SIP, SDP and other NGN Protocols

-

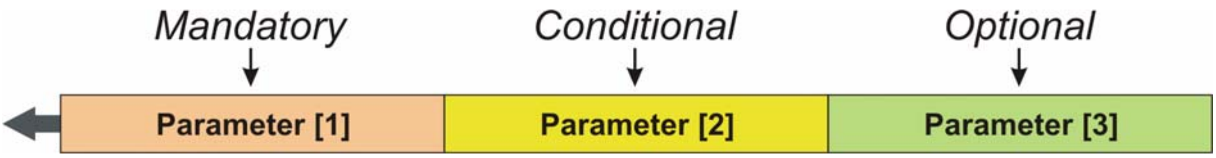
Signaling & Protocol Analysis

INACON GmbH
Kriegsstrasse 154
76133 Karlsruhe
Germany
www.inacon.com
e-mail: inacon@inacon.de

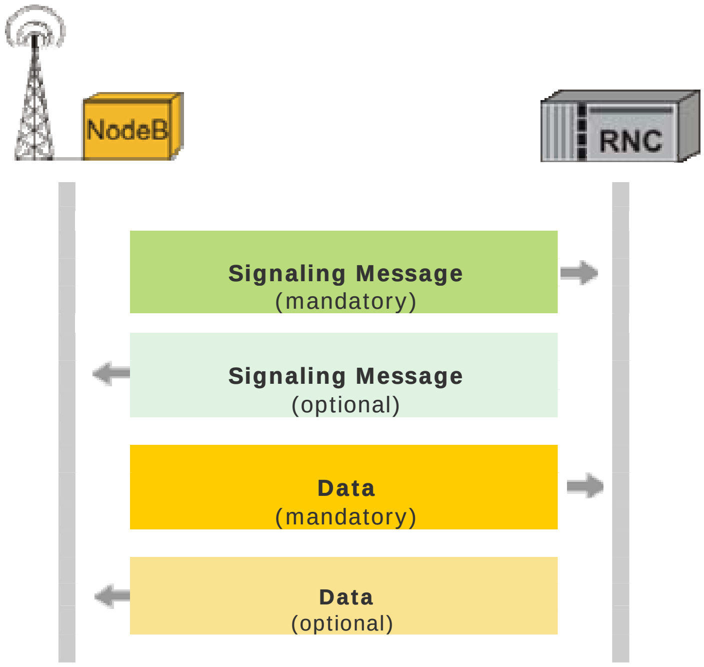
Legend:

All INACON publications use the same color codes to distinguish mandatory from optional or conditional parts in frame formats or optional from mandatory data blocks or signaling messages in scenarios. The different color codes are explained underneath:

- **Color Codes in Frame Formats:**

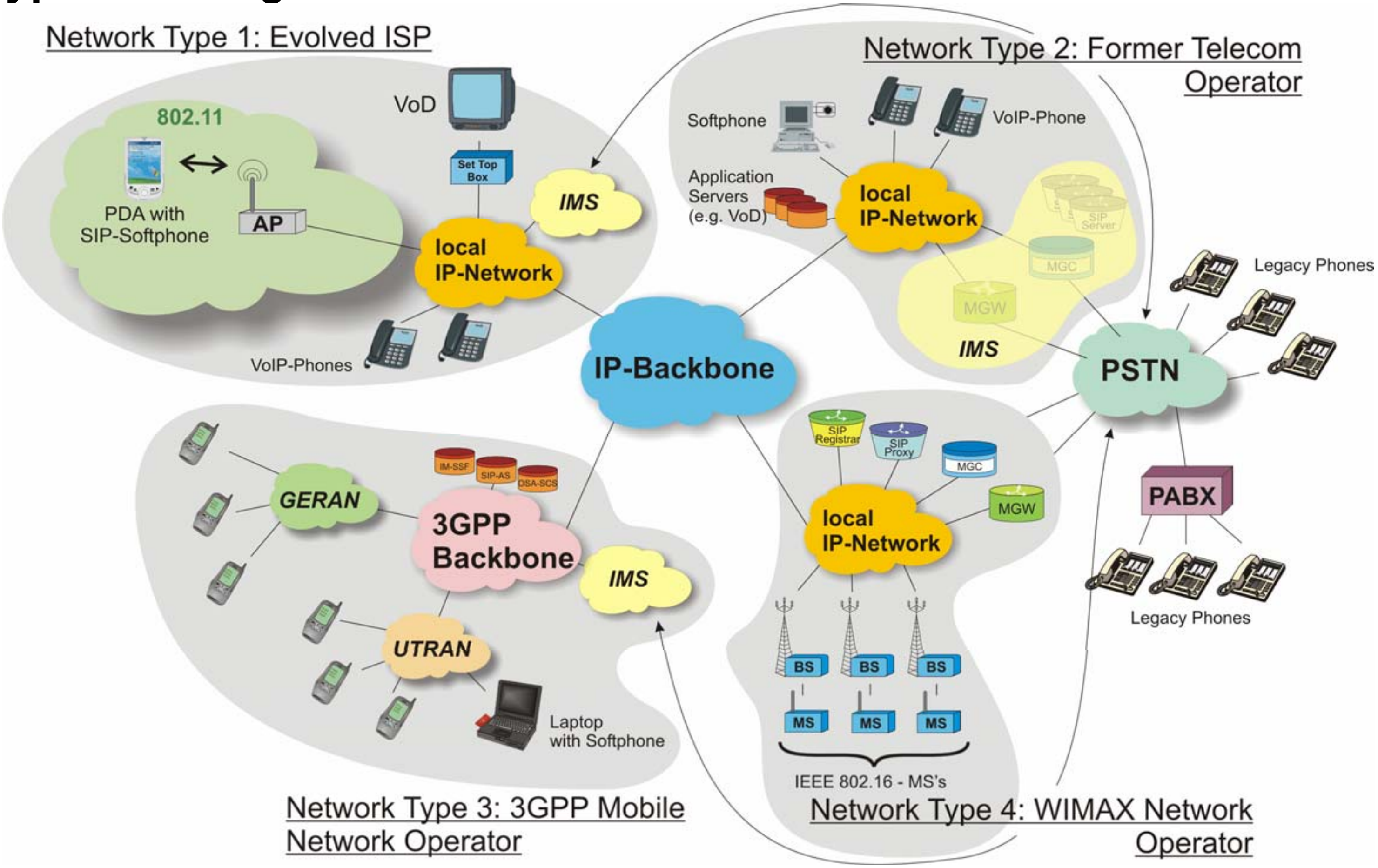


- **Color Codes in Scenarios:**



Next Generation Networks and their Components

- **Typical Configuration and Interconnection of Next Generation Networks**



Next Generation Networks and their Components

Typical Configuration and Interconnection of Next Generation Networks

The figure illustrates the most likely configuration of NGN's and it provides information about the services offered (Triple-Play). Most interestingly, the figure includes two wireless access networks of which only one is based on 3GPP while the other one is based on WIMAX.

Network Type 1: Evolved ISP

This type of network now provides telephone services, VoD-services (Video on Demand) and obviously still standard ISP-services (not shown). Through the operation of public hotspots, the ISP also gets the flavor of wireless operation. All multimedia services and the VoIP-services are controlled through the operator owned IMS (IP Multimedia Subsystem).

Network Type 2: Former Telecom-Operator

The former Telecom-Operator still has strong ties towards the PSTN. As the figure illustrates, part of the IMS is a soft switch (⇔ combination of Media Gateway and Media Gateway Controller) which allows the VoIP-subscribers the communication with regular PSTN-subscribers. Note that this Telecom-Operator also operates a number of application servers for all kinds of services like VoD or Presence Services. Nothing hinders the Telecom-Operator to allow other operators like the Network Type 1 operator to also use these application servers.

