Istruzioni per l'utilizzo coordinato della Telefonia su IP nel GARR

Antonio Pinizzotto
IIT-CNR, Pisa
antonio.pinizzotto@iit.cnr.it

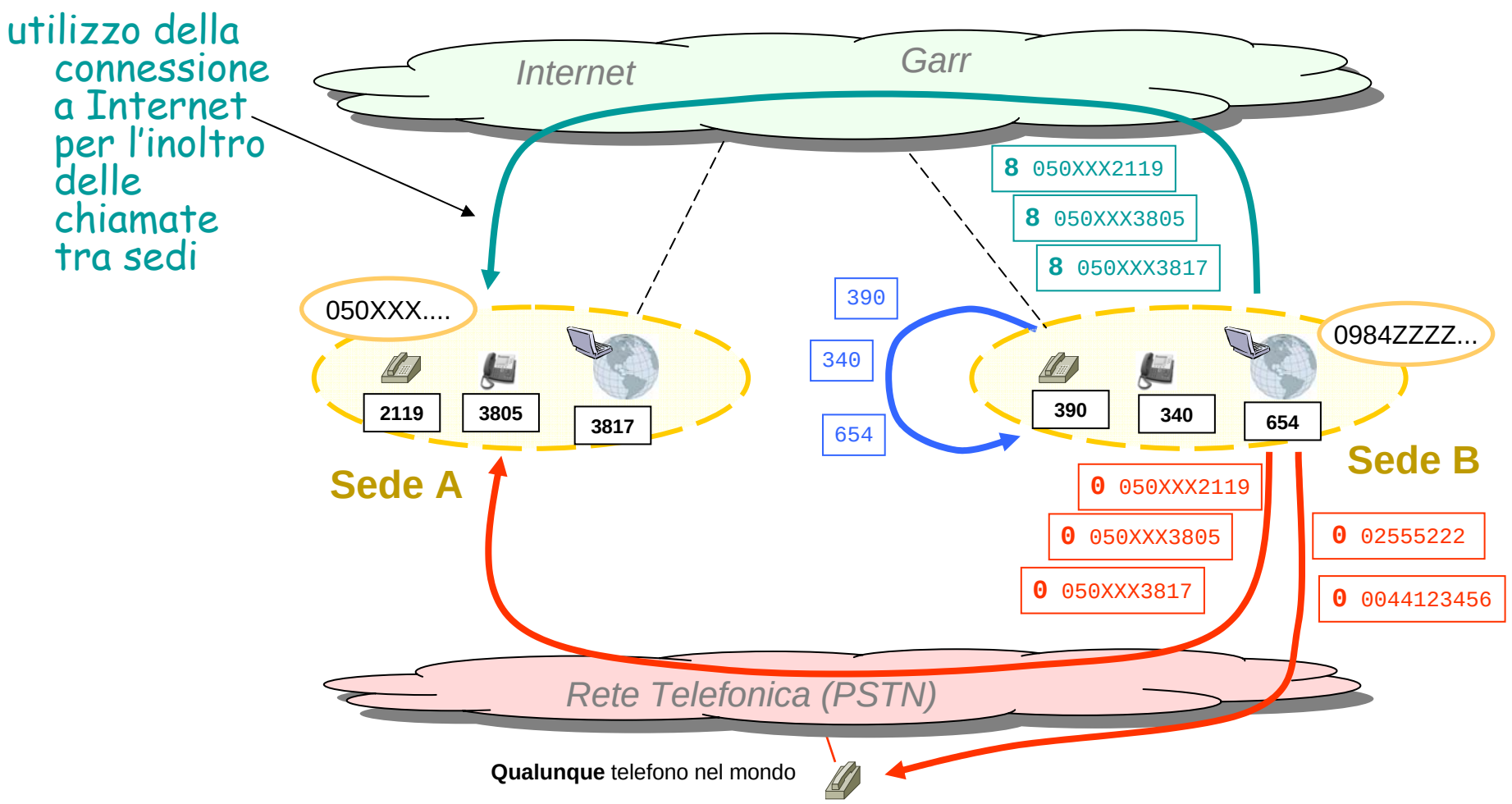
GARR_WS6 – Roma, 16 novembre 2005

Obiettivo del Tutorial

- **Presentazione del piano di integrazione della telefonia tradizionale con quella su IP già in atto nel GARR**
- Prima parte "teorica" necessaria alla piena comprensione della seconda parte più descrittiva dell'implementazione
- **Come partecipare al progetto**: con un investimento minimo (o inizialmente nullo) qualunque sede può partecipare.

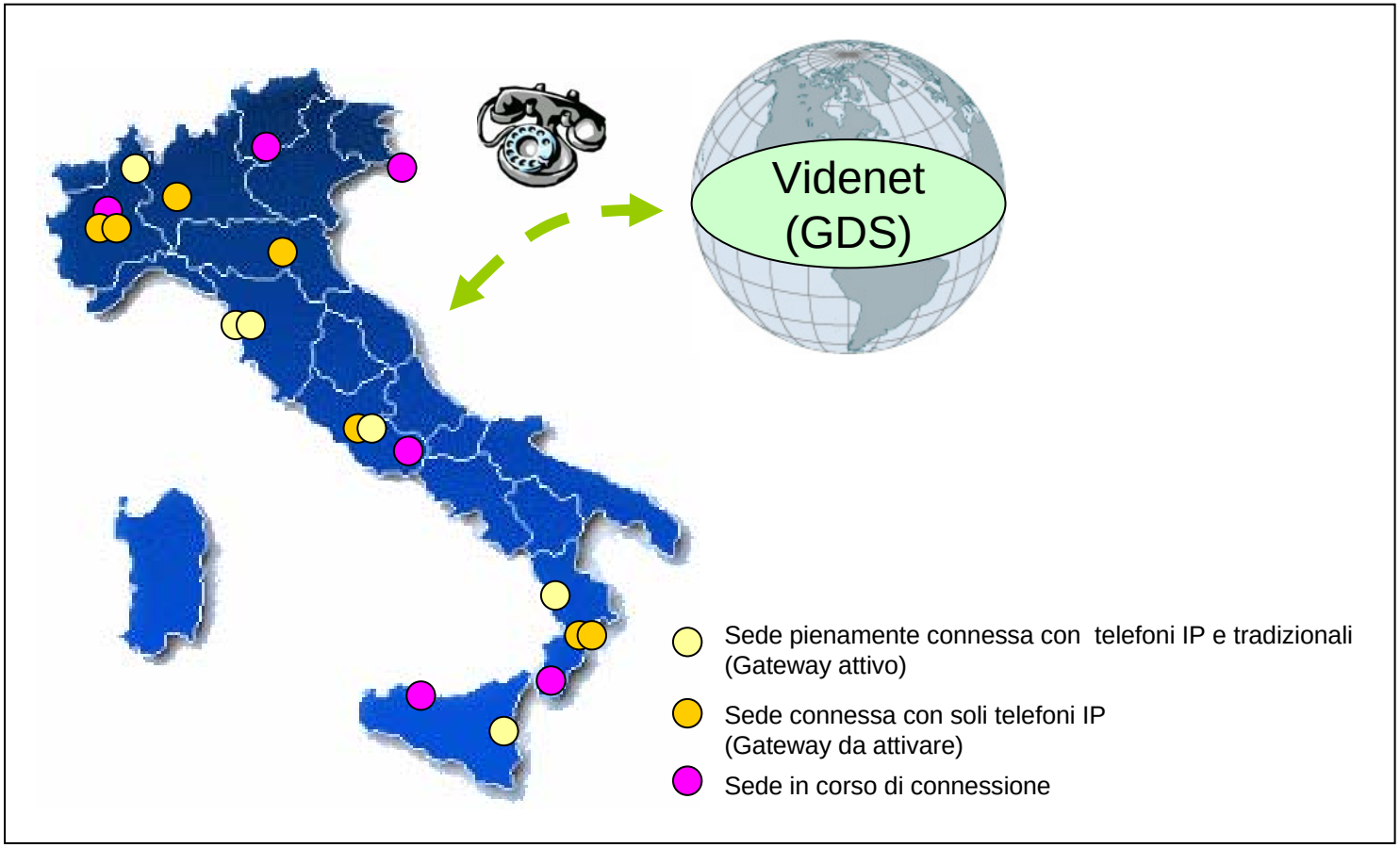
- In sintesi le finalità del progetto:
  - Introduzione graduale della tecnologia VoIP tra le sedi afferenti al GARR integrandola con le infrastrutture telefoniche già esistenti
  - Introduzione di telefoni IP hardware e software integrati con il piano di numerazione già esistente
  - Facilità d'uso per l'utente finale
  - Doppia raggiungibilità: ogni telefono è raggiungibile con lo stesso numero sia via IP che via rete telefonica tradizionale
  - Integrazione con il piano di numerazione nazionale esistente
  - Utilizzo di soluzioni OpenSource e multi-vendor
  - Mobilità
  - Scalabilità
  - Integrazione con il Global Dialling Scheme (chiamate su IP in tutto il mondo)

Via alternative (IP) per le telefonate fra sedi
(... e molto di più)



Stato attuale delle sedi italiane

aggiornato al 2 novembre 2005



Sommario

- Introduzione
- Protocolli standard: H.323 e SIP
- ENUM
- GDS e sua evoluzione con ENUM
- Applicazioni: SIP.EDU
- Problemi e soluzioni con NAT e Firewall
- Implementazioni OpenSource: gnugk, ser, asterisk
- Alcuni prodotti commerciali: Cisco, Innovaphone
- Client testati
- Utilizzo del GDS per l'inoltro delle chiamate VoIP nel GARR
  - Obiettivi
  - Implementazione
  - Requisiti minimi per partecipare
    - Gatekeeper
    - La scelta del Gateway
  - Stato attuale delle sedi connesse
  - Integrazione con ENUM e SIP
  - Configurazione di filtri e firewall
  - Autenticazione
  - Integrazione del CallManager
  - Come aderire

Introduzione

- I termini VoIP e IP Telephony, anche se simili, hanno significati leggermente diversi
  - Entrambi si riferiscono al trasporto di servizi di fonia su reti IP (Internet Protocol) in alternativa al trasporto su PSTN (Public Switched Telephone Network)

- Con il termine VoIP (Voice over IP) si intende il trasporto su reti IP di servizi voce real-time interattivi
  - localizzazione del chiamato
  - trasporto della voce: conversione A/D (analogico/digitale) e D/A della voce, codifica, trasporto su pacchetti IP

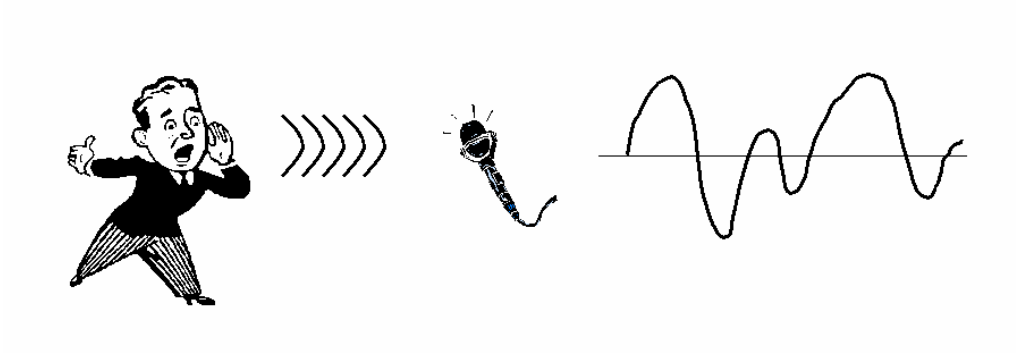
Introduzione (2)

- Il termine IP Telephony si riferisce principalmente all'insieme di tutti i servizi correlati al trasporto della voce su IP:
  - interoperabilità
  - scalabilità
  - affidabilità
  - soprattutto **integrazione** con la **telefonia tradizionale**

- In breve, mentre la Telefonia su IP si prefigge lo scopo di riprodurre, integrare ed estendere su IP i servizi della telefonia tradizionale, il VoIP si può considerare svincolato dalla telefonia tradizionale

VoIP minimale

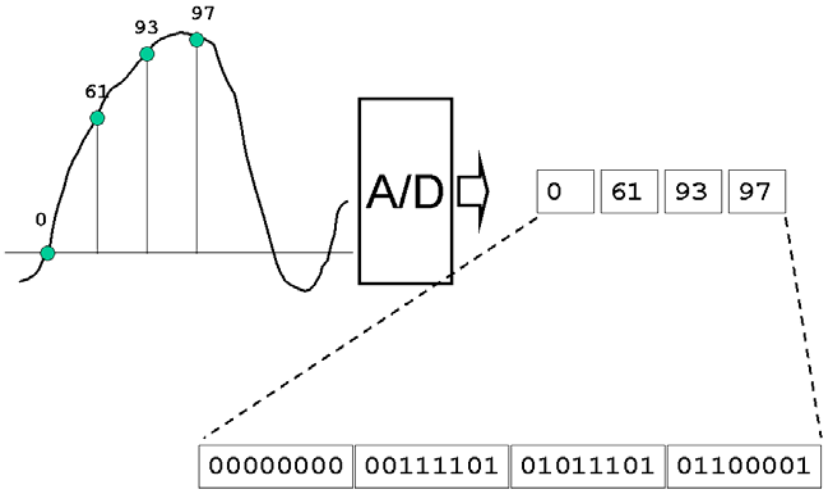
- *Breve parentesi*: elementi minimi per poter trasmettere la voce su IP (VoIP)
- La voce per poter essere trasmessa deve essere digitalizzata
  - segnale analogico
    - pressione dell'aria
    - attraverso il mocrofono diventa un segnale elettrico
    - ma è sempre un segnale analogico, ovvero un segnale che varia con continuità



VoIP minimale (2)

- Per ottenere la digitalizzazione del segnale audio analogico occorre eseguire una operazione di:
  - campionamento
  - e quantizzazione

  Eseguita da un chip, ad esempio quello della scheda audio di un PC
- Campionamento
  - frequenza di campionamento (almeno doppia della banda del segnale, teorema di Nyquist). Es.:
    - 44.1 kHz per l'audio di un CD
    - 8 kHz per l'audio del telefono
- Quantizzazione
  - suddivisione dei livelli di ampiezza in un numero finito di valori; es.:
    - CD: 19 bit: 2^{19} = 524288 livelli possibili
    - telefono: 8 bit: 2^{8} = 256 livelli possibili

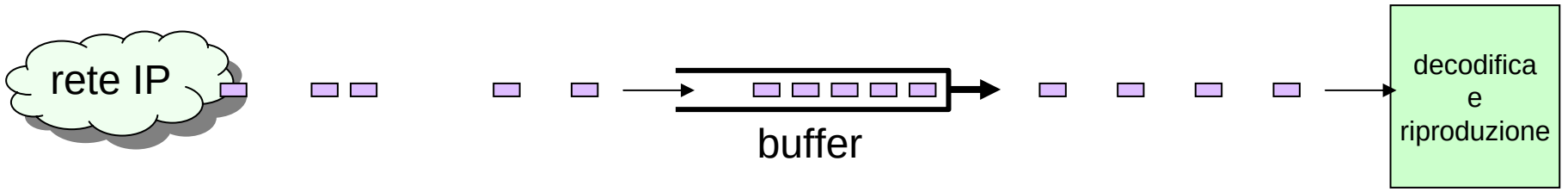


VoIP minimale (3)

- La sequenza di bit così ottenuta corrisponde ad una sequenza non codificata. Es.:
  - voce telefono: 8 kHz x 8 bit = 64 kbps
  - CD: 2 canali x 44.1 kHz x 16 bit = ~ 1.4 Mbps
- La sequenza di bit non viene trasmessa così com'è ma viene prima codificata. I motivi sono diversi:
  - compressione (motivo principale): determina l'occupazione di banda
  - adattamento alle condizioni di QoS di una rete IP (tiene conto del fatto che il flusso viaggia su una rete a commutazione di pacchetto e non di circuito)
- La sequenza codificata viene suddivisa in pacchetti che sono quindi trasmessi sulla rete IP

VoIP minimale (4)

- In ricezione il processo è quello inverso:
  - decodifica
  - conversione digitale/analogica (A/D)
  - riproduzione in altoparlante
- Per migliorare la qualità audio in ricezione possono essere presi alcuni accorgimenti quali la bufferizzazione
  - l'audio contenuto nei pacchetti non viene riprodotto subito ma memorizzato in una coda locale e riprodotto con un certo ritardo
  - questo migliora gli effetti sgradevoli dovuto al fatto che i pacchetti possono arrivare con tempi diversi (jitter) ma introduce un ritardo costante che per non essere percepito non dovrebbe superare i 100-150 ms.



VoIP minimale (4b): Codec Audio

- Alcuni codec audio utilizzati nella telefonia

Encoding/Compression	Bit Rate
G.711 PCM A-Law / µ-Law	64 kbps
G.726 ADPCM	16, 24, 32, 40 kbps
G.727 E-ADPCM	16, 24, 32, 40 kbps
G.729 CS-ACELP	8 kbps
G.728 LD-CELP	16 kbps
G.723.1 CELP	5.3 / 6.3 kbps (variabile)

VoIP minimale (5)

- Quali sono gli elementi non presi in considerazione in questo esempio?
  - localizzazione dell'utente destinatario (occorre conoscere a priori l'indirizzo IP)
  - negoziazione dei canali di trasporto (codec audio, protocolli di trasporto, ..)
    - esistono vari codec corispondenti a diverse occupazioni di banda (bit rate) ed i due utenti potrebbero avere preferenze diverse
  - **integrazione con la rete telefonica tradizionale**?
- L'ultimo punto, in particolare, è un obiettivo della Telefonia su IP

Obiettivi della Telefonia su IP

- Protocolli standardizzati che supportino l'interoperabilità fra dispositivi diversi e di diversi costruttori
- Integrazione trasparente con la PSTN (Public Switched Telephone Network)
- Scalabilità: gli operatori telefonici hanno normalmente milioni di utenti
- Affidabilità: necessità di riprodurre quella della telefonia classica (>99,999%)

Segnalazione e Trasporto

- Le due componenti fondamentali della telefonia su IP sono:
  - segnalazione
  - trasporto audio/video

- Segnalazione
  - prima che audio (e video) possano fluire fra due terminali occorre:
    - localizzare il dispositivo remoto
    - negoziare i “mezzi” su cui trasportare i canali di comunicazione
- Trasporto audio/video
  - utilizza protocolli progettati per la trasmissione in tempo reale di audio e video su reti IP

I piani di segnalazione e trasporto sono ovviamente presenti anche nella telefonia classica (reti a commutazione di circuito)

Protocolli

- Standardizzati
  - H.323, H.248, SIP, MGCP, MEGACO, ...
- Proprietari
  - Skinny (Cisco), Skype, IAX (Asterisk), CorNet (Siemens)... e moltissimi altri

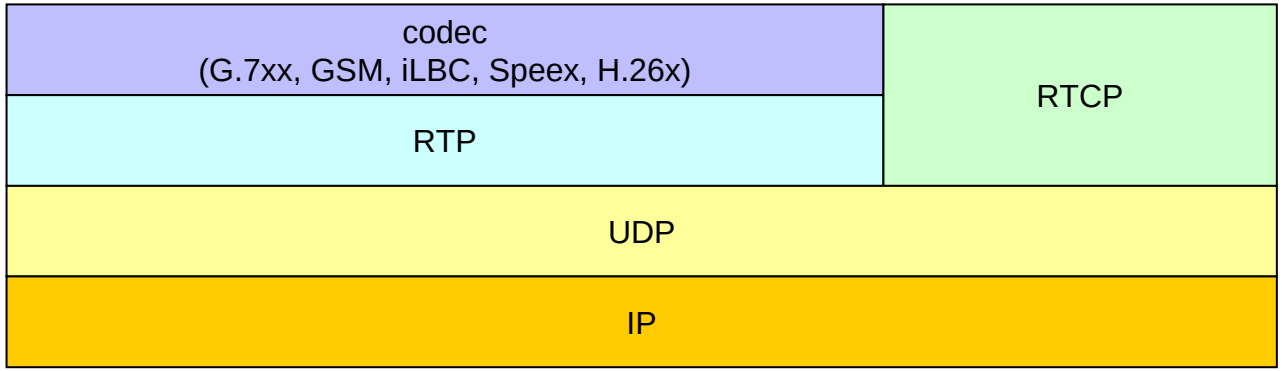
- I protocolli proprietari sono “chiusi”
  - comunità di sviluppo ristrette, poco innovativi
  - supporto minimo verso i protocolli standard: insieme limitato di funzionalità (limite: notevole sforzo nell’implementazione di Gateway fra diversi protocolli)

Protocolli standardizzati

	ITU-T	**IETF**
Trasporto audio/video	RTP/RTCP	RTP/RTCP
Segnalazione	H.323 H.248	SIP MGCP MEGACO

Protocolli standard di trasporto

- Protocolli progettati per la trasmissione in tempo reale di dati su reti IP (RFC 3550, 2003)
  - RTP (Real Time Protocol)
    - fornisce funzioni di trasporto end-to-end su rete IP adatte a flussi audio/video
  - RTCP (Real time Control Protocol)
    - protocollo con funzioni di controllo e monitoraggio
- Entrambi utilizzano il protocollo UDP



- Importante: il QoS non è un compito dei protocolli di trasporto!
- Se si vuole garantire il QoS occorre configurare opportunamente i dispositivi di rete (router: code con priorità, DiffServ, banda, MPLS, ...)

Protocolli standard di segnalazione

- Prima di poter trasmettere i flussi audio/video fra due terminali occorre svolgere due funzioni fondamentali:
  - la localizzazione del terminale remoto
    - in pratica determinare l'indirizzo IP a partire dall'identificativo del terminale
  - la negoziazione dei mezzi su cui trasportare i flussi audio e video
- I protocolli che svolgono queste funzioni sono detti protocolli di segnalazione. I due principali protocolli standardizzati sono:
  - **H.323** (ITU-T Study Group)
    - inizio lavori nel 1995, prima versione nel 1996; versione attuale: ver. 5, 2003)
  - **SIP** (Session Initiation Protocol, IETF)
    - inizio lavori nel 1995; prima versione nel 1999; versione attuale: RFC 3261, 2002)

Altri protocolli standard di segnalazione

- Nelle reti telefoniche tradizionali il piano di segnalazione e quello di trasporto sono separati; inoltre il backbone è ormai digitale, mentre l'"ultimo miglio" di utente è in una situazione mista (analogica e digitale)
  - Il piano di segnalazione (SS7, Signalling System N. 7, o sue varianti) è utilizzato per istruire i vari switch del backbone di trasporto su come instradare il traffico audio ed effettuare eventuali processi di compressione, rilevamento di segnalazione in banda, ecc.
  - Il piano di trasporto soddisfa requisiti di QoS che una normale rete IP generalmente non fornisce
- Molti operatori telefonici si propongono di realizzare in IP una situazione analoga
  - ad es. una rete IP basata su MPLS, con richiesta esplicita di QoS fra qualunque coppia di punti della rete
  - Elementi fondamentali del protocollo:
    - SG - Signalling Gateways: SS7 <-> H.323, SIP o SS7 in IP
    - MG - Media Gateways: trans-codifica
    - MGC - Media Gateway Controllers

Altri protocolli standard di segnalazione

- Il protocollo di comunicazione fra i MGC è l'oggetto di standardizzazione
  - MGCP (Media Gateway Control Protocol, IETF RFC 3661)
  - E' in corso uno sforzo congiunto di ITU-T e IETF, che però mantiene al protocollo due nomi identificativi differenti:
    - MEGACO (IETF MEGACO working group, RFC 3525)
    - H.248 (ITU-T Study Group 16)
  - E' attualmente sotto discussione il MEGACO/H.248 versione 2

Terminologia

- Terminale (end-point)
  - elemento finale di una comunicazione
  - hardware o software
  - supporti
    - audio, video, dati (chat, whiteboard)
  - ogni terminale ha associato almeno un indirizzo IP
- Server
  - Nodo su cui i terminali si registrano con un identificativo
    - impensabile chiamare direttamente l'IP del terminale
    - estremamente complicato se in presenza di DHCP
  - il server crea un mapping fra IP e Alias
  - Alla ricezione di una chiamata il server cerca di risolvere l'indirizzo IP del destinatario (local lookup, interrogazione DNS, location request ad altri server, ...)
  - Autenticazione, Registrazione, Autorizzazione, Accounting

Terminologia (2)

- Gateway
  - Possono essere visti come terminali telefonici che consentono a telefoni di diverse tecnologie di comunicare tra di loro
    - ISDN-H.323, H.323-SIP, IPv4-IPv6, etc..
  - Devono occuparsi della trasposizione di segnalazione e codifica
    - Segnalazione:
      - problemi dovuti a standard non pienamente implementati
    - Codifica:
      - carico computazionale (meglio se DSP dedicati)
- Conference Bridging
  - consentono conferenze audio/video (e dati) a 3 o più partecipanti
    - mix di audio e video
    - carico computazionale: server dedicati

Indirizzamento

- Un utente deve poter essere identificato univocamente indipendentemente dalla sua posizione (ovvero dal suo indirizzo IP)
- La telefonia tradizionale utilizza lo standard E.164 per la numerazione
  - identificativo composto al massimo da 15 cifre con un segno “+” iniziale:

    +39021234567

    in cui il “+” viene sostituito da “00”
- Al momento la Telefonia su IP utilizza 2 tipi di identificativi:
  - URI (RFC 2396)
  - numeri (E.164)
- Alcuni sistemi hanno provato ad utilizzare nomi (alfanumerici) ma questo conduce ad un sistema flat, non scalabile, applicabile solo localmente.

Indirizzamento (2)

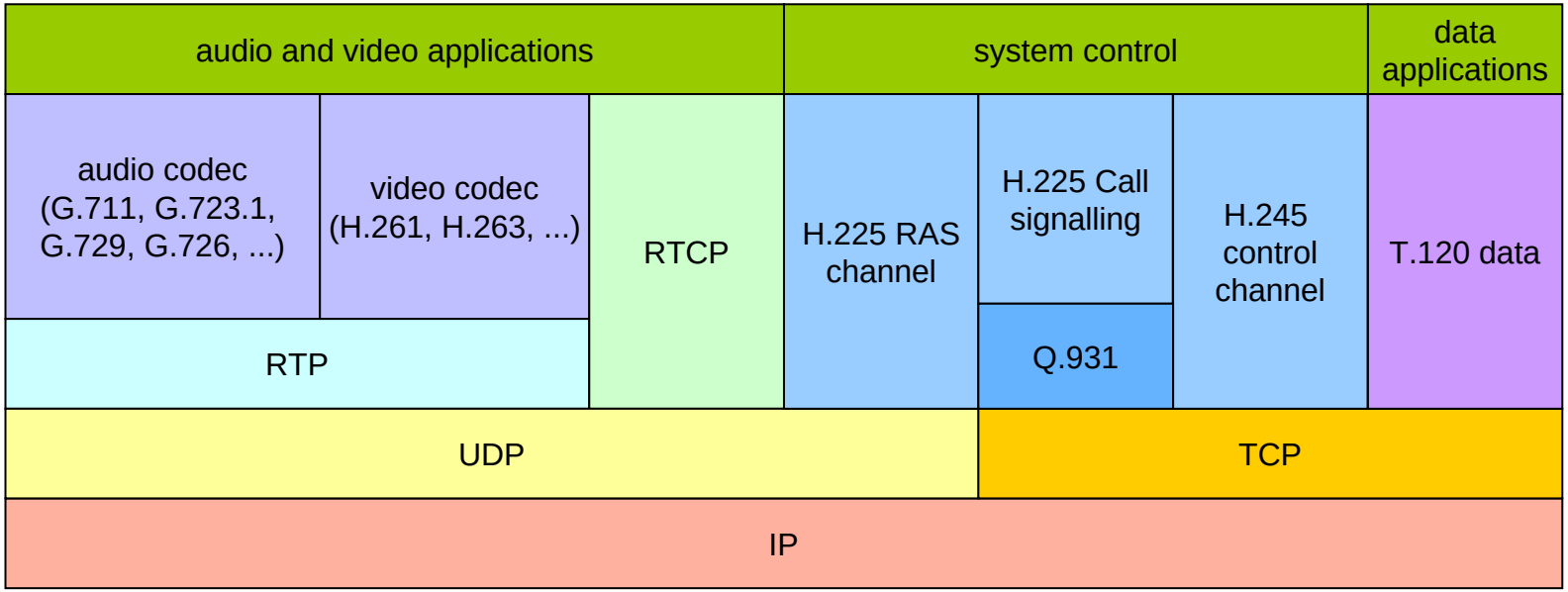
- URI (Universal Resource Identifier)
  - ha lo stile di un indirizzo di posta elettronica: utilizza uno spazio di nomi registrati e serve a descrivere una risorsa indipendentemente dalla sua posizione
    - mailto:user@domain.cc
    - sip:user@domain.cc
  - ha il vantaggio di essere mnemonico
  - ha lo svantaggio di rappresentare male la numerazione della telefonia tradizionale, aspetto necessario quando si vuole integrare telefonia tradizionale e telefonia IP
    - Soluzioni: ENUM

Protocolli standard:
H.323

Cosa è H.323?

