SIP(Session Initiation Protocol)

(RFC3261)

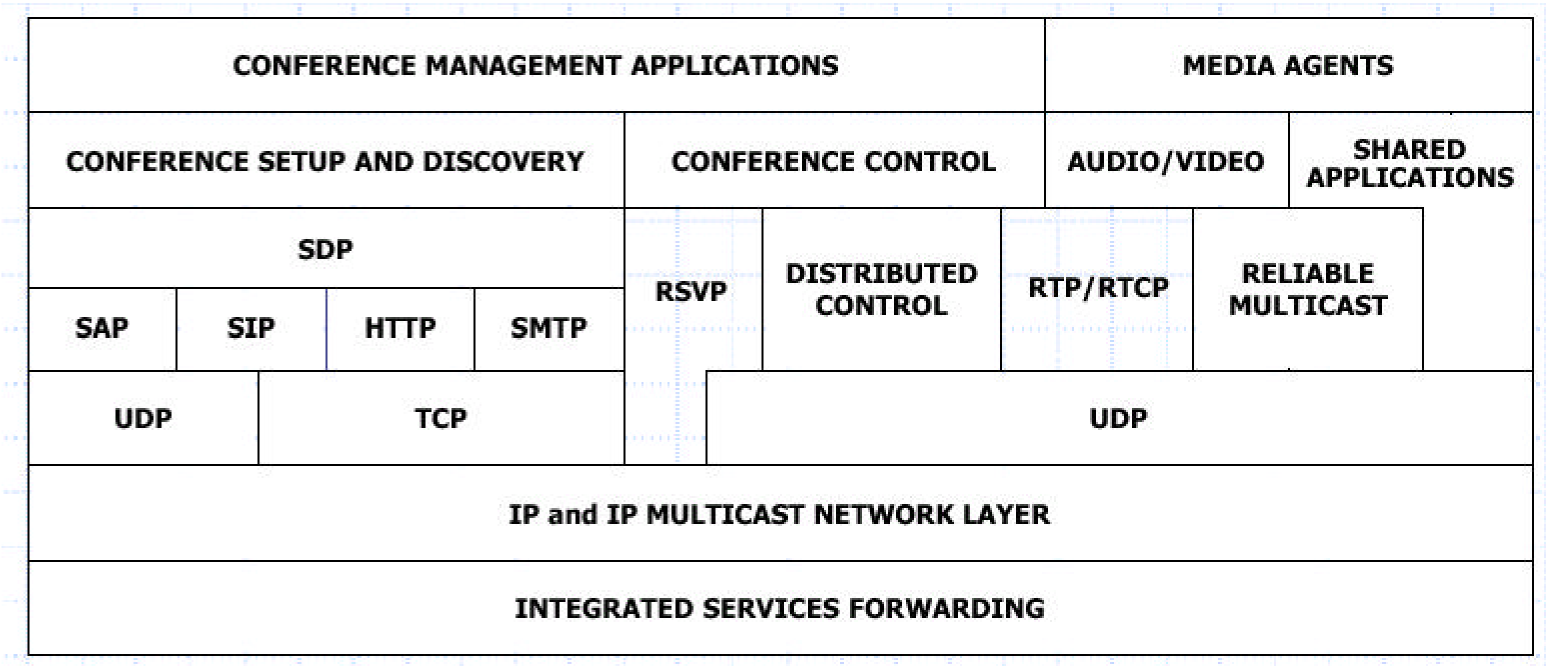
이 경 희

MIPTelecom

(khlee@miptel.com)

KRnet2002

Internet Multimedia Protocol



Internet Protocol

SIP	Session Initiation Protocol
SDP	Session Description Protocol
SAP	Session Announcement Protocol
RTSP	Real Time Streaming Protocol
HTTP	HyperText Transfer Protocol
RTP	Real-time Transport Protocol
RTCP	RTP Control Protocol
RSVP	Resource reSerVation Protocol
UDP	User Datagram Protocol
TCP	Transmission Control Protocol
IGMP	Internet Group Management Protocol

SIP Timeline

- 1996
  - Mark Hadley's SIP(Session Invitation Protocol)
  - Henning Schulzrinne's SCIP(Simple Conference Control Protocol)
- 1996-1997
  - Interest grows in academic circles over MBone
- 1998
  - MCI's Henry Sinnreich => VoIP Appliance
- 1999. 3.
  - RFC 2543 by IETF MMUSIC WG
- 1999. 9.
  - IETF SIP WG
- 2000. 6
  - RFC 2543bis

SIP Timeline

• 2000. 6	RFC 2543bis
• 2000. 11	RFC 2543bis-02
• 2001. 3	RFC 2543bis- 03
• 2001. 6	RFC 2543bis- 04
• 2001. 10	RFC 2543bis- 05
• **2002. 1**	**RFC 2543bis- 06**
• **2002. 2. 4**	**RFC 2543bis- 07**
• **2002. 2.18**	**RFC 2543bis- 07.9**
• **2002. 2.21**	**RFC 2543bis- 08**
• **2002. 2.27**	**RFC 2543bis- 09**
• **2002. 7.04**	**RFC 3261**

A Short History of SIP

- Internet Engineering Task Force (IETF)protocol
- Developed as Internet Draft (I-D)within MMUSIC (Multiparty Multimedia Session Control)Work Group
- Inventors:M.Handley,H.Schulzrinne,E.Schooler, and J.Rosenberg
- Became “Proposed Standard ” and RFC 2543 in March 1999.
- Separate SIP WG established in September 1999.
- Now new SIPPING (applications)and SIMPLE (presence and instant messaging)WGs using SIP.
- RFC 2543 obsolete, RFC 3261 July 2002.

RFC 3261 Major Change

- Protocol Structure : layer structure
- Routing rule: Record-Route, Route
- Via branch
- Transaction detail specify
- URI Comparison detail specify
- Locating SIP Server
- SDP offer/answer model
- Reliability of Provisional

SIP Functions

- IETF-standardized peer-to-peer signaling protocol
- User location
  - Locate user given email-style address
  - Personal mobility
    - Different terminal
    - Same identifier
    - Any location
- User capabilities
  - (re)-negotiate session parameters
- User availability
  - Determination of willingness of the called party to engage in communications
- Call setup
- Call handling
  - Manual and automatic forwarding ("name/number mapping")
  - "forking" of calls
  - Terminate and transfer calls

SIP Overview

- IETF-standardized peer-to-peer signaling protocol
  - Locate user given email-style address
  - Set up session
  - (re)-negotiate session parameters
  - Manual and automatic forwarding (“name/number mapping”)
  - Personal mobility
    - Different terminal
    - Same identifier
    - Any location
  - “forking: of calls
  - Terminate and transfer calls

SIP Protocol Overview

- Part of IETF conference control architecture:
  - SAP for advertising multimedia sessions
  - RTSP for controlling delivery of streaming media [RFC 2326]
  - SDP for describing multimedia sessions [RFC 2327]
  - RTP for transporting real-time data and providing QoS feedback [RFC 1889]
  - RSVP for reserving network resources [RFC 2205]
  - others: malloc, multicast, conference bus, …
- HTTP-like [RFC 2616]
- URI and URL [RFC 2396]

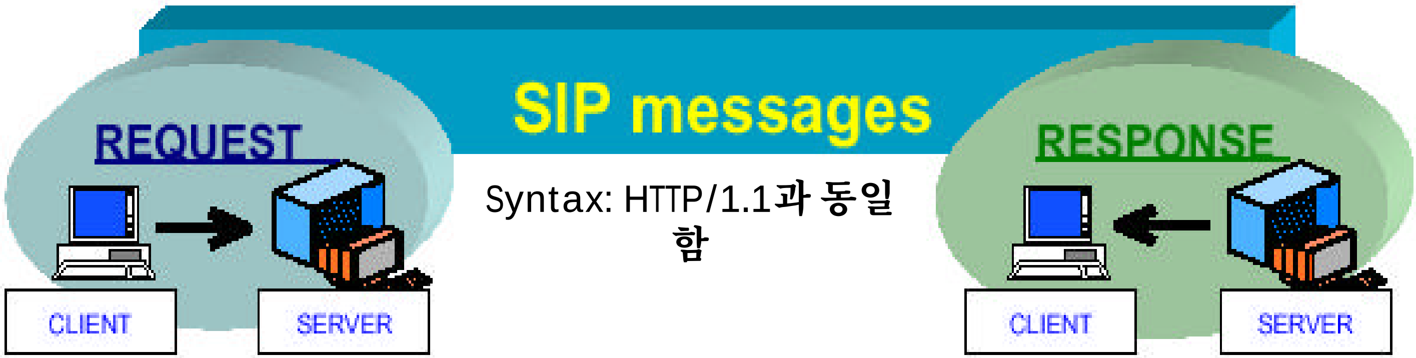
SIP and H.323

H.323	SIP + SDP
H.225.0 + RAS	SIP
H.245	SDP
Gatekeeper	Proxy

SIP Protocol

- text based (~ HTTP)
- Methods
  - INVITE initiate call
  - ACK confirm final response
  - BYE terminate (and transfer) call
  - CANCEL cancel searches and “ringing”
  - OPTIONS features support by other side
  - REGISTER register with location service

SIP Message



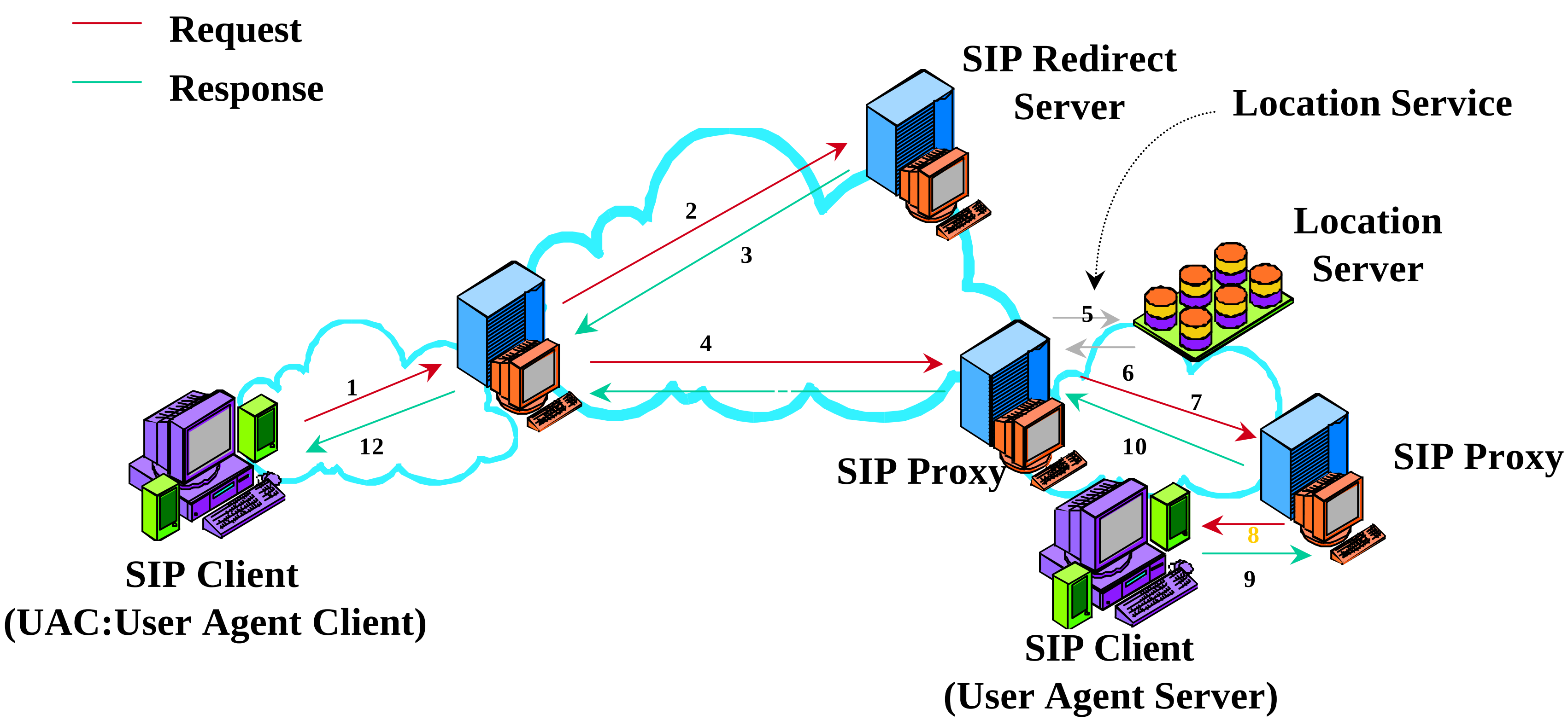
INVITE
ACK
BYE
CANCEL
REGISTER
OPTIONS

1xx : Provisional
2xx : Success
3xx : Redirection
4xx : Client Error
5xx : Server Error
6xx : Global Failure