Network Type 3: 3GPP Mobile Network Operator

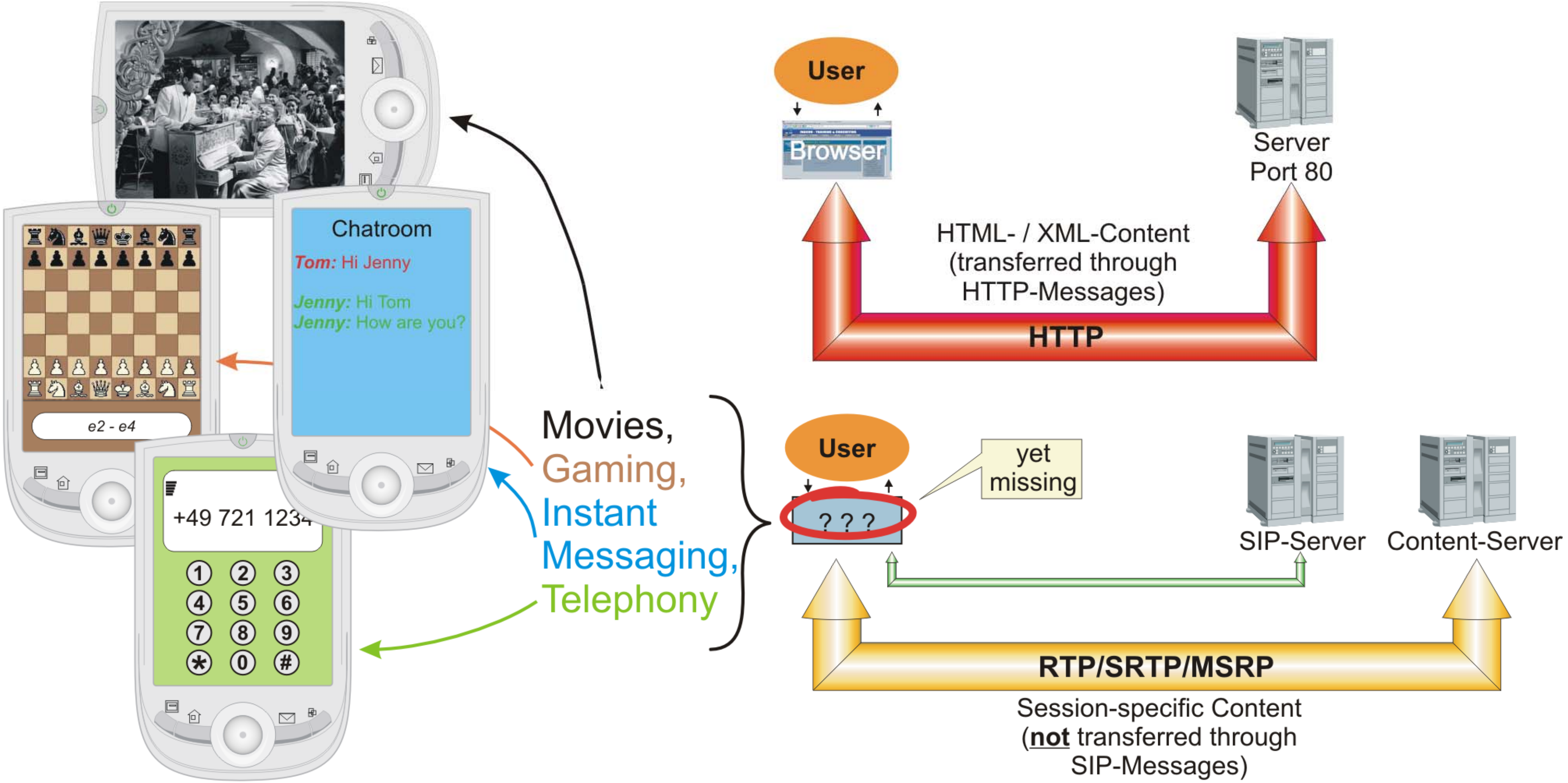
The 3GPP mobile network operator is no different from the previously mentioned operators with one exception: The primary way of accessing an IMS is through GERAN or UTRAN. With bandwidths of up to 2 Mbit/s, the 3GPP-network operator can offer similar or the same services as wireline operators (who are bandwidth limited through the physical limitations of DSL). In the long run, only the operator owned IMS interconnects calling mobile subscribers towards the PSTN. That's why we did not include the circuit-switched core network domain of the mobile network operator.

Network Type 4: WIMAX Network Operator

The upcoming WIMAX-network operators may emerge to a combination of a wireless network operator and an evolved ISP. WIMAX is a very strong DSL-competitor and WIMAX has the potential to become a cellular standard. Note that in case of network type 4 we did not put in an IMS. Its functions are accomplished through a series of dedicated SIP-servers and soft switches.

Note: The IP-CAN towards the customer for wireline operators is usually realized through DSL. Still, cable-TV operators may rather use their evolved cable-TV lines. And WIMAX or UMTS are yet other options of IP-CAN's.

Comparison between SIP and HTTP



Comparison between SIP and HTTP

- **To get a better feeling about SIP a comparison with HTTP is helpful**
  - ⇒ HTTP is the protocol behind and underneath web browsing and it is extremely prominent on the World Wide Web. HTTP enables *all* applications on the internet that are accessed through a web browser. Typical examples for web browsers are the Internet Explorer or Firefox.
  - ⇒ The message format, the request/response philosophy and many response codes (e.g. 200-OK) of SIP have been inherited from HTTP. One can legitimately say that SIP is at least formally based on HTTP.
  - ⇒ Note that HTTP is the transfer and control protocol for the "content" but the content itself is not HTTP. It consists of HTML-/XML-encoded information which in turn may incorporate other information like JAVA or JAVA-SCRIPT.

Note:

- In difference to SIP, HTTP is also used to embed the previously mentioned "content" into HTTP-packets for its transfer between peers.
- In HTTP, a "session setup" occurs inherently and static with the first HTTP-message exchange that identifies the capabilities of both peers and which clarifies whether these peers can exchange "content" in the first place. In other words, the actual session setup phase is very simple.