- E’ una raccomandazione dell’ITU-T per sistemi di comunicazioni multimediali basati su reti a commutazione di pacchetto (reti IP)
- Inizio lavori nel 1995; prima versione nel 1996
- Non è un protocollo ma una raccomandazione (*umbrella specification*) di una serie di protocolli:
  - H.225.0 RAS, Q.931
  - H.245
  - RTP/RTCP
  - audio/video codec
  - T.120
- H.323 fa parte di una serie di standard di comunicazione che consentono videoconferenza su una gamma di reti. La serie e’ nota come H.32X e include H.320 (per ISDN) e H.324 (per PSTN, Public Switched Telephone Network).
  - Questo garantisce una buona interoperabilità con la PSTN

H.323: architettura del protocollo



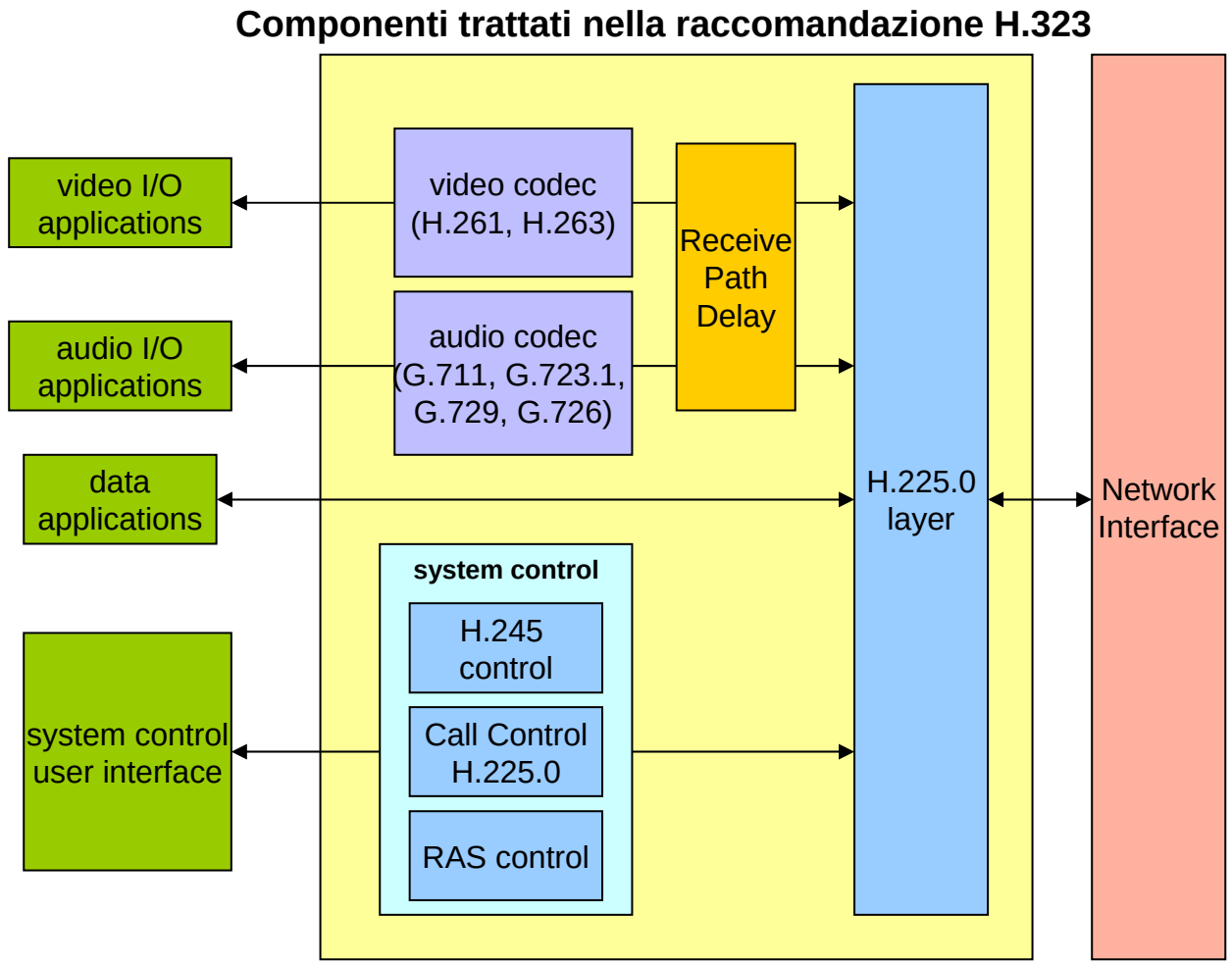
- Q.931
  - utilizzato per la fase di instaurazione e terminazione di chiamata
- H.245
  - utilizzato per lo scambio di capabilities tra i terminali

- RTP: Real Time Transmission Protocol
- RTCP: Real Time Control Protocol
- RAS: Registration, Admission, Status

H.323: elementi architetturali - **Terminale**

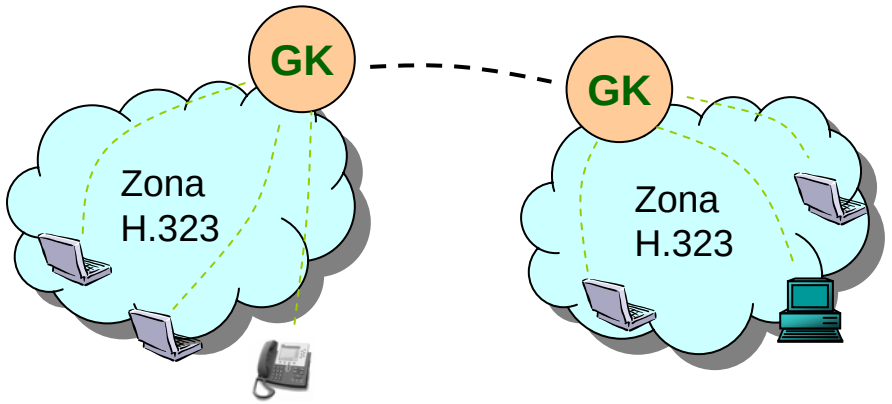
- **Terminale**
  - Supporta comunicazioni real-time bidirezionali con altre entità H.323
  - hardware clients
  - software clients
  - caratteristiche richieste
    - codec audio (almeno G.711)
      - RTP, RTCP
    - Protocolli di segnalazione H.323
      - H.225 (RAS, Q.931)
      - H.245
  - caratteristiche opzionali
    - codec video
    - trasmissione dati (chat, whiteboard, file sharing)

H.323: elementi architetturali - **Terminale**



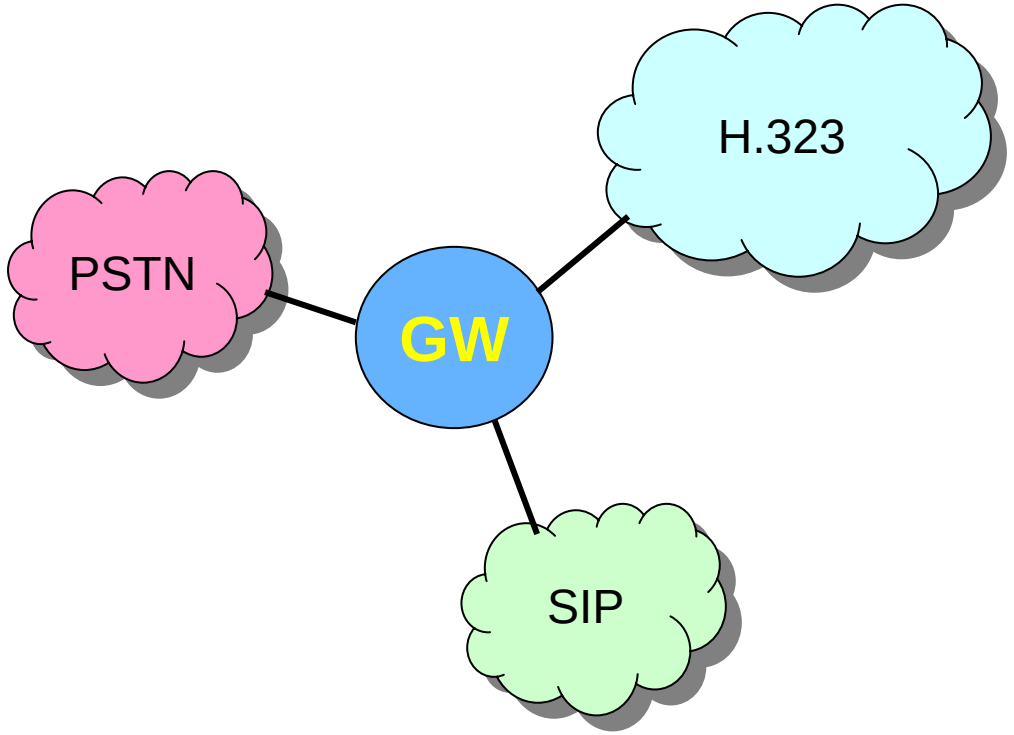
H.323: elementi architetturali - **Gatekeeper**

- E' l'elemento chiave di una architettura H.323 (anche se nella specifica è considerato opzionale)
- Funzioni
  - Gestione di una zona (ovvero delle entità H.323 ad esso registrati)
    - registrazione di endpoint
    - E' possibile avere gatekeeper di backup
  - Traslazione degli indirizzi: (H323ID e/o E.164) --> indirizzo IP
    - localizzazione
  - Instradamento delle chiamate
    - next hop location
  - Admission Control
  - Authorization control
  - Gestione della Banda
  - etc.



H.323: elementi architetturali - **Gateway**

- E’ una interfaccia fra mondi diversi
  - SIP <-> H.323
  - H.323 <-> ISDN
  - .....
- Si occupa di tradurre:
  - segnalazione
  - trasporto (transcodifica)

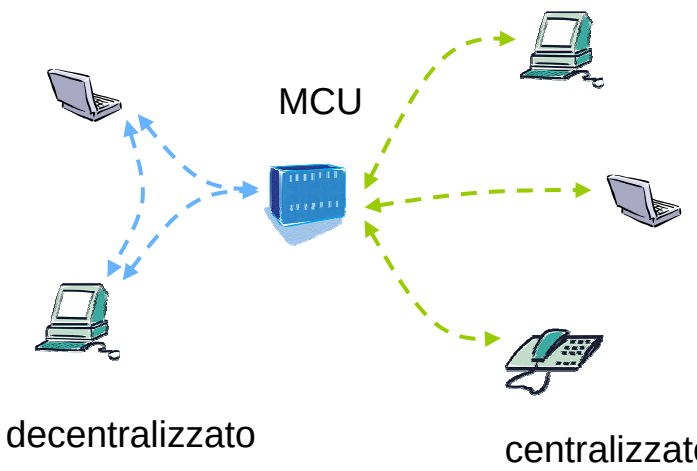


- Non è sempre possibile garantire una piena compatibilità di tutte le funzionalità

H.323: elementi architetturali - **MCU**

- Multipoint Control Unit
  - Server per conferenze multi-punto
  - Centralizzata: riceve i segnali (audio/video) di ogni terminali, effettua il mixing e lo rimanda a tutti
  - Decentralizzata: gestisce solo la segnalazione; il traffico audio/video è inviato in multicast fra i vari terminali
  - Mista: alcuni nodi sono in modalità centralizzata, altri in modalità decentralizzata

- La modalità decentralizzata richiede ai terminali una maggiore capacità computazionale (mixing) ed una maggiore banda; inoltre richiede una rete abilitata al multicast



Indirizzamento H.323

- Alias H323-ID
  - Qualunque sequenza di caratteri - *Basic ISO/IEC 10646-1 (Unicode)*
    - antonio-phone
    - pippo@aaa.cnr.it
- Alias E.164
  - 3451
  - 0503152288
  - 00390503151234

H.323: protocollo H.225.0

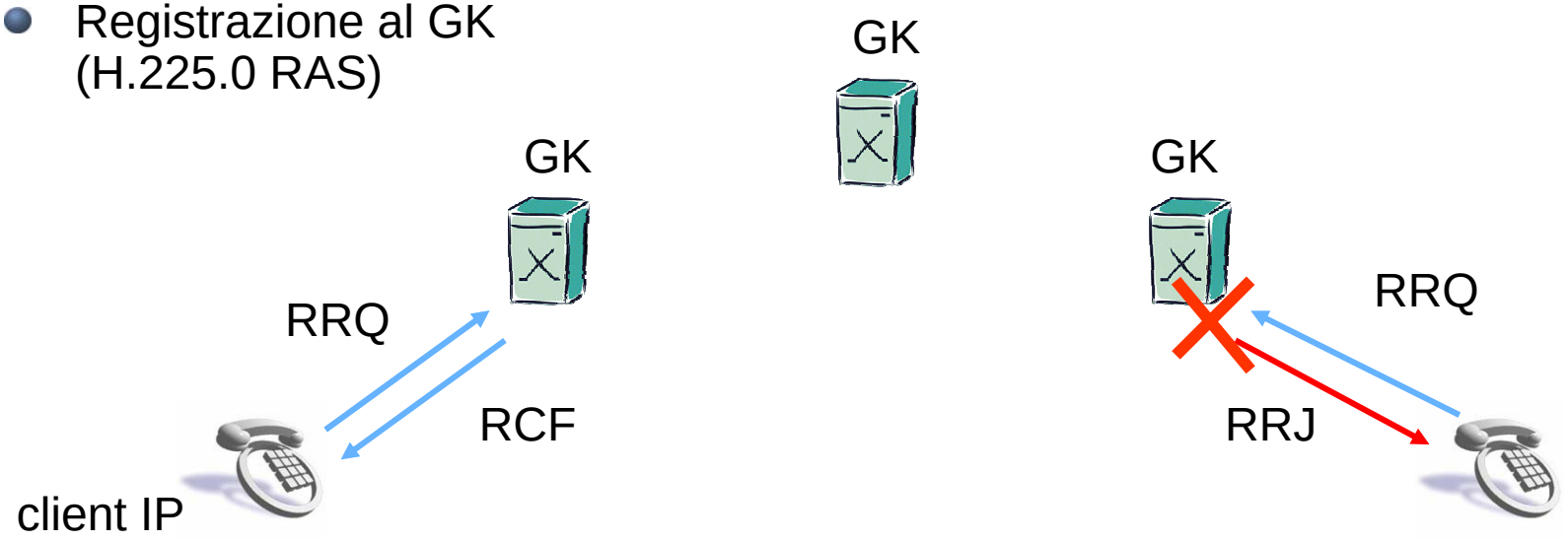
- H.225.0
  - Registration, Admission e Status (RAS)
    - End point
      - Registrazione, richiesta di permesso di utilizzo di risorse, richiesta di localizzare utenti remoti (mapping alias <-> IP), meccanismi di autenticazione
    - Gatekeeper
      - Monitoraggio dello stato degli end point associati
  - Call Signalling
    - Richiesta di call setup (chiamata), notifica di successo o fallimento di chiamata, trasporto di informazioni supplementari
    - Sono un sotto insieme di messaggi derivati dal Q.931 della segnalazione ISDN
    - Sono messaggi scambiati end-to-end, eventualmente attraverso i Gatekeeper

H.323: protocollo H.245

- H.245: Conference Control
  - Negoziazione end-to-end della codifica da usare per audio/video
  - Configurazione dei parametri da usare per i flussi multimediali (audio/video): porte UDP su cui gli end-point invieranno e riceveranno i flussi multimediali
  - Può trasportare altre informazioni supplementari quali la codifica dei toni DTMF
  - Questi messaggi sono end-to-end e possono transitare attraverso i gatekeeper
  - I messaggi di call setup H.225.0 contengo i parametri (porte TCP) per poter instaurare le connessioni H.245.
  - **Ottimizzazioni:**
    - **Tunneling H.245**: le informazioni dei messaggi H.245 sono direttamente incapsulate nei messaggi di call setup H.225.0 (riduce il numero di connessioni TCP da dover aprire)
    - **Early Connect**: i messaggi H.245 iniziano ad essere scambiati in parallelo ai messaggi di call setup H.225.0 (ottimizzazione sui tempi di setup)
    - **FastConnect** (o FastStart): nei messaggi di call setup H.225.0 vengono proposti i parametri dei canali dautilizzare. Successivamente parte la fase H.245 effettiva (ottimizzazione sui tempi di setup).
  - **Nota:** H.245 è un protocollo che eredita meccanismi derivati da altri protocolli utilizzati su ATM, PSTN. Pertanto è molto ricco di informazioni non tutte utilizzate dalla raccomandazione H.323

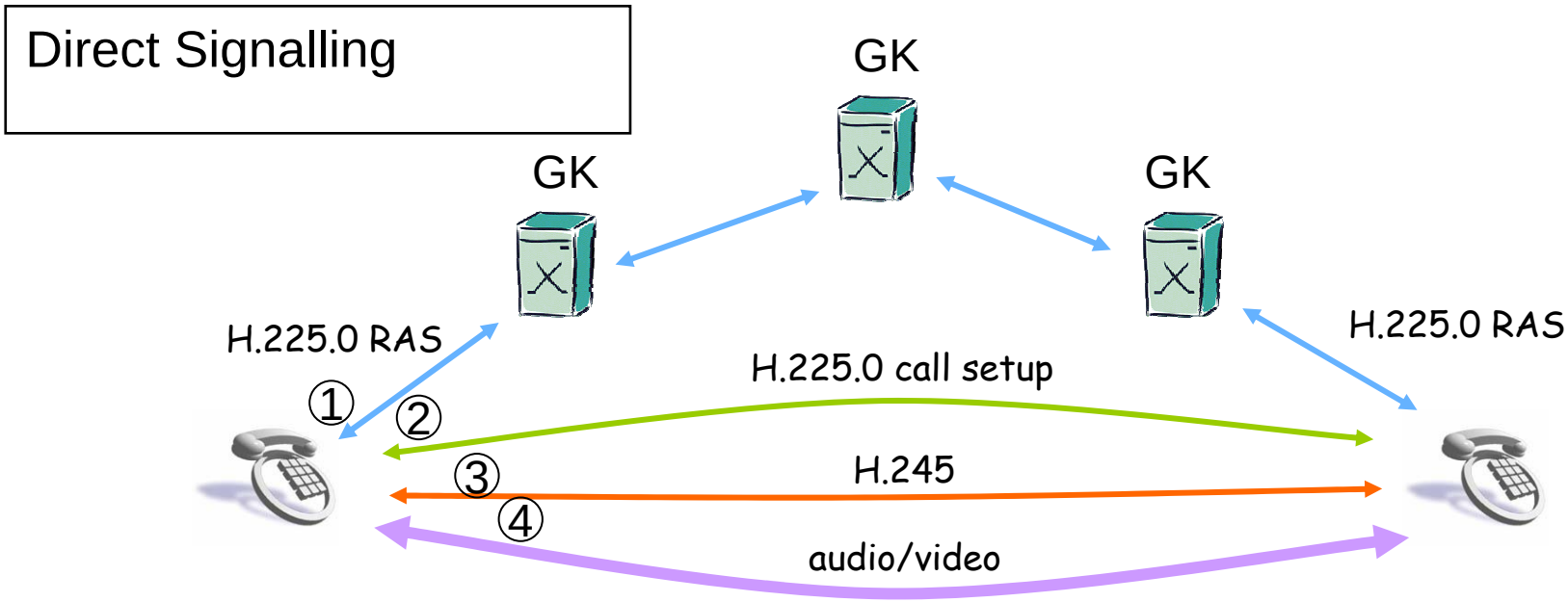
Messaggi H.323

- Registrazione al GK (H.225.0 RAS)



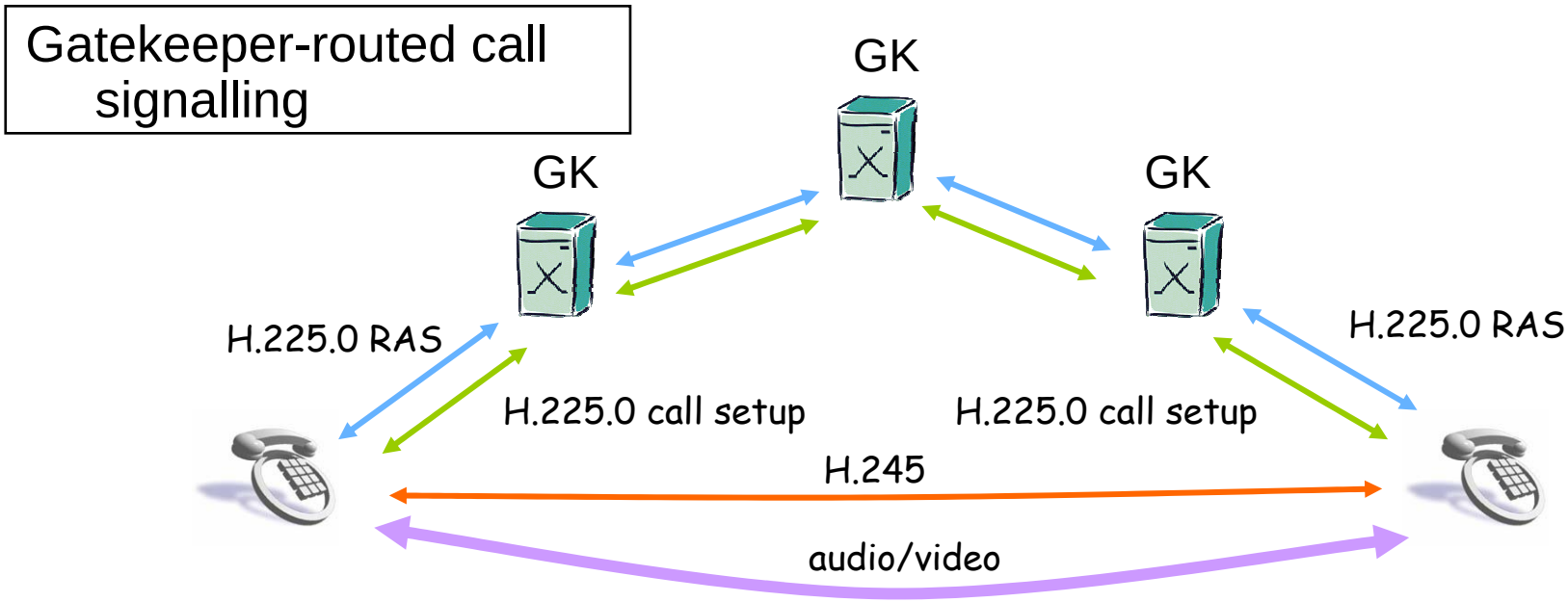
- **RRQ (Registration ReQuest): pacchetto UDP, porta 1719 del GK**
  - E' **fortemente sconsigliato** cambiare il numero di porta (1719), in quanto sui client questo valore potrebbe non essere configurabile
  - Nel payload del pacchetto UDP è contenuto anche il valore della porta TCP su cui il client starà in ascolto (generalmente 1720 o 1721; ma potrebbe essere qualsiasi valore configurato o scelto dal client). Analoga informazione è contenuta nel RCF del GK.
- Possibili risposte: RCF (Registration ConFirm), RRJ (Registration ReJect)
- Mantenimento della registrazione: KeepAlive

H.323: modi di segnalazione



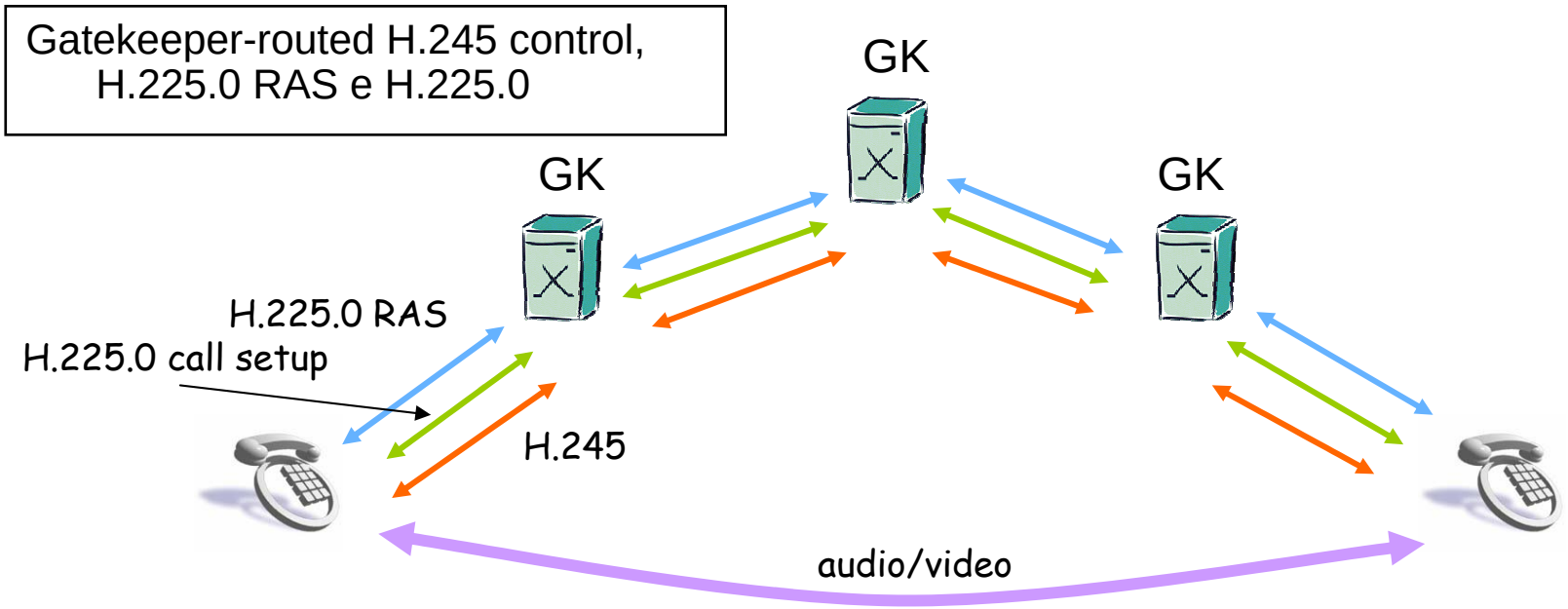
- **Solo i messaggi H.225.0 RAS transitano attraverso i gatekeeper**
- **Tutti gli altri messaggi sono diretti end-to-end**
- **Il traffico audio/video è end-to-end**

H.323: modi di segnalazione



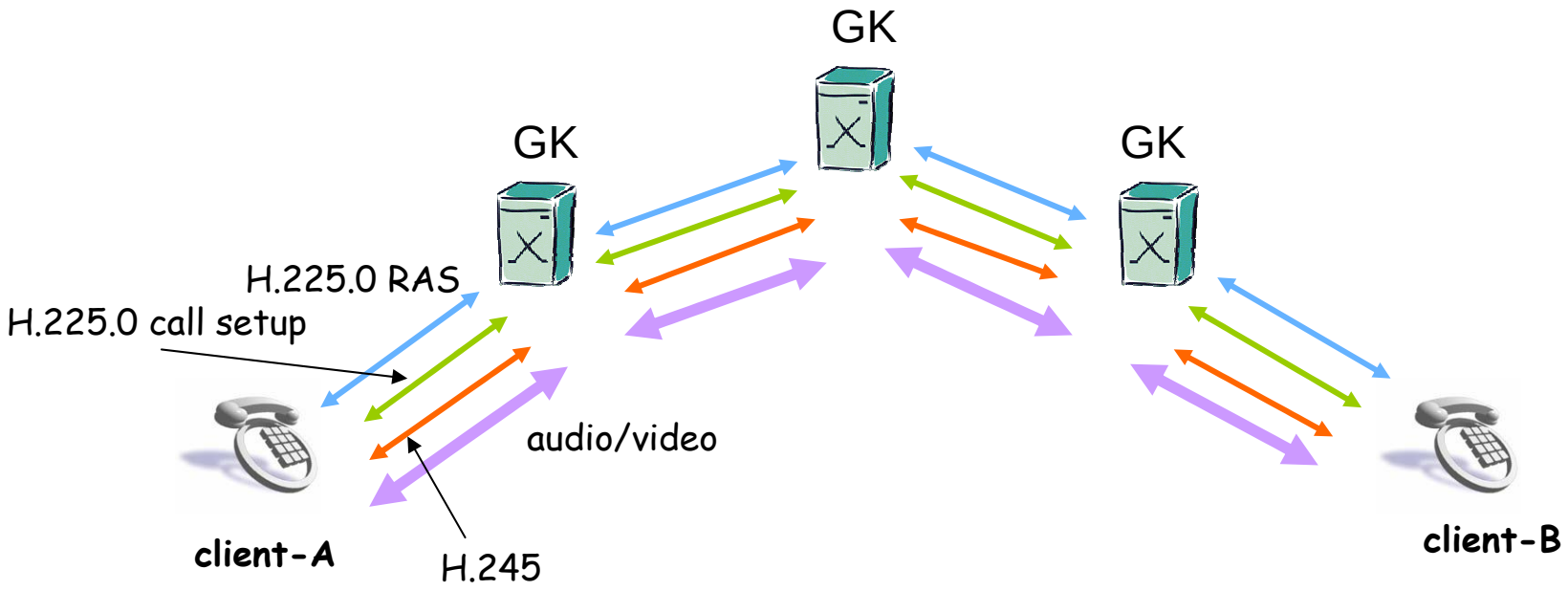
- **I messaggi H.225.0 RAS e H.225.0 call setup transitano attraverso i gatekeeper**
- **I messaggi H.245 sono diretti end-to-end**
- **Il traffico audio/video è end-to-end**

H.323: modi di segnalazione



- **Tutti i messaggi di segnalazione sono scambiati attraverso i gatekeeper**
- **Solo il traffico audio/video è end-to-end**