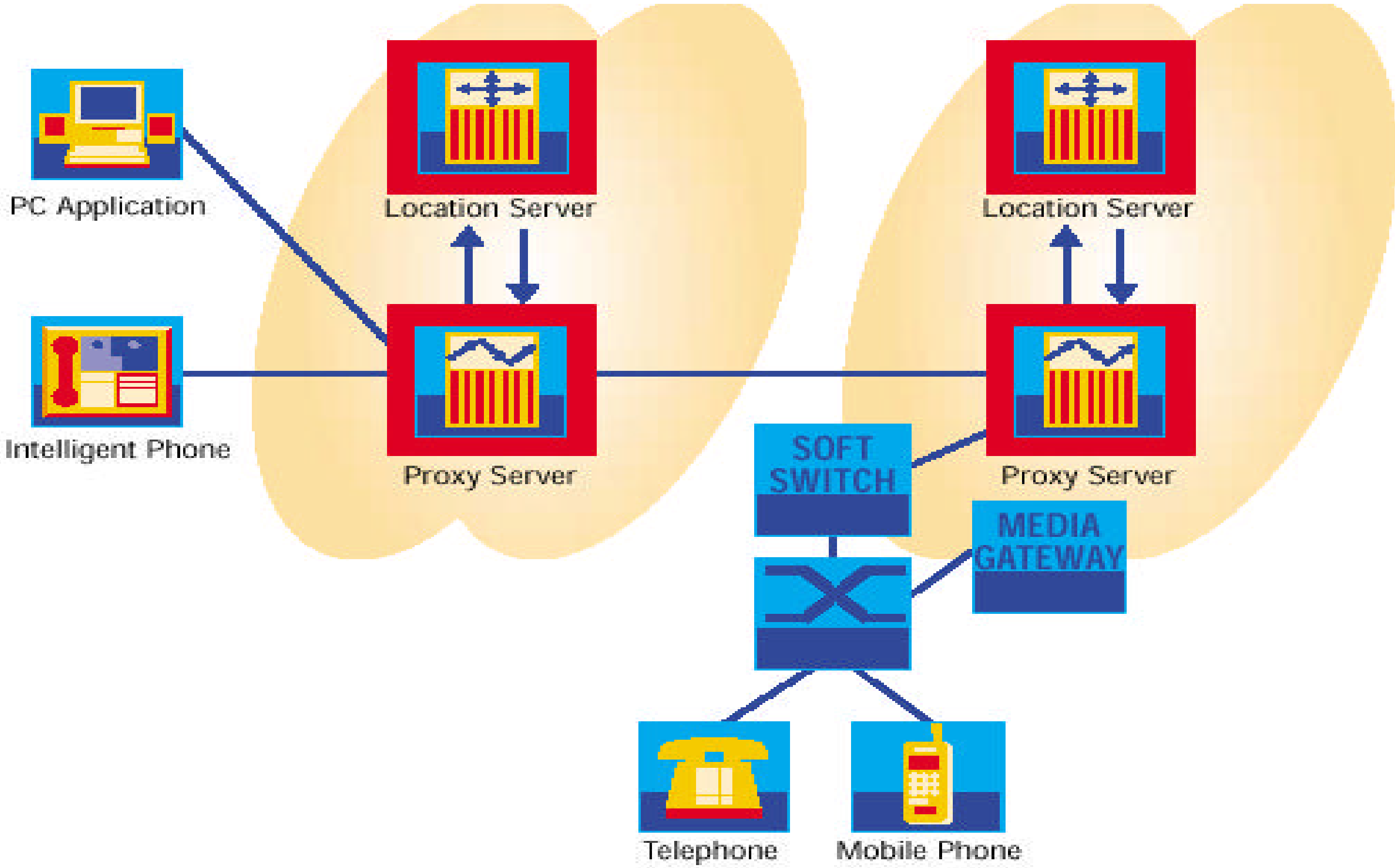
SIP Servers and Clients

- UAC (User-Agent Client)
  - caller application, call initiation
- UAS (User-Agent Server)
  - caller application, accept, redirect, refuse call
- Redirect Server
  - Redirect requests
- Proxy Server
  - Server + client
- Registrar
  - Track user locations
- User agent = UAC + UAS
- Often combine registrar + (proxy or redirect server)

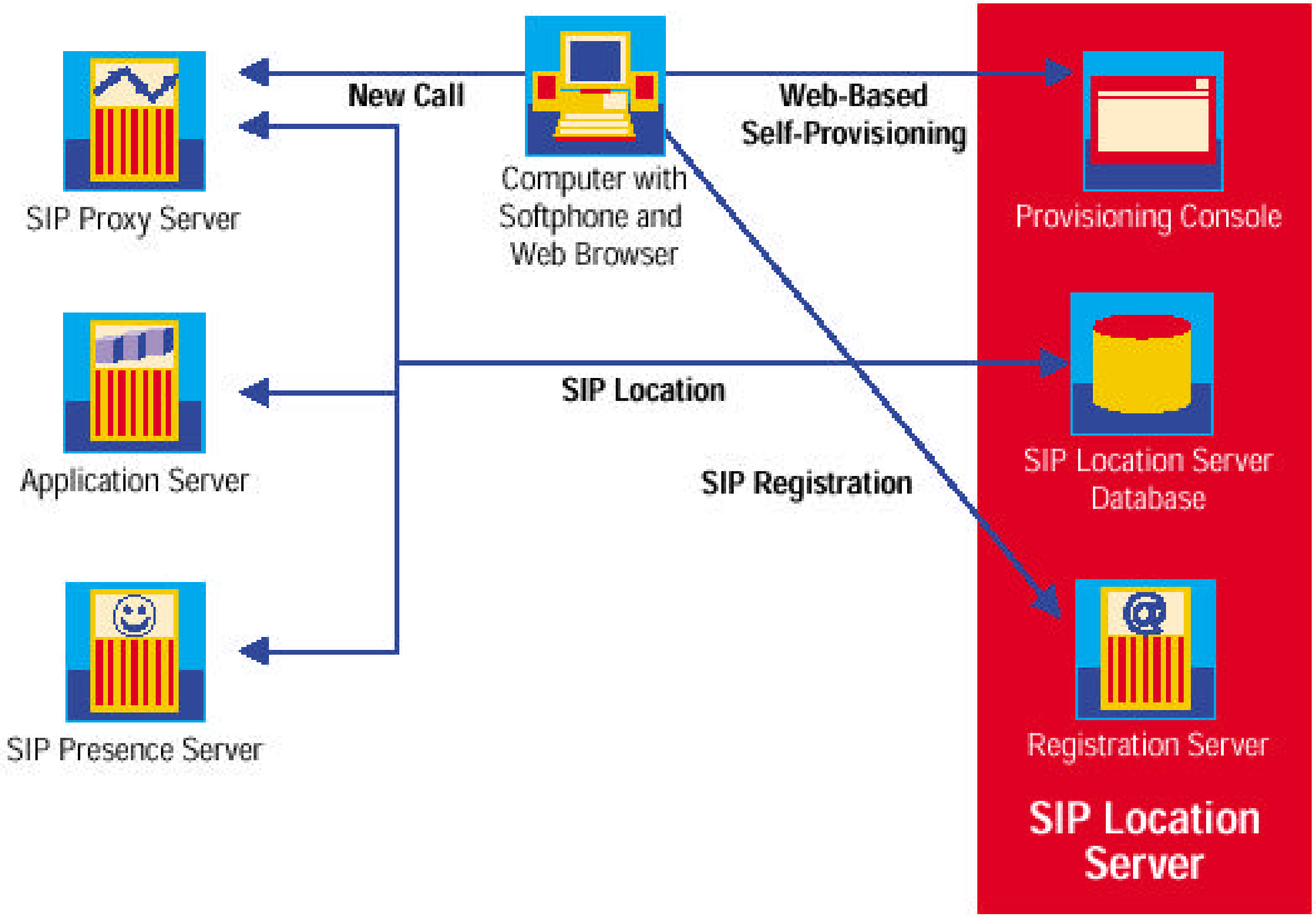
SIP Architecture



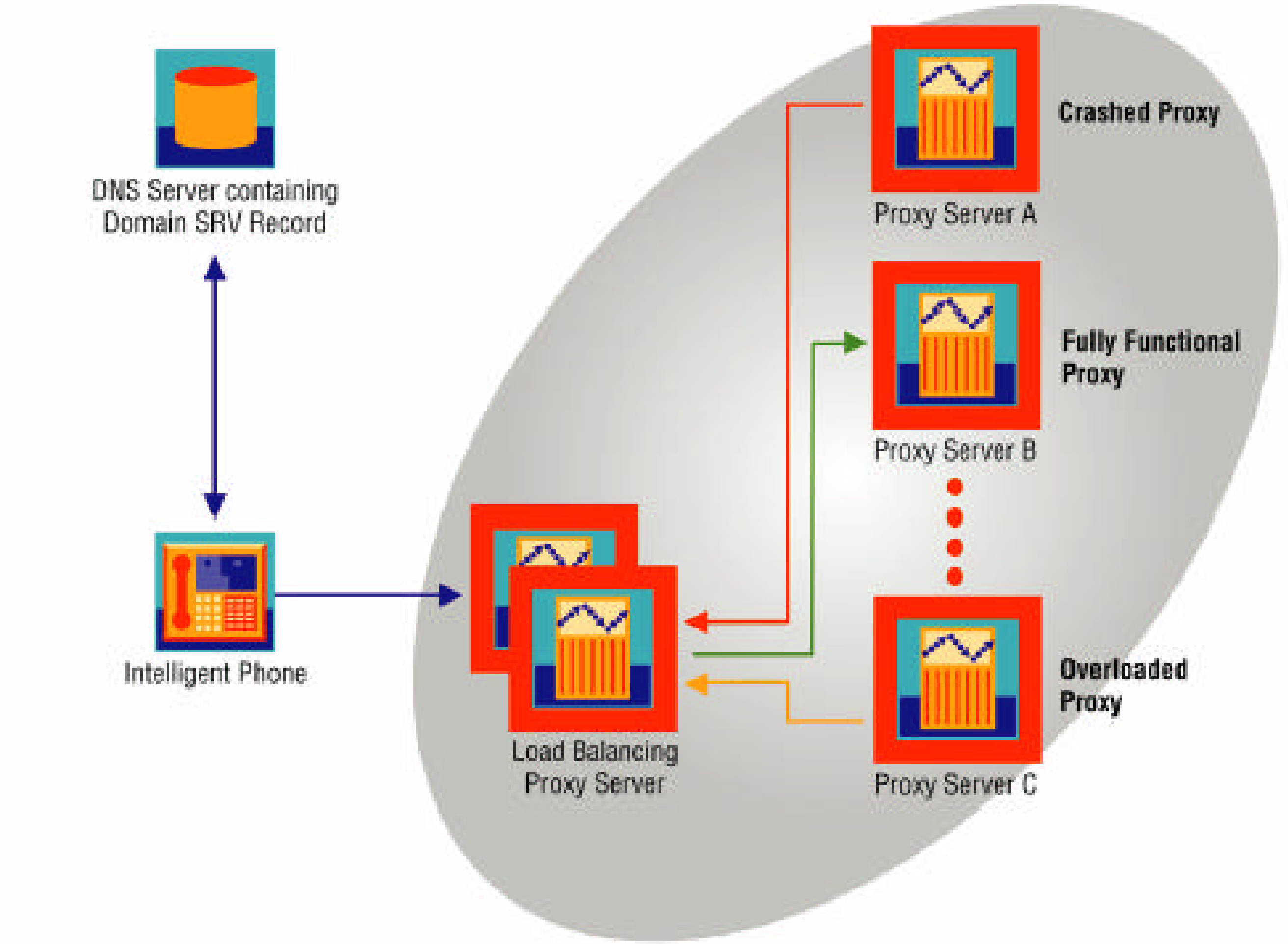
SIP Proxy Server



SIP Location Server



DNS & SIP Server



SIP URI Overview

- Format

sip:user:password@host:port;uri-parameters?headers

- SIP에서 사용자 identifer로 사용, e-mail Style
- Request-URI, From, To, Contact 등에 사용
- web page or email message에 URI 사용에 의해 시작되는 세션 가능
- FQDN(fully-qualified domain), IPv4, IPv6 address 지원
- Mandatory
  - only host part
- Optional or Disallowed
  - host part를 제외한 나머지 부분
- sips scheme 지원

SIP URL Syntax

```
SIP-URL        = "sip:" [ userinfo "@" ] hostport   url-parameters [ headers ]
 userinfo      = user [ ":" password ]
 user          = *( unreserved | escaped | "&" | "=" | "+" | "$" | "," )
 password      = *( unreserved | escaped    | "&" | "=" | "+" | "$" | "," )
 hostport      = host [ ":" port ]
 host          = hostname | IPv4address | IPv6reference
 hostname      = *( domainlabel "." ) toplabel [ "." ]
 domainlabel   = alphanum | alphanum *( alphanum | "-" ) alphanum
 toplabel      = alpha | alpha *( alphanum | "-" ) alphanum
 IPv4address   = 1*3digit "." 1*3digit "." 1*3digit "." 1*3digit
 IPv6reference = “[“ IPv6address “]”
 IPv6address   = hexpart [“:” IPv4address]
 hexapart      = hexaseq—hexaseq “::” [hexseq] – “::” [hexaseq]
 hexaseq       = hex4 *(“:” hex4)
 hex4          = 1*4HEX
 port          = *digit
```

SIP URL Syntax (Cont'd)

```
url-parameters   = *( ";" url-parameter )
url-parameter    = transport-param | user-param | method-param
                   | ttl-param | maddr-param | other-param
transport-param  = "transport=" ( "udp" | "tcp" )
ttl-param        = "ttl=" ttl
ttl              = 1*3DIGIT      ; 0 to 255
maddr-param      = "maddr=" host
user-param       = "user=" ( "phone" | "ip" )
method-param     = "method=" Method
tag-param        = "tag=" UUID
UUID             = 1*( hex | "-" )
other-param      = ( token | ( token "=" ( token | quoted-string )))
headers          = "?" header *( "&" header )
header           = hname "=" hvalue
hname            = 1*uric
hvalue           = *uric
uric             = reserved | unreserved | escaped
reserved         = ";" | "/" | "?" | ":" | "@" | "&" | "=" | "+" | "$" | ","
digits           = 1*DIGIT
```

SIP URL Syntax (Cont'd)

- Host
  - FQDM (Fully Qualified Domain Name) or numeric IP address
- Port
  - The port number to send a request to
- URL parameters
  - Specific parameter of the request
  - If not present, special procedure required
- Headers
  - ? Mechanism
- Method
  - The method of the SIP request

SIP URL: Examples

sip:j.doe@big.com

sip:j.doe:secret@big.com;transport=tcp

sip:j.doe@big.com;maddr=239.255.255.1;ttl=15

sip:j.doe@big.com?subject=project%20x&priority=urgent

sip:+1-212-555-1212:1234@gateway.com;user=phone

sip:1212@gateway.com

sip:alice@10.1.2.3

sip:alice@example.com

sip:alice%40example.com@gateway.com

sip:alice@registrar.com;method=REGISTER

%20 => space
%40 => @

SIP URI Comparison (bis09)

- **이해하지 못하는** uri-parameter **포함시 무시** (MUST)
- disallowed uri-parameter **포함시에도 무시** (SHOULD)
- **Comparison**
  - SIP and SIPS URI : not equivalent
  - userinfo : case-sensitive, others: case-insensitive
  - “reserved”, “unsafe” set : “% HEX HEX”**와** equivalent

URI Comparison (Cont.)

