario SIP 2	28
2.6 network H.323 di riferimento	38
2.7 H.323 Direct Signaling	39
2.8 H.323 Routed H.225.0 Signaling	40
2.9 H.323 Routed H.225.0/H.245 Signaling	41
2.10 SCCP Call Manager Gateway	43
2.11 network SCCP di riferimento	45
2.12 SCCP Routed Signaling	46
4.1 MITM via ARP Poisoning: situazione prima dell'attacco	69
4.2 MITM via ARP Poisoning: esecuzione dell'attacco	69
4.3 MITM via ARP Poisoning: situazione dopo l'attacco	70

Bibliografia

- [1] Augmented BNF for syntax specifications: ABNF. RFC 2234, Internet Engineering Task Force, November 1997.
- [2] T. Dierks and C. Allen. The TLS protocol version 1.0. RFC 2246, Internet Engineering Task Force, January 1999.
- [3] M. Handley and V. Jacobson. SDP: session description protocol. RFC 2327, Internet Engineering Task Force, April 1998.
- [4] International Telecommunication Union. Digital subscriber signalling system no. 1 (DSS 1) - ISDN user-network interface layer 3 specification for basic call control. Recommendation Q.931, International Telecommunication Union, Geneva, Switzerland, March 1993.
- [5] International Telecommunication Union. Media stream packetization and synchronization on non-guaranteed quality of service LANs. Recommendation H.225.0, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, November 1996.
- [6] International Telecommunication Union. ASN.1 encoding rules - specification of packed encoding rules (PER). Recommendation X.691, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, December 1997.
- [7] International Telecommunication Union. The international public telecommunication numbering plan. Recommendation E.164, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, May 1997.
- [8] International Telecommunication Union. Control protocol for multimedia communication. Recommendation H.245, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, February 1998.

- [9] International Telecommunication Union. Packet based multimedia communication systems. Recommendation H.323, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, November 2000.
- [10] D. Plummer. Ethernet address resolution protocol: Or converting network protocol addresses to 48.bit ethernet address for transmission on ethernet hardware. RFC 826, Internet Engineering Task Force, November 1982.
- [11] Marshall Rose and D. Cass. ISO transport services on top of the TCP: version 3. RFC 1006, Internet Engineering Task Force, May 1987.
- [12] J. Rosenberg, Henning Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: session initiation protocol. RFC 3261, Internet Engineering Task Force, June 2002.
- [13] Henning Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: a transport protocol for Real-Time applications. RFC 1889, Internet Engineering Task Force, January 1996.
- [14] F. Yergeau. UTF-8, a transformation format of ISO 10646. RFC 3629, Internet Engineering Task Force, November 2003.