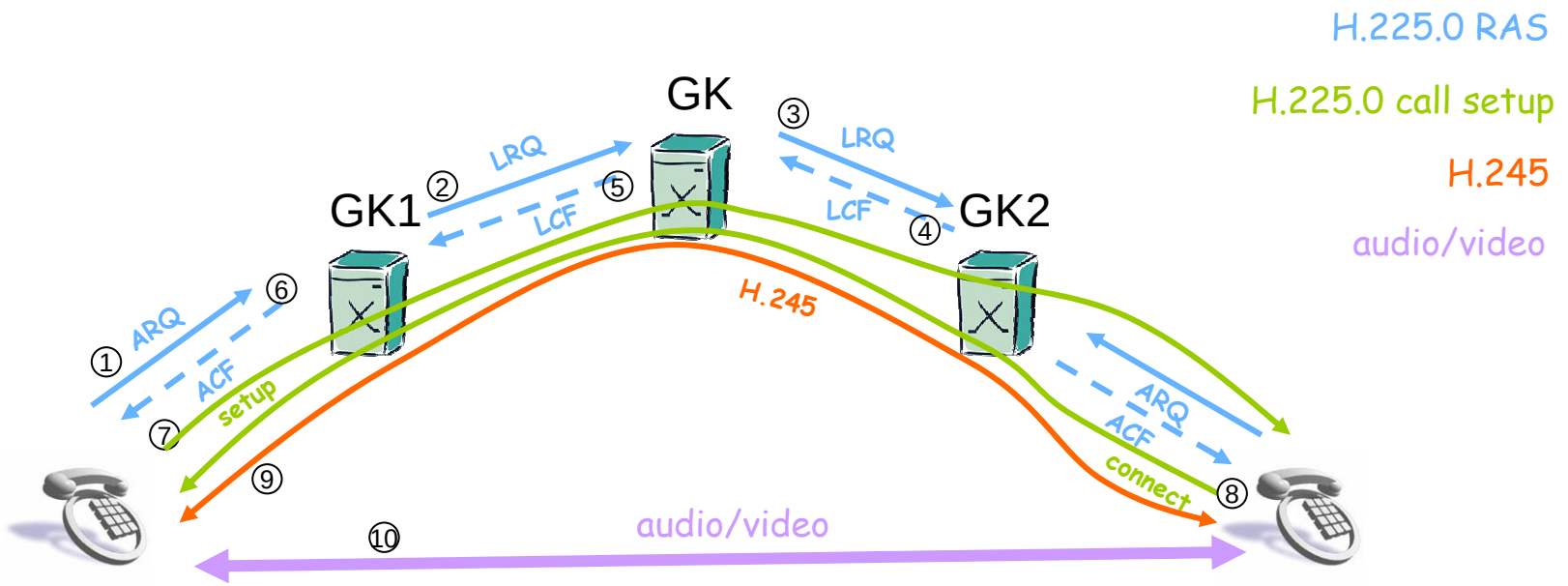
Gatekeeper in Proxy Mode



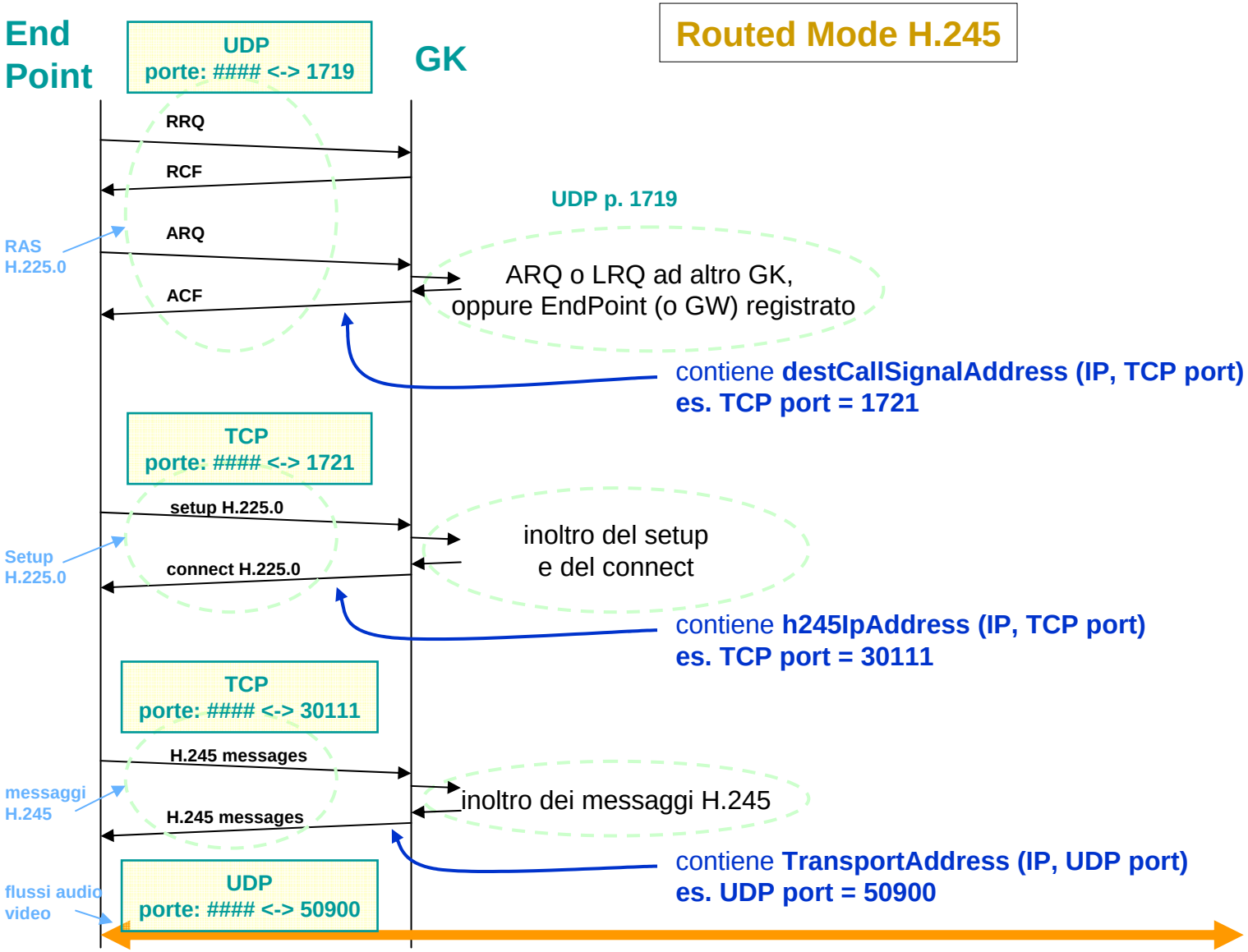
- **In questa modalità tutti i pacchetti, di segnalazione ma anche di traffico, sono scambiati attraverso i gatekeeper**
- **Non è una modalità prevista dallo standard, ma è offerta da molte implementazioni di gatekeeper e può essere utile in presenza di NAT e/o Firewall.**

Esempio di chiamata

- Routed Mode H.245



Messaggi, protocolli e porte



Messaggi, protocolli e porte

- Per la determinazione di IP/porta da usare per l'apertura di nuove connessioni TCP (o invio di pacchetti UDP) viene sfruttata l'informazione ricavata dal messaggio precedente.
- Se si utilizza la modalità "Direct Mode", IP e porta dipendono dall'altro end point.
- Se si passa dalla modalità "Route Mode", a quella "Route Mode H.245" o a quella "Proxy Mode", si può avere un controllo sempre maggiore su valori di IP e porte, facilitando la configurazione di Firewall e NAT.

Esempio di messaggio (connect H.225.0)

```
Ethernet II, Src: 00:50:56:0f:16:01 (00:50:56:0f:16:01), Dst: 00:50:56:0f:16:02 (00:50:56:0f:16:02)
Internet Protocol, Src: 146.48.96.183 (146.48.96.183), Dst: 146.48.98.111 (146.48.98.111)
Transmission Control Protocol, Src Port: 1721 (1721), Dst Port: 20018 (20018), Seq: 1, Ack: 198, Len: 133
TPKT, Version: 3, Length: 133
Q.931
    Protocol discriminator: Q.931
    Call reference value length: 2
    Call reference flag: Message sent to originating side
    Call reference value: 1000
    Message type: CONNECT (0x07)
    Display  '0649937124'
        Information element: Display
        Length: 10
        Display information: 0649933124
    User-user
        Information element: User-user
        Length: 109
        Protocol discriminator: X.208 and X.209 coded user information
H.225.0 CS
    H323_UserInformation
        h323-uu-pdu
            T_h323_message_body
                h323-message-body: connect (2)
                    connect
                        protocolIdentifier: 0.0.8.2250.0.4 (itu-t(0) recommendation(0) h(8) h225-0(2250) version(0) 4)
                        H245TransportAddress
                            h245Address: h245ipAddress (0)
                                h245ipAddress
                                    h245ipv4: 146.48.96.183 (146.48.96.183)
                                    h245ipv4port: 30216
                                ..........................
```

Protocolli standard:
SIP

SIP

- SIP (Session Initiation Protocol) è un protocollo utilizzato per iniziare una sessione. (IETF RFC 3261)
- E’ un Application Layer protocol per:
  - stabilire
  - modificare
  - terminare

    una sessione multimediale (come ad es. una telefonata)
- Utilizza SDP (Session Description Protocol, IETF RFC 3227) anziché inviare “session capabilities (come in H.323)
- Utilizza un indirizzamento stile e-mail; URL (Uniform Resource Locators); es.:
  - sip:antonio.pinizzotto@iit.cnr.it
  - sip:+004437612234@sip-proxy.org:5062
- Per il traffico multimediale utilizza RTP ed RTSP.

Metodi SIP

- SIP utilizza differenti tipi di messaggi (metodi) per le comunicazioni fra le varie entità:
  - **INVITE**
    - inizia le sessioni (nel messaggio è inclusa una descrizione del messaggio utilizzando la codifica SDP)
  - **ACK**
    - conferma dell'instaurazione della sessione
  - **BYE**
    - termina la sessione
  - **CANCEL**
    - cancella un INVITE pendente
  - **REGISTER**
    - mappa un indirizzo SIP (permanente) ad un IP temporaneo
  - **OPTIONS**
    - richiesta di caratteristiche (capability)
  - A questi metodi ne sono stati aggiunti altri (standard):
    - es.: INFO, UPDATE, MESSAGE, ...

Elementi architetturali del SIP

- User Agent Client (UAC)
  - entità logica che crea una nuova richiesta e la invia al destinatario
- User Agent Server (UAS)
  - entità logica che genera una risposta ad una richiesta SIP. La risposta può accettare, respingere o ridirezionare la richiesta.
- Proxy Server
  - entità logica che instrada le richieste SIP verso gli UAS e le risposte SIP verso gli UAC
  - Può rispondere direttamente ad una richiesta SIP (in tal caso opera come UAS)
  - Può operare in due modalità:
    - Stateless
      - inoltra le richieste dimenticandosene immediatamente
    - Statefull
      - inoltra le richieste ma ne tiene memoria, influenzando eventuali messaggi futuri

Elementi architetturali del SIP

- User Agent (UA)
  - combina le funzioni di UAC e UAS
  - Di fatto tutti i client SIP sono UA:
    - telefoni hardware:
      - Cisco 7960, 7920
      - Zyxel WiFi phone
      - etc.
    - telefoni software:
      - X-Lite, X-pro, Eyebeam
      - SJPhone
      - etc.
- Registrar Server
  - tipo speciale di UAS che accetta le richieste REGISTER e memotizza le richieste in esse contenute in un “location service”

Altri elementi architetturali del SIP

- Altri elementi derivati da quelli standard ma non definiti nello standard dell'RFC 3261
  - Redirect Server
    - un tipo speciale di UAS che ridireziona le richieste basandosi su un "location service" (utile per migliorare la scalabilità)
  - Outbound Proxy
    - Un tipo speciale di proxy che riceve le richieste da un client e le reinvia a destinazione (tipicamente si configura manualmente sul client UA per agirare firewall e NAT)

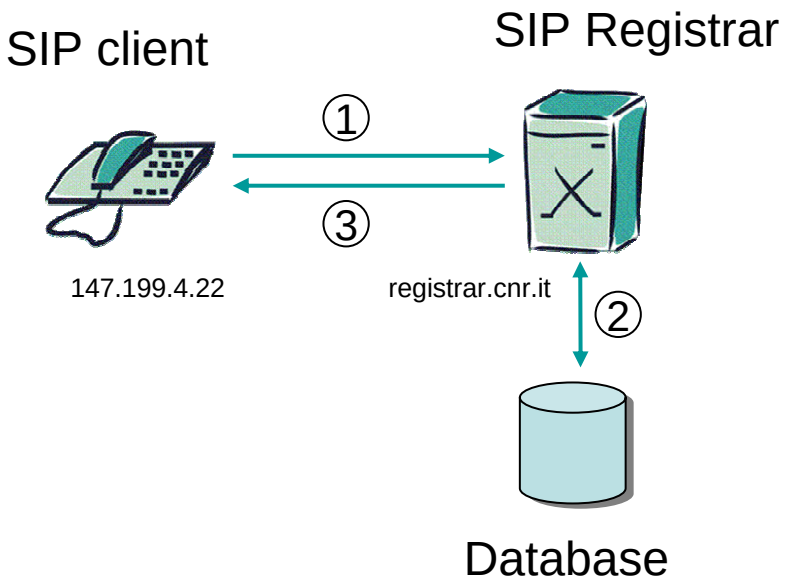
SIP: esempio di registrazione

1. Il client SIP invia un messaggio REGISTER al SIP proxy server, che agisce da SIP registrar

```
REGISTER sip:registrar.cnr.it SIP/2.0
To: sip:alberto@cnr.it
Contact: sip:alberto@147.199.4.22:5060; exiperes=1800
.....
```

2. Il SIP registrar memorizza la "posizione" corrente nel database
3. Il SIP registrar informa il client SIP che la registrazione ha avuto successo

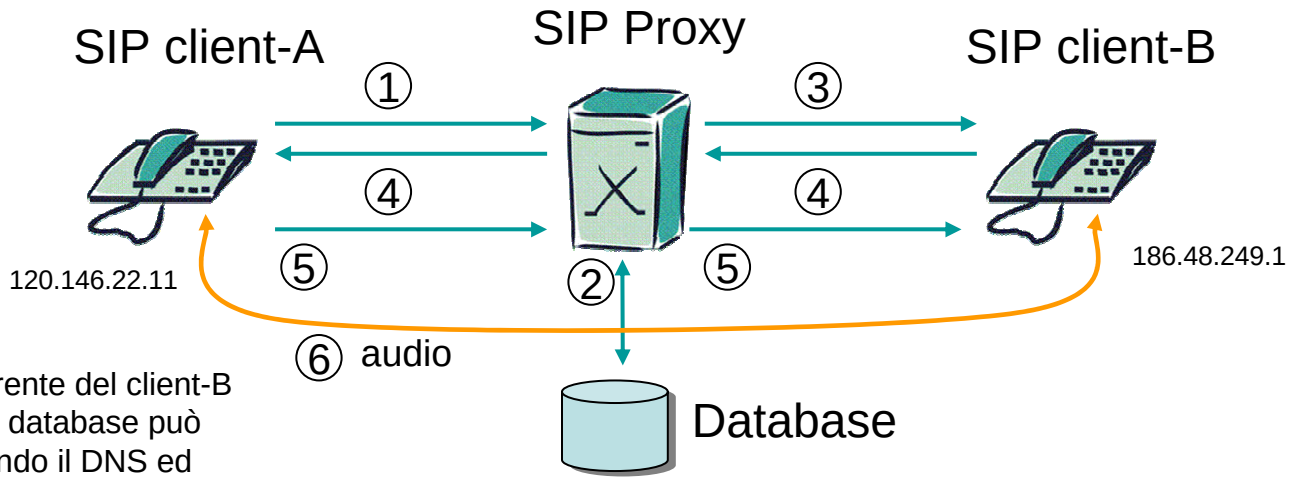
```
SIP/2.0 200 OK
To: sip:alberto@cnr.it
Contact: sip:alberto@147.199.4.22:5060; exiperes=1800
.....
```



SIP: esempio di sessione

1. Il client-A (alice) invia un SIP INVITE al SIP proxy server

```
INVITE sip:bob@milano.it SIP/2.0
To: sip:bob@milano.it
From: sip:alice@roma.it
Contact: sip:alice@120.146.22.11:5060
.....
```



2. Il SIP proxy cerca la posizione corrente del client-B nel suo database (se non è nel database può risolvere la posizione interrogando il DNS ed inoltrando la richiesta al successivo proxy)

3. Il SIP proxy inoltra l'INVITE alla posizione corrente del client-B

```
INVITE sip:bob@186.48.249.1 SIP/2.0
To: sip:bob@milano.it
From: sip:alice@roma.it
Contact: sip:alice@120.146.22.11:5060
.....
```

4. Quando l'utente B risponde, il client-B invia un messaggio di OK al client-A

```
SIP/2.0 200 OK
To: sip:alice@roma.it
From: sip:bob@milano.it
Contact: sip:bob@186.48.249.1:5060
.....
```

5. Il client-A invia un acknowledgment al messaggio con 200 OK, al client-B

```
ACK sip:bob@186.48.249.1:5060 SIP/2.0
.....
```

5. Il flusso audio viene inviato end-to-end in accordo con IP/port negoziati durante la segnalazione. Tali valori sono contenuti negli SDP inclusi negli INVITE e 200 OK, o in qualunque altro messaggio.

Esempio di messaggio SIP (INVITE)

```
INVITE sip:alberto.bianchi@iit.cnr.it SIP/2.0
Content-Length: 268
Contact: <sip:3820@146.48.124.84:5060>
Call-ID: 82FBF93D-1B36-4730-87F0-BB3263742C9D@146.48.124.84
Content-Type: application/sdp
CSeq: 1 INVITE
From: <sip:3820@bh3.iit.cnr.it>;tag=5983986524418
Max-Forwards: 70
To: <sip:alberto.bianchi@iit.cnr.it>
User-Agent: SJphone/1.60.289a (SJ Labs)

v=0
o=- 3340834275 3340834275 IN IP4 146.48.124.84
s=SJphone
c=IN IP4 146.48.124.84
t=0 0
a=direction:active
m=audio 49156 RTP/AVP 8 0 3 101
a=rtpmap:8 PCMA/8000
a=rtpmap:0 PCMU/8000
a=rtpmap:3 GSM/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-11,16
```

SDP

Confronto H.323 - SIP

- H.323:
  - è stato rilasciato prima per cui è stato anche implementato e diffuso prima (1996)
  - è nato per integrare il mondo legacy con Internet per cui è più maturo nell'interoperabilità con la PSTN
  - è molto complesso e per motivi di compatibilità con le versioni precedenti diventa sempre più complesso
- SIP:
  - è stato rilasciato dopo H.323 (1999)
  - è stato progettato pensando solo a Internet
  - è meno complesso di H.323
- Tutto lascia pensare che SIP sarà il futuro, ma sul lungo termine. H.323 continuerà ad esistere ancora per lungo tempo

Soluzioni per integrazione PSTN – Telefonia IP: **ENUM**

ENUM

- ENUM è parte di un ampio progetto che ha come obiettivo quello di fornire un meccanismo per l'identificazione di servizi su Internet
- Il mezzo scelto per questo scopo è il DNS mediante l'utilizzo di due tipi di record:
  - **SRV record**
  - **NAPTR record**
- Senza entrare nel merito di tutte le potenzialità di ENUM vediamo una applicazione pratica utilizzabile per la localizzazione di un utente telefonico (sia esso legacy che IP)

- IMPORTANTE: NON occorre inserire nel DNS un singolo record NAPTR per ogni utente telefonico!
  - Può essere sufficiente un **unico record** per **l'intero prefisso** gestito da un centralino.

ENUM (esempio)

- L'esempio riportato è relativo al protocollo SIP ma nel caso di applicazione al protocollo H.323 la soluzione è identica (a parte la sostituzione della keyword "sip" con "h323" e l'utilizzo dei Gatekeeper al posto dei SIP Proxy)

- Consideriamo una sede che abbia un centralino associato al prefisso telefonico della PSTN italiana: "0503152"
- Nella file di zona del dominio "2.5.1.3.0.5.0.9.3.e164.arpa" (corrispondente al prefisso +390503152) viene inserito il record:

```
$ORIGIN  2.5.1.3.0.5.0.9.3.e164.arpa.
*  IN NAPTR 10 10 "u" "e2u+sip“ "!^\\+390503152(.*)$!sip:\\1@psip.iit.cnr.it!" .
```

(per la sintassi del record NAPTR si veda più avanti)

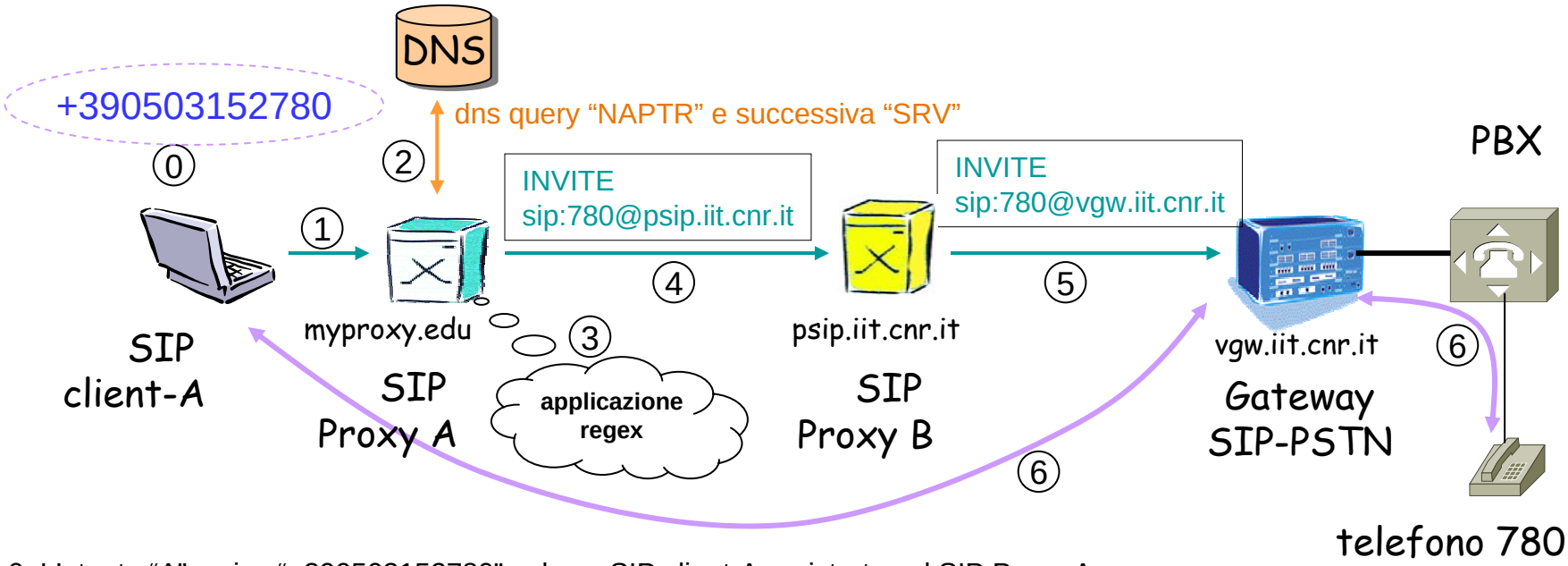
ENUM (esempio)

Vedi figura slide successiva

- Supponiamo che un client SIP registrato ad un SIP Proxy di una qualunque sede nel mondo voglia chiamare il numero +390503152780.
- Il suo SIP proxy per scoprire a chi inoltrare la richiesta inoltra una query DNS di tipo NAPTR per il nome a dominio:
  - 0.8.7.2.5.1.3.0.5.0.9.3.e164.arpa. (corrispondente al numero +390503152780)
- la risposta del DNS contiene la stringa:
  - **"!^\\+390503152(.*)$!sip:\\1@psip.iit.cnr.it!"**
  - fatta da due parti:
    - la prima è una regular expression da applicare alla chiave originaria delle query, ovvero il numero "+390503152780".
    - La seconda parte è il replacemente che però è parametrizzato in quanto contiene una back reference ("**\\1**") alla regular expression ("**(.*)**"). In questo caso il valore del parametro estratto è "**780**".
- All fine il risultato dell'applicazione di questa espressione è:
  - **sip:780@psip.iit.cnr.it** (780 è il numero breve interno dell'utente sul centralino)
- Nota: a questo punto il SIP Proxy esegue anche una query DNS di tipo SRV per scoprire se al dominio psip.iit.cnr.it è associato un server (diverso da psip.iit.cnr.it) che offra servizio sip su udp. In assenza di risposta DNS si assume che il server sia proprio psip.iit.cnr.it.

ENUM (esempio)

- Rivediamo lo stesso esempio graficamente



0. L’utente “A” scrive “+390503152780” sul suo SIP client-A registrato sul SIP Proxy A.
1. Il client-A invia un messaggio INVITE +390503152780@myproxy.edu al SIP Proxy A.
2. Il SIP Proxy A invia una query DNS NAPTR per il dominio 0.8.7.2.5.1.3.0.5.0.9.3.e164.arpa. (segue una query DNS di tipo SRV, come spiegato nella precedente slide).
3. La risposta viene elaborata come esposto nella precedente slide.
4. il SIP Proxy A invia un INVITE sip:780@psip.iit.cnr.it al SIP Proxy B (competente per il centralino dell’utente da chiamare).
5. Il SIP Proxy B sa che il numero 780 è fra quelli gestiti dal centralino (per configurazione) ed invia quindi un INVITE al GW.
6. Il GW apre una connessione con il client-A ed il telefono legacy del numero 780 attraverso il centralino.

Record NAPTR

I record NAPTR (specificati nell'IETF RFC 3403) hanno la seguente forma:

```
"domain-name TTL Class NAPTR order preference flags service regexp replacement"
```

Per esempio:

```
$ORIGIN 5.6.7.8.9.3.e164.arpa.
1.2.3.4 43200 IN NAPTR 60 50 "u" "e2u+sip" "!^.*$!sip:389001@sip.bigu.edu!" .
```

In cui:

domain-name è il dominio da interrogare: corrisponde a "1.2.3.4. 5.6.7.8.9.3.e164.arpa."

TTL è il tempo di vita nella cache: 43200 sec (12 ore)

Class è la classe: IN

NAPTR è il tipo di record in esame

order: 60.

preference: 50.

Ordine e preferenza hanno significati diversi da priorità e peso dei record SRV. L'ordine indica l'ordine con cui vanno letti i record; prima quelli a valore più basso. Non appena ne viene trovato uno che soddisfa, va preso ed i successivi vanno scartati. A parità di ordine si guarda la preferenza (prima quelli a valore più elevato) ma il client può ignorarne il valore di preferenza.

Record NAPTR

(continua)

"domain-name TTL Class NAPTR order preference flags service regexp replacement"

Esempio:

```
$ORIGIN 5.6.7.8.9.3.e164.arpa.
1.2.3.4 43200 IN NAPTR 60 50 "u" "e2u+sip" "!^.*$!sip:389001@sip.bigu.edu!" .
```

flag: "u"***.*** E' una stringa alfanumerica che determina il modo di interpretare ed eventualmente riscrivere un record. I flag S, A e U sono utilizzati come flag per terminare il ciclo DDDS (RFC 3402). Il più usato sembra essere U che non nega una successiva query di un record SRV.

service: **"E2U+SIP".** E2U è mandatorio per determinare la traduzione da E.164 a URI. SIP specifica il tipo di servizio. Le combinazioni sono molteplici: "E2U+h323", "E2U+msg:mailto", etc.

regexp: **"!^.*$!".** E' una regular expression che il client applica alla stringa originale. la stringa **"!^.*$!"** è la più generale possibile che lascia la stringa originale invariata.

replacement: "**sip:389001@sip.bigu.edu**". In generale può contenere dei back reference alla regular expression per generare un target in funzione della query fatta.

Albero/i per ENUM

- Il protocollo ENUM è soggetto a problemi legali, differenti da paese a paese.
- L'albero dovrebbe essere unico per tutto il mondo (e164.arpa), ma non è chiaro chi è autorizzato a gestire la zona di ogni nazione (es.: 9.3.e164.arpa).
- Per questo motivo sono nate molte iniziative di sperimentazioni che utilizzano alberi privati.
- Una di queste è la sperimentazione del Piano (Privato Nazionale) di Numerazione VoIP, con utilizzo di ENUM e SIP, promossa dal Namex. L'albero scelto per tale sperimentazione è:
  - e164.namex.it

GDS e sua evoluzione con ENUM

GDS e sua evoluzione con ENUM

- Il GDS (Global Dialling Scheme) è una gerarchia di gatekeeper H.323 ad estensione mondiale che oggi consente di veicolare semplici telefonate o videoconferenze attraverso Internet. (http://www.vide.net)
  - E' basata sui meccanismi di localizzazione H.323 sfruttando la struttura gerarchica. Ad esempio in Italia il Cineca (Bologna) ospita il gatekeeper competente per il prefisso italiano 0039.
  - Anche la rete VoIP del GARR è connessa al GDS consentendo anche ai telefoni legacy di effettuare telefonate ad enti stranieri.
- Una estensione del GDS, basata su ENUM, è attualmente oggetto di discussione nella TERENA Task Force Voice Video and Collaboration (TF-VVC)
  - http://www.terena.nl/tech/task-forces/tf-vvc/

GDS e sua evoluzione con ENUM

- Il problema maggiore potrebbe essere quello legato alla non univocità dell'albero di interrogazione per ENUM.
- Tuttavia al momento esistono due alberi principali (maggiormente popolati):
  - in Giappone (APAN)
  - in USA (Internet2)
- L'idea è quella di agganciarsi, per l'Europa, all'albero USA.
- I Gatekeeper H.323 del GDS andrebbero quindi configurati per interrogare in sequenza i due alberi ENUM e come ultima possibilità inoltrare una location sulla gerarchia del GDS.

Applicazioni:
SIP.EDU

SIP.EDU

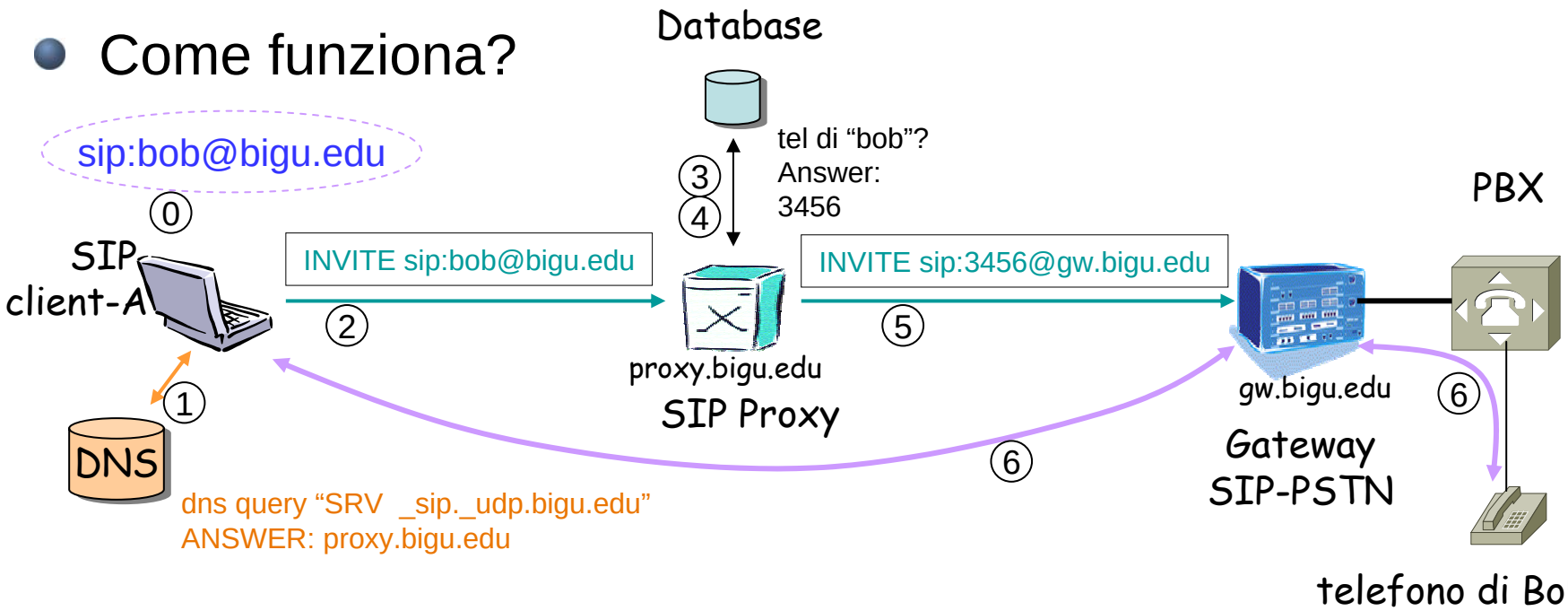
- Progetto di Internet2 VoIP WG
  - http://voip.internet2.edu, http://mit.edu/sip/sip.edu/
- Obiettivi
  - incrementare il numero di utenti SIP raggiungibili (all'interno della comunità Internet2, ma aperta a tutte le istituzioni accademiche)
  - promuovere la convergenza fra voce ed e-mail
- Abilitandosi a SIP.EDU un ente rende raggiungibili via SIP i propri telefoni (compresi quelli legacy)

SIP.EDU

- Di cosa si tratta?
- Esempio:
  - L'email di Bob è bob@bigu.edu
  - Il telefono legacy della scrivania di Bob è +39555123456
  - Bob decide di associare la sua e-mail al suo telefono utilizzando il servizio SIP.EDU
  - Chiunque, utilizzando un qualunque client SIP, chiami sip:bob@bigu.edu farà squillare il telefono (legacy) di Bob e potrà quindi parlarci.