- user, password, host, port : must match
  - **예: 한쪽에 명시적으로 5060 값이 존재하고, 다른 한쪽에 값이 없는 경우 : not match**
- transport-parameter, ttl-parameter, user-parameter, method, Header : must match
  - **예: 한쪽에 default 값이 명시가 되어있고, 다른쪽에는 값이 없는 경우 : not match**
- sip:bob@biloxi.com
- sip:bob@biloxi.com:5060

not match

- All others uri-parameter
  - **한쪽에 값이 존재하고 다른쪽에 없는 경우 무시**

URI Comparison - Example

- Equivalent
  - Domain name and parameter name is case-insensitive

    %65 => e

    sip: juser@%65xample.com

    sip: juser@ExAmPle.coM;Transport=UDp

- Not Equivalent
  - User name is case-sensitive

    sip: JUSER@ ExAmPle.coM;Transport=UDp

SIP Syntax - Usage

- Field names and some tokens (e.g., media type) are case-insensitive
- Everything else is case-sensitive
- White space doesn't matter except in first line
- Lines can be folded
- Multi-valued header fields can be combined as a comma-list

SIP Message Format

start-line
*message header
CRLF
[message body]

SIP Syntax

request	response
method *Request-URI* SIP/2.0	SIP/2.0 *status* *reason*

Via:	**SIP/2.0 protocol host:port**
From:	**user <sip:from_user@source>**
To:	**user <sip:to_user@destination>**
Call-ID:	**localid@host**
CSeq:	**seq# method**
Content-Length:	**length of body**
Content-Type:	**media type of body**
Header:	**parameter;par="value";par2="value"; par3="value folded into next line"**

Blank line

V= **0**
O= **Origin_user timestamp timestamp IN IN4 host**
C= **IN IN4 media destination address**
T= **0 0**
M= **Media type port RTP/AVP payload types**

Request-URI

- Description
  - SIP URL (user@host)
  - General URI [rfc 2396]
- The user or service to which this request is being addressed

Basic Operation

- Callers and callees are identified by SIP addresses
  - SIP URL
- When making a call, a caller
  - first locates the appropriate server
  - then send a SIP request
- The most common SIP operation is
  - invitation
- Instead of directly reaching the intended callee, a SIP request
  - may be redirected or
  - may trigger a chain of new SIP requests by proxies
- User can
  - register their location(s) with SIP servers

Locating a SIP Server

- Locally configured SIP proxy server
  - Request-URI independent
- Using Request-URI (sunchoi@anyang.ac.kr)
  1. If (host portion of the Request-URI == IP address)
     - Client contacts the server at the given IP address
  2. Otherwise
     ① If Request-URI contains an explicit port number, skip ② & ③
     ② If Request-URI does not specify a protocol (TCP or UDP),
        1) Client queries the name server for SRV records for both UDP (if supported by client) and TCP(if supported by client) servers.
        2) DNS does not return any records, goto step ④.
        3) DNS returns server lists, user attempts to contact each server in the order listed. If none of the server can be contacted, the user gives up.

Locating a Server (Cont'd)

③ If Request-URI specifies a protocol (TCP or UDP) that is supported by the client,

1) The client queries the name server for SRV records for SIP servers of that protocol type only.
2) If the client does not support the protocol specified in the Request-URI => it gives up.
3) DNS does not return any records, goto step ④
4) DNS returns the list of servers, the user attempts to contact each server in the order listed.
5) If no server is contacted, the user gives up.

Locating a Server (Cont'd)

④ The client queries the DNS server for address records for the host portion of the Request-URI

1) If the DNS server returns no address records

   ⇒ Client stops, it has been unable to locate a server.

2) The returned address records include A RR's, AAAA RR's, or other similar records, chosen according to the client's network protocol capabilities

DNS (Domain Name System)

- SRV RR(Resource Record)
  - RFC 2782, A DNS RR for specifying the location of services (DNS SRV)
  - Format

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

    - Service: the symbolic name of the desired service
    - Proto: the symbolic name of the desired protocol
    - Name: the domain this RR refers to.
    - TTL: time to live of the RR => cache
    - Class: IN(Internet Family), CH(Chaos Family)
    - Priority: the priority of the target host
      - » client must attempt to contact the target host with the lowest-number priority
    - Weight: server selection mechanism (within same priority)
    - Port
    - Target: the domain name of the target

SIP Transaction

- Occurs between a client and server
- Comprises all messages from the first request sent from the client to the server up to a final (non-1xx) response sent from the server the client
- Identified by the Cseq sequence number within a single leg.
  - Call leg: <Call-ID, To, From>

SIP Invitation

- Successful SIP invitation
  - INVITE -> ACK
  - BYE
- Session Description written in SDP [RFC 2327] format
  - Capability negotiation

SDP [RFC 2327]

- Session Information Description Format ?
- to *convey* information about media streams in multimedia sessions, to *allow* the recipients of a session description to participate in the session
- in multicast based sessions on the internet
  - a mean to communicate the *existence* of a session
  - a mean to convey sufficient information to enable

    joining and participating in the session

What is Session ?

- A multimedia session
  - A set of multimedia senders and receivers and the data streams flowing from senders and receivers
- A multimedia conference
  - An example of a multimedia session
- A session
  - Can comprise one or more RTP sessions
  - Origin field in SDP
    - <user name, session id, network type, address type, address>

SDP SPECIFICATION (I)

- Descriptions are Textual: <type>=<value>
  - session description
  - Time description
  - media description

Session Description

*: options

v= (protocol version)
o= (owner/creator and session identifier)
s= (session name)
i=* (session information)
u=* (URL of description)
e=* (email address)
p=* (phone number)
v= (connection information)
b=* bandwidth information)
t=* (zeor or more times)
k=* (encryption key)
a=* (zero or more session attribute lines)

SDP SPECIFICATION (II)

- Time description

```
t =   (time the session is active)
r =*  (zero or more repeat times)
```

- Media description

```
m=   (media name and transport address)
i=*  (media title)
c=*  (connection information)
b=*  (bandwidth information)
k=*  (encryption key)
a=*  (zero or more media attribute lines)
```

SDP DESCRIPTION - EXAMPLE

User id Session-id Version Network type Address type

```
v=  0
o=  khlee 2890844526 2890842087 IN IP4 126.16.64.4
s=  SIP Tutorial
I=  A Turorial on the VoIP Protocol
u=  http://www.miptel.com/presentation/sip.ps
e=  khlee@miptel.com
c=  IN IP4 224.2.17.12/127
t=  2873397496 2873404696
a=  recvonly
m=  audio 3456 RTP/AVT  4  18
a=  rtpmap:4 G723/8000
a=  rtpmap:18 G729/8000
m=  video 2232 RTP H261
m=  whiteboard 32146 UDP WB
a=  orient:landscape
```

Address

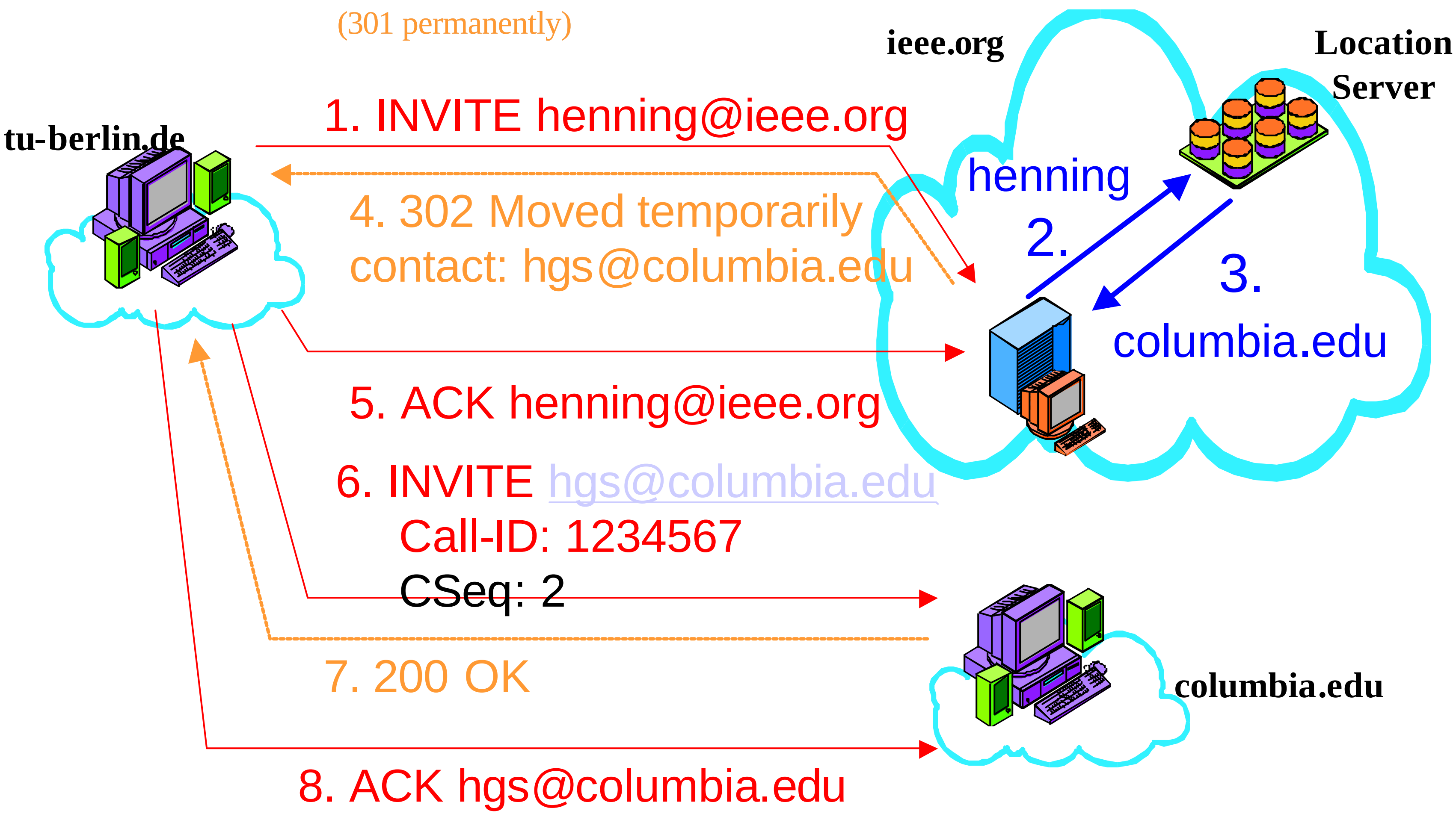
Connection info.

NTP timestamp

G.723

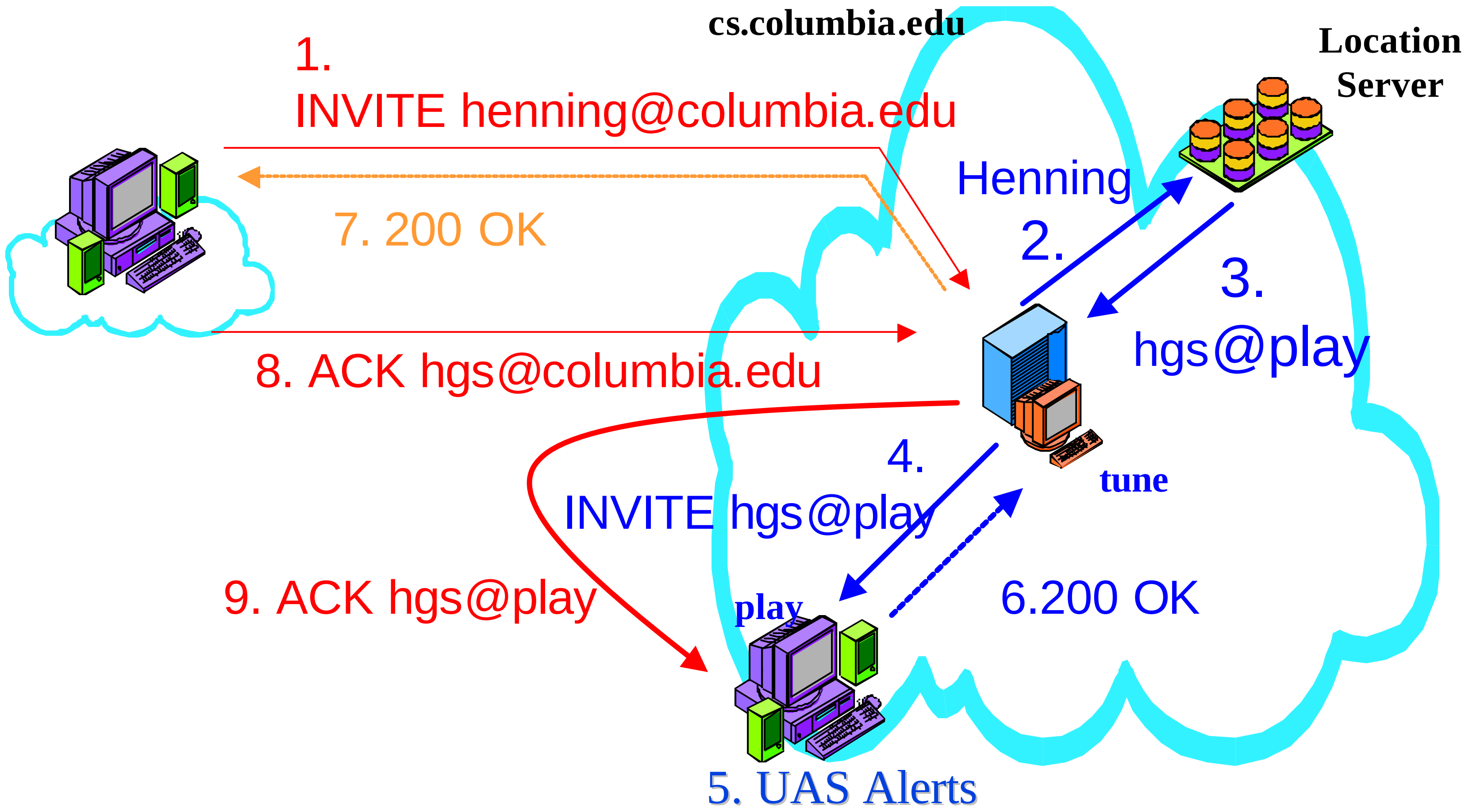
G.729

SIP Operation in Redirect Mode



SIP Operation in Proxy Mode

All requests and responses have the same Call-ID



Locating a User

- A callee may move between a number of different end systems over time
  - Locations can be dynamically registered.
  - Location server may use
    - Finger [RFC 1288]
    - Rwhois [RFC 2167]
    - LDAP [RFC 1777]
    - OS dependent mechanisms
  - Location server may return several locations
    - User is logged in at several hosts simultaneously
    - Location server has (temporarily) inaccurate information