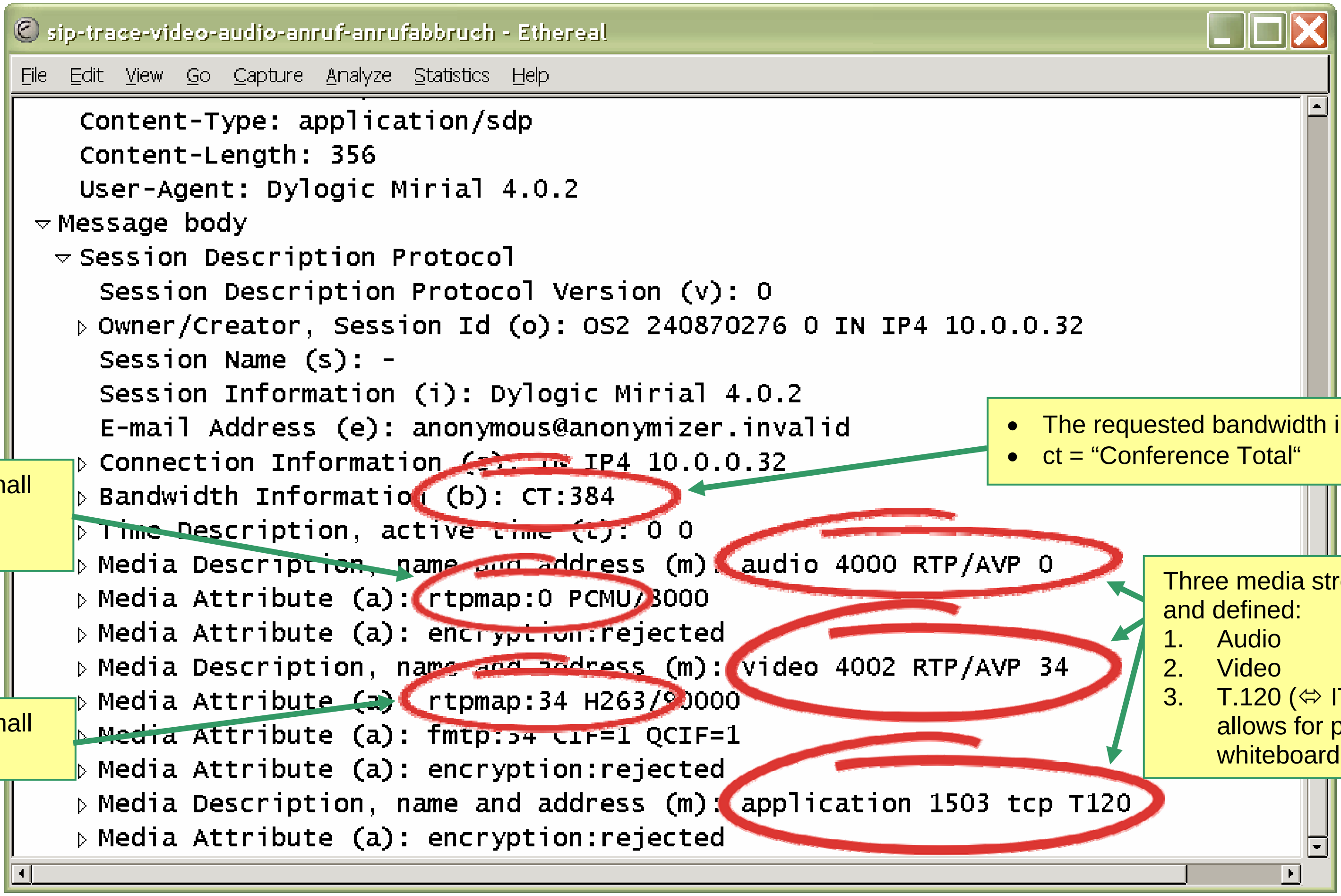
- ⇒ Usually, a client or user accesses an HTTP-server from his/her web browser to download a website from that server to the browser and afterwards the client will, possibly interactive, process the downloaded content through his/her web browser.
- ⇒ SIP on the other hand uses a "user client device" to manage a session between that "user client device" and other "user client devices" or towards an application server.

Problem for SIP:

- The vast number of possible *session types* (e.g. telephone calls, games, location services using RTP, MSRP or SRTP as bearers) to be established through SIP makes it difficult to provide or define a standardized generic "user client device" like the web browser for HTTP to support all possible session types.
- Still, we are almost convinced that such a definition will occur to enable the development of an unlimited number of applications on top of SIP. After all, it was only the invention of the web browser about 10 years ago with its GUI which made the internet gain speed for applications beyond mail.

Conclusion: Those who want to use SIP as basis for application development need to clarify in a first step which additional features SIP provides compared to HTTP (e.g. multi-party vs. two-party, user-to-user communication is possible…). In a second step a generic "SIP-browser" may need to be invented that will allow for the necessary economies of scale and that will enable the application development itself rather seamlessly.

The Related Session Description Protocol (SDP) Contents


sip-trace-video-audio-anruf-anrufabbruch - Ethereal
File Edit View Go Capture Analyze Statistics Help

```
    Content-Type: application/sdp
    Content-Length: 356
    User-Agent: Dylogic Mirial 4.0.2
▽ Message body
  ▽ Session Description Protocol
      Session Description Protocol Version (v): 0
    ▷ Owner/Creator, Session Id (o): OS2 240870276 0 IN IP4 10.0.0.32
      Session Name (s): -
      Session Information (i): Dylogic Mirial 4.0.2
      E-mail Address (e): anonymous@anonymizer.invalid
    ▷ Connection Information (c): IN IP4 10.0.0.32
    ▷ Bandwidth Information (b): CT:384
    ▷ Time Description, active time (t): 0 0
    ▷ Media Description, name and address (m): audio 4000 RTP/AVP 0
    ▷ Media Attribute (a): rtpmap:0 PCMU/8000
    ▷ Media Attribute (a): encryption:rejected
    ▷ Media Description, name and address (m): video 4002 RTP/AVP 34
    ▷ Media Attribute (a): rtpmap:34 H263/90000
    ▷ Media Attribute (a): fmtp:34 CIF=1 QCIF=1
    ▷ Media Attribute (a): encryption:rejected
    ▷ Media Description, name and address (m): application 1503 tcp T120
    ▷ Media Attribute (a): encryption:rejected
```

- The requested bandwidth in kbit/s
- ct = “Conference Total“

The audio codec shall be PCM / µ-law encoded speech.

The video codec shall be H.263.

Three media streams are requested and defined:

1. Audio
2. Video
3. T.120 (⇔ ITU-T standard which allows for program sharing and whiteboard functionality)

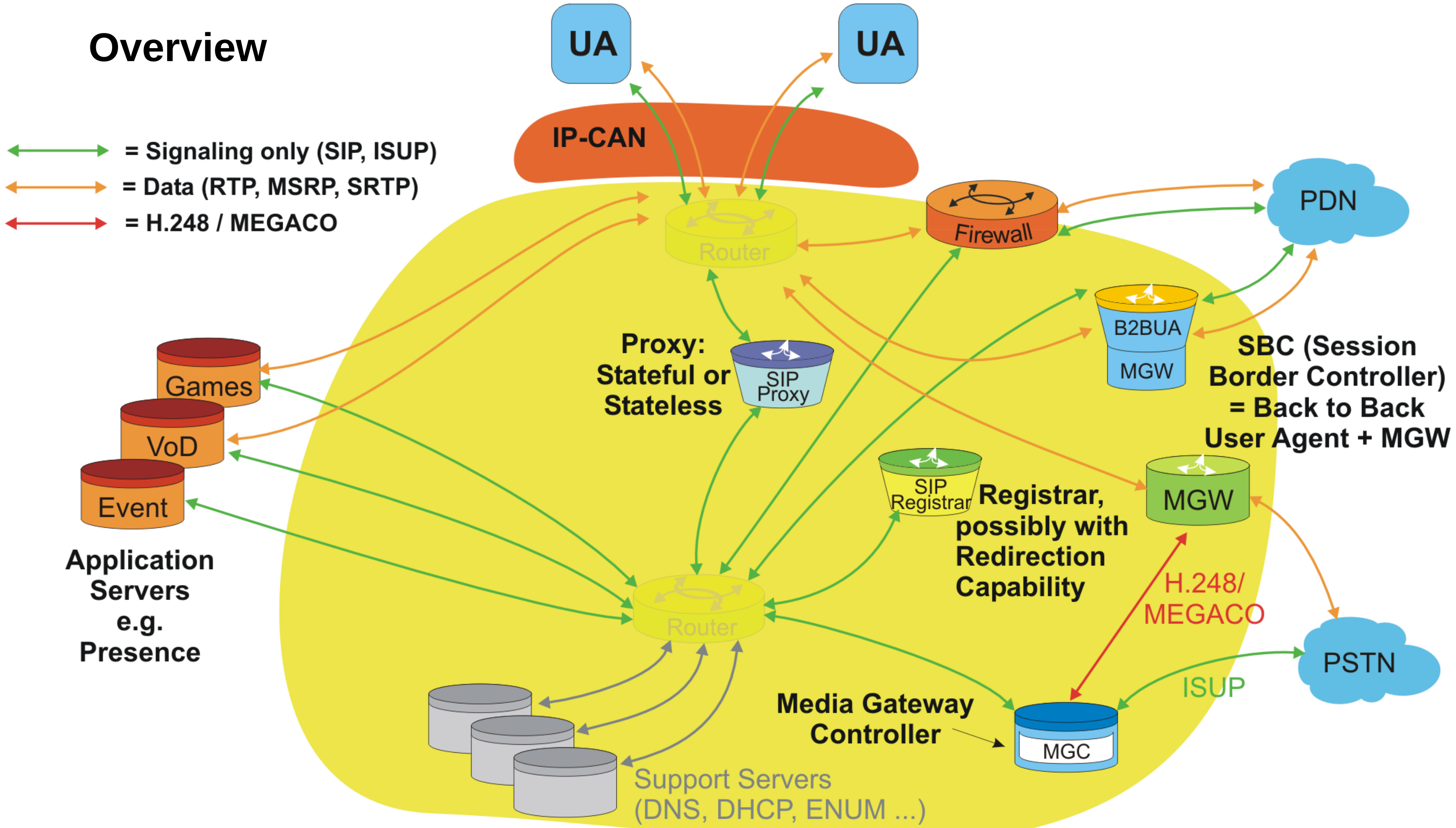
The Related Session Description Protocol (SDP) Contents