SIP.EDU

- Come funziona?



0. Alice scrive *sip:bob@bigu.edu* sul suo SIP client
1. Il SIP client di Alice interroga il DNS per determinare se esiste un servizio di tipo sip su udp associato a bigu.edu; la risposta è proxy.bigu.edu
2. Il SIP client di Alice invia un “INVITE sip:bob@bigu.edu” al proxy.bigu.edu
3. Il proxy interroga il suo database per determinare il numero di telefono associato a bob
4. Il database fornisce per bob il numero di telefono 3456 (numero breve interno utilizzato sul centralino)
5. Il proxy riscrive l’INVITE sostituendo la parte user con il numero di telefono ed il dominio con quello del GW: “INVITE sip:3456@gw.bigu.edu” che invia al GW
6. Il GW apre una connessione con il client-A ed il telefono legacy di Bob attraverso il centralino

Configurazione del DNS per SIP.EDU

- Esempio:
  - "_sip._udp.bigu.edu 43200 IN SRV 10 10 5060 proxy.bigu.edu.“
-
- La query del record SRV _sip._udp.bigu.edu consente di determinare a quale SIP Proxy (proxy.bigu.edu:5060) inviare l’INVITE, in UDP, per gli utenti con dominio “bigu.edu”.

Record SRV

I record SRV (specificati nell'IETF RFC 2782) hanno la seguente forma:

"_Service._Proto.Name TTL Class SRV Priority Weight Port Target"

Per esempio:

"_sip._udp.bigu.edu 43200 IN SRV 10 10 5060 proxy.bigu.edu."

In cui:

Service è il servizio: SIP

Proto.Name è il trasporto: UDP (può anche essere TCP, SCTP o TLS)

TTL è il tempo di vita nella cache: 43200 sec (12 ore)

Class è la classe: IN

SRV è il tipo di record in esame

Priority è la priorità: 10. Ha senso nel caso di più record SRV per determinare l'ordine di interrogazione. Quelli con valore più basso sono interrogati prima.

Weight è il peso: 10. Ha senso nel caso di più record SRV con stessa priorità, per determinarne la probabilità con cui un proxy viene utilizzato. Quelli con valori più alti sono interrogati più frequentemente.

Port è la porta: 5060.

Target è il nome del server: proxy.bigu.edu. (In base alla sintassi del DNS la riga va terminata con un punto.)

La domanda a cui questo record risponde è: "qual è il server e la sua porta UDP che forniscono un servizio SIP per il dominio bigu.edu?"

Problematiche in presenza di Firewall e NAT

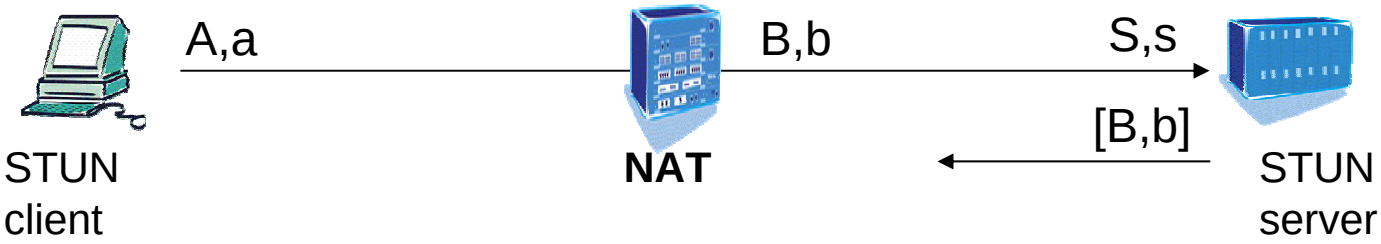
Problemi in presenza di Firewall e/o NAT

- Il problema nasce dal fatto che il payload dei pacchetti di segnalazione contiene informazioni relative a indirizzo IP e porta del client. Il NAT modifica l'header dei pacchetti ma non il payload. Inoltre l'apertura dinamica di connessioni TCP e porte UDP crea ulteriori problemi con NAT e Firewall.
- Non è possibile dare soluzioni generiche in quanto ogni soluzione dipende dal tipo di Firewall o NAT e dalle funzionalità più o meno complesse implementate

- Alcuni Firewall e NAT con funzionalità evolute sono in grado di riconoscere e interpretare il payload dei pacchetti di segnalazione H.323. Le azioni sono:
  - **FW**: apertura dinamica di opportune porte TCP e/o UDP (statefull Firewall), sia per messaggi H.225.0 e H.245 che per i flussi multimediali
  - **NAT**: creazione di mapping dinamici sia per TCP che UDP, sia per messaggi H.225.0 e H.245 che per i flussi multimediali; inoltre, manipolazione opportuna del payload.

Soluzioni per NAT: STUN

- STUN: Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs) - IETF RFC 3489 (Marzo 2003)



- Mediante un server esterno, il client scopre il suo IP publico e le regole di mapping delle porte usate dal NAT.
- Svantaggi:
  - non funziona con NAT simmetrici (oggi largamente utilizzati!)
  - non funziona fra due client entrambi dietro lo stesso NAT
  - è pensato solo per UDP (la segnalazione H.323 usa anche TCP)
- Vantaggi
  - implementato in molti client: Xten, Zysel WiFi phone, SJPhone, OhphoneX
  - open-source stund server

Soluzioni per NAT: non ancora standard

- **TURN** - IETF MIDCOM draft “Traversal Using Relay NAT)
- **ICE** - IETF MMUSIC draft “Interactive Connectivity Establishment: A Methodology for Network Address Translator (NAT) Traversal for Multimedia Session Establishment Protocols
- **B2BUA** – Back-to-back User Agent

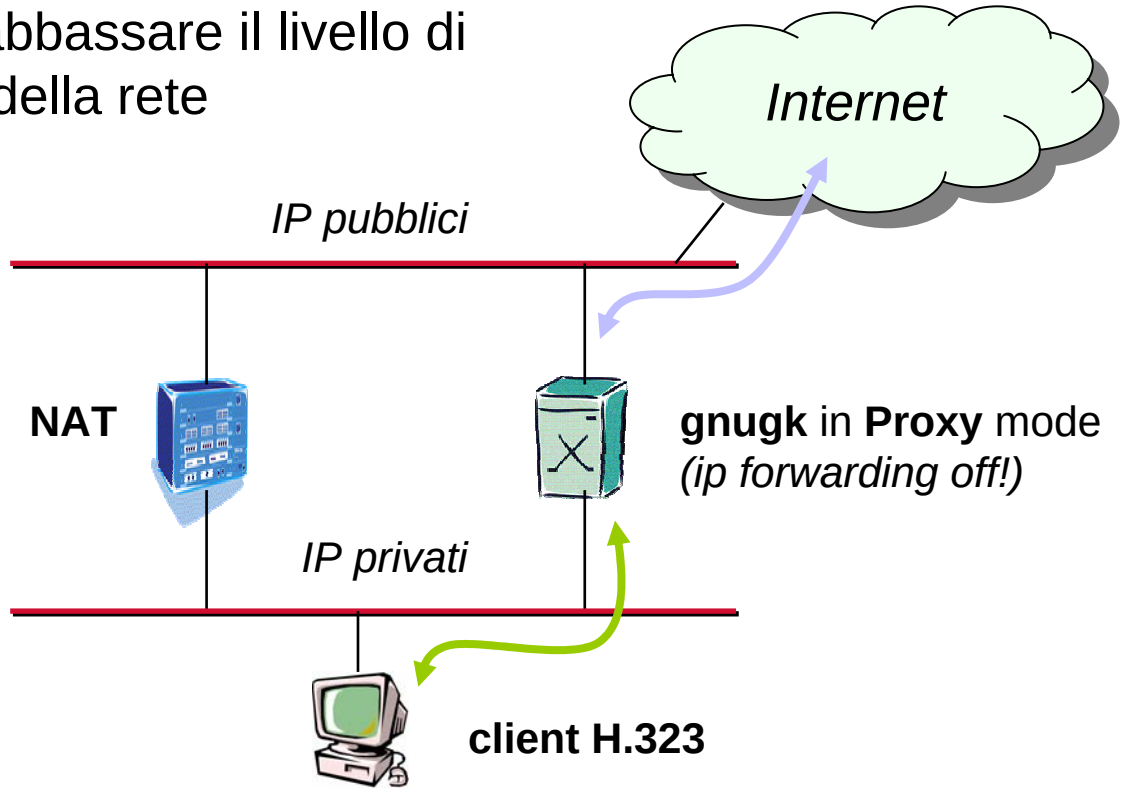
- Limiti:
  - Non sono ancora standard
  - Ancora poco utilizzabili perché non implementati nei client
  - Per alcuni il server non è disponibile in open-source
  - ....

Altre soluzioni per Firewall/NAT Traversal

- Application Level Gateway
  - Interpretazione dinamica del payload
    - SIP (o H.323) **aware** NAT/FW che modificano opportunamente i messaggi SIP (o H.323) e aprono opportuni mapping o IP/porte
  - Configurazione statica
    - Utilizzo di modalità Proxy sui server (SIP o H.323) e configurazione **statica** di opportuni mapping (NAT port forwarding) o aperture di IP/porte (open pinholes su Firewall)

Alcune soluzioni con gnugk (1)

- Doppia interfaccia
  - potrebbe abbassare il livello di sicurezza della rete

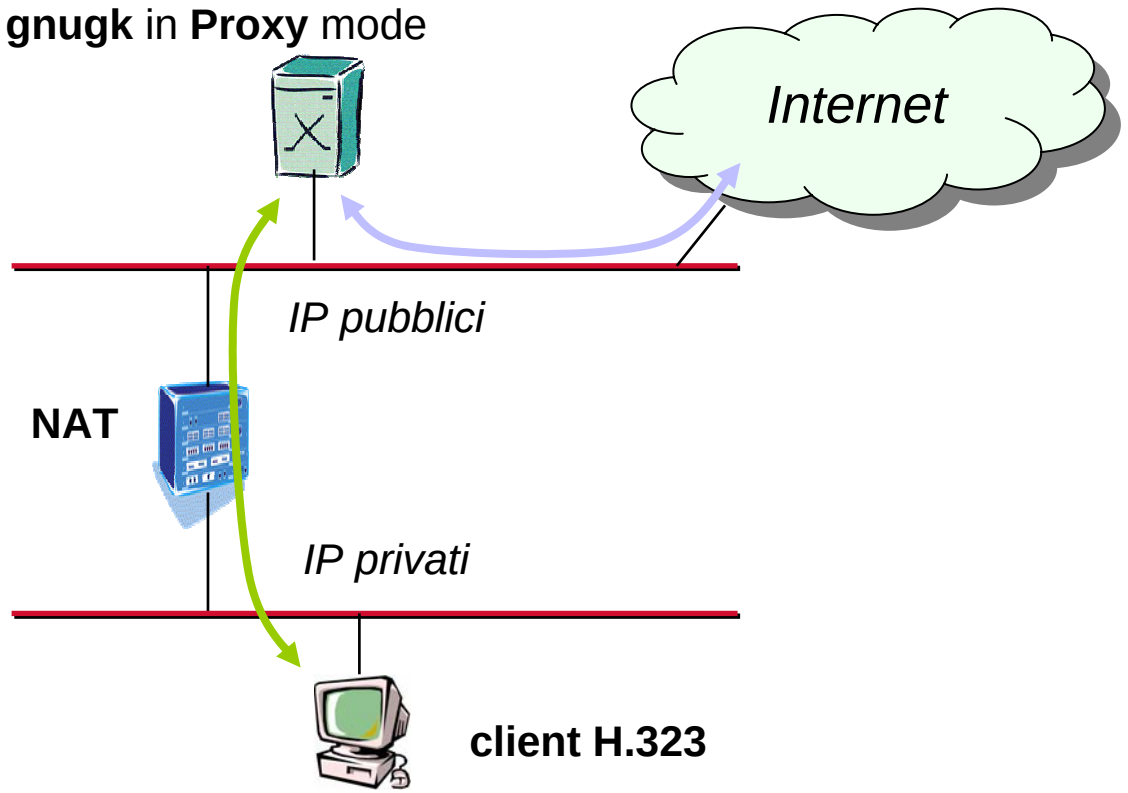


(1) gnugk: gatekeeper open-source descritto nel seguito delle slides

Alcune soluzioni con gnugk

- gnugk nella zona pubblica
  - dalla versione 2.0.2, gnugk è in grado di capire che il client è dietro un NAT

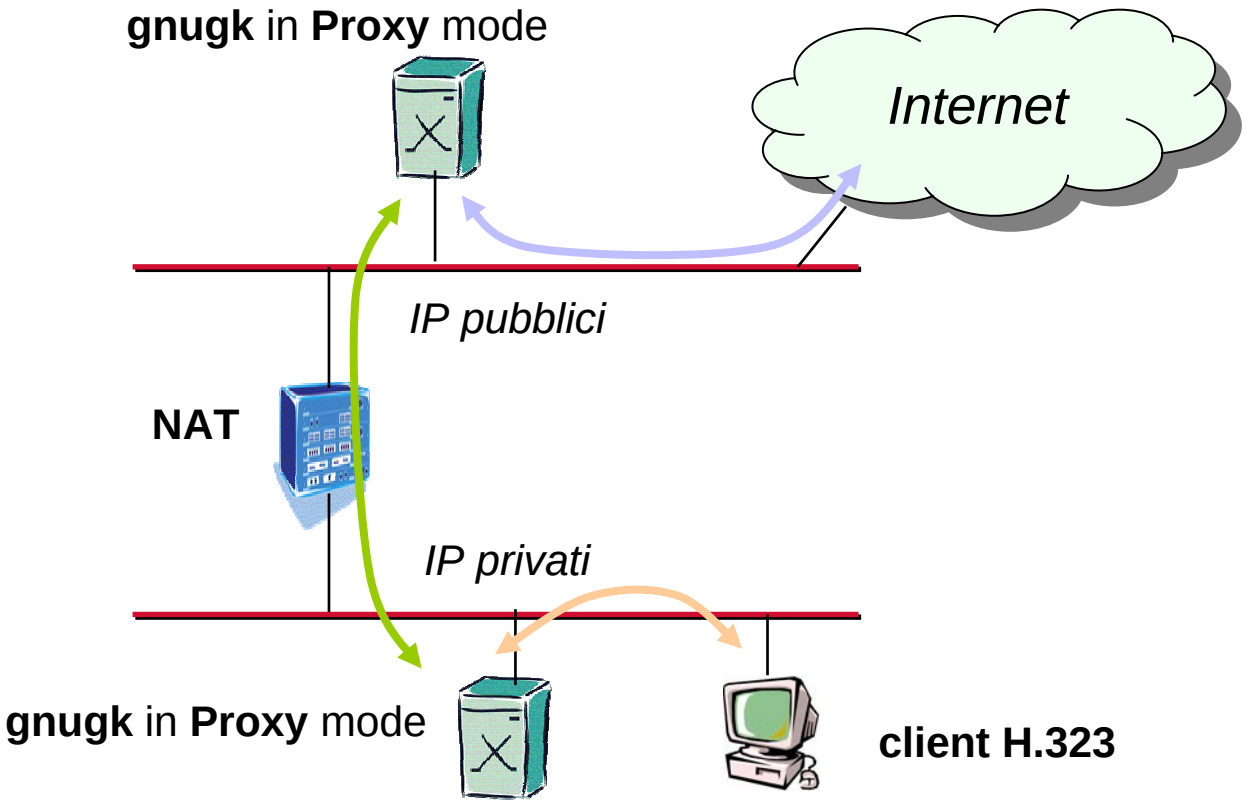
- chiamate solo in uscita a meno di un opportuno mapping statico sul NAT



Alcune soluzioni con gnugk

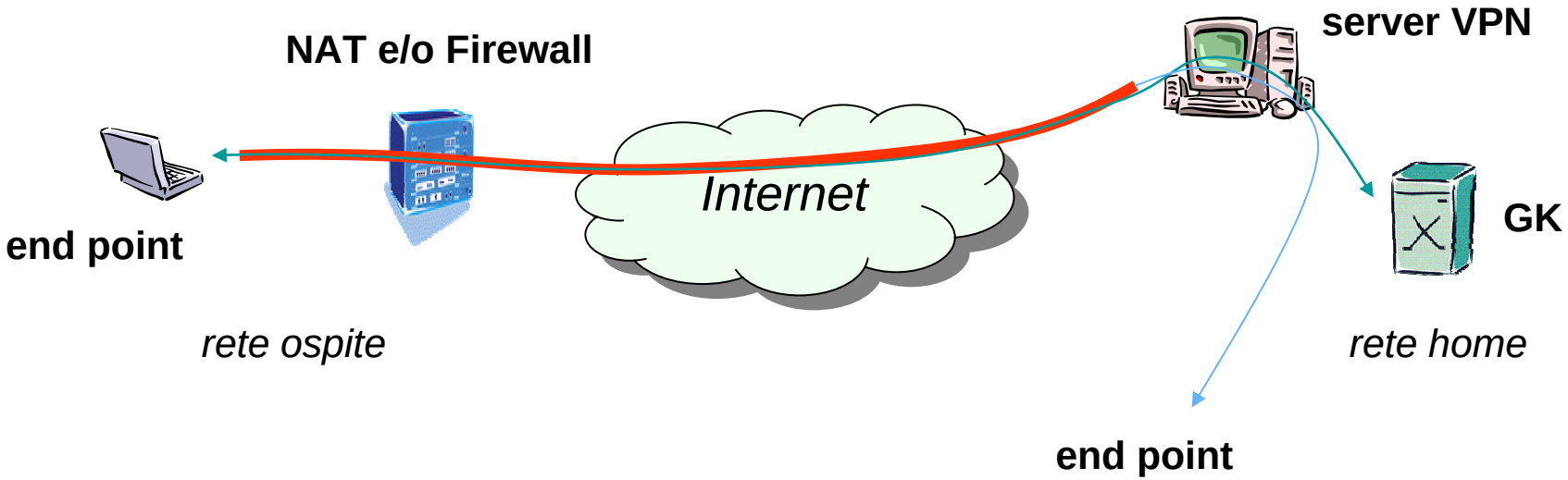
- 2 gnugk: uno nella zona pubblica e uno in quella privata

- Il NAT è attraversato dal solo traffico fra i due gnugk
- Il gnugk interno è child del gnugk esterno
- Il gnugk esterno è configurato per accettare Natted end-points
- Occorre configurare opportunamente il NAT per il forwarding delle porte



Soluzione con VPN

- Nel caso in cui un utente mobile si trovi al di fuori della propria rete (in cui risiede il proprio GK) e possibile utilizzare un tunnel (VPN) per superare eventuali NAT o Firewall.
- Limiti
  - La rete ospite potrebbe non consentire l'apertura di tunnel per VPN
  - Se il tunnel è basato su TCP, questo potrebbe avere effetti dannosi sulla qualità audio (TCP usa gli ACK, inutili per l'audio/video)



Implementazione Gatekeeper: **gnugk**

gnugk

- OpenH323 Gatekeper – The GNU Gatekeeper (in breve gnugk) è un progetto open-source che implementa un gatekeeper H.323
- E’ basato sullo stack protocol OpenH323 <http://sourceforge.net/projects/openh323>
- E’ disponibile già compilato per Linux, Windows, FreeBSD, Solaris e MacOS X

- Download, documentazione, mailing-list, etc.:
  - <http://www.gnugk.org/>

gnugk

- può girare su Windows come servizio
- supporta accounting e autorizzazione via SQL database, Radius, file o applicazioni esterne
- supporta un call routing flessibile
- number rewriting (calling and called)
- supporta NAT traversal
- può funzionare come full H.323 proxy
- supporto a gatekeeper clustering (neighbors, parent/child, alternates)
- H.235 security

Compilazione di gnugk

- Se non si vuole utilizzare la versione già compilata, per compilare su Linux occorre:
  - scaricare dal sito http://sourceforge.net/projects/openh323 le seguenti librerie:
    - PWLib (release + debug version)
    - OpenH323
  - copiare i file tar.gz delle librerie nella home directory dell'utente con cui ci si è connessi all'host Linux, e compilarle nell'ordine indicato sopra, seguendo le rispettive istruzioni.
  - compilare quindi gnugk con i comandi:
    - **`configure`**
      - se si desidera il supporto a radius o un database occorre specificarlo con le opzioni di configure (la versione pre-compilata ha già il supporto a radius e database)
    - **`make opt`**
  - per lanciare gnugk:
    - copiare obj_linux_x86_r/gnugk in /usr/sbin/ e creare un file di configurazione in /etc/gnugk.ini
    - **`$ /usr/sbin/gnugk –c /etc/gnugk.ini –o var/log/gnugk.log -ttt`**

Configurazione di gnugk

- La configurazione è suddivisa in sezioni:
  - [Nome Sezione]
  - comandi sezione

- Segue una breve analisi delle principali sezioni (per le altre si rimanda alla documentazione ufficiale; esempi di configurazione sono riportati nel seguito di queste slides)
- [Gatekeeper::Main]
  - contiene le configurazioni a carattere globale
    - Name=gk_nomesede
    - .....
- [RoutedMode]
  - Definisce se gnugk lavora in routed mode (e se anche per h245); definisce anche quali porte vengono utilizzate per i vari messaggi
    - GKRouted=1
    - H245Routed=1
    - CallSignalPort=1721
    - ....
- [Proxy]
  - Definisce se gnugk lavora in proxy mode; definendo eventualmente le porte UDP per il traffico multimediale
    - Enable=1
    - RTPPortRange=50000-59999
    - ......

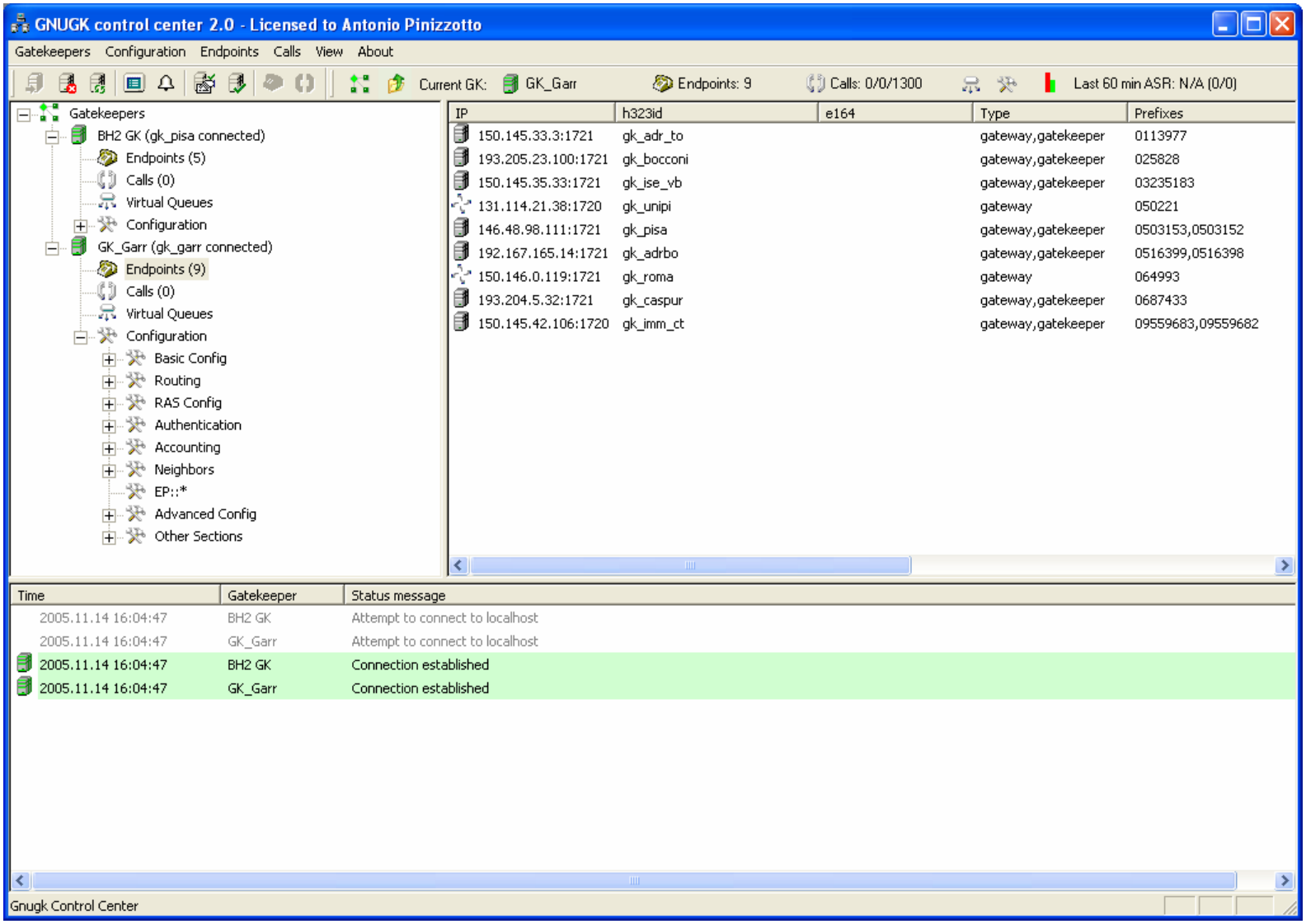
Configurazione di gnugk

- [Endpoint]
  - Definisce se gnugk si registra su un altro gatekeeper come child consentendo così di creare una gerarchia di gatekeeper
    - Gatekeeper=192.168.33.22
    - Type=Gateway
    - H323ID=gk_nomesede
    - Prefix=0987653, 0987654
- [Endpoint::RewriteE164]
  - Definisce evenutali manipolazioni dei numeri chiamante/chiamato per le chiamate che transitano fra questo gnugk (child) ed il gnugk father
    - 0987653=3
    - 0987654=4
    - sintassi: **numero esterno = numero interno**; definisce le regole di manipolazione dei numeri delle chiamate fra esterno e interno e viceversa; ovvero, cosa va tolto dal numero chiamato per le chiamate entranti, e cosa va aggiunto al numero chiamante per le chiamate uscenti
- [GkStatus::Auth]
  - Per motivi di sicurezza si consiglia di inserire:
    - rule=regex
    - regex=^(127\.0\.0\.1)$
    - In questo modo non è possibile connettersi alla console se non dalla stessa macchina a cui magari si è connessi in ssh. Se però si utilizza gnugk-cc (descritto nel seguito) occorre abilitare l'indirizzo IP su cui gira gnugk-cc

Monitoraggio e gestione di gnugk

- console interattiva (minimale):
  - telnet localhost 7000
  - log dei messaggi relativi a registrazioni o chiamate
  - Alcuni comandi:
    - ? o ??: fornisce l'elenco dei client/gateway registrati
    - help: fornisce un elenco dei comandi
    - reload: rilegge la configurazione in maniera soft (senza restart)
- gnugk-cc (gnugk control center) <http://www.gnugk-cc.org>
  - ottima interfaccia grafica per il controllo remoto di più gnugk
    - sia monitoraggio dettagliato che configurazione

gnugk control center



Implementazione SIP Proxy:
ser

ser

- SIP Express Router (SER) è un server VoIP gratuito basato su protocollo SIP.
- E’ realizzato da IPTEL.ORG: <http://www.iptel.org>
- SER può funzionare come SIP registrar, proxy o redirect server.
- Tra le varie funzioni supporta RADIUS/syslog accounting e authorization, server status monitoring.

Compilazione e installazione di ser

- SER è un VoIP server modulare. Pertanto occorre compilare oltre a SER anche un certo numero di moduli
- La procedura di compilazione può richiedere un minimo di attenzione a causa delle dipendenze dei moduli da altri moduli
- Se si installa su Linux Debian la procedura risulta più semplice:
  - Nel file /etc/apt/sources.list aggiungere le righe:

```
deb http://apt.sip-router.org/debian/ stable main contrib non-free
deb-src http://apt.sip-router.org/debian/ stable main contrib non-free
```

  - Quindi seguire le istruzioni riportate su
    - http://www.iptel.org/ser/ al link INSTALL

Configurazione di SER

- La funzione principale di ogni SIP server è quella dell'instradamento delle richieste, che ne determina il next-hop.
- La logica può essere complessa dovendo tener conto di route statiche a gateway verso PSTN, route dinamiche ad utenti registrati, politiche di autenticazione, etc.
- Per questo SER utilizza un "linguaggio di routing" che consente di definire la logica di route in maniera molto dettagliata.
- La sintassi prevede costrutti per azioni che possono essere eseguito sulla base di condizioni logiche i cui argomenti possono essere anche delle regular expression.
- Es.:

```
if (uri=~"sip:0[0-9]*@iptel.org") {
     forward(192.168.99.3, 5080);
}
```

- La sintassi è molto ricca e prevede l'uso di molti operatori, operandi, etc. Può essere paragonato ad un linguaggio di programmazione che una volta appreso si rivela molto potente.
- Le espressioni utilizzate nelle azioni e nelle istruzioni condizionali fanno uso di moduli. I moduli vanno dichiarati nel file di configurazione prima di poter essere usati. Inoltre è possibile scrivere dei nuovi moduli a seconda delle esigenze.