Locating a User (Cont'd)

- Action on receiving a list of location
  - SIP redirect server
    - Returns **Contact** headers
    - Contact: sunchoi@chollian.net
  - SIP proxy server
    - Sequential try/Parallel try
    - Forwards SIP request
      - Add **Via** headers before the previous Via header
        - » **Via: SIP/2.0/UDP sip.ieee.org**
        - » **Via: SIP/2.0/UDP c.bell-tel.com**
      - Response path
        - » Each host removes its Via => internal information hidden from the callee

Locating a User (Cont'd)

– SIP proxy server (cont'd)
  - Forking
    – SIP invitation may traverse more than one SIP proxy server
    – Forks the request
      » Same Call-ID
      » User Agent MUST return the same status returned in the first response
      » Duplicate requests are not an error

Changing an Existing Session

- Change the parameters
  - INVITE
    - Same Call-ID
    - New or different body or header fields with the new information
    - Higher CSeq
  - Example: new member Join
    - INVITE (new multicast session description, old Call-ID)

SIP: Basic Operation

1. Use directory service (e.g., LDAP) to map name to user@domain
2. Locate SIP servers using DNS SRV, CNAME
3. Called server may map name to user@host using aliases, LDAP, canonical program
4. Callee accepts, rejects, forward (-> new address)
5. If new address, go to step 2
6. If accept, caller confirms
7. Conversation
8. Caller or callee sends BYE

SIP Message Overview

- Test-based protocol
- ISO 10646 character set in UTF-8 encoding
  - F. Yergeau, UTF-8, a transformation format of ISO 10646(Unicode), RFC 2279, Jan. 1998.
- Senders MUST terminate line with CRLF
- Receivers MUST also interpret CR and LF
- Except for the above difference in character sets and line termination, much of the message syntax is and header fields are identical to HTTP/1.1

SIP Message - BNF

```
generic message = start-line
                  * message-header
                  CRLF
                  [message-body]
start-line = Request-Line | Status-Line
message-header = (general-header | request-header | entity-header)
```

```
SIP-message = Request | Response
Request = Request-Line
          *(general-header) | request-header | entity-header)
          CRLF
          [message=body]
Response=Status-Line
          *(general-header) | response-header | entity-header)
          CRLF
          [message=body]
```

Header Field Format

message-header = field-name ":" [field-value] CRLF
field-name = token
field-value = *(field-content | LWS)
field-content = <the OCTETs making up the field-value
and consisting of either *TEXT-UTF8 or
combinations of token, separators,
and quoted-string>

LWS: Leading White Space

SIP Request Message

Start - LIne	Method **Request-Line**	INVITE | ACK | OPTIONS | BYE | CANCEL | REGISTER space Request-URI space SIP-version		
		CRLF		
***(message header)**		**General Header** Accept | Accept-Encoding | Accept-Language | Call-ID | Contact | CSeq | Date | Encryption | Expires | From | Record-Route | Require | Supported | Timestamp | To | User Agent | Via	**Request Header** Authorization | Contact | Hide | Max-Forwards | Organization | Priority | Proxy-Authorization | Proxy-Require | Route | Require | Response-Key | Subject |	**Entity Header** Allow Content-Encoding | Content-Length | Content-Type Expires
			CRLF	
[message body]		**[BODY (SDP)]**		

SIP Request Message

- Many header fields from http
- New ones (Also, Replaces, Via) are SIP specific
- Payload contains a media description

```
INVITE ann@lucent.com SIP/2.0
Via: SIP/2.0/UDP bell-labs.com:5060
From: jdrosen@bell-labs.com (Jonathan
Rosenberg)
Subject: SIP will be discussed, too
To: ann@lucent.com (Arun Netravali)
Call-ID: 1997234505@bell-labs.com
Content-type: application/sdp
CSeq: 4711
Content-Length: 187

v=0
o=user1 53655765 2353687637 IN IP4
128.3.4.5
s=Mbone Audio
i=Discussion of Mbone Engineering
Issues
e=mbone@somewhere.com
c=IN IP4 224.2.0.1/127
t=0 0
m=audio 3456 RTP/AVP 0
```

SIP Response Message

Start - LIne	**Status-Line**	SIP Version space status-code (1xx|2xx|3xx|4xx|5xx|6xx) space *<TEXT-UTF8, excluding CR, LF> **CRLF**		
***(message header)**	**General Header** Accept | Accept-Encoding | Accept-Language | Call-ID | Contact | CSeq | Date | Encryption | Expires | From | Record-Route | Require | Supported | Timestamp | To | User-Agent | Via	**Response Header** Proxy-Authorization | Retry-After | Server | Unsupported | Warning | WWW-Authenticate	**Entity Header** Allow | Content-Encoding | Content-Length | Content-Type Expires |	
		CRLF		
[message body]		**[BODY (SDP)]**		

KRnet2002

SIP Response Message

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP here.com:5060
From: BigGuy <sip:UserA@here.com>
To: LittleGuy <sip:UserB@there.com>
Call-ID: 12345601@here.com
CSeq: 1 INVITE
Contact: LittleGuy <sip:UserB@there.com>
Content-Type: application/sdp
Content-Length: 134

v=0
o=UserB 2890844527 2890844527 IN IP4 there.com
s=Session SDP
c=IN IP4 110.111.112.113
t=0 0
m=audio 3456 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

Status Code: 1xx, 2xx, 3xx

```
Informational  =  "100"  ;  Trying
                |  "180"  ;  Ringing
                |  "181"  ;  Call Is Being Forwarded
                |  "182"  ;  Queued
                |  "183"  ;  Session Progress

Success      =  "200"  ;  OK
Redirection  =  "300"  ;  Multiple Choices
                |  "301"  ;  Moved Permanently
                |  "302"  ;  Moved Temporarily
                |  "303"  ;  See Other
                |  "305"  ;  Use Proxy
                |  "380"  ;  Alternative Service
```

Status Code: 4xx

```
Client-Error  =   "400"  ;  Bad Request
              |  "401"  ;  Unauthorized
              |  "402"  ;  Payment Required
              |  "403"  ;  Forbidden
              |  "404"  ;  Not Found
              |  "405"  ;  Method Not Allowed
              |  "406"  ;  Not Acceptable
              |  "407"  ;  Proxy Authentication Required
              |  "408"  ;  Request Timeout
              |  "409"  ;  Conflict
              |  "410"  ;  Gone
              |  "411"  ;  Length Required
              |  "413"  ;  Request Entity Too Large
              |  "414"  ;  Request-URI Too Large
              |  "415"  ;  Unsupported Media Type
```

Status Code: 4xx (Cont'd)

- "420" ; Bad Extension
- "480" ; Temporarily not available
- "481" ; Call Leg/Transaction Does Not Exist
- "482" ; Loop Detected
- "483" ; Too Many Hops
- "484" ; Address Incomplete
- "485" ; Ambiguous
- "486" ; Busy Here

Status Code: 5xx, 6xx

```
Server-Error  =  "500"  ;  Internal Server Error
                 |  "501"  ;  Not Implemented
                 |  "502"  ;  Bad Gateway
                 |  "503"  ;  Service Unavailable
                 |  "504"  ;  Gateway Time-out
                 |  "505"  ;  SIP Version not supported

Global-Failure =  "600"  ;  Busy Everywhere
                 |  "603"  ;  Decline
                 |  "604"  ;  Does not exist anywhere
                 |  "606"  ;  Not Acceptable
```

Header: Call and Request Identification

- Call-ID: globally(time, space) unique call identifier
- To: logical call destination
- From: call source
- CSeq: request within call leg
- Call leg = Call-ID + To + From

Header Type

- General header field
  - Request & response message
  - Change in the protocol version
- Entity header field
  - Meta-information about the message body
- Request header field
  - Client passes additional information about the request to the server
  - Request modifier => like parameter of a programming language

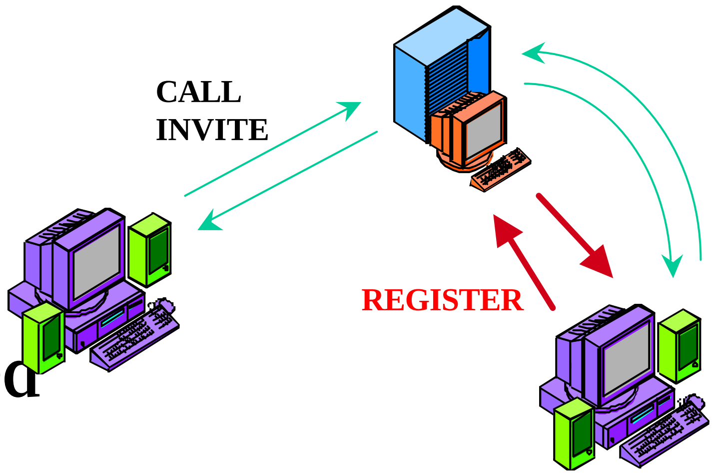
Header Type (Cont.)

- Response header field
  - server passes additional information about the response which cannot be Status-Line
- End-to-End headers and Hop-by-Hop headers
  - End-to-end headers: MUST be transmitted unmodified all proxies
  - Hop-by-hop headers: MAY be modified or added by proxies

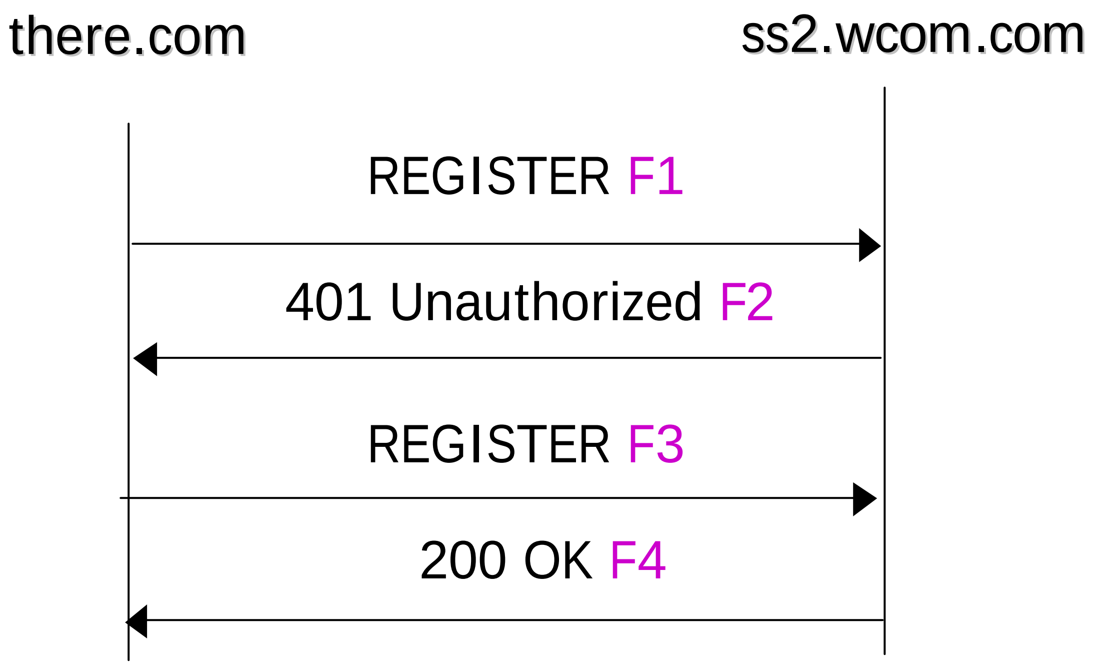
Registration

- REGISTER message allows clients to inform proxy or redirect server of my address (presence)
- on startup, send registration to sip.mcast.net via multicast

registration expire - determined by server
returns list of current registrations
registrations may be proxied
$\Rightarrow$ mobility



REGISTER example



REGISTER F1

F1 REGISTER there.com -> ss2.wcom.com

```
REGISTER sip:ss2.wcom.com SIP/2.0
Via: SIP/2.0/UDP there.com:5060
From: LittleGuy <sip:UserB@there.com>
To: LittleGuy <sip:UserB@there.com>
Call-ID: 123456789@here.com
CSeq: 1 REGISTER
Contact: LittleGuy <sip:UserB@there.com>
Contact: sip:+1-972-555-2222@gw1.wcom.com;user=phone
Contact: tel:+1-972-555-2222
Content-Length: 0
```

REGISTER F2

F2 401 Unauthorized ss2.wcom.com -> there.com

SIP/2.0 401 Unauthorized
Via: SIP/2.0/UDP there.com:5060
From: LittleGuy <sip:UserB@there.com>
To: LittleGuy <sip:UserB@there.com>
Call-ID: 123456789@here.com
CSeq: 1 REGISTER
WWW-Authenticate: Digest realm="MCI WorldCom SIP",
domain="wcom.com",
nonce="ea9c8e88df84f1cec4341ae6cbe5a359",
opaque="", stale="FALSE",
algorithm="MD5"
Content-Length: 0