Additional Information

- ⇒ Detailed information about the meaning of the different SDP-parameters like “ct” or “RTP/AVP” is provided later in this book.
- ⇒ Unfortunately, the IETF never updates RFC’s but rather issues new RFC’s if new parameters as in case of SDP are added. Still, on the website http://www.iana.org/assignments/sdp-parameters the IETF at least provides a summary of all SDP-parameters.
- ⇒ Similarly, for SIP the IETF maintains the website http://www.iana.org/assignments/sip-parameters where all SIP-parameters, header fields etc. can be found.
- ⇒ Note the numbers ‘0’ and ‘34’ at the end of two media lines audio and video: These numbers represent the payload type values which will be used in the header of the upcoming RTP-frames to identify the respective media and codec types.
- ⇒ Implicitly, the indicated receiver port numbers 4000 (⇔ audio) and 4002 (⇔ video) also determine to which port number any RTCP-messages shall be sent (Real-time Transport Control Protocol). These shall be sent to the next higher port number, respectively [RFC 3264 (5.1)]. That is port number 4001 for the audio stream and 4003 for the video stream.
- ⇒ The numbers “8000” and “90000” behind the codec identifiers PCMU and H263 relate to the number of samples per second.
- ⇒ The terms “CIF = 1” and “QCIF = 1” in the fmtp-line for the H.263 video-codec (⇔ 34) indicate support of that peer for QCIF (Quarter Common Intermediate Format) and for CIF (Common Intermediate Format). QCIF and CIF relate to video resolutions in terms of number of lines and pixels per line.

SIP-Network Architecture

- **Overview**


= Signaling only (SIP, ISUP)
= Data (RTP, MSRP, SRTP)
= H.248 / MEGACO
UA
UA
IP-CAN
Router
Firewall
PDN
B2BUA
MGW
SBC (Session Border Controller) = Back to Back User Agent + MGW
Proxy: Stateful or Stateless
SIP Proxy
Games
VoD
Event
Application Servers e.g. Presence
SIP Registrar
Registrar, possibly with Redirection Capability
MGW
H.248/ MEGACO
Router
PSTN
ISUP
Media Gateway Controller
MGC
Support Servers (DNS, DHCP, ENUM ...)

SIP-Network Architecture

Overview

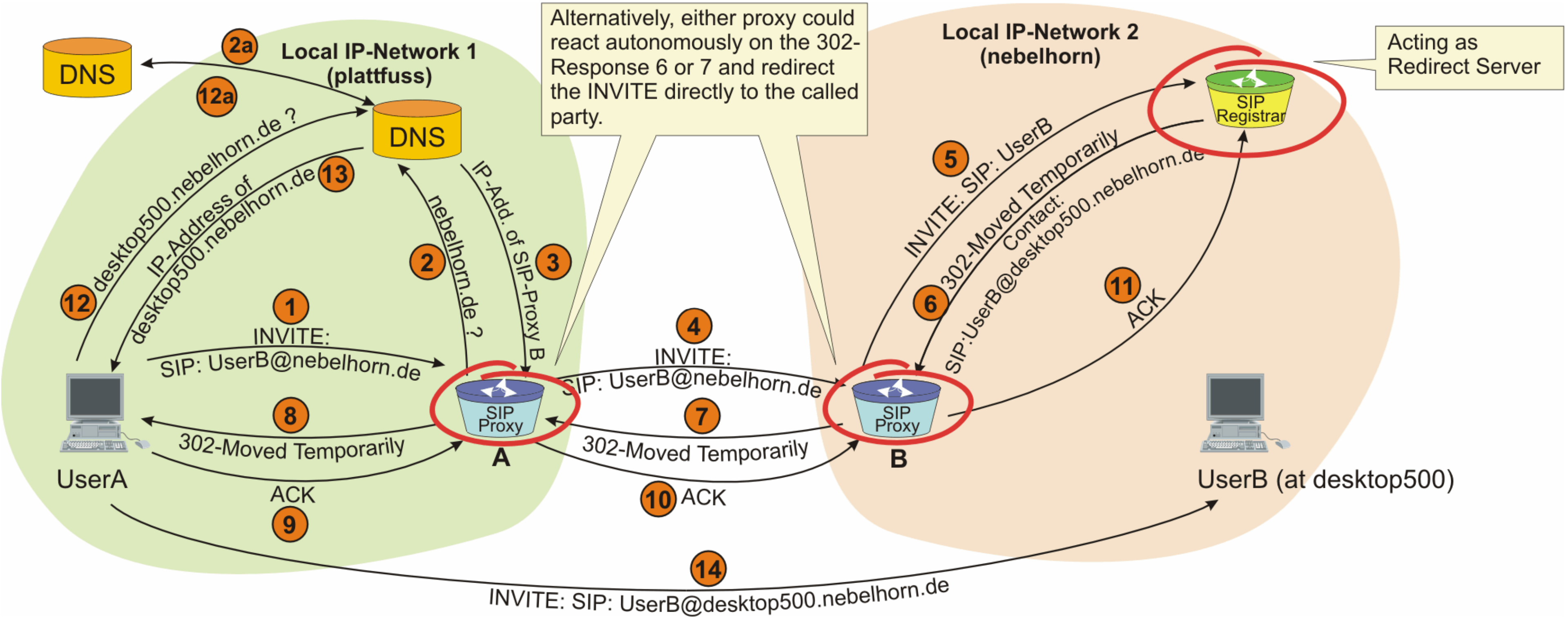
The figure illustrates a typical network which uses SIP for session management functions. Please note the following information:

- ⇒ We specifically distinguished between the orange colored lines that the data take on one hand and the green colored lines for SIP-signaling messages. To be more precise: With the exception of an SBC (Session Border Controller), no SIP-proxy server or MGC deals with the data themselves.
- ⇒ The implicated *physical* network architecture with the two (SIP-independent) standard routers is an arbitrary possibility to interconnect the various network elements. We only added these routers to avoid irritations. Still, our focus is the logical network architecture. That's why we hid these routers behind shades.
- ⇒ Sessions towards the PSTN require by default the interaction of soft switches (⇔ MGC + MGW).
- ⇒ Sessions towards external PDN's or to the internet will either traverse a firewall or they will traverse an SBC.