Esempio di configurazione per ENUM

```
.......................
# Load enum module
loadmodule "/usr/lib/ser/modules/enum.so"
.......................
# -- enum params --
modparam("enum", "domain_suffix", "e164.namex.it.")
.......................
# attempt to use ENUM
if (uri=~"^sip:\+396[0-9]+@") {
     log("Using ENUM\n");
            if (enum_query()) {
            if (uri!=myself || uri!=~"[2-3][0-9]+@.*$") {
                        append_hf("P-hint: outbound ENUM\r\n");
                        route(1);
                        break;
            };
     };
};
.......................
route[1]
{
     # send it out now; use stateful forwarding as it works reliably
     # even for UDP2TCP
     if (!t_relay()) {
            sl_reply_error();
     };
}
.......................
```

Asterisk PBX

Asterisk & Linux

- Supporto hardware PSTN: solo per linux
- Su altri sistemi principalmente limitato a VoIP
  - Porting per FreeBSD, OpenBSD and OS X in corso (esistono versioni a pagamento)

Architetura di Asterisk

- Asterisk è un software PBX
  - Canale: È un canale di comunicazione tra un sorgente esterna (telefono, VoIP provider, telefono IP …) ed asterisk
  - Molti canali hanno uno o più file specifici di configurazione
  - Ciascuna chiamata concorrente ha un proprio canale
    - le conversazioni telefoniche tra due punti terminali passano tramite asterisk: trans-codifica di voce, codecs e conversioni
  - Protocolli
      - SIP, IAX, H.323, Skinny, GCMP, CAPI (Common Application Interface)
  - Instradamento delle chiamate (extensions.conf)

Architetura di Asterisk (2)

- Software Asterisk è modulare
  - Asterisk (PBX e Channels, dialplan)
  - Zaptel (driver per hardware Digium)
  - Libpri (ISDN PRI Drivers for Zaptel)
- Zaptel Hardware: prodotti da Digium
  - X100P e X101P scheda PCI (solo FXO)
  - S100U (FXS via USB port)
  - TDM400P PCI (4 porte FXS o FXO)
  - T100P/E100P (una porta PRI T1 oppure E1)
  - T400P, E400P, TE410P e TE405P (4 porte PRI)
- Componenti opzionali
  - Soft phones
  - Management tools
  - Hardware (altre ditte)

Architetura di Asterisk (3)

- Applicazioni
  - Usate per manipolare le chiamate e dare interattività

  - Legacy PBX/IVR services
  - Voice-over Internet Protocol (VoIP) services
  - Complex IVR Trees
  - “Meet-me-Bridge” conferencing
  - VoIP Gateways (supports SIP, H.323 and IAX)
  - Calling Card Platforms
  - Voice/Data Router (replace expensive routers)

Installazione di Asterisk come H.323 - SIP gateway

- Provato su RedHat, Fedora, Debian, Mandrake e Gentoo:
  - supporto per linux kernel >= 2.4
    - 2.4 consigliato
    - 2.6 installato da noi ed è stabile
- Molti protocolli VOIP sono supportati per default:
  - SIP, IAX, skinny, mgcp
- Asterisk ha bisogno di driver esterni:
  - Zaptel se si usano schede digium
  - Libpri se si usano schede PRI T1/E1
  - OpenH323 e Pwlib se si usa il canale h323

Scaricare Asterisk

- CVS
  - Una modifica applicata è subito disponibile
  - Modifiche introdotte possono introdurre altri problemi
  - Quindi scaricare l'ultima funzionante

```
#!/bin/bash

export CVSROOT=:pserver:anoncvs@cvs.digium.com:/usr/cvsroot
echo "Please use anoncvs as password."
cvs login
cvs checkout zaptel libpri asterisk
```

- Zaptel serve se si ha scheda Digium mentre Libpri se si ha una scheda PRI ISDN

Open H.323 Channel Driver per Asterisk

- Non è compilato per default
- Ha bisogno di due package esterni:
  - Open H.323 version v1.17.1, PWLib v1.9.0 e GCC v3.2.2.
- Dipendenze neccesarie del canale asterisk h323: openssl-0.9.6b+, openssl-devel-0.9.6b+, expat-1.95+, expat-dev-1.95+
  - Asicurarsi che questi package siano installati precedentemente

Compilazione di asterisk con supporto di H.323

- È importante installare le versioni esatte delle librerie consigliate per fare funzionare h.323
  - Il file README sotto la directory /asterisk/channels/h323, che indica le versioni delle librerie di PWLIB e OPENH323 da installare prima di poter compilare asterisk reperibili dal sito: http://sourceforge.net/project/openh323
    - le versioni da noi provate e funzionanti con il nostro asterisk CVS sono PWLib v1.9.0 e Open H.323 version v1.17.1
- Scaricare le librerie in formato tar.gz e posizionate sotto la directory /usr/src e si procede con l'installazione nel seguente ordine:
  - pwlib, openh323 ed infine asterisk

Compilazione di asterisk con supporto di H.323 (2)

- dobbiamo settare le variabili d'ambiente, esempio per bash

  PWLIBDIR=/usr/src/pwlib_v1_9_0

  export PWLIBDIR

  OPENH323DIR=/usr/src/openh323_v1_17_1

  export OPENH323DIR
  LD_LIBRARY_PATH=$PWLIBDIR/lib:$OPENH323DIR/lib

  export LD_LIBRARY_PATH

- cd $PWLIBDIR
  - ./configure
  - make
- cd $OPENH323DIR
  - ./configure
  - make opt

Compilazione di asterisk con supporto di H.323 (3)

- Se vogliamo utilizzare il protocollo H.323, dobbiamo abilitare il modulo apposito sotto la directory
  - cd /usr/src/asterisk/channels/h323
  - make all (dovrebbe andare a buon fine, se non va bisogna trovare un modulo h323 funzionante… a volte fallisce)
    - Una versione da noi usata è quella del 15 giugno 2005
- Se tutto va a buon fine, a questo punto, si passa alla directory principale di asterisk /usr/src/asterisk e procediamo con l'installazione definitiva con i comandi:
  - make; make install; make samples

/etc/asterisk/h323.conf

```
;-----------Inizio file h323.conf------------------
; The NuFone Network's
; Open H.323 driver configuration
;
[general]
port = 1720
bindaddr = X.X.X.X ;IP address di asterisk non lasciare 0.0.0.0
disallow=all
allow=ulaw
allow=alaw
language=it
gatekeeper = X.X.X.X ; IP address del tuo GK
AllowGKRouted = yes
;
;esempio di un Softphone con interno 5555. Se si hanno piu softphone
;aggiungere altre entry simili per ciascun softphone. Comunque  nel caso
;in cui si abbia un GK su cui registrare normalmente i softphone
;queste entry non ci dovrebbero essere.
;
;[5555]
;type=friend
;host=X.X.X.X ;IP address dell'host in cui e` installato il softphone
;context=default
;
[asterisk]
type=h323 ; registra I prefissi
prefix=YYY,ZZZ ; dove YYY e ZZZ sono i prefissi gestiti da asterisk
FastStart=no
context=default
;-----------Fine file h323.conf------------------
```

/etc/asterisk/sip.conf

```
;-----------Inizio file sip.conf------------------
;
; SIP Configuration example for Asterisk
;

[general]
context=default              ; Default context for incoming calls
bindport=5060                ; UDP Port to bind to (SIP standard port is
        5060)
bindaddr=X.X.X.X             ; IP address to bind to (0.0.0.0 binds to all)
srvlookup=yes                ; Enable DNS SRV lookups on outbound calls
disallow=all                 ; First disallow all codecs
allow=ulaw
allow=alaw                   ; Allow codecs in order of preference
;---------------------------------------------------------------------------
```

```
; Definitions of locally connected SIP phones
;
; type = user   a device that calls us
; type = peer   a device we place calls to
; type = friend two configurations (peer+user) in one

[2222]
type=friend
username=abraham
secret=zzzz ; password del client 2222 per registrarsi ad asterisk
host=dynamic
context=default
[3333]
type=friend
username=claudio
secret=zzzzzzzz
host=dynamic
context=default
[4444]
type=friend
username=marco
secret=zzzzzzzzzz
context=default
host=dynamic

;-----------Fine file sip.conf------------------
```

/etc/asterisk/extensions.conf

```
;-----------Inizio file extensions.conf------------------
[general]
static=yes
writeprotect=no

[default]
exten => 2222,1,Dial(SIP/2222,15) ; chiama il softphone 2222
exten => 3333,1,Dial(SIP/3333,15)
exten => 4444,1,Dial(SIP/4444,15)
;exten => 5555,1,Dial(H323/5555) : chiama estenzione 5555
exten => _X.,1,Dial(H323/${EXTEN}) ; una regola che dice chiamate verso
;numeri sconoscuti mandali a GK

;-----------Fine file extensions.conf------------------
```

Fine installazione GW H.323 - SIP

Installazione Hardware

- IRQ
  - Le schede digium tendono a generare molti IRQ
  - Fare assegnazione manuale dal BIOS specifico per la scheda non condivisa da altri
  - cat /proc/interrupts
- Una volta risolto il problem del IRQ:
  - Installare i driver zaptel per le schede

Schede PRI Digium

- Esistono 3 tipi di schede PCI: universale, 3 volt e 5 volt.
- per asterisk ci solo due tipi di schede PRI PCI:

  - da 3 volt oppure da 5 volt
- verificare il bus PCI e comprare la scheda adata. Di solito in Italia sono più diffuse quelle da 5 volt, ma per noi al CNR di Pisa per esempio serviva quella da 3 volt.

Compatibilità Asterisk <-> PBX

- Verifica se esiste un attacco RJ45 sul lato del centralino. In tal caso è possibile costruire un cavo crociato come segue:
  - RJ45 PRI Cross Over Cable
    - 2------5
    - 1------4
    - 5------2
    - 4------1
- Altrimenti se l'attacco del centralino è di tipo proprietario, diverso da RJ45, bisogna dotarsi di un cavo dal fornitore che abbia un lato di tipo RJ45.

Compatibilità asterisk <-> PBX

- asterisk supporta molti tipi di ISDN switch type. Questo è configurabile via software quindi basta verificare che il centralino supporti almeno uno di questi tipi:
  - 4ESS
  - DMS100
  - EuroISDN
  - Lucent 5E
  - National ISDN2
  - NFAS
- qsig e altri tipi di switch sono stati recentemente inclusi nella lista, da verificare sul sito di Digium.

Scheda PRI TE410P/TE405P DIGIUM

PCI
The TE410P is for use only with a 3.3 volt PCI slot.
The TE405P is for use only with a ćnormalĻ 5volt PCI slot.

TE410P / TE405P (caratteristiche della scheda)

- svolge più attività in hardware con conseguente riduzione di utilizzo della CPU e aumento della densità di schede sul server.

- **PRI Switch Compatibility** Network or CPE (customer premise equipment ????)
  - EuroISDN (PRI or PRA) — Q.931/Q.921
  - AT&T 4ESS
  - DMS 100
  - Lucent 5E
  - National ISDN 2
- **CAS Voice Modes**
  - Feature Group D
  - E&M Wink
  - A-Law, Mu-Law, and Linear Modes Supported
- **Data Modes**
  - SyncPPP (both Fixed and Dialup)
  - Frame Relay
  - Cisco HDLC
  - Multi-link PPP

Installazione scheda PRI TE410P DIGIUM

- Ipotiziamo di volere aggiungere al PBX asterisk una scheda PRI, al fine di poter allacciare la nostra rete VoIP con la rete telefonica ISDN tradizionale avendo a disposizione:
  - scheda 4 porte PRI (TE410P compatibile con Asterisk)
    - PCI slot da 3.3 Volts (esiste anche versione da 5 volt)
    - porte configurabili per supportare E1/T1, tramite alcuni jumpers, selezionabili per porta
  - BIOS che permette di assegnare manualmente l'IRQ
  - PWLIB e OpenH323 gia installate al fine di poter utilizzare il protocollo H.323
  - librerie Libpri e Zaptel (scaricati dal sito digium)

Installazione scheda PRI TE410P DIGIUM (2)

- Si configurano i jumpers per abilitare la scheda a funzionare secondo modalità E1
- si alloggia la scheda nello slot PCI (da 3,3 Volt)
  - Al riavvio del PC, assegnare un IRQ unico alla scheda digium
  - Riavviato il sistema dopo aver installato la scheda notiamo i 4 led lampeggiare con luce rossa, in maniera intermittente e con frequenza elevata (il software di plug and play di linux: kudzu lo rileva e chiede di configurarlo, rispondendo si).
- Installare i moduli necessari all'utilizzo della scheda da parte del Asterisk PBX
  - cd /usr/src/zaptel
    - Tale driver serve per interfacciare la scheda al sistema operativo linux
    - more README (indica quale modulo caricare per i vari tipi di schede; il nostro caso *wct4xxp*)
  - Procediamo quindi alla compilazione del modulo zaptel seguente modo:
    - make
    - make install
  - A questo punto i led delle schede sono spenti e viene creato il files zaptel.conf (file di configurazione Digium) sotto la directory /etc
    - modprobe wct4xxp
    - ztcfg (legge il file di configurazione /etc/zaptel.conf)
  - A questo punto i led sono rossi e lampeggiano lentamente
  - Se colleghiamo una porta della scheda con la linea ISDN del provider, che fornisce il servizio, il led della porta corrispondente diventa verde

Installazione scheda PRI TE410P DIGIUM (3)

- *lsmod*, verifichiamo il caricamento del driver wct4xxp usato dal modulo zaptel.
- *cat /proc/interrupts* visualizziamo il numero di IRQ assegnati alla scheda PRI.
- Ci spostiamo sotto la directory /usr/src/libpri
- Tale libreria verrà usata da asterisk per interfacciarsi con la scheda digium
  - ed eseguiamo i comandi:
    - make
    - make install
    - cd..

  A questo punto si procede con l'installazione di asterisk come visto precedentemente (esempio: pwlib, openh323, canale h323 ed infine asterisk)

Installazione di Asterisk con scheda DIGIUM

- entriamo nella directory /usr/src/asterisk
  - make clean
- entriamo nella directory /usr/src/asterisk/channels/h323
  - make clean
  - make all
- torniamo nella directory /usr/src/asterisk
  - make
  - make install
  - make samples (solo alla prima installazione)
    - Configurare il file /etc/zaptel.conf ed /etc/asterisk/zapata.conf
- **N.B.** Quest'ultima direttiva crea dei files di configurazione di asterisk, rinominando quelli già presenti come .old (es. extensions.conf.old, sip.conf.old, etc).

Fine installazione scheda PRI TE410P

Installazione Schede Junghanns quadBRI (non Digium)

- Digium non produce schede ISDN BRI
- Configurare i jumpers per la scheda:
- Ipotesi: i jumpers sono settati come
  - TE no power supply, power feeding 3 volt, S/T interface on (scaricati lo user guide per le schede Junghanns, è spiegato bene ...)
  - (È possibile anche configurare la scheda come NT),
- La scheda ha bisogno di un suo IRQ (questo in generale per tutte le schede ISDN)
- Nel caso del CNR di Pisa, IRQ 3 settato dal BIOS (Il PC deve essere in grado di assegnare IRQ dal BIOS)
  - Per esempio alla UNIPI la scheda PRI condivideva un IRQ con altre due schede gigaEthernet e come conseguenza la scheda PRI si riavviava spesso rendendolo quasi inutilizzabile al crescere del'utenza. Quindi fare questa verifica quando si scegle il PC.

Schede Junghanns quadBRI

- Andare sul sito:
  - http://www.junghanns.net/en/download.html
  - Scaricare bristuff-0.2.0-RC8f-CVS.tar.gz
    - Si trova sotto BRIstuff for CVS-HEAD experimental version of BRIstuff CVS (la versione STABLE non funziona il modulo h323).
  - Tar xvfz bristuff-0.2.0-RC8f-CVS.tar.gz
  - Cd bristuff-0.2.0-RC8f-CVS
- serve il sorgente del kernel per compilare il driver ISDN
  - per il sorgente del kernel 2.6 installato con Fedora core 4, valgono i seguenti comandi
    - Creare un link simbolico sotto /usr/src/
      - linux-2.6 -> ./kernels/2.6.11-1.1369_FC4-i686/
    - Creare un altro link simbolico sotto /usr/src/linux
      - include -> /usr/src/kernels/2.6.11-1.1369_FC4-i686/include/

Schede Junghanns quadBRI

- Compilare il modulo ISDN come segue:
  - cd qozap
  - make clean all
  - make install (DA NON FARE!!!)
- Caricare i moduli neccesari come segue:
  - Cd qozap
  - Modprobe zaptel
  - Insmode qozap.o (for kernel 2.4)
  - Insmod qozap.ko (for kernel 2.6)
  - Ztcfg (ATTEN se uno aveva gia schede DIGIUM oppure tutte le volte successive usare l'opzione -s ---> ztcfg -s)
- Se tutto ha funzionato bene attaccando l' ISDN uscita borchia di tipo S i led devono diventare VERDI.

Schede Junghanns quadBRI

- Torna una directory sopra: cd .. (vedi slides precedenti)
- Lanciare download.sh (che applica i patch per le schede junghanns) che ha come contenuto qualcosa di simile ...

```
************************************************************
#!/bin/bash
Export CVSROOT=:pserver:anoncvs@cvs.digium.com:/usr/cvsroot
echo "Please use anoncvs as password."
cvs login
cvs co -D 05/29/05 zaptel
cvs co -D 05/29/05 libpri
cvs co -D 05/29/05 asterisk
cd zaptel
patch -p1 < ../patches/zaptel.patch
cd ..
cd libpri
patch -p1 < ../patches/libpri.patch
cd ..
cd asterisk
patch -p1 < ../patches/asterisk.patch
cd ..
echo "******************************************************"
echo "        Downloading and patching finished."
echo "******************************************************"
```

Schede Junghanns quadBRI

- NON lanciare compile.sh
- Fare a mano come segue:

  cd zaptel

  make clean all

  make install

  cd ..

  cd libpri

  make clean all

  make install

  cd ..

- Quelli sopra si compilano bene ma ASTERISK per il supporto H323 ha bisogno di librerie esterne.
  - Per default asterisk non supporta h323 come detto precedentemente

Moduli Kernel Linux: Schede Junghanns quadBRI

- Asicurarsi che il sorgente del kernel sia installato e configurato!
- Per default se si esegue “make clean; make; make install” sono installati moduli in più che non servono, per levarlo:
  - rmmod hfc4s8s_l1, hisax, crc_ccitt, isdn, slhc
- Per insttalare solo quelli neccessari
  - modprobe zaptel, insmod qozap.ko (kernel 2.6 oppure insmod qozap.o per kernel 2.4), ztcfg (oppure ztcfg -c se non è la prima volta)
- lsmod (per listare i moduli installati)

```
Module                  Size  Used by
qozap                  20120  12
zaptel                211204  31 qozap
crc_ccitt              2113  1 zaptel
```

Moduli kernel Linux al Boot

- /etc/rc.local

  /sbin/modprobe zaptel

  /sbin/insmod /usr/src/bristuff-0.2.0-RC8f-CVS/qozap/qozap.ko

  /sbin/ztcfg

  /usr/sbin/asterisk

- Se ci sono moduli non desiderati che partono automaticamente commentare le righe corrispondenti a tali moduli, esempio:
  - cd /lib/modules/2.6.11-1.1369_FC4/
  - Editare il file corrispondente

Compilare Asterisk

- cd /usr/src/zaptel
  make clean
  make install
- cd /usr/src/libpri
  make clean
  make install
- /usr/src/asterisk/channels/h323 (dopo aver installato Pwlib e OpenH323)
  make clean
  make
- cd /usr/src/asterisk
  make clean
  make install
- cd /usr/src/asterisk
  - make samples (making the samples)

Aggiornare Asterisk

- make update
  - Aggiorna Asterisk dal server CVS, lo compila ed installa
  - Fare ripartire asterisk
- Oppure conviene scaricare l'ultima CVS fare la compilazione a mano ed installare asterisk
- In tutte e due i casi:
  - NON c'e' bisogno di cambiare I file di configurazioni

Avviare asterisk

- /usr/sbin/asterisk
- Opzioni
  - c console
  - v modalità verbosa
  - d modalità di debug
  - g core dump al termine di asterisk
  - C <config file>: parte con un file di configurazione diverso da quello di default /etc/asterisk/asterisk.conf
  - r controlo remoto per collegarsi al CLI di un processo asterisk già attivo
  - n disabilitare i colori di CLI
- Le opzioni si possono concatenare:
  - asterisk -cvvvvd

Avviare asterisk usando safe_asterisk

- /usr/src/asterisk/contrib/scripts
  - safe_asterisk
    - Si puo parametrizzare
    - Per ogni crash invia un e-mail ??? Da correggere lo script??? ed automaticamente fa ripartire asterisk

- In caso di partenza al boot conviene usare lo script:
  - cp /usr/src/asterisk/contrib/init.drc.redhat.asterisk /etc/rc.d/init.d/asterisk
  - /sbin/chkconfig --add asterisk (aggiunge il link di startup su tutti i liveli di init)
    - Es: /etc/rc.d/rc5.d/S40asterisk

Esempio di utilizzo di applicazioni con asterisk

```
Le applicazioni si usano quasi sempre editando il file
/etc/asterisk/extensions.conf

-----

Esempio di chiamata verso un utente 3815 definito su asterisk apartenente ad
un contesto default.

exten => 3815,1,Set(LANGUAGE()=it)
exten => 3815,2,Dial(SIP/3815,10)
exten => 3815,3,Dial(H323/2127,20)
exten => 3815,4,Answer()
exten => 3815,5,Voicemail(u3815)
exten => 3815,6,Hangup()


-----------inizio /etc/asterisk/voicemail.conf------------------
;
; Voicemail Configuration
;
[general]
language=it
; Default formats for writing Voicemail
;format=g723sf|wav49|wav
format=wav49|gsm|wav
; Who the e-mail notification should appear to come from
serveremail=asterisk
```

Esempio di utilizzo di applicazioni con asterisk

```
; Should the email contain the voicemail as an attachment
attach=yes
; How many miliseconds to skip forward/back when rew/ff in message playback
skipms=3000
; How many seconds of silence before we end the recording
maxsilence=10
; Silence threshold (what we consider silence, the lower, the more
sensitive)
silencethreshold=128
; Max number of failed login attempts
maxlogins=3
sendvoicemail=yes       ; Context to Send voicemail from [option 5 from the
advanced menu]
                        ; if not listed, sending messages from inside
voicemail will not be
                        ; permitted
[default]
3816 => 4242,Abraham Gebrehiwot,Abraham.Gebrehiwot@iit.cnr.it
3815 => 4242,Marco Sommani,Marco.Sommani@iit.cnr.it

-----------fine /etc/asterisk/voicemail.conf-----------------
```

Esempio di utilizzo di applicazioni con asterisk

Esempio di enum lookup di tutti i numeri che cominciano con 396923

```
exten => _396923X.,1,EnumLookup(${EXTEN})
exten => _396923X.,2,Dial(${ENUM})
exten => _396923X.,3,Hangup
```

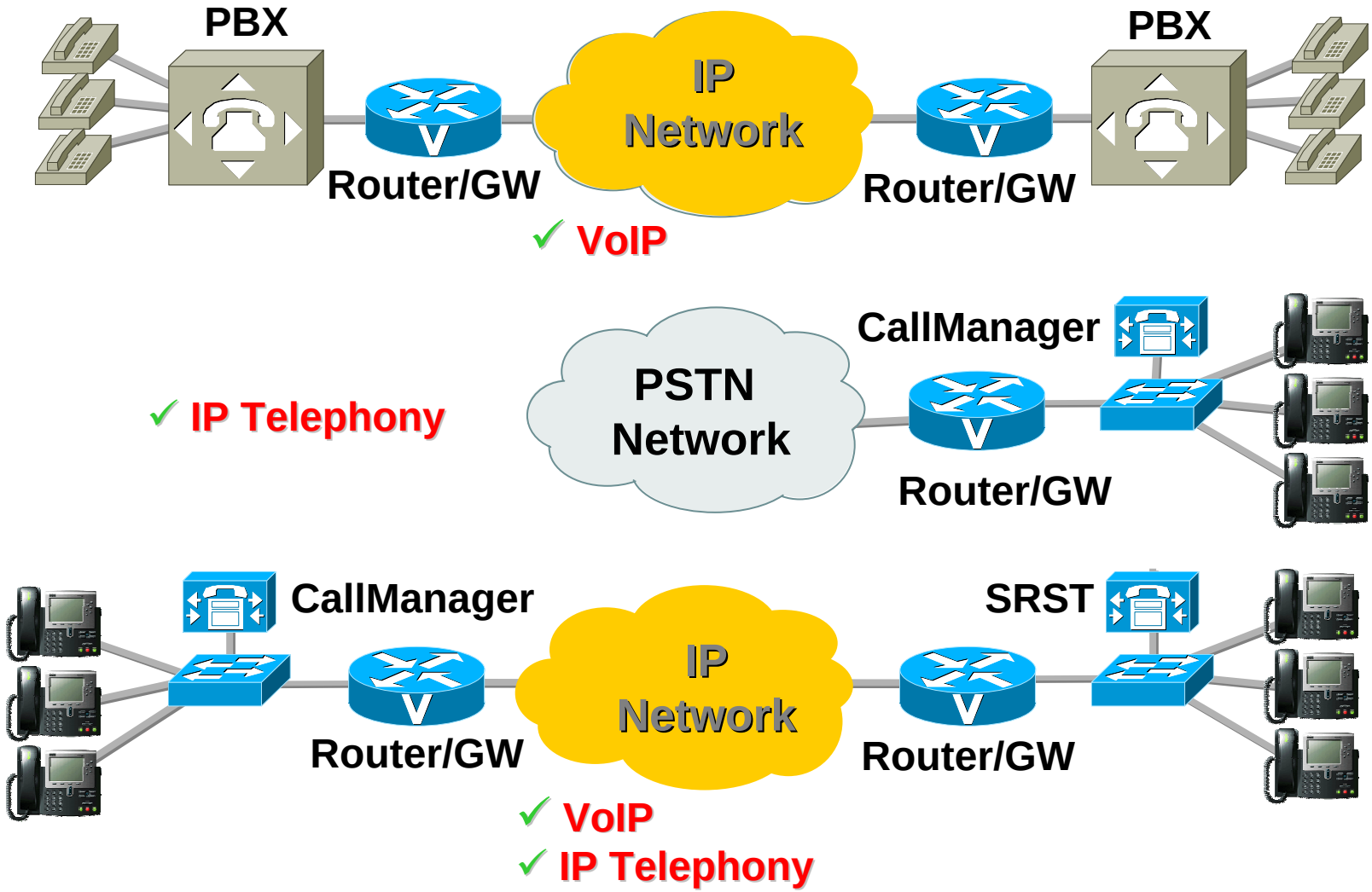
```
----------inizio /etc/asterisk/enum.conf----------
; ENUM Configuration for resolving phone numbers over DNS
; Sample config for Asterisk
; This file is reloaded at "reload enum" in the CLI
[general]
;
; The search list for domains may be customized.  Domains are searched
; in the order they are listed here.
;
search => e164.arpa
search => e164.namex.it
search => namex.e164.arpa
search => e164.org
search => enum.fierymoon.com
search => rfc2916.net
;
; If you'd like to use the E.164.org public ENUM registery in addition
; to the official e164.arpa one, uncomment the following line
;
;search => e164.org
; As there are more H323 drivers available you have to select to which
; drive a H323 URI will map. Default is "H323".
;
h323driver => H323
----------fine /etc/asterisk/enum.conf----------
```

codec per Asterisk

INT	BINARY	HEX	TYPE	NAME	DESC
1	(1 << 0)	(0x1)	audio	g723	(G.723.1)
2	(1 << 1)	(0x2)	audio	gsm	(GSM)
4	(1 << 2)	(0x4)	audio	ulaw	(G.711 u-law)
8	(1 << 3)	(0x8)	audio	alaw	(G.711 A-law)
16	(1 << 4)	(0x10)	audio	g726	(G.726)
32	(1 << 5)	(0x20)	audio	adpcm	(ADPCM)
64	(1 << 6)	(0x40)	audio	slin	(16 bit Signed Linear PCM)
128	(1 << 7)	(0x80)	audio	lpc10	(LPC10)
256	(1 << 8)	(0x100)	audio	g729	(G.729A)
512	(1 << 9)	(0x200)	audio	speex	(SpeeX)
1024	(1 << 10)	(0x400)	audio	ilbc	(iLBC)
65536	(1 << 16)	(0x10000)	image	jpeg	(JPEG image)
131072	(1 << 17)	(0x20000)	image	png	(PNG image)
262144	(1 << 18)	(0x40000)	video	h261	(H.261 Video)
524288	(1 << 19)	(0x80000)	video	h263	(H.263 Video)
1048576	(1 << 20)	(0x100000)	video	h263p	(H.263+ Video)

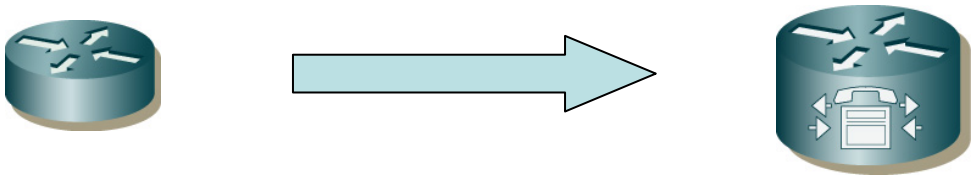
Soluzioni Cisco VoIP

Soluzioni Cisco: VoIP e IP Telephony



Funzionalità integrate nel router

- Gatekeeper
- Gateway (H.323, SIP, MGCP)
- IP PBX (Call Manager Express)
- Interactive Voice Response IVR



Funzionalità e router

Feature solo software dell'IOS:

- Gatekeeper
- Call Manager Express (CME)
- IVR

Feature HW e SW dell'IOS:

- Gateway
- Answer Machine

Gatekeeper

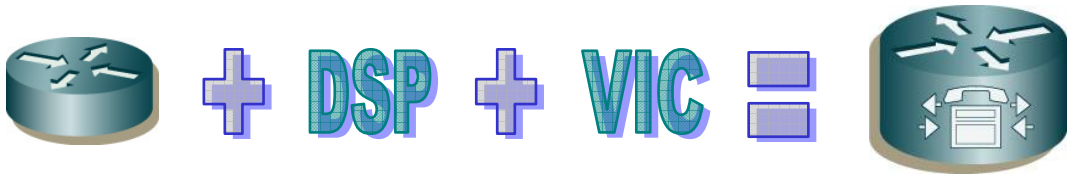
- Ultimi aggiornamenti versione 12.3.8T
- Piattaforme su cui “gira”: 2500, 2600, 2800, 3600, 3700, 3800
- Necessita della feature IP/H323, IP/MCM H323

Esempio di configurazione:

```
gatekeeper
 zone local gk_cosenza gk.isn.cnr.it 194.119.199.100
 zone remote gk_garr gk.iit.cnr.it 146.48.96.183 1719
 zone prefix gk_cosenza 09849801... gw-priority 10 voip.cs.cnr.it
 zone prefix gk_garr 901*
no shutdown
```

Gateway

- Ultimi aggiornamenti versione 12.4.5T
- Piattaforme su cui “gira”: 2600, 2800, 3600, 3700, 3800



Esempio Gateway 2 ISDN BRI:

•2600, 3600, 3700: Router + NM-2V + VIC-2BRI-S/T

•2800, 3800: Router + PVDM2-8 + VIC2-2BRI-S/T

IP PBX (CME) 1/2

- Funzionalità introdotta nella versione 12.1 con poche funzionalità
- Ultimi aggiornamenti versione 12.4.5T
- Piattaforme su cui “gira”: 1751, 2600XM, 2800, 3600, 3700, 3800
- Max 240 telefoni (Cisco 3845)
- Complete funzionalità di Call Processing
- Per soluzioni da 240 a 10.000 telefoni si passa a Call Manager su HW dedicato

IP PBX (CME) 2/2

Esempio di configurazione:

```
telephony-service
 load 7960-7940 P00303020214
 load 7905 flash:CP79050101SCCP030530B.
 max-ephones 10
 max-dn 20
 ip source-address 150.145.61.65 port 2000
 time-format 24
 date-format dd-mm-yy
 system message Benvenuto sul VoIP CNR-ISN
 user-locale IT
 network-locale IT
 create cnf-files version-stamp 7960 Ott 30 2005 13:28:40
 dialplan-pattern 1 5 extension-length 3
 mwi relay
 mwi expires 600
 max-conferences 2
 moh music-on-hold.au
 web admin system name prova password prova
 dn-webedit
 time-webedit
 directory last-name-first
 directory entry 1 301 name Duca, Ivan
```

```
!-DEFINIZIONE DEI NUMERI E.164 GESTITI
!
ephone-dn  1
 number 09849801310
 name Ivan Server Farm
 translate calling 901
 hold-alert 30 originator

!- BUTTON ASSEGNA I NUMERI E.164 AI TELEFONI
!- TYPE DEFINISCE IL TIPO DI TELEFONO (7960 / 7905 / ECC.)
!- MAC-ADDRESS ASSOCIA IN MANIERA STATICA LA DEFINIZIONE AL TELEFONO
!- SPEED-DIAL DEFINISCE I NUMERI BREVI
!
ephone  2
 username "ivan" password ivan
 mac-address 0007.5052.8457
 speed-dial 1 09849801301 label "Ivan Duca 2 VoIP"
 speed-dial 2 9010503152158 label "Lorenzo Rossi VoIP"
 speed-dial 3 9010503152127 label "Marco Sommani VoIP"
 speed-dial 4 09849801267 label "Ivan Studio"
 type 7960
 button  1:1
```

Soluzioni Innovaphone

(accenno)

- Specializzati in prodotti H.323
- Gateway/PBX
  - IP3000 (PRI)
  - IP400 (BRI)
  - .....
- Telefoni
  - IP200 TipTel
  - .....
- Gateway/PBX configurabili con interfaccia WEB
  - Configurazione molto flessibile e articolata
  - boot in meno di 10 sec
  - se inserito fra PBX e PSTN, i 2 PRI si “cortocircuitano” se manca la corrente
  - .....