REGISTER F3

F3 REGISTER there.com -> ss2.wcom.com

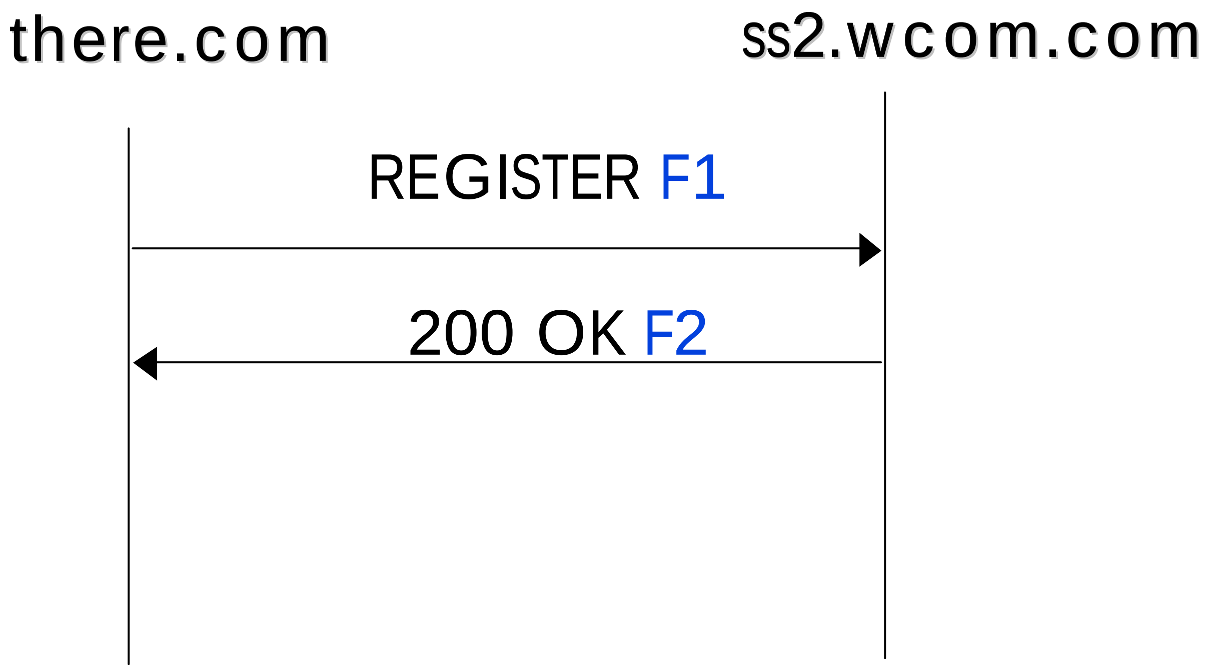
REGISTER sip:ss2.wcom.com SIP/2.0
Via: SIP/2.0/UDP there.com:5060
From: LittleGuy <sip:UserB@there.com>
To: LittleGuy <sip:UserB@there.com>
Call-ID: 123456790@here.com
CSeq: 1 REGISTER
Contact: LittleGuy <sip:UserB@there.com>
Contact: sip:+1-972-555-2222@gw1.wcom.com;user=phone
Contact: tel:+1-972-555-2222
Authorization:Digest username="UserB",
realm="MCI WorldCom SIP",
nonce="ea9c8e88df84f1cec4341ae6cbe5a359",
opaque="",uri="sip:ss2.wcom.com",
response="dfe56131d1958046689cd83306477ecc"
Content-Length: 0

REGISTER F4

F4 200 OK ss2.wcom.com -> there.com

SIP/2.0 200 OK
Via: SIP/2.0/UDP there.com:5060
From: LittleGuy <sip:UserB@there.com>
To: LittleGuy <sip:UserB@there.com>
Call-ID: 1234567890@here.com
CSeq: 1 REGISTER
Contact: LittleGuy <sip:UserB@there.com>
Contact: sip:+1-972-555-2222@gw1.wcom.com;user=phone
Contact: tel:+1-972-555-2222
Content-Length: 0

Register:
User Updates Contact List



REGISTER Update F1

F1 REGISTER there.com -> ss2.wcom.com

REGISTER sip:ss2.wcom.com SIP/2.0
Via: SIP/2.0/UDP there.com:5060
From: LittleGuy <sip:UserB@there.com>
To: LittleGuy <sip:UserB@there.com>
Call-ID: 123456791@here.com
CSeq: 1 REGISTER
Contact: mailto:UserB@there.com
Authorization:Digest username="UserB",
realm="MCI WorldCom SIP",
nonce="1cec4341ae6cbe5a359ea9c8e88df84f",
opaque="",uri="sip:ss2.wcom.com",
response="71ba27c64bd01de719686aa4590d5824"
Content-Length: 0

REGISTER Update F2

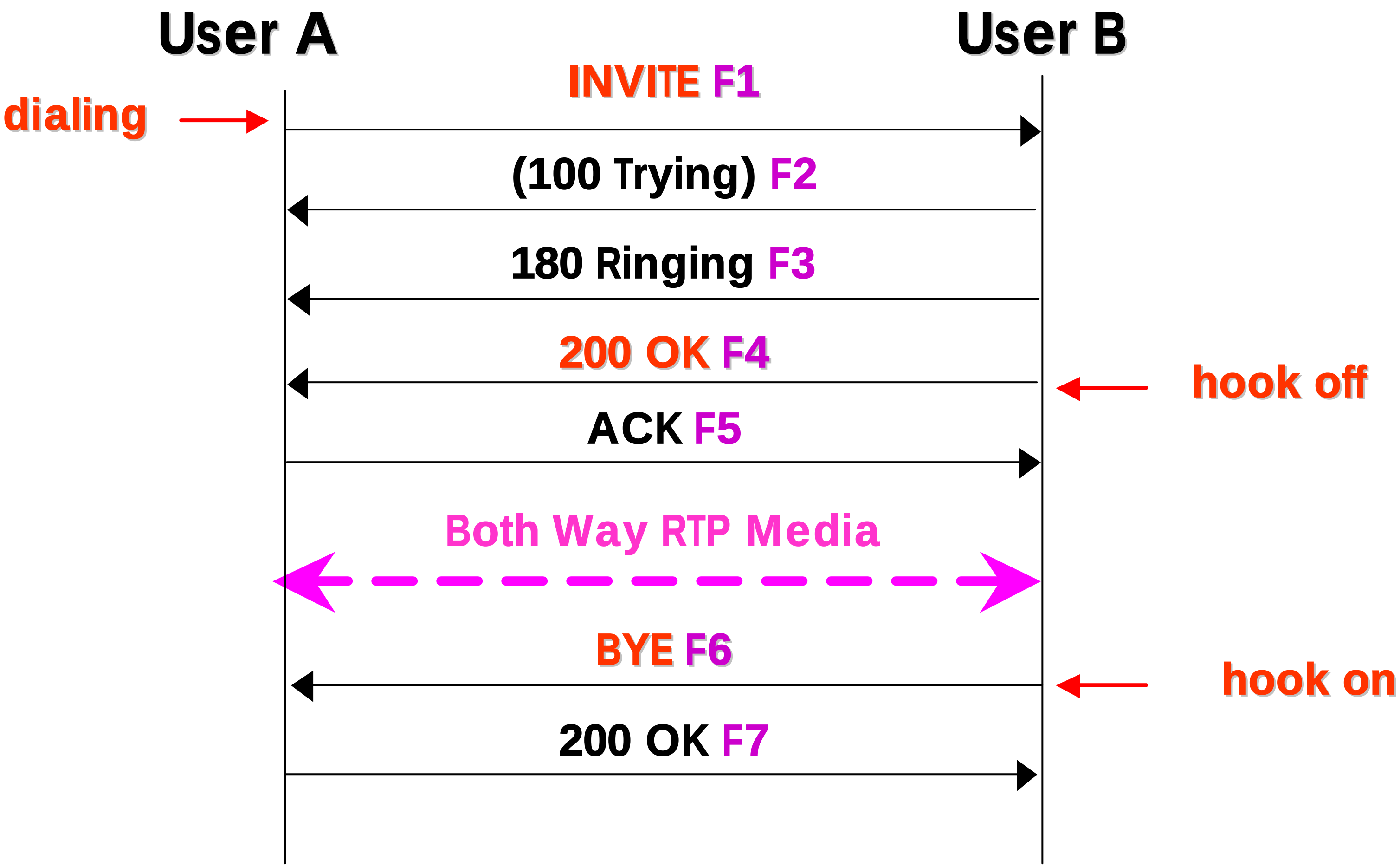
F2 200 OK ss2.wcom.com -> there.com

SIP/2.0 200 OK
Via: SIP/2.0/UDP there.com:5060
From: LittleGuy <sip:UserB@there.com>
To: LittleGuy <sip:UserB@there.com>
Call-ID: 1234567891@here.com
CSeq: 1 REGISTER
Contact: LittleGuy <sip:UserB@there.com>
Contact: sip:+1-972-555-2222@gw1.wcom.com;user=phone
Contact: tel:+1-972-555-2222
Contact: mailto:UserB@there.com
Content-Length: 0

SIP to SIP Dialing

- User A
  - BigGuy
  - sip:UserA@here.com
  - Registrar -> ss1.wcom.com
- User B
  - LittleGuy
  - sip:UserB@there.com
  - Registrar -> ss2.wcom.com
- Assumption
  - Call route using at least one SIP Proxy server
  - SIP digest authentication is used by the first Proxy Server to authenticate the caller user A

Simple Successful Call Flow



Successful Simple SIP to SIP: F1

F1 INVITE A -> Proxy 1 (ss1.wcom.com)

INVITE sip:UserB@ss1.wcom.com SIP/2.0
Via: SIP/2.0/UDP here.com:5060
From: BigGuy <sip:UserA@here.com>
To: LittleGuy <sip:UserB@there.com>
Call-ID: 12345600@here.com
CSeq: 1 INVITE
Contact: BigGuy <sip:UserA@here.com>
Content-Type: application/sdp
Content-Length: 147

v=0
o=UserA 2890844526 2890844526 IN IP4 here.com
s=Session SDP
c=IN IP4 100.101.102.103
t=0 0
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000

Successful Simple SIP to SIP: F2, F3

F2 (100 Trying) User B -> User A

SIP/2.0 100 Trying
Via: SIP/2.0/UDP here.com:5060
From: BigGuy <sip:UserA@here.com>
To: LittleGuy <sip:UserB@there.com>
Call-ID: 12345600@here.com
CSeq: 1 INVITE
Content-Length: 0

F3 180 Ringing User B -> User A

SIP/2.0 180 Ringing
Via: SIP/2.0/UDP here.com:5060
From: BigGuy <sip:UserA@here.com>
To: LittleGuy <sip:UserB@there.com>
Call-ID: 12345600@here.com
CSeq: 1 INVITE
Content-Length: 0

Successful Simple SIP to SIP: F4

F4 200 OK User B -> User A

SIP/2.0 200 OK
Via: SIP/2.0/UDP here.com:5060
From: BigGuy <sip:UserA@here.com>
To: LittleGuy <sip:UserB@there.com>
Call-ID: 12345600@here.com
CSeq: 1 INVITE
Contact: LittleGuy <sip:UserB@there.com>
Content-Type: application/sdp
Content-Length: 134

v=0
o=UserB 2890844527 2890844527 IN IP4 there.com
s=Session SDP
c=IN IP4 110.111.112.113
t=0 0
m=audio 3456 RTP/AVP 0
a=rtpmap:0 PCMU/8000

Successful Simple SIP to SIP: F5

F5 ACK User A -> User B

ACK sip:UserB@ss1.wcom.com SIP/2.0
Via: SIP/2.0/UDP here.com:5060
From: BigGuy <sip:UserA@here.com>
To: LittleGuy <sip:UserB@there.com>
Call-ID: 12345600@here.com
CSeq: 1 ACK
Content-Length: 0

RTP Streams Between A and B

Successful Simple SIP to SIP: F6, F7

F6 BYE User B -> User A

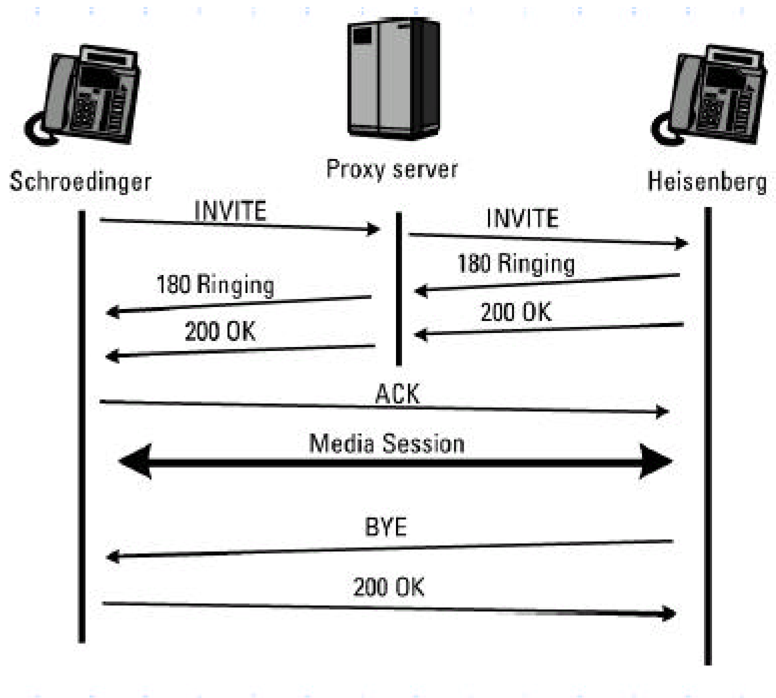
BYE sip: UserA@ss2.wcom.com SIP/2.0
Via: SIP/2.0/UDP there.com:5060
From: LittleGuy <sip:UserB@there.com>
To: BigGuy <sip:UserA@here.com>
Call-ID: 12345601@here.com
CSeq: 1 BYE
Content-Length: 0

F7 200 OK User A -> User B

SIP/2.0 200 OK
Via: SIP/2.0/UDP there.com:5060
From: LittleGuy <sip:UserB@there.com>
To: BigGuy <sip:UserA@here.com>
Call-ID: 12345600@here.com
CSeq: 1 BYE
Content-Length: 0

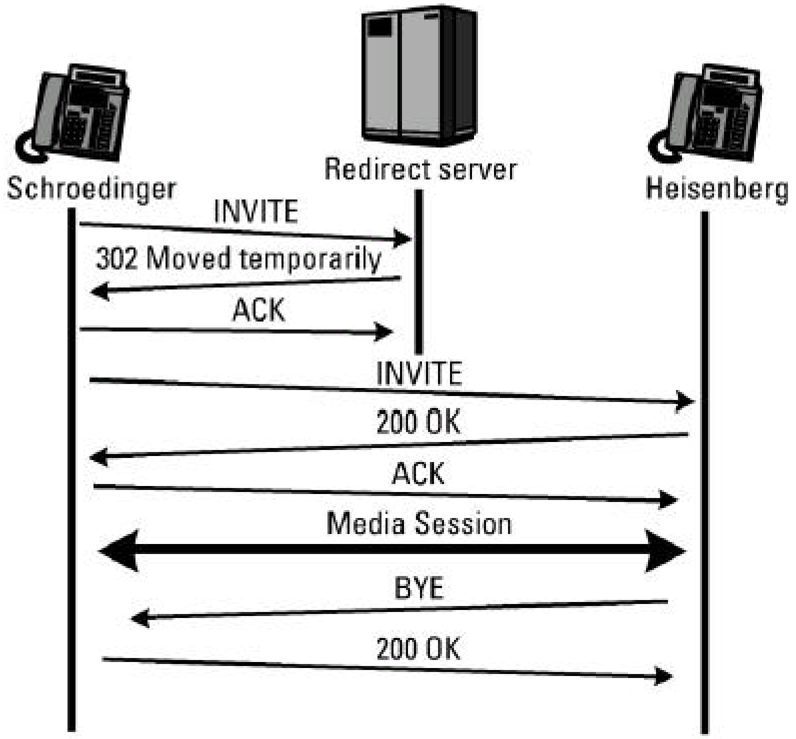
Call Flow with Proxy

- **INVITE** is sent to Proxy instead of to Heisenberg directly.
- Proxy looks up address of Heisenberg and forwards **INVITE** to that IP Address.
- Responses to INVITE route back through the Proxy: **180 Ringing** and **200 OK**
- **200 OK** contains a **Contact** header which allows the **ACK** and all future requests to go directly bypassing Proxy.