???? Question Section 1 ????

- ⇒ Why are there SIP-proxies to relay SIP-messages? Why is this task not taken care of by simple IP-routers?

Operation of Redirect Servers



Alternatively, either proxy could react autonomously on the 302-Response 6 or 7 and redirect the INVITE directly to the called party.

Acting as Redirect Server

Operation of Redirect Servers

Introduction

A redirect server is a SIP-proxy server or a SIP-registrar that responds an incoming Request: INVITE with a Response: 3XX (e.g. 302-“Moved Temporarily”). This response includes in the “Contact:”-header field the one or more current user device’s addresses that shall be contacted instead or directly by the originating party.

Procedure Description

As illustrated in the figure, UserA sends an INVITE / sip: UserB@nebelhorn.de to its SIP-proxy server A (⇔ message 1). The proxy server invokes the help of one or more DNS-servers to resolve the IP-address of nebelhorn.de (⇔ message 2, 2a and 3). Consequently, proxy A relays the INVITE-message to SIP-proxy server B (⇔ message 4).

⇒ Proxy B sends the INVITE-message to the responsible SIP-registrar (⇔ message 5).

⇒ The SIP-registrar will issue a final Response: 302-“Moved Temporarily” which traverses all the way back to UserA (⇔ message 6, 7, 8) and which ultimately is the message that will trigger the redirection of the request.

Note the comment on the graphics slide: Either SIP-proxy server could react on the Response: 302-Moved Temporarily autonomously and redirect the Request: INVITE to its new destination directly.

⇒ As mentioned before, this response message type always carries in its “Contact:”-header field the current IP-address or FQDN where the requested part can be found. In our case, this is the fully qualified domain name desktop500.nebelhorn.de.

⇒ Before another INVITE to UserB at desktop500.nebelhorn.de can be sent, UserA needs to finish the previous INVITE-transaction by issuing a Request: ACK and sending it to the registrar in local IP-network 2 (⇔ message 9, 10, 11).

⇒ To be able to send an INVITE-message to sip: UserB@desktop500.nebelhorn.de, UserA invokes the support of one ore more DNS-servers to resolve the FQDN into an IP-address (⇔ messages 12, 12a and 13).

⇒ Finally, UserA can send a Request: INVITE / sip: UserB@desktop500.nebelhorn.de directly to UserB. *No SIP-proxy server is used* (⇔ message 14).

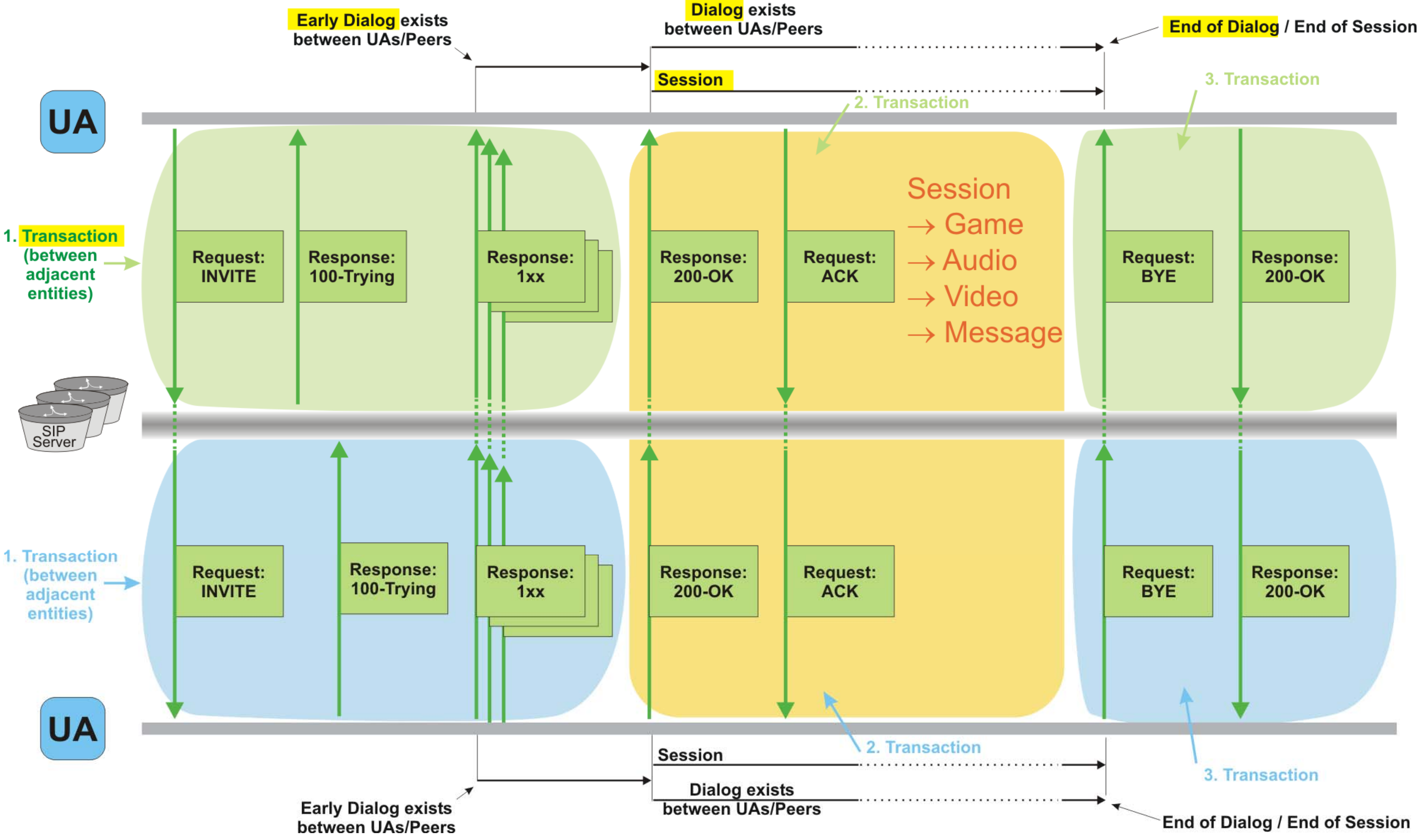
Redirection is well suited to reduce the load of SIP-proxies but it is not well suited for carrier based services which require at least a SIP-proxy server for charging purposes.

[RFC 3261 (8.3)]

???? Question Section 2 ????

⇒ If either SIP-proxy would autonomously redirect the INVITE to its new destination, would this be a stateful or a stateless proxy server or both?

Important SIP-Terminology / Step 1: Two UA's …



Important SIP-Terminology / Step 1: Two UA's …

We start with the limitation of only two users (no SIP-forking) to simplify things initially. In a second step we shall handle the case of multiple users).

Transaction)

Each SIP-transaction consists of a single request message which is sent by a UAC (User Agent Client) and the related final response message which is sent by the adjacent UAS (User Agent Server). If the request message is an INVITE, then there are zero or more provisional responses between the INVITE and the final response message. Note that the term "adjacent" in the previous sentence means that transactions ultimately exist between adjacent SIP-entities (e.g. UA and proxy) and not necessarily between peers (the two UA's in the graphics) [RFC 3261 (p.24)].

The exception to this rule is indicated in the figure: The successful dialog establishment through the initial INVITE-transaction shall be acknowledged by a Request: ACK-message which is considered to be a second transaction but which is not responded at all (although it is a SIP-Request). The Request: ACK really is a new transaction, considering the fact that a new branch-value is used. However, as we will see later, the transaction number (⇔ CSeq) does not get incremented from the Request: INVITE that the Request: ACK relates to [RFC 3261 (p.24)].