Softphone gratuiti testati

- SJPhone <http://www.sjlabs.com>
  - client H.323 e SIP
  - solo audio
  - piattaforme: Windows, Linux, Mac OSX
- OhphoneX <http://xmeeting.sourceforge.net/>
  - client H.323
  - audio e video
  - piattaforma: Mac OSX
- MyPhone <http://myphone.sourceforge.net/>
  - client H.323
  - audio e video
  - piattaforma: Windows
- X-lite <http://www.xten.com/>
  - client SIP
  - audio (video nella versione a pagamento)
  - piattaforme: Windows e Mac OSX
- .....

Telefoni IP e Gateway testati

- Telefoni
  - Cisco IP Phone 7960, ...
  - videotelefoni IP H.323 (Aethra)
  - Innovaphone IP200, TipTel
  - stazione di videoconferenza FalconIP VCON
  - .....
- Gateway
  - Innovaphone IP3000
  - Innovaphone IP400
  - Cisco 2651XM + scheda VoIP NM-HDV-E1-30
  - Asterisk
  - .....

Utilizzo del GDS per l'inoltro delle chiamate VoIP nel GARR

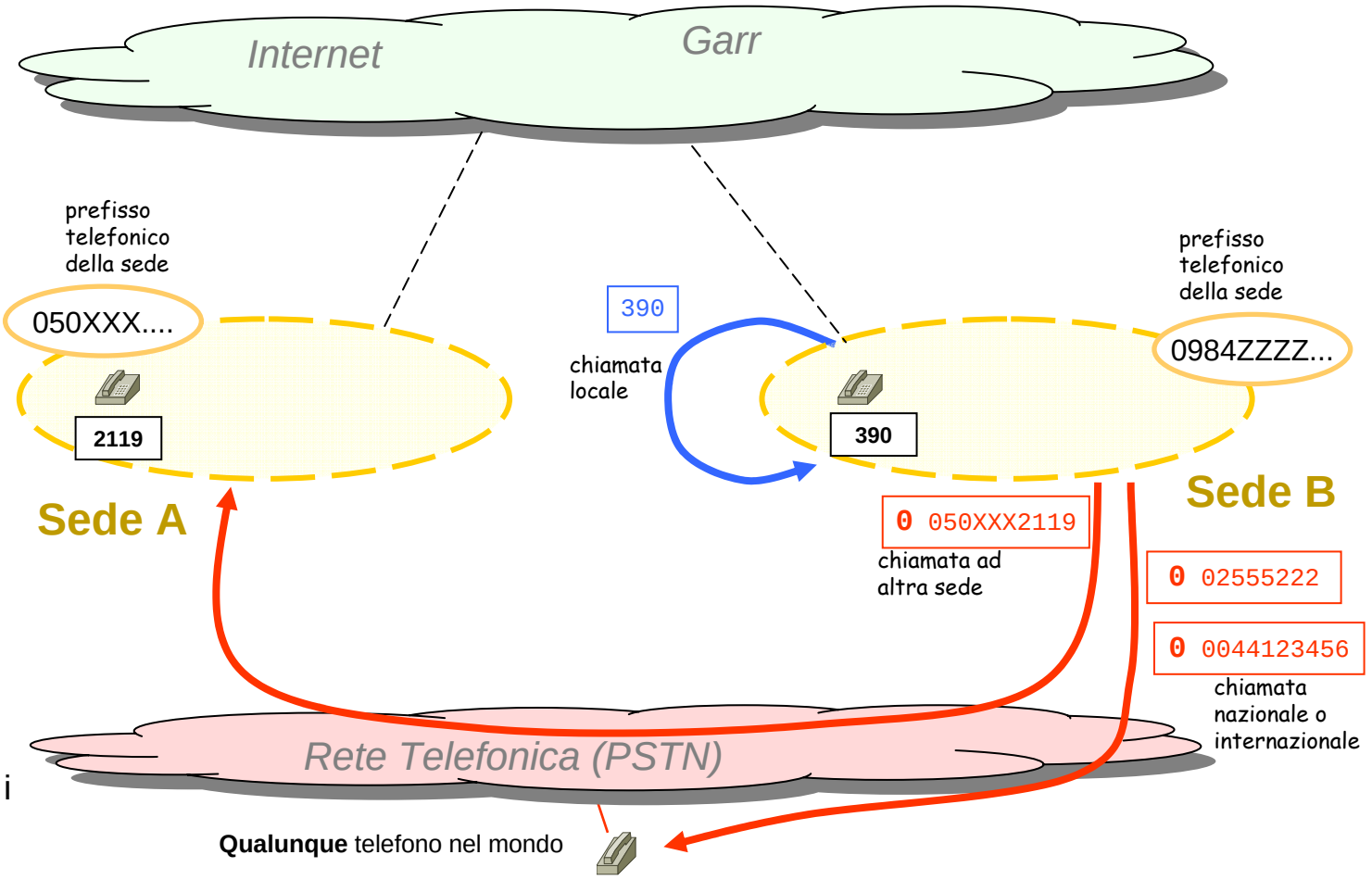
Uso del GDS per le chiamate VoIP nel GARR

- Obiettivi
  - Introduzione graduale della tecnologia VoIP tra le sedi afferenti al GARR integrandola con le infrastrutture telefoniche già esistenti
  - Introduzione di telefoni IP hardware e software integrati con il piano di numerazione già esistente
  - Facilità d'uso per l'utente finale
  - Doppia raggiungibilità: ogni telefono è raggiungibile con lo stesso numero sia via IP che via rete telefonica tradizionale
  - Apertura ai protocolli standard H.323 e SIP
  - Integrazione con il piano di numerazione nazionale esistente
  - Utilizzo di soluzioni OpenSource e multi-vendor
  - Mobilità
  - Scalabilità
  - Integrazione con il Global Dialling Scheme
- Scenario
  - Numero elevato di sedi per gli enti afferenti al GARR (solo il CNR conta circa 400 sedi)
  - Alcune sedi hanno già fatto scelte VoIP indipendenti
  - Gestione autonoma di ogni sede

Servizi aggiunti

- <u>Situazione iniziale</u>: sedi senza alcuna tecnologia VoIP

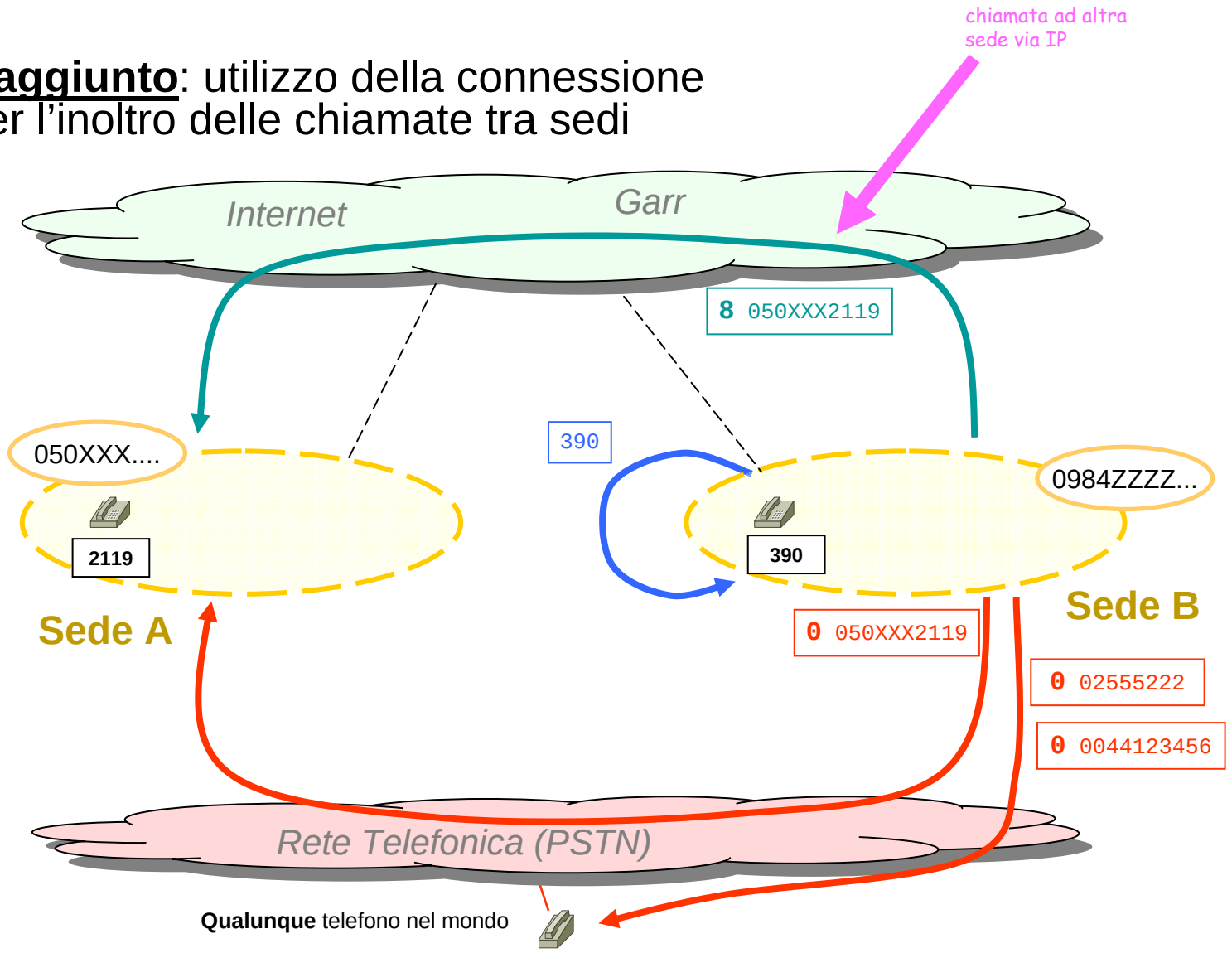
- Le chiamate tra sedi sono effettuate attraverso la PSTN anteponendo, ad es., lo 0 per chiamare attraverso il centralino.
- Le chiamate interne sono fatte con i numeri brevi attraverso il centralino locale
- La connessione a Internet è utilizzata solo per i dati



Servizi aggiunti

- **1. Servizio aggiunto**: utilizzo della connessione a Internet per l'inoltro delle chiamate tra sedi

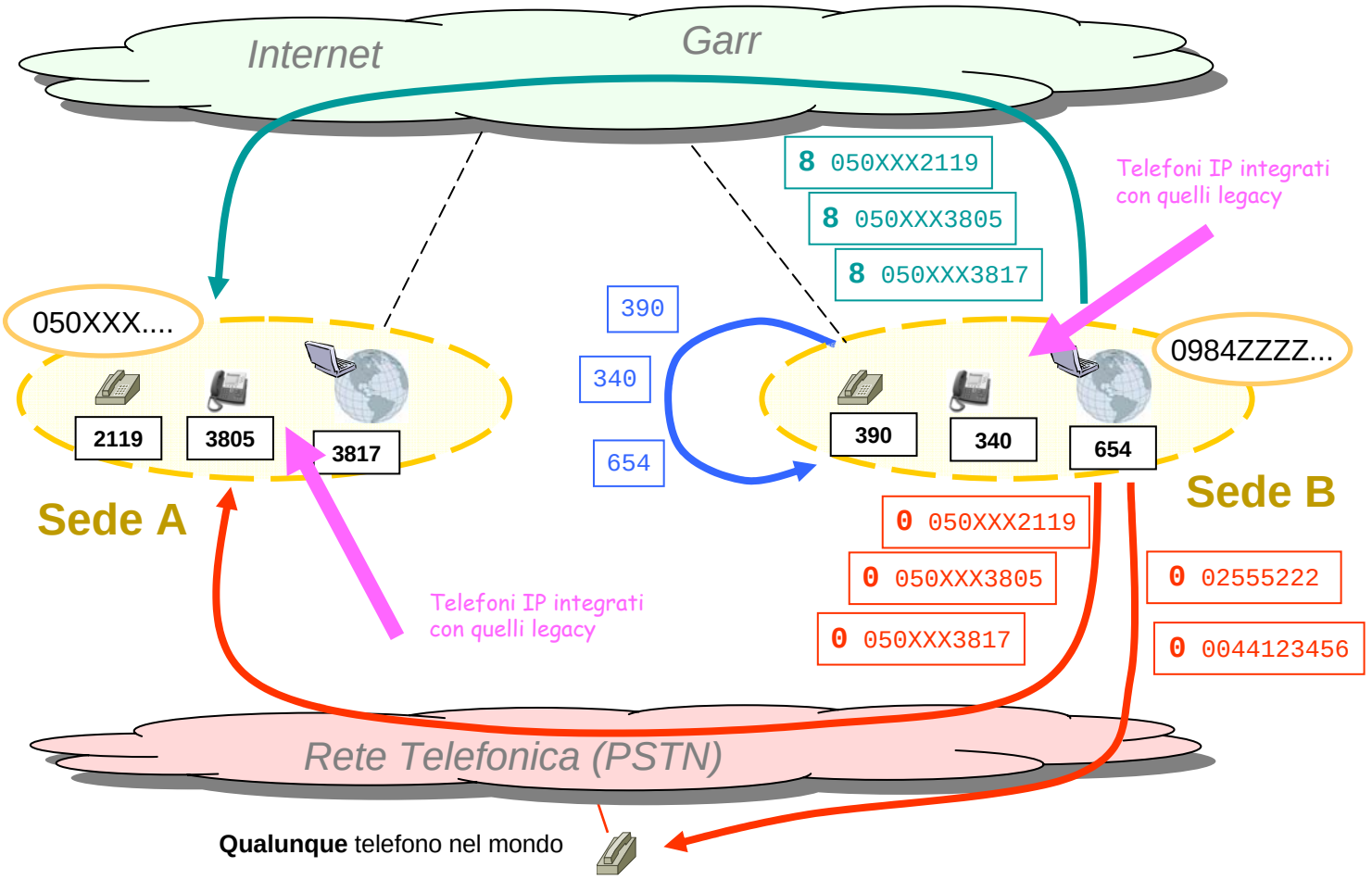
- Il numero da comporre è identico; cambia solo il prefisso da anteporre: ad esempio 0 per il centralino (PSTN) e 8 per VoIP. La scelta di questi numeri è indipendente per ogni sede.



Servizi aggiunti

- **2. Servizio aggiunto**: integrazione di telefoni IP da affiancare ai telefoni legacy

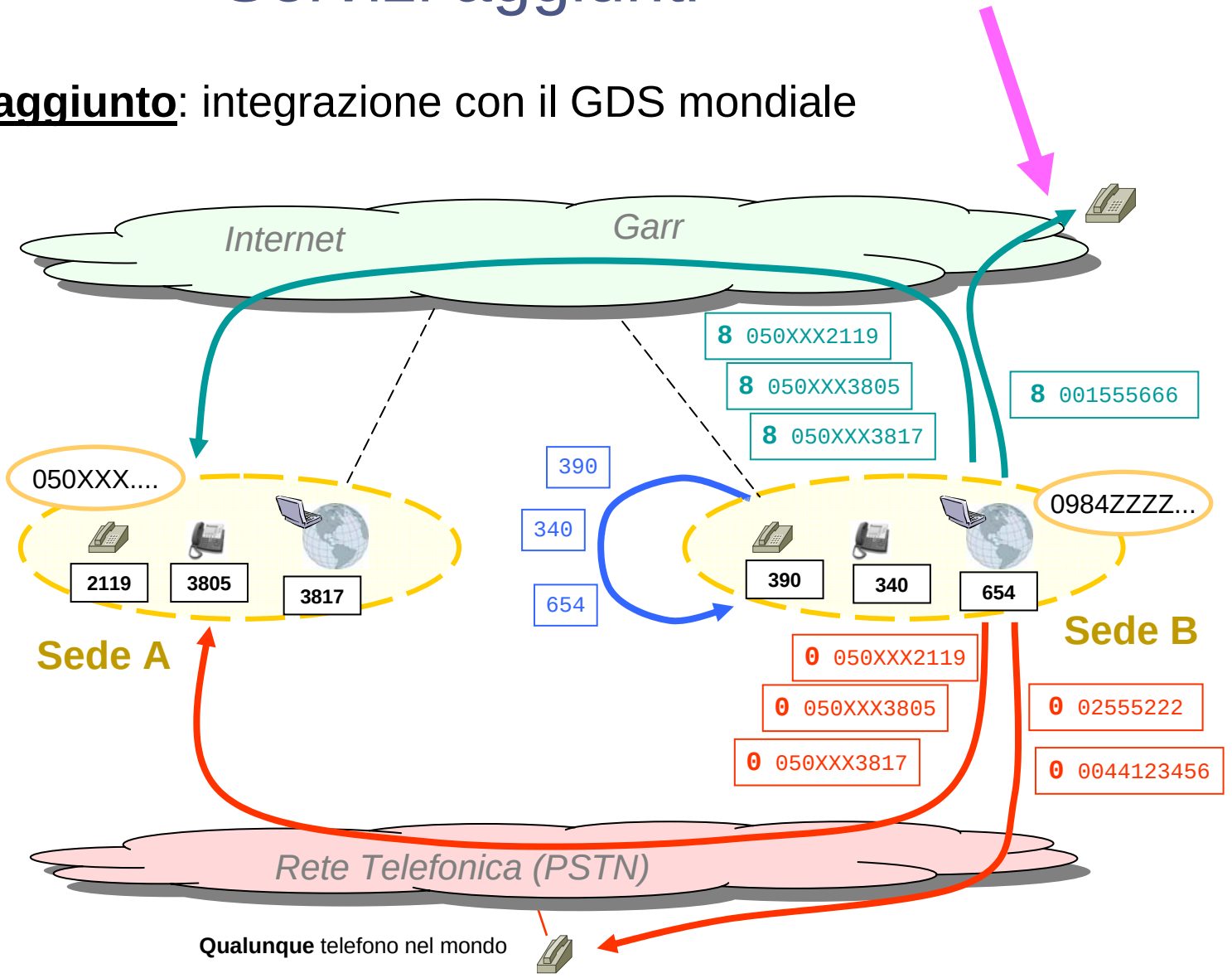
- I telefoni IP aggiunti ed integrati con quelli legacy sono utilizzati con le stesse identiche regole di numerazione di quelli legacy. L'utente non è tenuto a sapere se sta chiamando da un telefono IP né tanto meno se sta chiamando un telefono IP.
- I telefoni IP aggiunti posso essere hardware ma anche software consentendo la mobilita' dell'utente. Possono inoltre avere anche il supporto video.



Servizi aggiunti

- **3. Servizio aggiunto**: integrazione con il GDS mondiale

- La connessione al GDS consente di chiamare, via IP, qualunque altra sede nel mondo che abbia aderito al GDS.



Implementazione

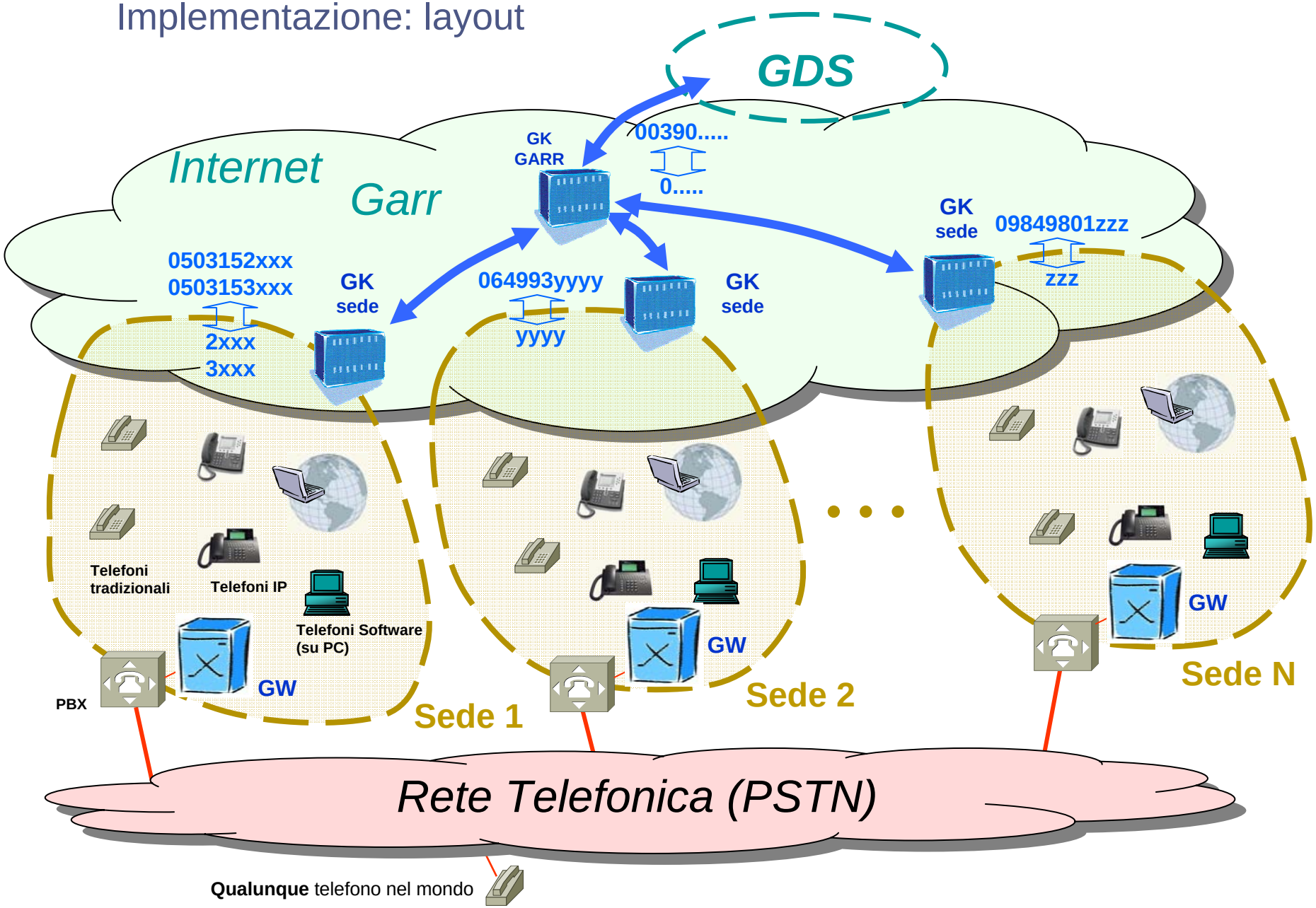
- Per la realizzazione dei servizi esposti, il protocollo standard scelto è stato H.323.
  - Tra i motivi: la possibilità di connettersi al GDS che è tutt'oggi implementato in H.323
- L'implementazione, comunque, si integra e può naturalmente evolvere con le soluzioni basata su ENUM e su SIP, come dettagliato nel seguito.
- **Implementazione attuale**:
  - Esiste un Gatekeeper (GK) H.323 gerarchicamente superiore per tutte le sedi GARR e competente per il prefisso (internazionale) "00390", che comprende quindi tutti i telefoni della rete fissa italiana. Allo stato attuale i soli telefoni di interesse sono quelli degli enti afferenti al GARR.
  - Ogni sede ha un proprio Gatekeeper competente per il prefisso già assegnato al centralino locale (in alcuni casi ci possono essere più prefissi assegnati). Ad es.: il GK del CNR di Pisa è competente per i prefissi (nazionali) "0503152" e "0503153".
  - La numerazione rispetta la gerarchia dei gatekeeper:
    - le chiamate locali interne ad una sede possono essere effettuate con la numerazione breve (2543 -> 3866); è comunque una scelta del gestore locale;
    - le chiamate fra sedi italiane sono effettuate con la numerazione nazionale (0503152244 -> 0649937862)
    - le chiamate fra sedi italiane e sedi estere sono effettuate con la numerazione completa internazionale (00390323518355 -> 004476543210)

Implementazione

(continua)

- **Inplementazione attuale**:
  - Il Gatekeeper di una sede può essere configurato come child (si registra al GK superiore, possibile con gnugk) o come neighbor
  - Sul Gatekeeper di una sede si registrano:
    - i telefoni IP locali, con una numerazione breve coerente con quella utilizzata per i telefoni legacy preesistenti.
    - il GW, competente del prefisso (o dei prefissi) corrispondente all'insieme di numeri dei telefoni legacy del PBX.
  - Ogni sede predispone un Gateway (GW) per connettere il PBX con la rete IP. Nel seguito alcuni consigli.
  - Le chiamate dalla rete IP verso la rete pubblica, attraverso il GW ed il PBX di una sede, sono consentite solo dai telefoni IP appartenenti a quella sede (in caso contrario si introdurrebbero grosse complicazioni nella gestione dei costi fra una sede e l'altra).