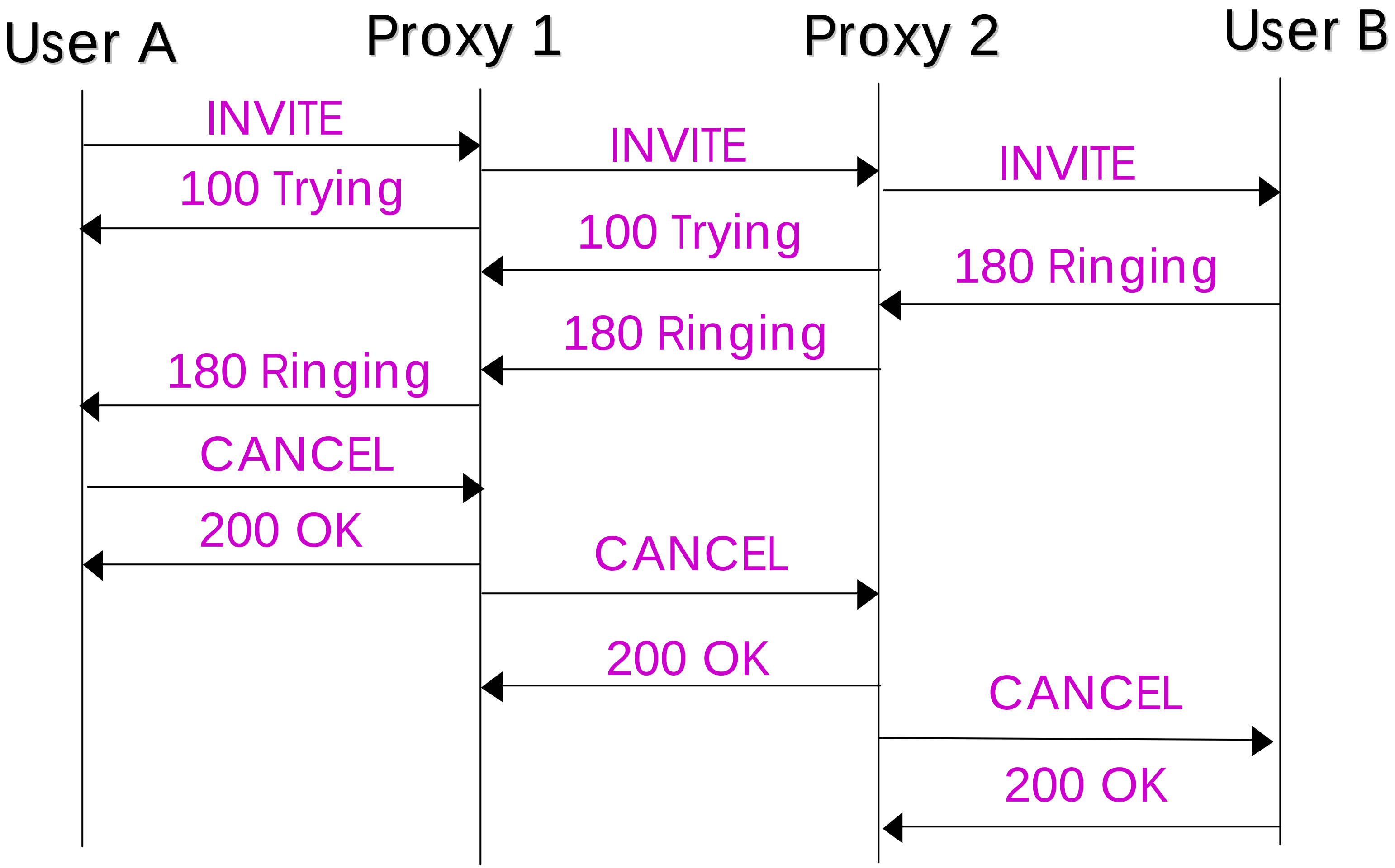


Call Flow with Redirect

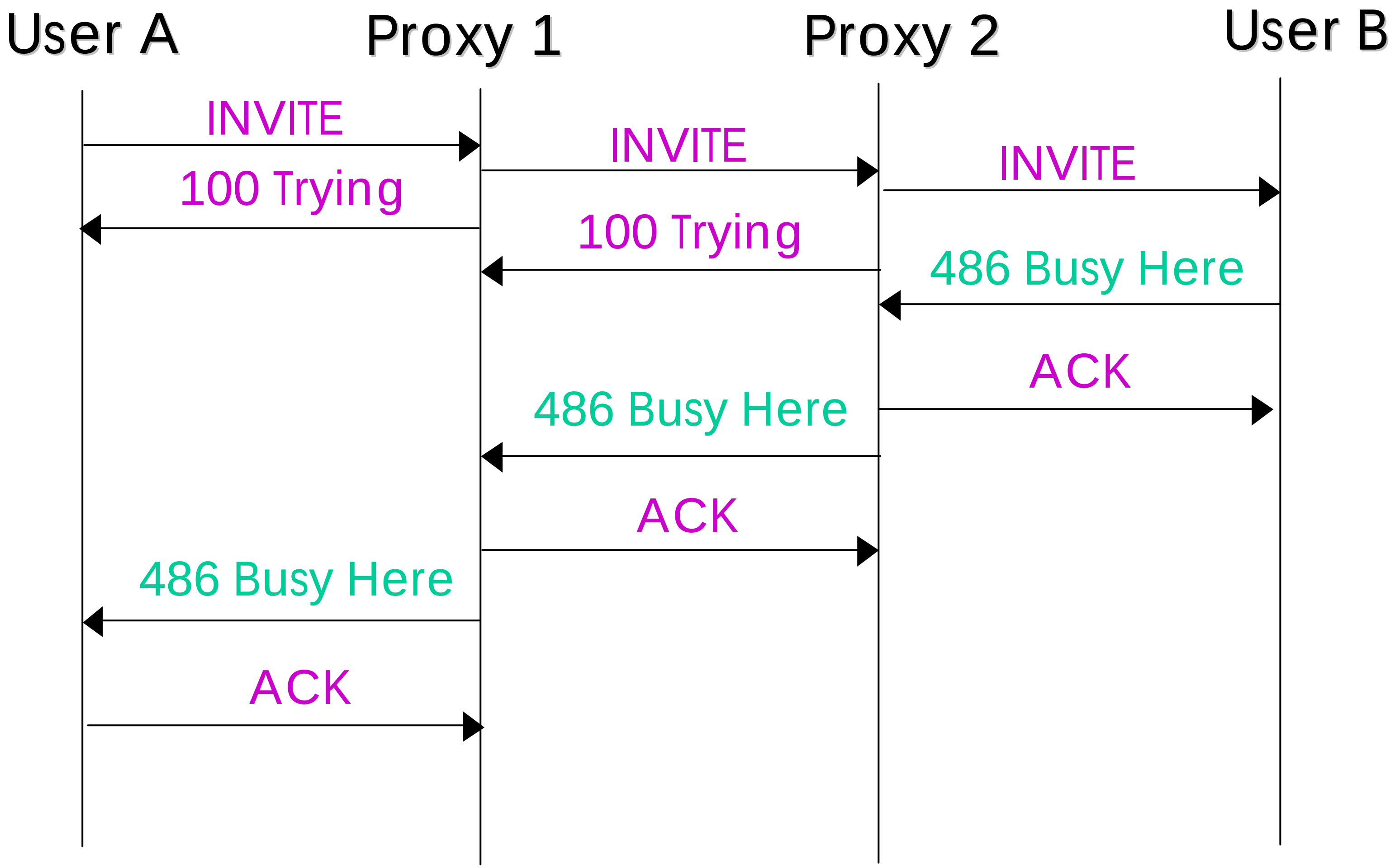
- **INVITE** is sent to Redirect Server
- Server looks up address of Heisenberg and returns that address in a **Contact** header in a **302 Moved Temporarily** response
- The **ACK** completes the transaction with the Server Schroedinger then re-sends the **INVITE** directly to Heisenberg