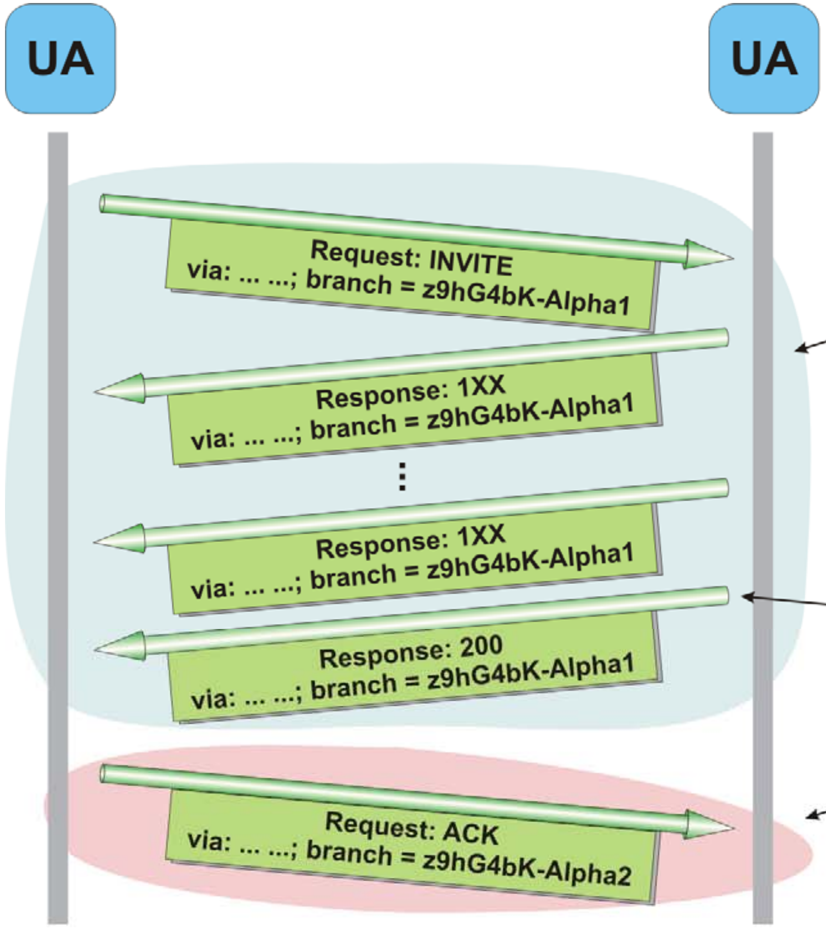
Dialog / Call / Early Dialog (Definition)

- Dialog establishment is initiated when a UAC sends a Request: INVITE-message towards another peer, with this message possibly traversing one or more SIP-proxies.
- Dialog establishment can only be triggered by Request: INVITE and (new with RFC 3515) also by Request: REFER.
- Dialogs only exist between two UA's / peers. There can be no dialogs between a UA and a SIP-proxy.
- A dialog has been established as soon as a UAS responds to a Request: INVITE with a non-failure final response message (⇔ 200-OK). This rule means that the reception of a 2XX-response by a UAC establishes a dialog between these two users. This also is the start of the session.
- An early dialog is there, if a UAS responds to a Request: INVITE with a provisional Response: "101 – 199" message (⇔ which excludes "100" (Trying)) [RFC 3261 (12.1)]. The benefit of the definition of an "early dialog" is that the UAC may send further SIP-Requests (e.g. UPDATE) to the UAS already while the dialog is in its early state [RFC 3261 (13.2.2.1)].
- In SIP, a call consists of one or more dialogs [RFC 3261 (p.78)]. More than one dialog per call is only possible for multiparty calls.
- Dialogs are terminated by either party by sending a Request: BYE-message. Early dialogs can be terminated by the UAC by sending a Request: CANCEL-message

Each dialog is identified by the Call-ID-value which is initially allocated by the peer that sent the Request: INVITE-message and by the "To:"- and "From:"-tag values. We will get back to these identifiers in a few slides.

Transaction-specific Messaging

- **Option 1: Request = INVITE / Transaction = successful**


UA
UA
Request: INVITE
via: ... ...; branch = z9hG4bK-Alpha1
Response: 1XX
via: ... ...; branch = z9hG4bK-Alpha1
⋮
Response: 1XX
via: ... ...; branch = z9hG4bK-Alpha1
Response: 200
via: ... ...; branch = z9hG4bK-Alpha1
Request: ACK
via: ... ...; branch = z9hG4bK-Alpha2

1. Transaction
⇔ **branch = z9hG4bK-Alpha1**
⇔ **CSeq: Integer1 INVITE**

Successful

2. Transaction
⇔ **another branch = z9hG4bK-Alpha2**
⇔ **CSeq: Integer1 ACK**
⇔ **may carry more than one "Via:"-header field**

Transaction-specific Messaging

Option 1: Request = INVITE / Transaction = successful

⇒ If a transaction is initiated by a Request: INVITE-message, then the final Response: 200 shall be acknowledged by the UAC through a Request: ACK-message.

⇒ Since this Request: ACK-message is considered as a new transaction, it shall be equipped with a new branch-parameter value (⇔ “z9hG4bK-Alpha2”). Note that each proxy between the two UA’s will add its own “Via:”-header field to the traversing Request: ACK-message which is different to an unsuccessful INVITE-transaction.

⇒ Despite this fact, the Request: ACK-message shall use the same “CSeq:”-value as the original Request: INVITE-message. However, the “CSeq:”-method shall be ACK [RFC 3261 (p.82)].

[RFC 3261 (17.1.1), (17.2.1)]