Implementazione: layout

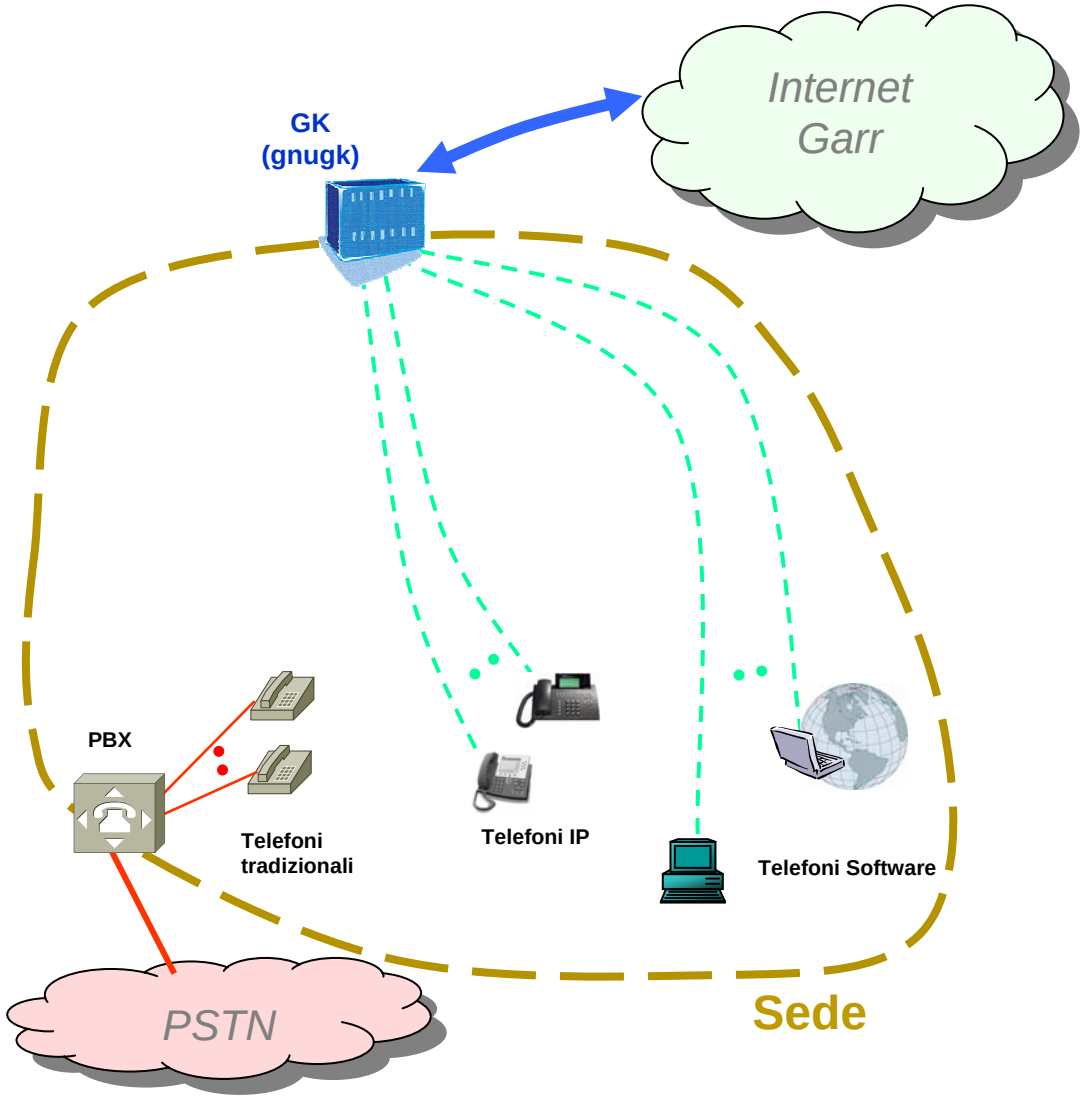


Requisiti minimi (1): Gatekeeper

- **Step 1: Attivazione del Gatekeeper**
  - è richiesto solo che il gatekeeper operi una opportuna manipolazione dei numeri in modo che per le chiamate esterne alla sede sia utilizzata una numerazione completa di tipo nazionale
  - il piano di numerazione interna è a discrezione del gestore locale
  - la soluzione OpenSource **gnugk** consente di partecipare immediatamente a costo praticamente nullo
  - Utilizzando telefoni software gratuiti è subito possibile scambiare telefonate con altre sedi
  - Il telefono software su PC consente la mobilità: ad esempio un utente può trovarsi all'estero ed avere un telefono software con cui chiamare i telefoni software della propria sede e qualunque altro telefono delle altre sedi connesse

Requisiti minimi (1): Gatekeeper

- Sede con GK attivo
  - GW assente
  - Si possono utilizzare telefoni IP sia hardware che software
  - I telefoni legacy e quelli IP della sede sono, però, due mondi separati



Configurazione di gnugk

[Gatekeeper::Main]
Fourtytwo=42
Name=gk_nuovasede

[RoutedMode]
GKRouted=1
H245Routed=1
CallSignalPort=1721
RemoveH245AddressOnTunneling=1
AcceptNeighborsCalls=1
AcceptUnregisteredCalls=1
DropCallsByReleaseComplete=1
SendReleaseCompleteOnDRQ=1
Q931PortRange=20000-20999
H245PortRange=30000-30999

[Endpoint]
Gatekeeper=146.48.96.183
Type=Gateway
H323ID=gk_nuovasede
Prefix=09876543,09876544
TimeToLive=60

[Endpoint::RewriteE164]
09876543=3
09876544=4

[RasSrv::ARQFeatures]
CallUnregisteredEndpoints=1
RemoveTrailingChar=#

[GkStatus::Auth]
rule=regex
regex=^(127\.0\.0\.1)$

[Gatekeeper::Acct]
FileAcct=required

[FileAcct]
DetailFile=/var/log/gk/cdr.log
Rotate=weekly
RotateDay=Sunday
RotateTime=23:59

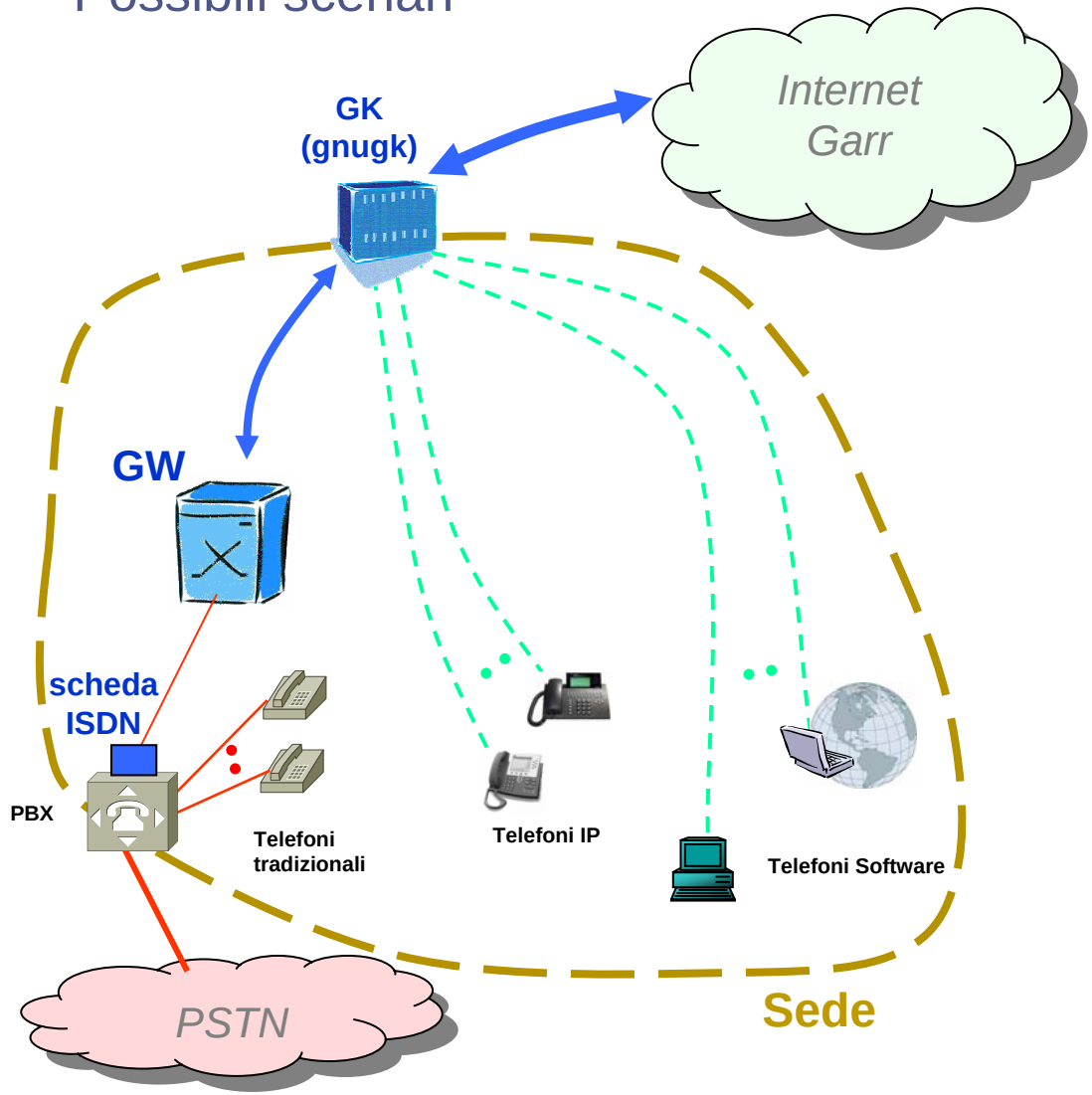
Requisiti minimi (2): Gateway

- **Step 2: Attivazione del Gateway**
- Per chi deve ancora realizzare il collegamento fra il PBX e la rete IP, valgono le seguenti considerazioni:
  - Preferire il Gateway esterno alla scheda VoIP integrata nel PBX, perché:
    - il Gateway esterno è più facile da gestire e configurare da parte di persone abituate a configurare i router;
    - le schede VoIP integrate di alcuni costruttori forniscono soluzioni proprietarie, difficili da integrare con il mondo H.323 puro; inoltre impongono limiti commerciali (licenze)
    - la soluzione del Gateway esterno rende più indolore la sostituzione o l'eliminazione, in tempi successivi, del PBX.
    - la soluzione del Gateway esterno consente una più facile integrazione con soluzioni basate su ENUM e SIP
  - Il Gateway H.323 esterno è attivabile a condizione che
    - sul PBX sia presente almeno una interfaccia per ulteriori collegamenti, preferibilmente di tipo esterno (ISDN PRI o BRI); in alternativa è possibile utilizzare una o più linee derivate attestate al centralino, di varie tecnologie sia analogiche che digitali (ISDN BRI, E&M, FXO), configurabili come unico derivato con numerazione interna breve (esempio"8");
    - il PBX sia configurato affinché invii determinati prefissi verso il Gateway H.323 esterno.
  - **Alcune sedi stanno già utilizzando Asterisk come GW di produzione**

Requisiti minimi (2): Gateway

Possibili scenari

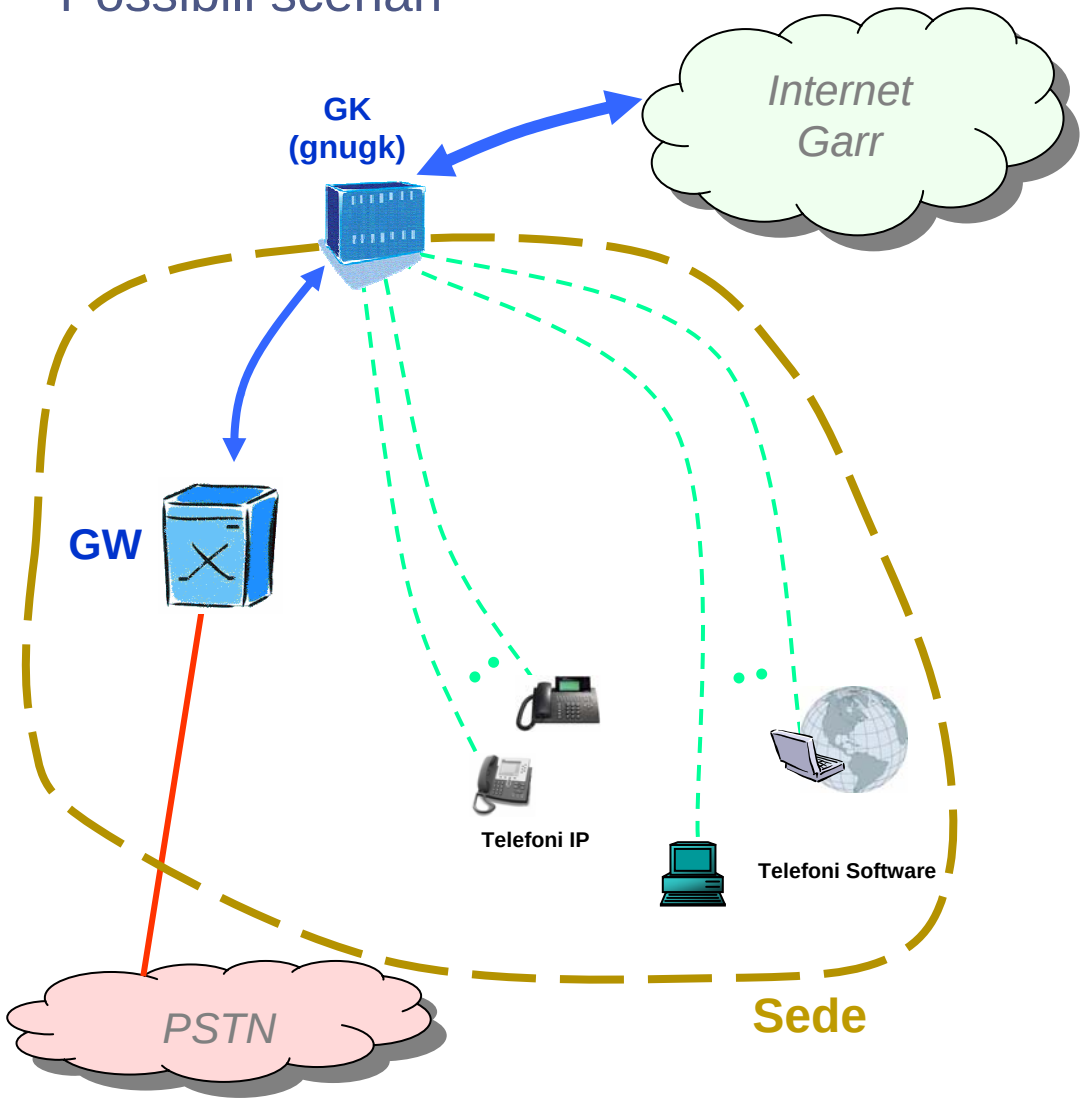
- Gateway esterno e connesso al centralino esistente



Requisiti minimi (2): Gateway

Possibili scenari

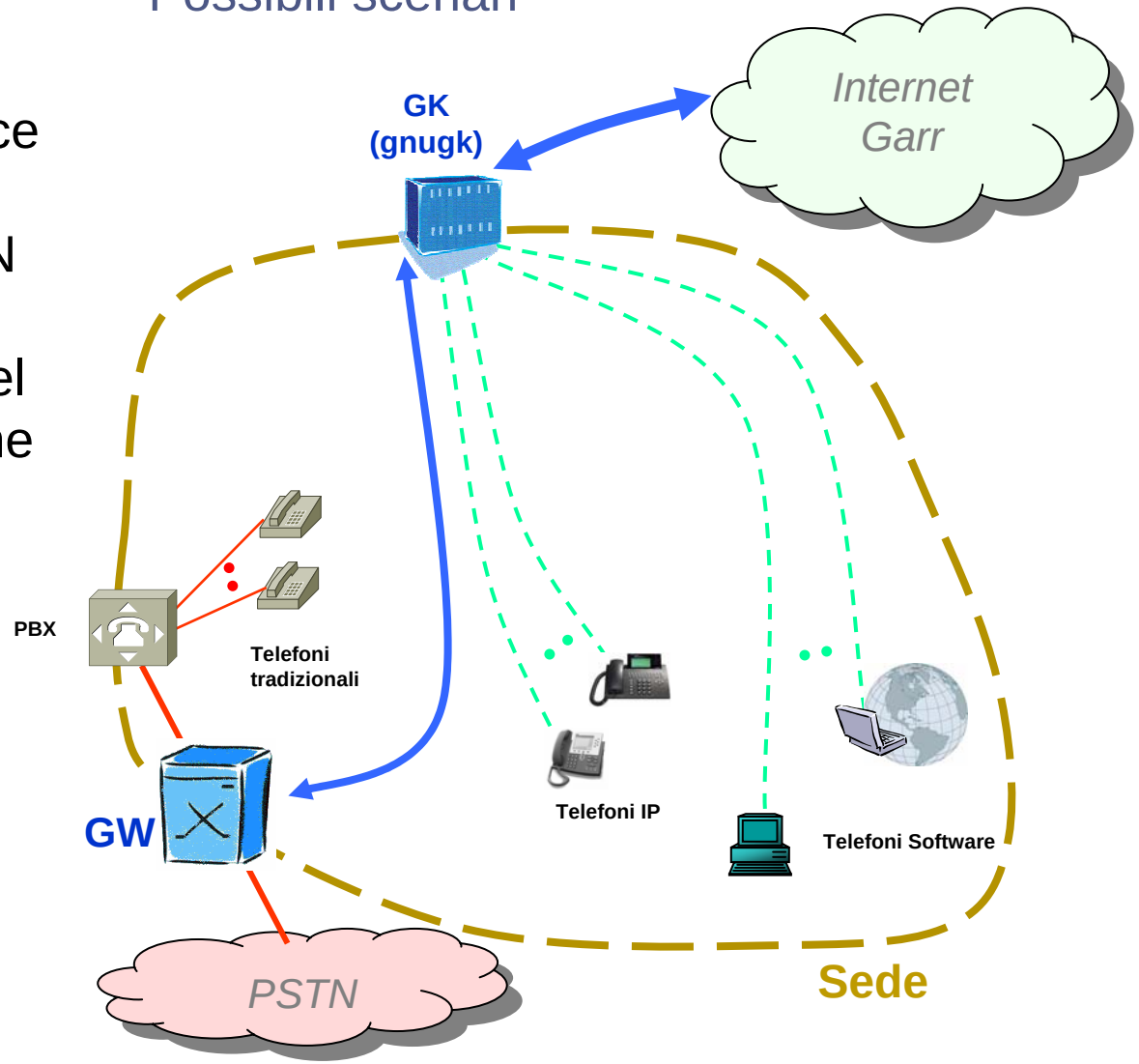
- Gateway in sostituzione del centralino telefonico
  - Può essere una nuova sede che decida di implementare direttamente solo Telefonia IP.



Requisiti minimi (2): Gateway

Possibili scenari

- Esistono soluzioni di GW con due interfacce ISDN che si possono interporre tra la PSTN ed il centralino, riducendo, ma non del tutto, la configurazione del PBX

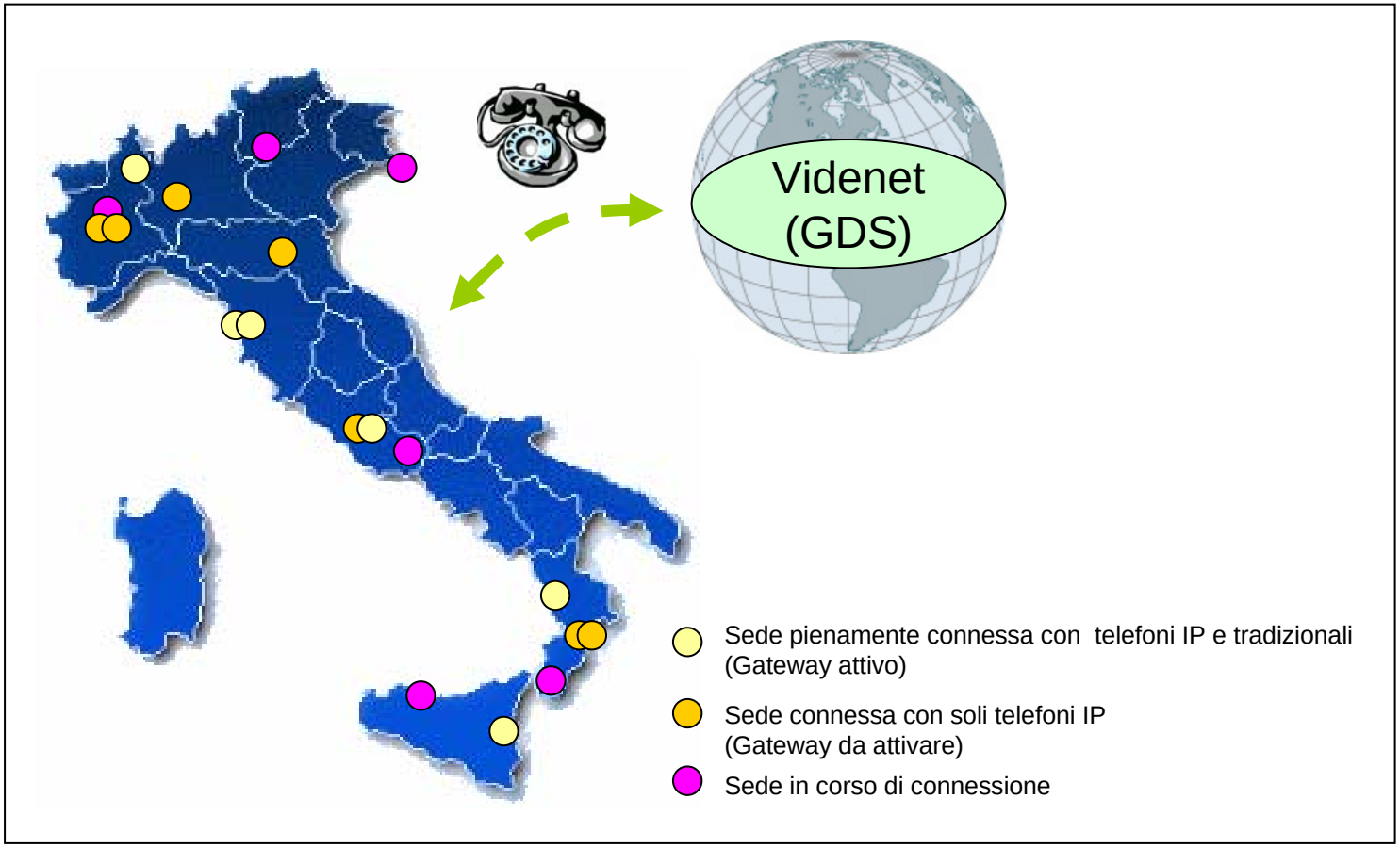


Stato attuale delle sedi italiane

- **Sedi pienamente connesse (GW attivo)**
  - CNR, Area della Ricerca di Pisa
  - CNR, Istituto di Scienze Neurologiche, Mangone (Cosenza)
  - CNR, Sede Centrale di Roma
  - CNR, Istituto per la Microelettronica e Microsistemi (Catania)
  - Università di Pisa
  - CNR, Istituto per lo Studio degli Ecosistemi (ISE), Pallanza (Verbania)
- **Sedi connesse con GW da attivare**
  - CNR, Area della Ricerca di Torino
  - CNR, Istituto di ricerca sull'impresa e lo sviluppo (CERIS), Moncalieri (Torino)
  - Area della Ricerca di Bologna (6 istituti CNR e 2 istituti INAF)
  - Università Bocconi (Milano)
  - CNR, ISN, Roccelletta di Borgia (Catanzaro)
  - Universita' di Catanzaro - Facoltà di Farmacia
  - Caspur, Roma
- **Sedi in corso di connessione**
  - CNR, Area della Ricerca di Palermo
  - Università di Trieste
  - Università di Cassino
  - Università di Trento
  - CSP – INLAB (Torino)
  - Università di Reggio Calabria

Stato attuale delle sedi italiane

aggiornato al 2 novembre 2005



- Situazione aggiornata:
  - http://reti4.iit.cnr.it/voipgarr

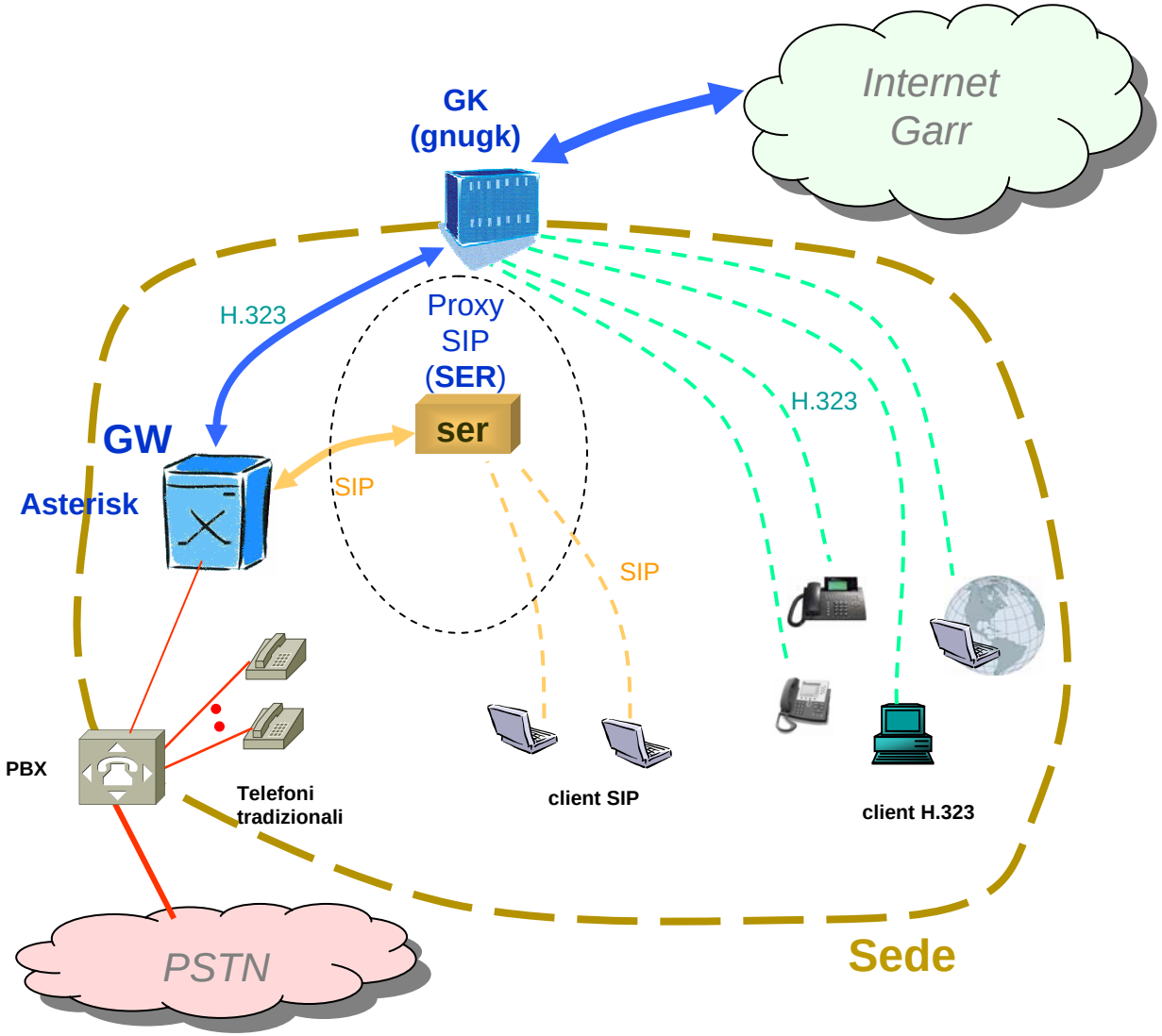
Integrazione con SIP

- L'attuale implementazione (in H.323) può essere facilmente estesa al protocollo SIP.
- E' sufficiente abilitare sul GW la funzione GW H.323<->SIP e configurare opportunamente l'instradamento
- Se il GW già utilizzato non supporta SIP si può installare un Asterisk che svolge molto bene questa funzione.
  - Per la registrazione dei client SIP si può utilizzare il proxy SIP "ser".
- Nella sede si può così avere una situazione mista di client SIP e H.323.
- Anche i client SIP sono perfettamente integrati con il resto della struttura:
  - possono fare e ricevere chiamate esattamente come tutti gli altri telefoni della sede

- **Integrazione con SIP già realizzata e funzionante in alcune sedi**

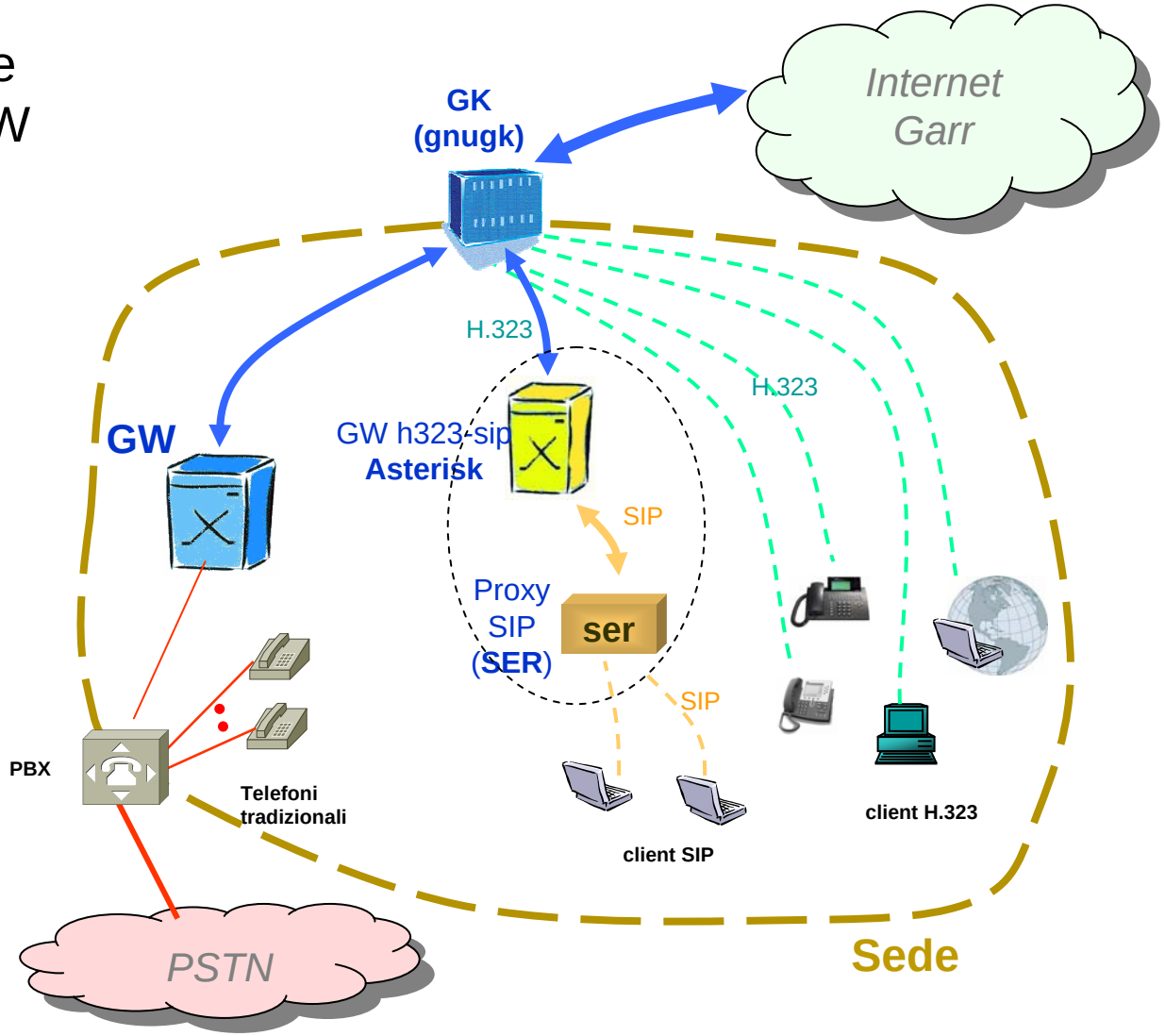
Integrazione con SIP

- Situazione mista di client H.323 e SIP:
  - Il GW (ad es. Asterisk) fa da gateway per PSTN, H323 e SIP.
  - Il Proxy SIP SER gestisce i client SIP
- Già realizzata e funzionante in alcune sedi.