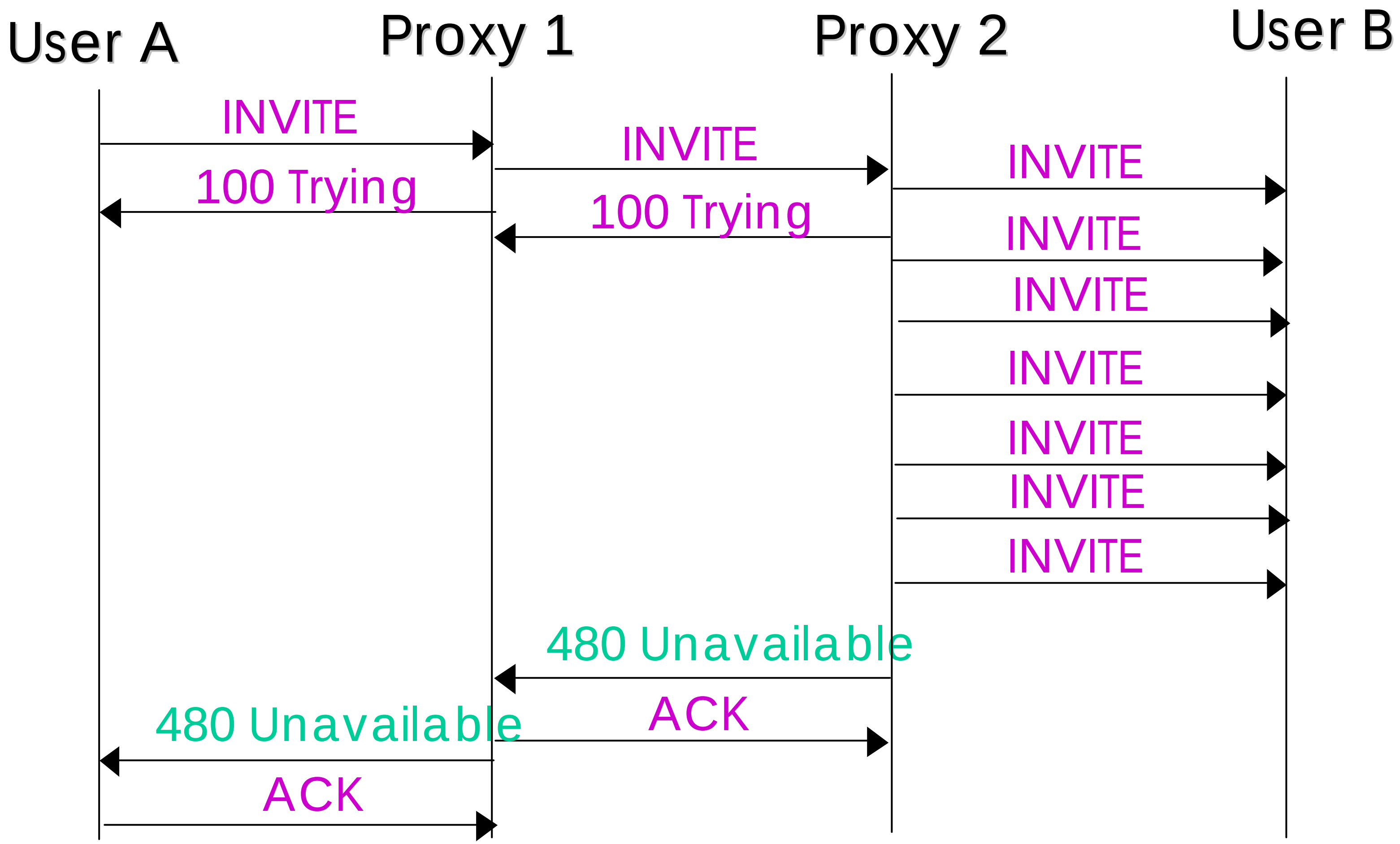
Unsuccessful Call Flow – No Answer



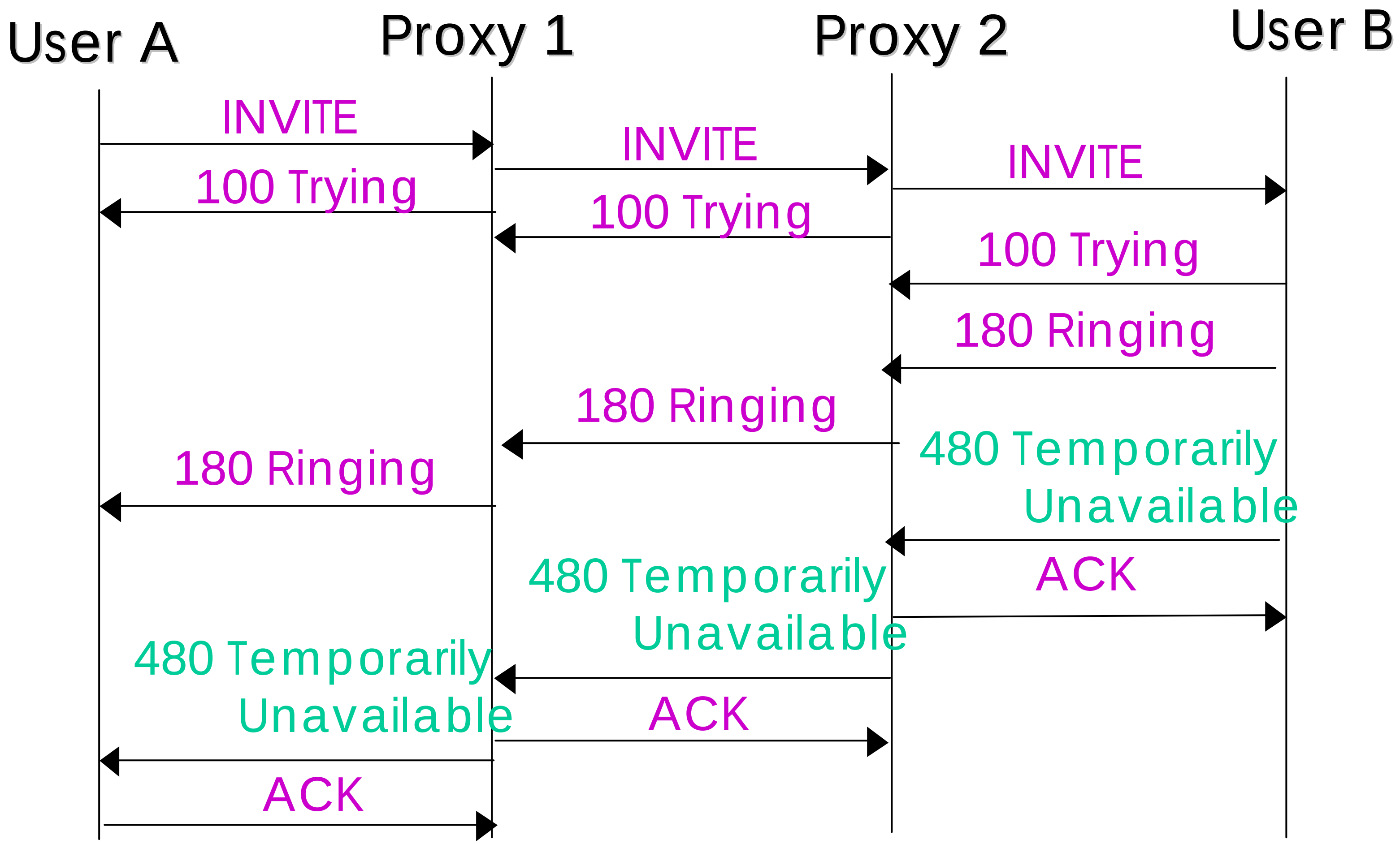
Unsuccessful Call Flow – Busy



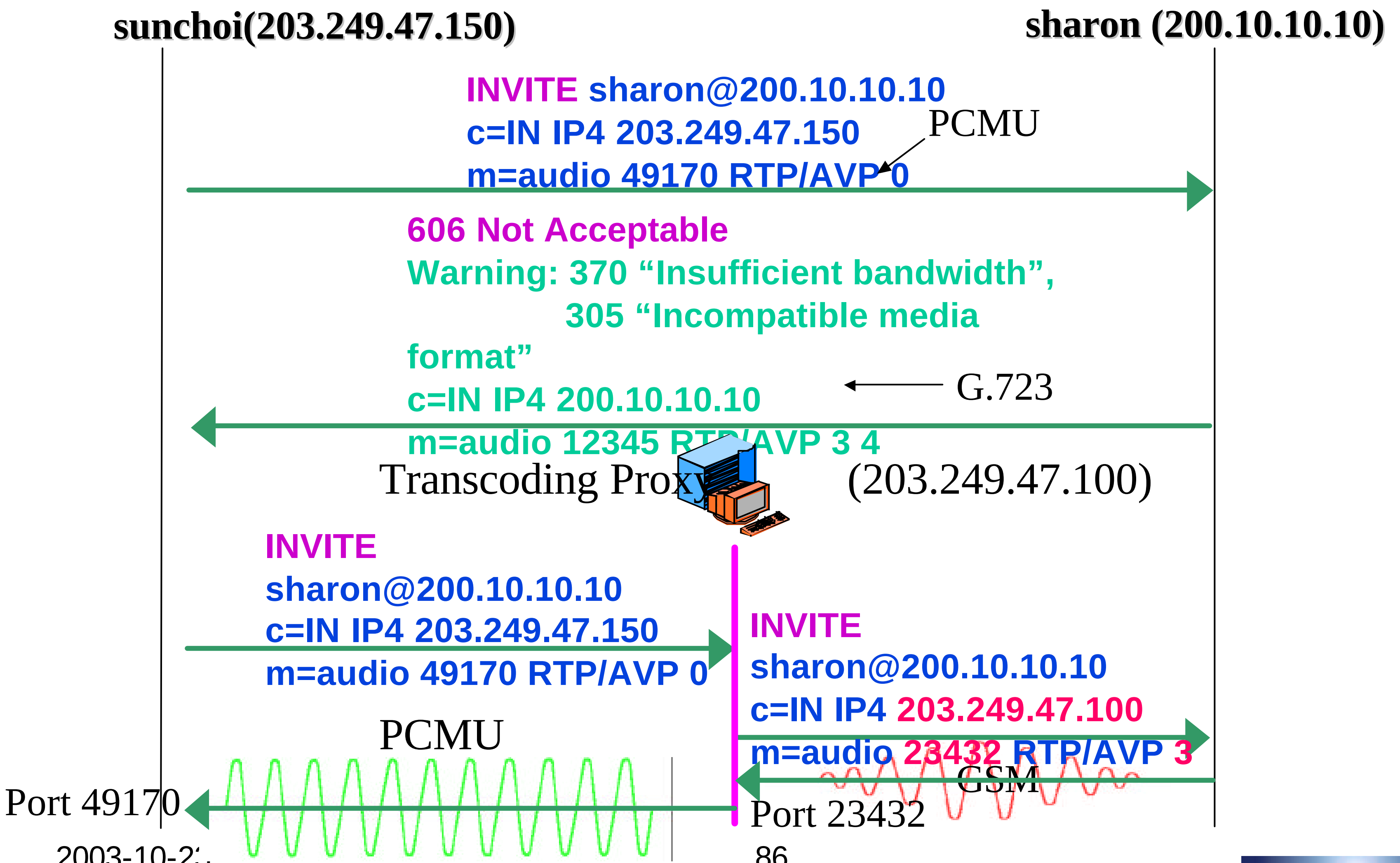
Unsuccessful Call Flow – No Response



Unsuccessful Call Flow – Temporarily Unavailable



Codec Negotiation

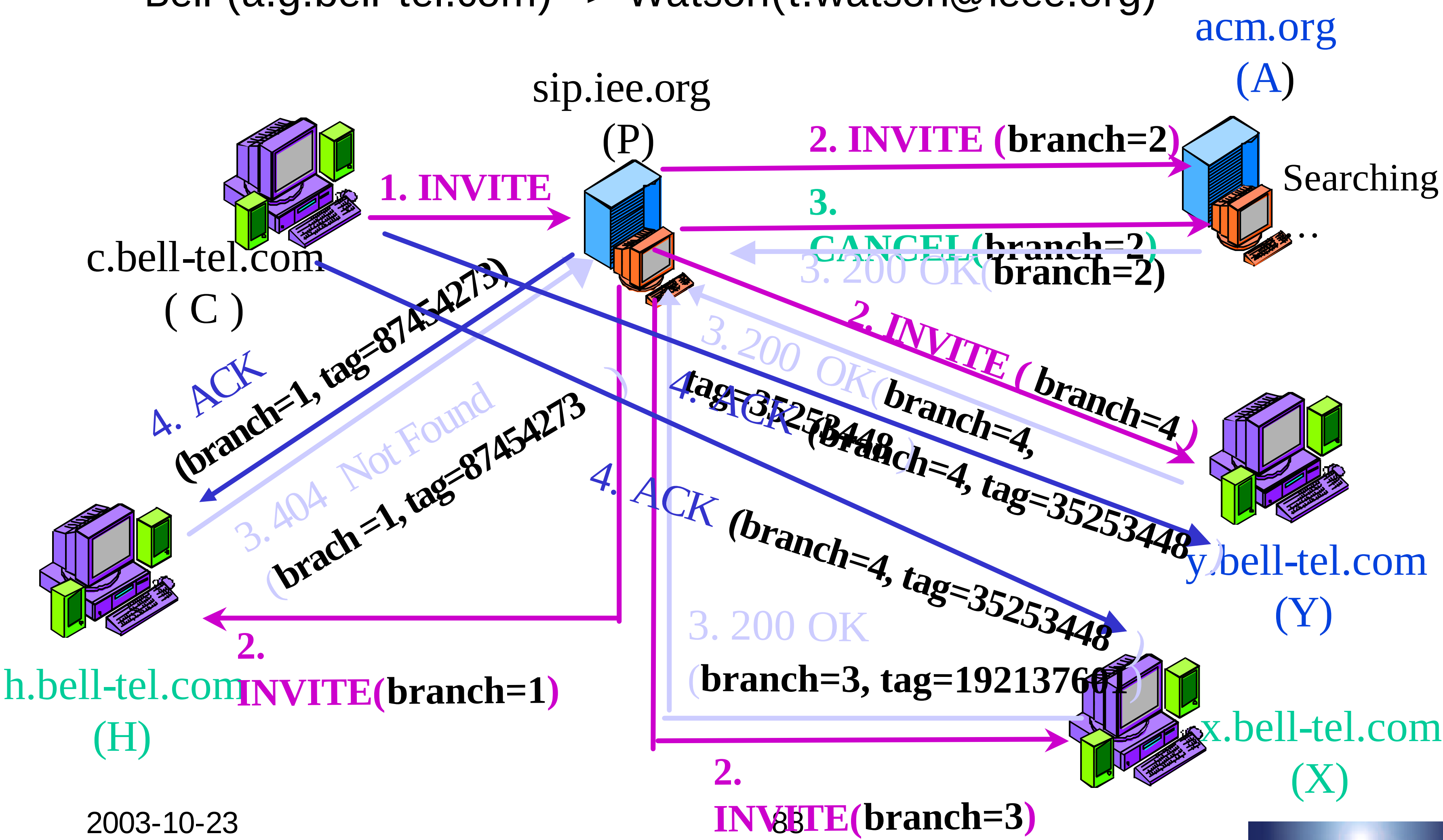


Forking Proxy

- Bell
  - a.g.bell@bell-tel.com
  - Currently seated at host c.bell-tel.com => C
- Watson
  - t.watson@ieee.org
  - Registrar: sip.ieee.org => P
  - Currently, logged in at two workstations
    - t.watson@x.bell-tel.com => X
    - t.watson@y.bell-tel.com => Y
  - Registered location in sip.ieee .org
    - t.watson@x.bell-tel.com
    - t.watson@y.bell-tel.com
    - watson@h.bell-tel.com (home machine) => H
  - Permanent registration
    - watson@acm.org (permanent registration) => A

Forking Proxy Example

- Bell (a.g.bell-tel.com) -> Watson(t.watson@ieee.org)



Redirects

P->C: SIP/2.0 302 Moved temporarily

Via: SIP/2.0/UDP c.bell-tel.com

From: A. Bell <sip:a.g.bell@bell-tel.com>

To: T. Watson <sip:t.watson@ieee.org>;tag=72538263

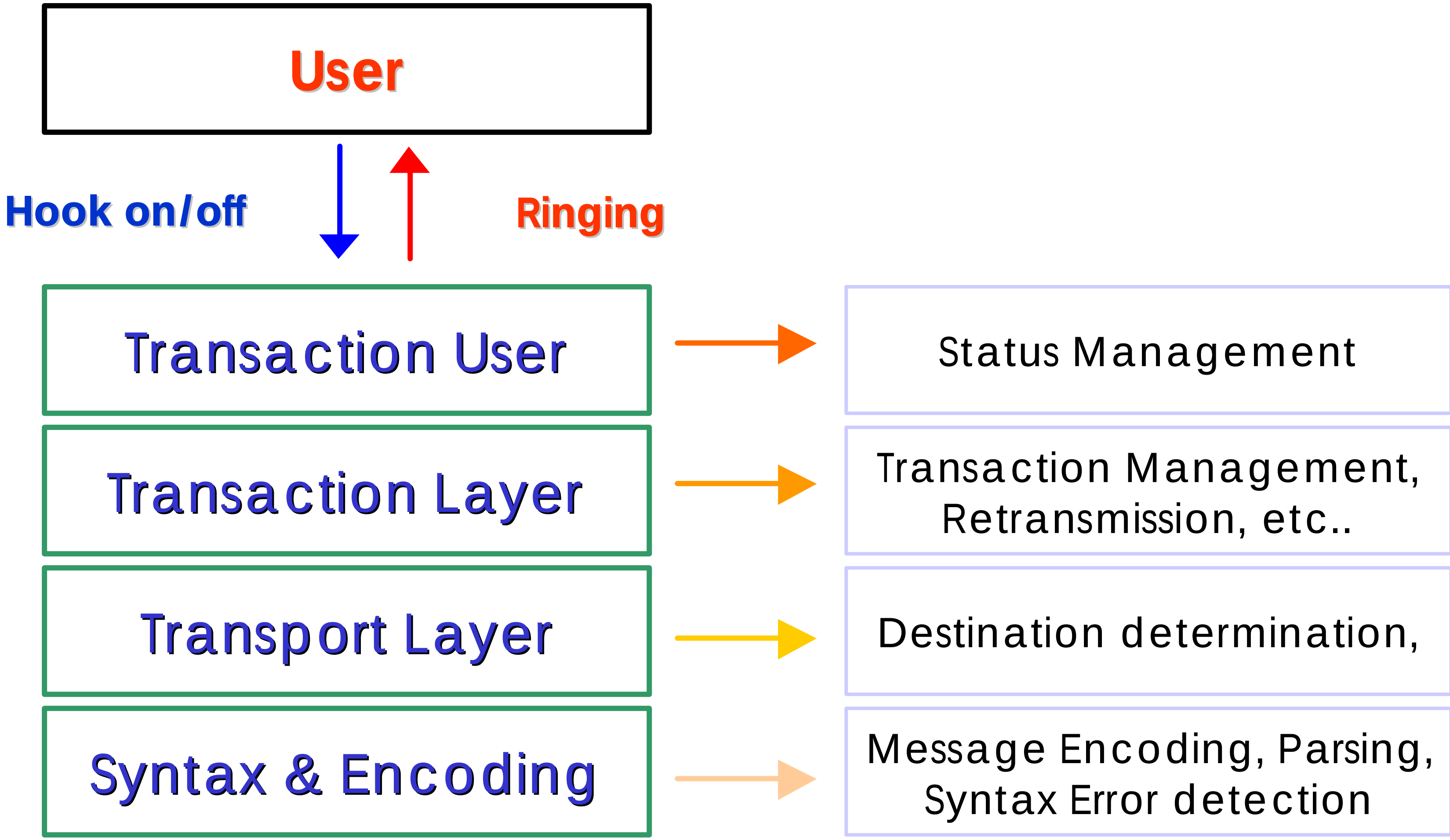
Call-ID: 31415@c.bell-tel.com

CSeq: 1 INVITE

Contact: sip:watson@h.bell-tel.com,
sip:watson@acm.org, sip:t.watson@x.bell-tel.com,
sip:watson@y.bell-tel.com

CSeq: 1 INVITE

SIP Structure - Layer Protocol



Record-Route (Cont.)