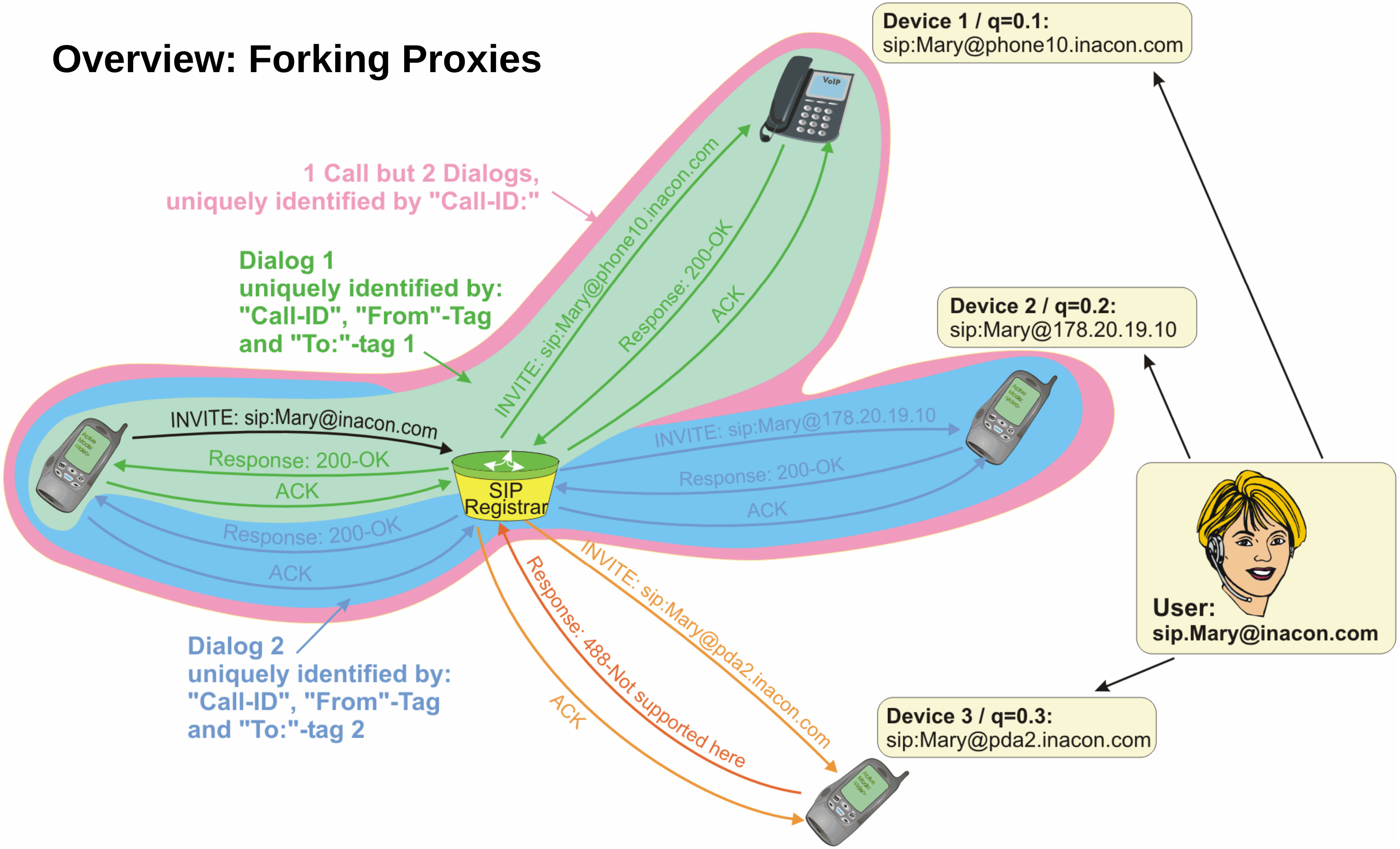
???? Question Section 6 ????

⇒ Why does SIP deploy a “3-Way Handshake” –procedure (1. INVITE / 2. 200-OK / 3. ACK) in the first place?

⇒ Why is ACK considered as a new transaction?

Amendments in case of more than two Peers

- **Overview: Forking Proxies**



1 Call but 2 Dialogs, uniquely identified by "Call-ID:"

Dialog 1 uniquely identified by: "Call-ID", "From"-Tag and "To:"-tag 1

Dialog 2 uniquely identified by: "Call-ID", "From"-Tag and "To:"-tag 2

Device 1 / q=0.1: sip:Mary@phone10.inacon.com

Device 2 / q=0.2: sip:Mary@178.20.19.10

Device 3 / q=0.3: sip:Mary@pda2.inacon.com

User: sip.Mary@inacon.com

VoIP
INVITE: sip:Mary@inacon.com
Response: 200-OK
ACK
Response: 200-OK
ACK
SIP Registrar
INVITE: sip:Mary@phone10.inacon.com
Response: 200-OK
ACK
INVITE: sip:Mary@178.20.19.10
Response: 200-OK
ACK
INVITE: sip:Mary@pda2.inacon.com
Response: 488-Not supported here
ACK

Amendments in case of more than two Peers

Introducing Different Contact Addresses per User

⇒ A user may have registered different contact addresses to his/her registrar. In our example, the user sip: Mary@inacon.com registered to her registrar from different SIP-devices of which each one conveyed a different SIP-URI in the "Contact:"-header field to the registrar (not shown). These "Contact:"-header SIP-URI's are the ones that we indicate as device SIP-addresses. Note that each of these "Contact:"-header SIP-URI's relates to a specific IP-address

⇒ As illustrated, there is also a 'q'-parameter (q = qualifier = 0.000 – 1.000) used during the registration to provide for a different prioritization of the different "Contact:"-addresses. The smaller 'q', the higher the priority. The forking registrar does not care in this case and relays the Request: INVITE to all destinations simultaneously. Alternatively, a predefined q=1.0 could be used as identifier for the voicemail URI of a user.

Behavior of Forking Proxies

⇒ Forking proxies either receive their location information from registrars or they are registrars themselves (the case which is illustrated in our example).

⇒ Such a forking proxy will relay a received Request: INVITE-message not only to a single destination but to several different ones.

The Terms Call, Dialog, Session and Transaction in case of Forking

The figure emphasizes what we already introduced earlier:

⇒ A *dialog* between two peers is established as soon as the Response: 200-OK is received from the called peer. Each dialog is uniquely identified by the "Call-ID:" and the "To:"- and "From:"-header field tags.

⇒ In case of multiparty (⇔ in our example let Mary herself answer at device 1 but somebody else answers at device 2) a *call* consists of all the different dialogs. Obviously, a *call* equals a *dialog* if only two parties are involved.

⇒ Although the detailed message parameters will be illustrated on the following slides we like to say that between the calling party and the registrar, all messages except the Request: ACK's for the Responses: 200-OK represent only a single transaction despite the fact that the two Response: 200-OK messages belong to two different dialogs!

⇒ Obviously, the registrar establishes a separate transaction to each of the called parties. And obviously, the relayed Request: ACK-messages towards device 1 and 2 are again separate transactions.

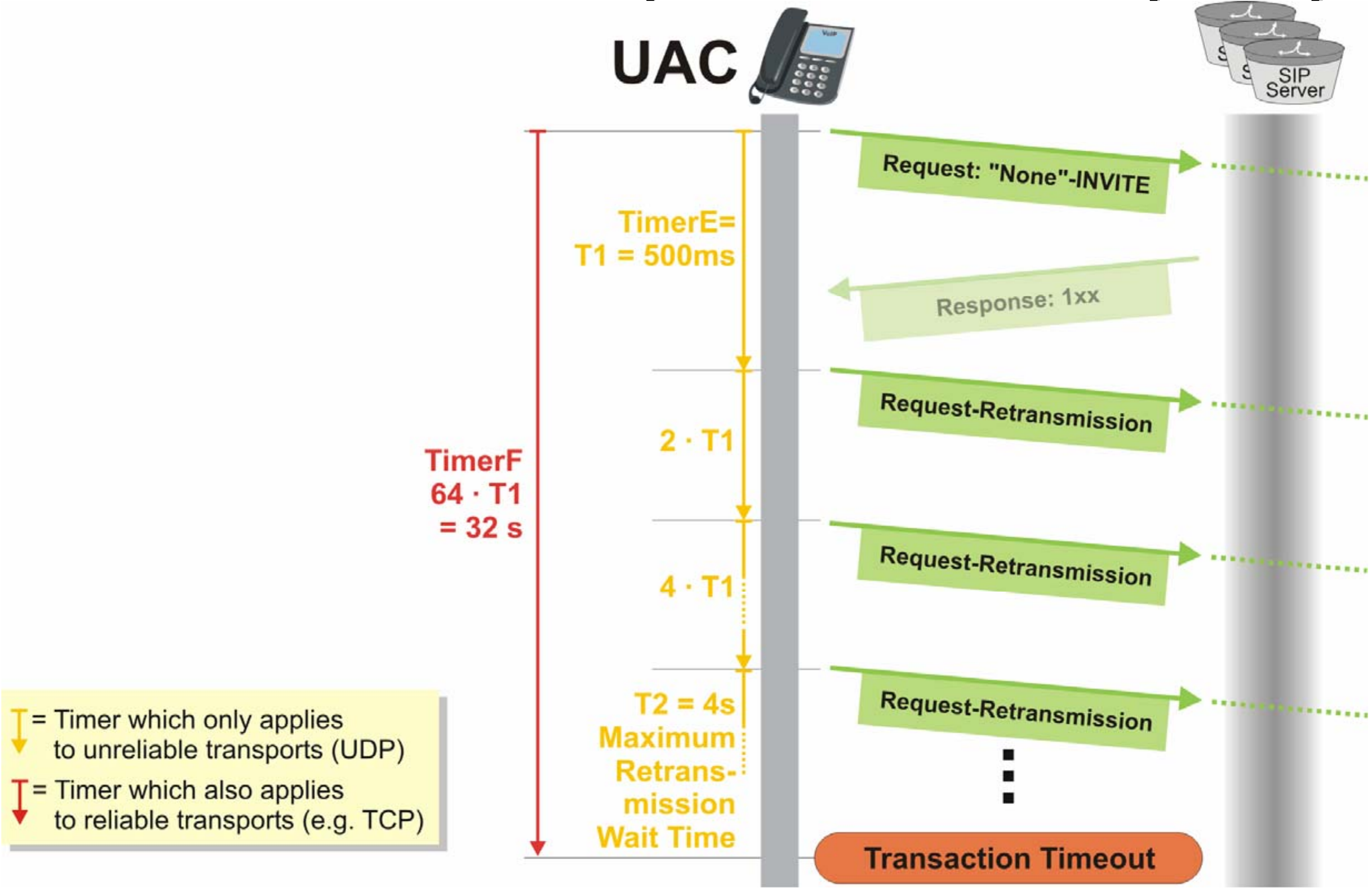
⇒ A very important new behavior of SIP-proxies in general is illustrated between the registrar and device 3. For one or another reason, device 3 rejects the incoming Request: INVITE with a final Response: 488-"Not Supported Here".