Integrazione con SIP

- Client misti H.323 e SIP, nel caso di GW già installato che non supporta SIP
- Asterisk e SER possono stare anche sulla stessa macchina

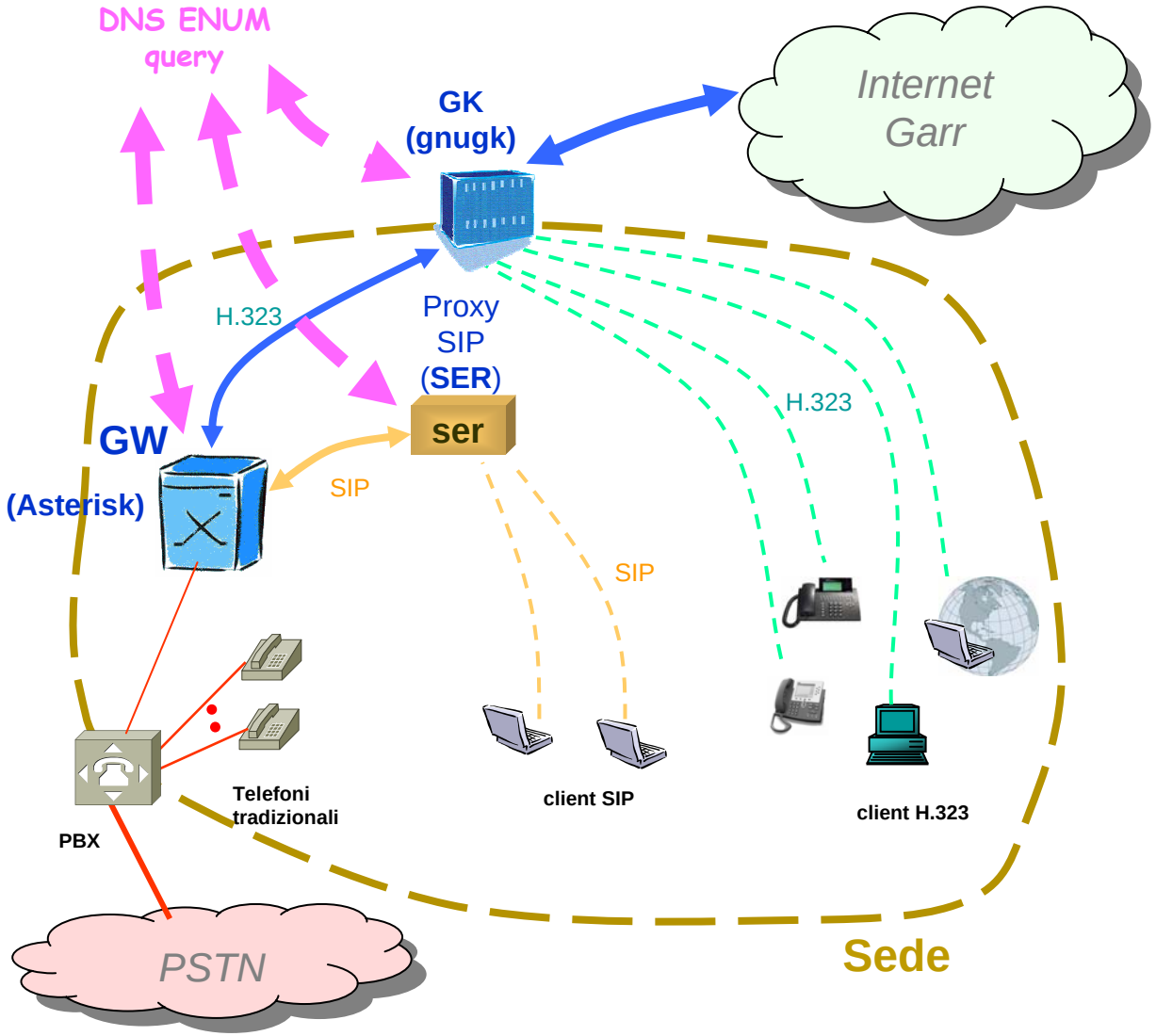


Integrazione con ENUM

- Il Global Dialling Scheme evolverà verso un meccanismo di localizzazione basato su ENUM, con una fase transitoria di convivenza con l'attuale meccanismo di localizzazione basato sulla gerarchia di gatekeeper.
- Le sperimentazioni basate su SIP utilizzano ENUM per la localizzazione
- Anche l'integrazione con ENUM è facilmente integrabile con l'implementazione attuale
- **Integrazione con ENUM già realizzata e funzionante in alcune sedi**

Integrazione con ENUM

- Sia il GW (asterisk), che il proxy sip (ser) che il gatekeeper (gnugk) possono fare una query al DNS per localizzare il chiamato.
- In caso di risposta negativa utilizzeranno i metodi alternativi (gerarchia di GK, ...)
- Gia implementato in alcune sedi



Configurazione di filtri su router e Firewall

- Per una più semplice gestione di porte e indirizzi IP da configurare su eventuali filtri su router e Firewall si consiglia di definire i range di porte utilizzate sul gatekeeper
- Esempio nel caso gnugk:

[RoutedMode]
GKRouted=1
H245Routed=1
CallSignalPort=**1721**
H245PortRange=**30000-30999**

[Proxy]
#Enable=1
#RTPPortRange=**50000-59999**

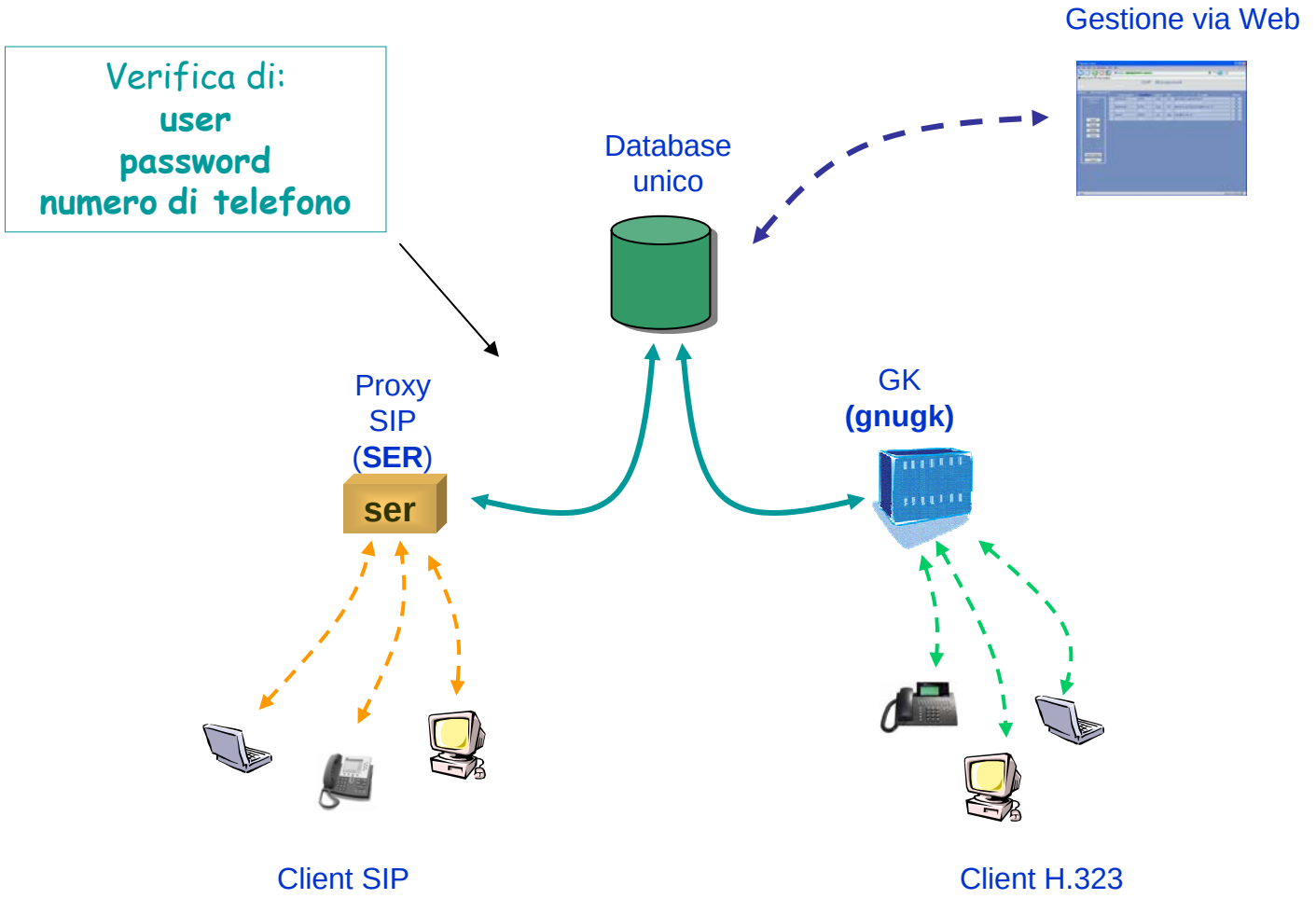
  - Il gatekeeper riceverà:
    - connessioni **TCP** sulle porte:
      - **1721** e range **[30000-30999]**
  - Se in Proxy mode riceverà:
    - pacchetti **UDP** sulle porte:
      - range **[50000-59999]**
  - Il gatekeeper dovrà inoltre inviare pacchetti **UDP** sulla porta **1719** del gatekeeper del GARR

- Per il proxy SIP (ser) i pacchetti di segnalazione sono scambiati sulla porta **UDP 5060** (se non diversamente configurato)

Autenticazione dei client H.323 e SIP

- Sistema implementato al CNR di Pisa
  - Consente una autenticazione integrata dei client H.323 e SIP
  - Unico Database a cui attingono sia il gatekeeper (gnugk) che il proxy sip (ser)
  - Gestione via web del database
    - Amministratore
    - Utente (può cambiare la password) ed aventualmente abilitare o disabilitare certe funzioni
  - I client si autenticano con una tripletta di valori:
    - user
    - password
    - numero di telefono
  - Per client particolari che non supportino questo tipo di autenticazione è prevista una autenticazione statica basata su user e indirizzo IP (ad esempio nel caso dei GW)

Autenticazione dei client H.323 e SIP



Autenticazione dei client H.323 e SIP

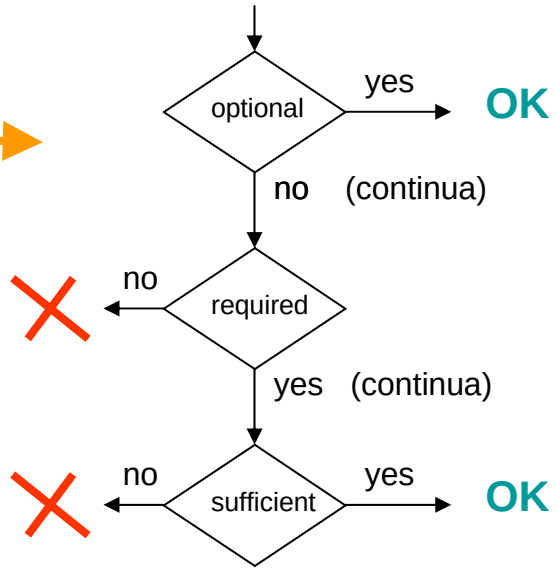
Esempio di autenticazione con gnugk

```
[RasSrv::RRQAuth]
gw_pisa=sigip:146.49.33.22:1720
asterisk=sigip:146.50.77.11:1720

[Gatekeeper::Auth]
AliasAuth=optional;RRQ
SQLPasswordAuth=required;RRQ
SQLAuth=sufficient;RRQ
default=allow

[SQLPasswordAuth]
Driver=PostgreSQL
Host=192.168.55.66:5432
Database=gk_cnr_pisa_db
Username=postgres
Password=
Query=SELECT passwd FROM h323_users WHERE
    username = '%1'
MinPoolSize=5
```

```
[SQLAuth]
Driver=PostgreSQL
Host=192.168.55.66:5432
Database=gk_cnr_pisa_db
Username=postgres
Password=
RegQuery=SELECT authorize FROM h323_users
    WHERE username = '%u' AND ( aliases =
    '%{aliases}' OR aliases2 = '%{aliases}')
```



- In questo esempio viene autenticato solo l'RRQ verificando: h323id, password e numero di telefono
- Nella sezione **`[Gatekeeper::Auth]`** il **default** è necessario per tutti gli altri tipi di messaggi che non siano RRQ (LRQ, ARQ, ...)

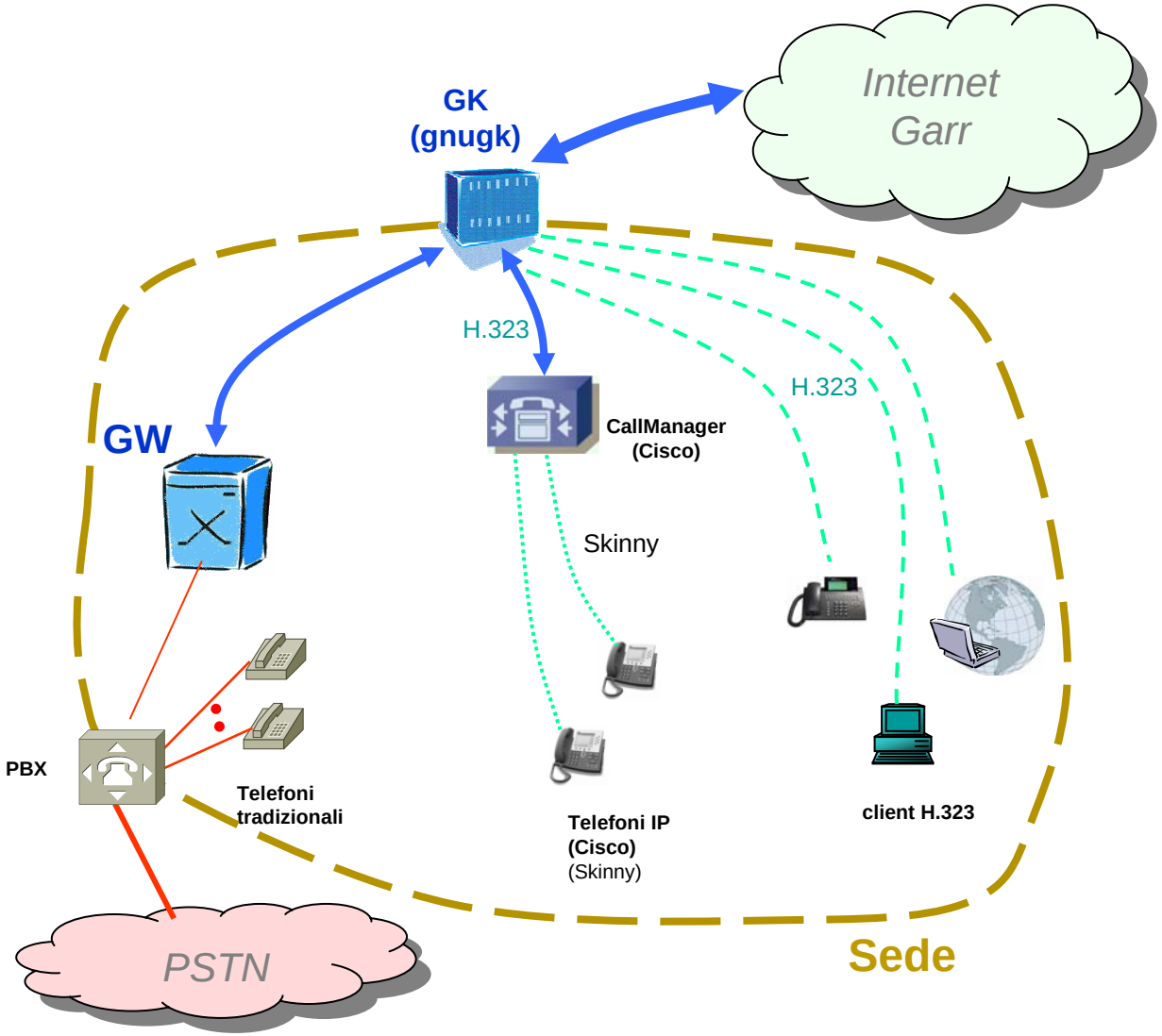
Esempio di autenticazione con gnugk

- L'esempio configurato su gnugk richiede un database Postgres (ma si può utilizzare anche MySQL) contenente una tabella creata come descritto dal seguente script:

```
-----------------------------------------------------------------
drop table Utenti_H323;
create table Utenti_H323 (
                            authorize varchar(32) not null,
                            username varchar(32) not null,
                            passwd varchar(32) not null,
                            e164 varchar(32) not null,
                            aliases varchar(128) not null,
                            aliases2 varchar(128) not null,
                            comment varchar(128) not null,
                            primary key (username, passwd, e164),
                            unique (e164)
                             );
-----------------------------------------------------------------
```

Integrazione di un Cisco Call Manager

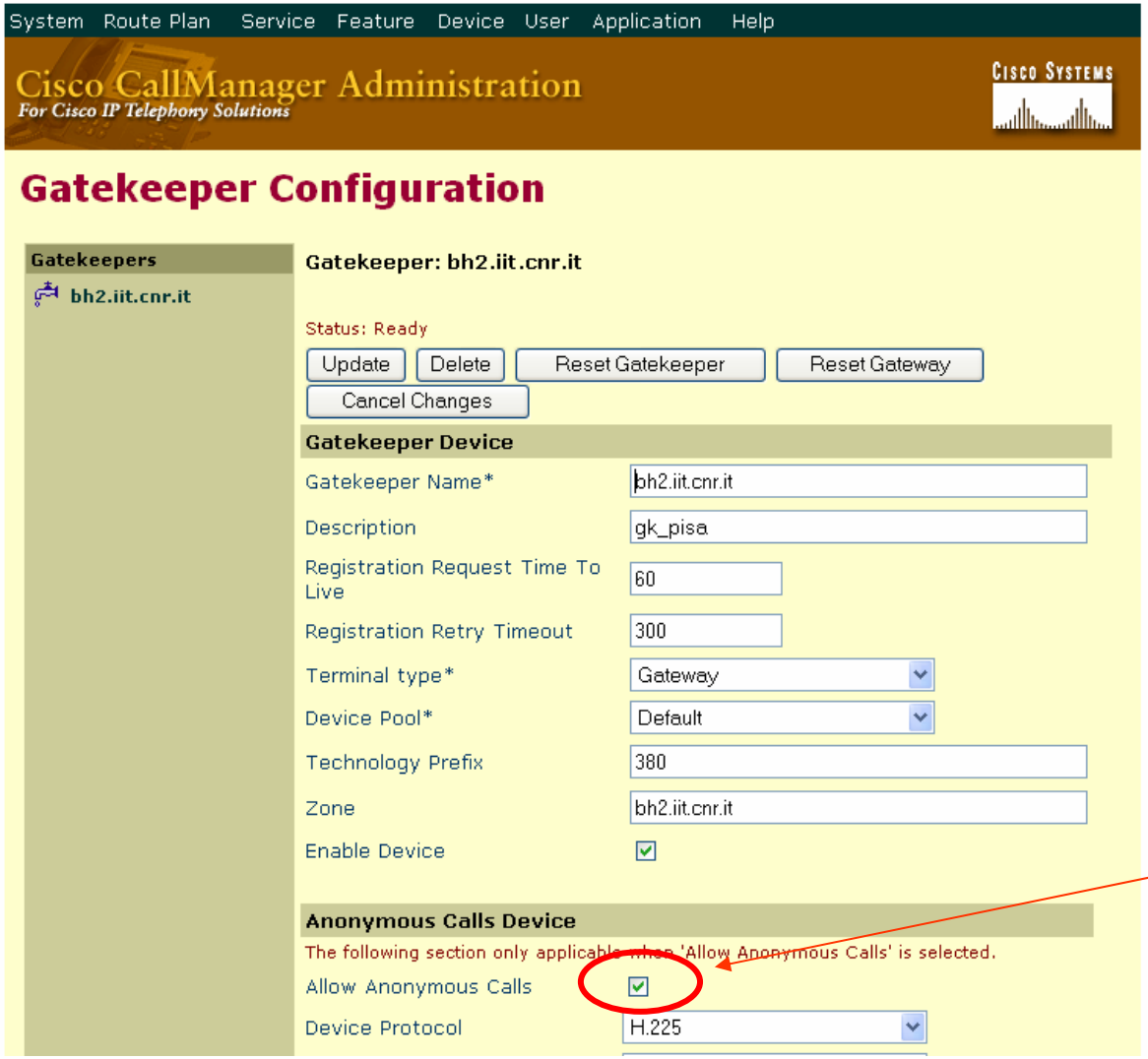
- Il Cisco Call Manager può facilmente essere integrato in quanto può registrarsi su un Gatekeeper come un Gateway



Cisco Call Manager registrato su GK

- **Come configurare il Cisco Call Manager per la sua registrazione su un Gatekeeper H.323:** (CCM ver 3.2)
  - Scegliere il menu “Device” -> “Gatekeeper”. Nel campo “Gatekeeper Name” inserire l'indirizzo IP del GK.
  - Accertarsi che la casella “Allow Anonymous Calls” sia selezionata.
  - Click sul bottone “Update” e quindi su “Reset Gatekeeper”. Apparirà un messaggio “Reset Device”. Selezionare “Reset”.
  - Scegliere il menu “Route Plan” -> “Route Pattern” -> “add a new route pattern”. Nel campo “Route Pattern”, inserire il prefisso per il quale si vuole l'inoltre al Gatekeeper.
    - Se ad esempio si vogliono inoltrare le chiamate a numeri che iniziano con “0”, inserire il pattern “0!”. Invece per numeri che iniziano con 2 e di lunghezza totale di 4 cifre inserire “2XXX”.
  - Nel campo “Gateway/Route List”, selezionare “Anonymous Device”.
  - Selezionare l’opzione “Provide Outside Dial Tone” nel caso di patern “0!” se si vuole fornire all’utente il tono di linea per le chiamate esterne (come già avviene con il centralino).
  - Fare click su “Insert”.
  -
  - Bene! A questo punto dovreste essere in grado di chiamare un telefono software H.323 da un telefono Cisco IP Phone.

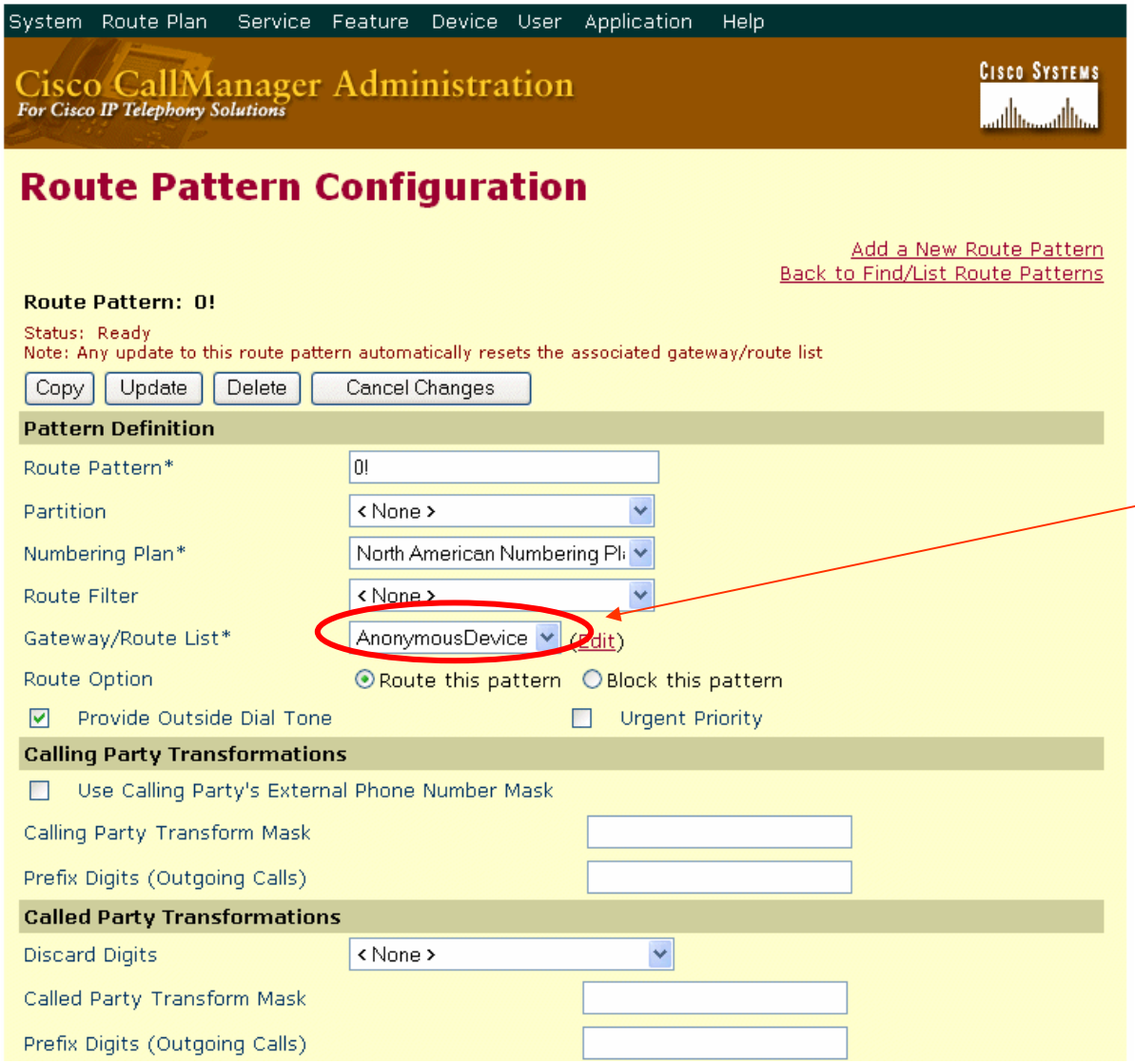
Cisco Call Manager registrato su GK



Registrazione del CCM su un Gatekeeper

Suggerimenti sulla configurazione

Cisco Call Manager registrato su GK

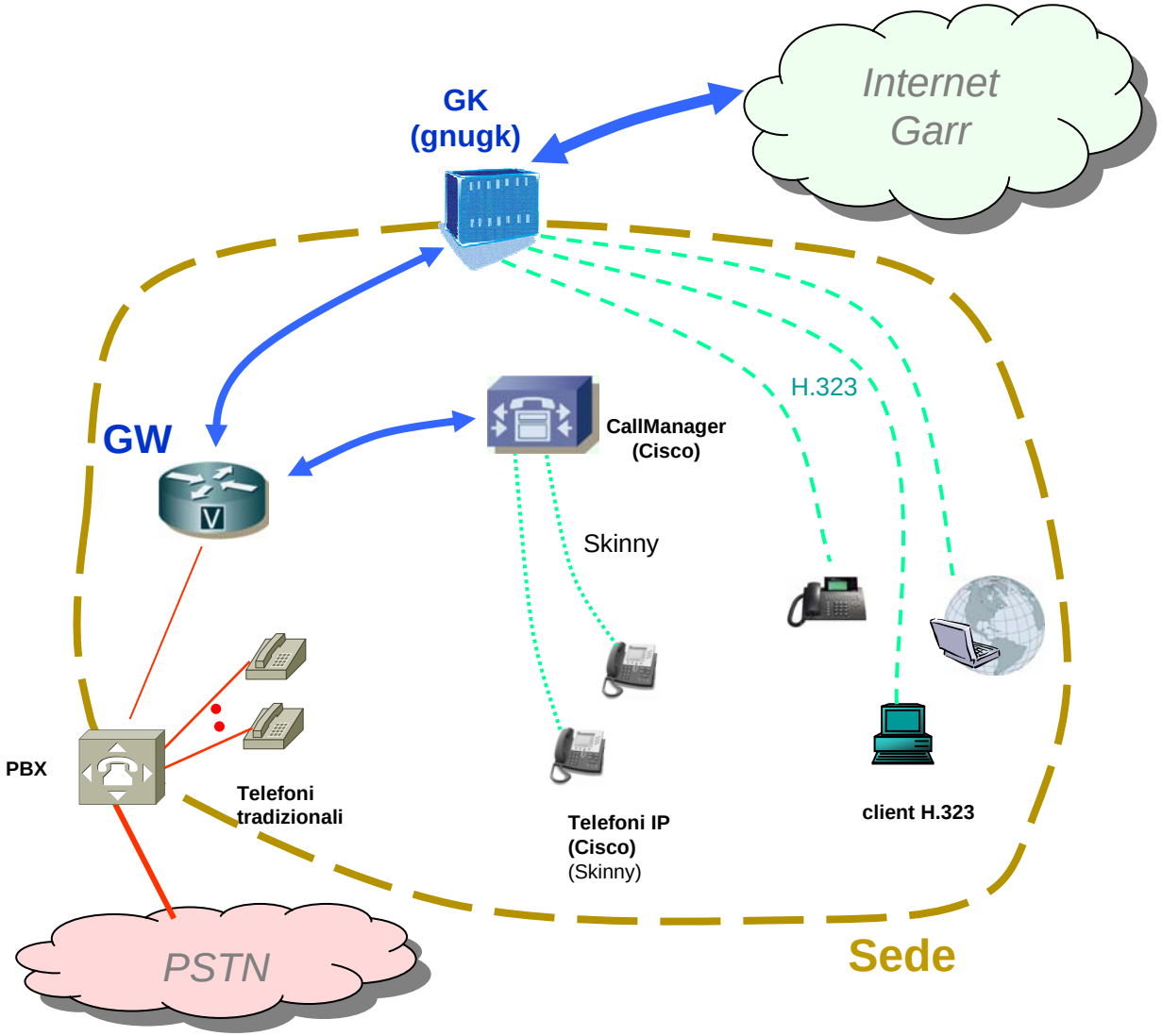


Registrazione del CCM su un Gatekeeper

Suggerimenti sulla configurazione

Integrazione di un Cisco Call Manager

- Se il Cisco Call Manager è già connesso al GW Cisco è sufficiente che il solo GW si registri sul GK
- Analogamente se si utilizza il Cisco Call Manager Express integrato nell'IOS dello stesso router



Ringraziamenti

- Desidero ringraziare Abraham Gebrehiwot (IIT-CNR, Pisa) e Ivan Duca (ISN-CNR, Mangone, Cosenza) per il contributo ed il materiale fornito.

Per aderire:

- Contatti per aderire all'utilizzo coordinato della telefonia su IP nel GARR:
  - http://reti4.iit.cnr.it/voipgarr