- REGISTER 메시지에는 삽입 불가 (MUST)
  - Registrar는 삽입된 Record-Route 무시 (MUST)
- 특정 request가 아닌 dialog 영역 내에서 유효
- 예제
  - Record-Route: <sip:p4.domain.com;lr>

Route

- pre-existing route set
  - single URI 포함
- Route 헤더 처리에서 SRV 사용
- **non-2xx에 대한 ACK, CANCEL은 original Request의 Route 헤더를 따름**
- **2xx에 대한 ACK은 2xx의 Record-Route를 따름** (bis09)
- stateless proxy에서도 CANCEL에 route processing logic 적용
- 예제
  - **Route: <sip:p4.domain.com;lr>**

Determination Target

- Sending the Request
  - DNS procedure 적용
  - route set의 first element가 strict route이면 request의 Request-URI에 적용되어야만 한다. (MUST)
  - otherwise, Route 헤더의 첫번째 값이 적용되거나 Route가 없으면 Request-URI가 적용된다.
  - Request-URI가 SIPS URI이면, TLS(Transport Layer Security)로 contact해야 한다. (MUST)

Determination Target (Cont.)

- Discovering a Registrar
  - Configuration
  - address-of-record
    - Request-URI의 address-of-record의 host part를 사용해야만 함 (SHOULD)
  - multicast
  - well-known "all SIP servers"인 "sip.mcast.net"으로 보냄

Call, Transaction identifier

- From, To, Call-ID ➔ Call-Leg
  - Call identifier로 사용
- Tag
  - Call-ID와의 조합으로 **dialog identify**
  - single request로부터의 **multiple dialog** 가능때문에 필요
  - **fault tolerant system에 도움**
- CSeq
  - Transaction identifier로 사용

Call, Transaction identifier (Cont.)

- Via의 branch parameter
  - request에 의해 생성된 그 **transaction을 identify** 하기 위해 사용
  - time and space에 unique
    - 예외 : CANCEL, ACK message
  - **non-2xx에 대한 ACK, CANCEL은 INVITE와 동일한 branch 사용**
- Others
  - Request-URI, Via 필드도 transaction을 위해 검사
  - 재전송, Loop Detect 검출

Transaction 관리

- Transaction
  - Request에 대한 final response까지
  - 예외: INVITE에 대한 non-2xx응답과 그에 대한 ACK은 한 transaction으로 관리 → **Transaction Layer에서 처리**
  - INVITE에 대한 2xx 응답과 그에 대한 ACK은 다른 transaction으로 관리 → **Transaction User에서 처리**
- ACK Request Processing
  - ACK Request에 대한 응답은 없음 (MUST)
- CANCEL Request Processing
  - **CANCEL을 보내기 전에 1xx (Provisional Response) 를 기다려야함 (MUST)**
  - CANCEL, INVITE transaction 분리
    - 다른 transaction을 생성, 유지하여야 함

Transaction 관리 (Cont.)

- BYE processing
  - **2xx에 대한 ACK을 받지 못하는 경우 UAS는 BYE를 보냄**
  - **UAC가 2xx를 받았을 경우에만 BYE를 보낼 수 있음**
  - tag 없는 BYE reject 가능
- non-INVITE Processing
  - 1xx 응답 생성하지 않음 (discouraged)
  - non-INVITE request는 처리를 바로 해야함
- 604 Does Not Exist Anywhere응답 처리
  - To가 아닌 Request-URI에 기초

Transaction 관리 (Cont.)

- Proxy Transaction
  - 101-199 응답 처리
    - **upstream으로 forwarding해야함** (MUST)
  - 6xx 응답 처리
    - proxy는 6xx를 받자마자 즉시 포워딩하지 않음. 대신에 pending branch를 즉시 CANCEL 함
    - 모든 branch CANCEL후에 6xx 응답 upstream으로 포워딩
  - 503 Service Unavailable 응답 처리
    - 503 응답만을(only) 수신하였을 경우에는 500 Server Internal Error 응답을 생성하여 upstream으로 포워딩 (SHOULD)

Transaction 관리 (Cont.)

- 즉시 upstream으로 포워딩하는 응답
    - **Any provisional response other than 100**
    - **Any 2xx response**
- Any final 응답이 Server에서 보내진 후에도 **Any 2xx response to an INVITE request**는 포워딩해야만 함 (MUST)

- Transaction 유지
    - client, server transaction은 Timer에 의존하여 유지
- Transport Layer 지원
    - proxy, redirect, registrar는 **TLS**, TCP, UDP 지원 (MUST)
    - UA도 TCP 지원 (SHOULD)

Capability & Negotiation

- OPTIONS 처리 proxy에서도 가능
  - Request-URI가 userinfo가 생략된sip:domain.com 형태인 경우
- Negotiation
  - draft-ietf-mmusic-sdp-offer-answer-02.txt 참조
  - offer에서 사용된 순서와 같은 순서로 응답
  - offer/answer model 구조
  - Initial offer가 **INVITE**인 경우
    - answer는 reliable non-failure message가 되어야만 한다 (MUST), 이 specification에서는 2xx 응답만 가능

Capability & Negotiation (Cont.)

- 같은 exact answer가 그 answer에 우선해서 보내진 provisional response에 놓여질 수도 있다. (MAY)
- UAC는 answer에게 받은 첫번째 session description만을 다루어야 한다(MUST)
- 그리고 initial INVITE로의 이어지는 응답의 session description은 무시 (MUST)

– initial offer가 **200**인 경우

- answer는 ACK request가 되어야 함
- UAS는 answer에게 받은 첫번째 session description만을 다루어야 한다(MUST)

REGISTER

- Expires Header and parameter
  - **relative value 사용** : delta value
  - **absolute value, malformed value** : 3600 으로 인식 (SHOULD)
  - **absent** : locally-configurd default value 사용 (MUST)
- Call-ID
  - **single boot cycle동안 같은 Call-ID 사용** (SHOULD)
  - delayed request의 순서를 알기 위해

REGISTER (Cont.)

- authenticated user's authorization
  - address-of-record의 수정권한 결정(SHOULD)
  - **third-party registration 제공**
  - 권한이 없는 경우에는 403 Forbidden 응답
- binding update
  - tentative binding : multi Contact인 경우에 임시적으로 binding을 한후 모두 성공적으로 처리된 경우에만 proxy 또는 redirect server에게 visible 하기 위해
  - **back-end database commit failed인 경우에 이전에 tentative binding을 모두 삭제하고 500 Server Error 응답** (MUST)
- REGISTER의 200 응답에 Date 헤더 제공 (SHOULD)

Header Field & Status Code

- Mandatory Fields
  - From, To, Call-ID, CSeq, Via, **Max-Forwards**
- **sips** scheme 추가
  - TLS를 통해서만 전송
- Added Header Field
  - **Min-Expires**
    - Expires가 너무 짧은 경우에 423 응답에 포함
- Added Status Code
  - **423** Interval Too Brief
  - **491** Request Pending
  - **493** Undecipherable

Message Generating & Parsing

- Header
  - **Max-Forwards mandatory**
  - proxy에서 Max-Forwards 삽입 허용
  - **하나의 헤더 파라미터 네임에 헤더 필드 value 하나만 가능**
  - Content-Length 존재는 TCP에서는 MUST, UDP는 SHOULD
- Message Body
  - CANCEL에도 포함 가능
- URI
  - "sips" URI scheme 추가
  - To/From 필드에 port 허용하지 않음 (MUST)
- tag 포함
  - **non-100 응답에는 MUST**
  - 100 응답에는 MAY

Message Generating & Parsing (Cont.)

- Proxy
  - stateless proxy도 top most Via match되어야 함
  - proxy는 **malformed header**를 삭제하지 않아야 함(MUST)
- UA
  - UA에서 **branch-id** 사용 (MUST)
  - S/MIME 지원
- Registrar
  - REGISTER의 2xx에는 현재 contact 포함
- terminator
  - CRLF만 가능 (MUST)
  - Message Body가 존재하지 않아도 empty line 필요(MUST)

Authorization

- hop-by-hop encryption & authentication: TLS (Transport Layer Security)
- proxy authentication: Proxy-Authentication
- end-to-end HTTP authentication: digest (challenge-response)

Authorization (Cont.)

- Authorization credential
  - To 헤더 전체가 아닌 URL을 기초로 cache
- ACK Authorization
  - ACK은 INVITE와 같은 credential을 사용해야함 (MUST)
- **Aggregate authorization** header field values if necessary
  - proxy는 모든 challenge를 collect해서 제공함
- Basic Authentication은 제공하지 않음

SIP Extensions and Feature Negotiation

```
C->S   INVITE sip:watson@bell-telephone.com SIP/2.0
       Require: com.example.billing
       Supported: 100rel
       Payment: Sheep_skins, conch_shells

S-C    SIP/2.0 420 Bad Extension
       Unsupported: com.example.billing

S-C    SIP/2.0 421 Extension Required
       Require: 183
```

Security & Privacy

- SIP signaling can be encrypted
  - S/MIME
- SIP can be transported over
  - IPSec
  - TLS
- SIP can carry encryption key for media in SDP
- “Anonymizer ” service can be used to conceal identity

SIP Services

- SIP service creation
- Gateway service to the PSTN
- User preferences
- Presence
- Instant communications
- Universal messaging – voice mail
- SIP-PSTN-mobile phone interworking
- Mobility and user location
- ENUM:Telephone number based directory
- Third party call control:PC -phone
- Voice portal

Extension

- INFO method
- PRACK
- TRANSFER method
- SESSION TIMER
- REFER method
- Caller preference & callee capabilities

Extensibility

- Primary Goal: Ensure Baseline Operation Always Works
  - Maintain “backward compatibility”
- Optional Features
  - Can be safely ignored by recipient
- Client Mandating Features
  - Require header in request
  - Unsupported header in response if incompatible
  - Client can resend request
- Server Mandating Features
  - Require header in response
  - Supported and unsupported header in request

Extensibility (Cont'd)

- Feature naming
  - IANA registered
- New Methods
  - UAS rejects requests with unknown methods
  - Allowed methods are listed
  - Does not affect proxy functionality
- New Bodies
  - Use MIME header techniques to indicate those which are acceptable

Extension

- INFO method
  - Used to transport mid-call signaling information
  - Only one pending **INFO** at a time.
  - Typical use – PSTN signaling message carried as MIME attachment
  - Defined by Donovan in RFC 2976 “The SIP INFO Method ”
- REFER method
  - Requests another User Agent to generate a new request to a specified URL
  - Typical Use:Call Transfer features in a SIP phone.
  - Allowed outside an established session
  - Defined by I-D Sparks “The REFER Method ”

Extension (Cont.)

- PRACK
  - Provisional response
  - QoS
  - Provisional Response ACK nowlegement.
  - Used to acknowledge receipt of a provisional response (eg 183) , otherwise response is retransmitted
  - RFC 3262
- Defined by I-D Rosenberg et al “Reliablilty of Provisional Responses ”

KRnet2002

Extesion (Cont.)

- COMET
  - Call pre **CO** nditions **MET**
  - Used to indicate the QoS or other preconditions on SIP call have been met and User Agent may now alert user
  - Defined by Marshall,et al,manyfolks I-D Distributed Call Control (DCS)

Extesion (Cont.)

- SUBSCRIBE, UNSUBSCRIBE, NOTIFY
  - Presence Extensions
  - **SUBSCRIBE** requests notification of when a particular event occurs.
  - A **NOTIFY** message is sent to indicate the event status.
  - **UNSUBSCRIBE** cancels a pending SUBSCRIBE
  - Defined in RFC 3265 “SIP-Specific Event Notification ”

Extension (Cont.)

- MESSAGE
  - Instant Message Extensions
  - **MESSAGE** carries an instant message (page)in the message body,formatted as HTML,plain text or other
  - Defined in Rosenberg et al I-D “SIP Extensions for Instant Messaging ”

Future of SIP

- Model
  - Component => flexible
  - Light-weight protocol => device
  - End-device is powerful => IN

Future of SIP

- System
  - Integration with legacy systems outlines a deployment path
    - SIP-T
    - PacketCable DCS
    - SIP-H.323 Interworking
    - 3GPP – All-IP
    - SIP call model mapping
  - Supporting infrastructure will be built
    - Integration with QoS
    - Management – MIBs
    - Accounting and billing
    - Security

Resources for SIP

- Henning Schulzrinne 's SIP Page http://www.cs.columbia.edu/sip
- IP Telephony Web Page http://iptel.org
- SIP Forum http://www.sipforum.vom
- SIP Center http://www.sipcenter.com
- SIP Products at Pulver.com http://www.pulver.com/sip/products.html
- Books
  - "SIP:Understanding the Session Initiation Protocol," Alan B.Johnston, Artech House,2001.
  - "Internet Communications Using SIP " by Henry Sinnreich and Alan B.. Johnston,Wiley,2001.
  - "SIP Demystified " by Gonzalo Camarillo,,McGraw-Hill,2001.