Note:

- In the example, the SIP-proxy / registrar does not relay this unsuccessful response to the calling party but rather handles the respective acknowledgement independently and by itself. Likewise, such a SIP-proxy / registrar could issue a Request: CANCEL-message.

[RFC 3261 (p.78)]

”None“-INVITE Transaction (UAC-Side - no Response)



↓ = Timer which only applies to unreliable transports (UDP)

↓ = Timer which also applies to reliable transports (e.g. TCP)

”None“-INVITE Transaction (UAC-Side - no Response)

Overview

When the UAC issues a “None”-INVITE-Request, it will start timer F and possibly timer E within its transaction management sublayer. This relates to all SIP-requests, except Request: ACK (⇔ CANCEL, MESSAGE, OPTIONS, INFO…).

Timer E

⇒ Timer E is there to control retransmissions of the Request: “None”-INVITE-message in case of unreliable transport protocols (⇔ UDP).
⇒ Timer F is the transaction surveillance timer.
⇒ As illustrated in the figure, the UAC will retransmit the Request: “None“-INVITE-message upon every expiry of timer E. The duration of timer E doubles with each expiry (⇔ exponential back off) until a maximum retransmit interval of 4 s is reached.
⇒ Note that the reception of provisional response messages for that Request has no impact on timer E.

Expiry of Timer F

⇒ Upon expiry of timer F, the UAC considers a transaction timeout and deletes the transaction.

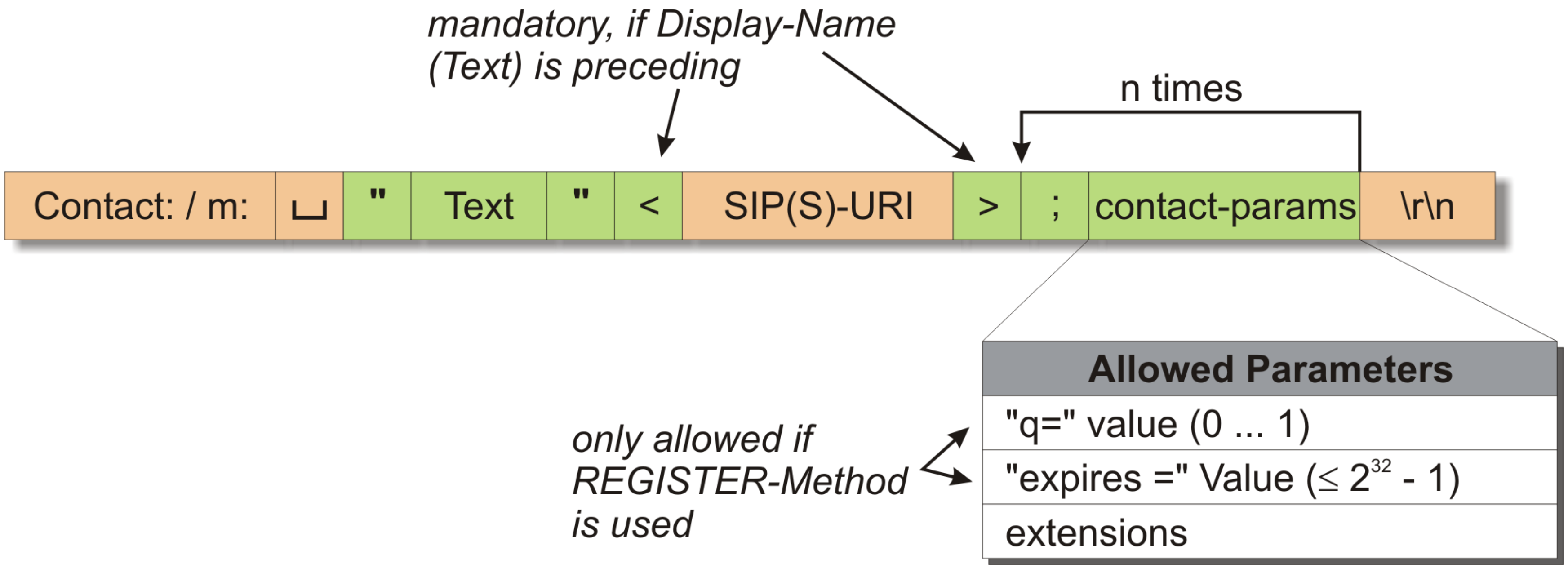
[RFC 3261 (17.1.2), Annex A]

Timer 1, Timer 2, Timer E and Timer F in case of 3GPP-Networks

Timer	Value to be used between P-CSCF and UE	Value to be used between SIP-nodes within the IMS
Timer 1 (T1)	2 s	0.5 s
Timer 2 (T2)	16 s	4 s
Timer E	Initially T1	Initially T1
Timer F	128 s	32 s

[3GTS 24.229 (7.7)]

The “Contact:” Header Field



mandatory, if Display-Name (Text) is preceding

n times

Contact: / m: | ␣ | " | Text | " | < | SIP(S)-URI | > | ; | contact-params | \r\n

only allowed if REGISTER-Method is used

Allowed Parameters
"q=" value (0 ... 1)
"expires =" Value ($\leq 2^{32} - 1$)
extensions

The “Contact:” Header Field

The “Contact:”-header field most importantly identifies the originator of that request to be used as identifier in subsequent requests. The “Contact:”-header field value may contain a display-name, a URI with URI parameters (see SIP(S)-URI) and additional header parameters. Only the provision of the SIP(S)-URI is mandatory. The enclosing “<” and “>” are mandatory, if a display-name is preceding.

As the figure illustrates, “Contact:”-specific parameters shall be delimited from each other through a “;”-character. Context-specific parameters may be defined but the two indicated parameters “q” and “expires” must only be present in REGISTER-request messages.

The parameter “q” allows to relatively prioritizing subsequent REGISTER-requests from different locations to be contacted in case of a terminating transaction.

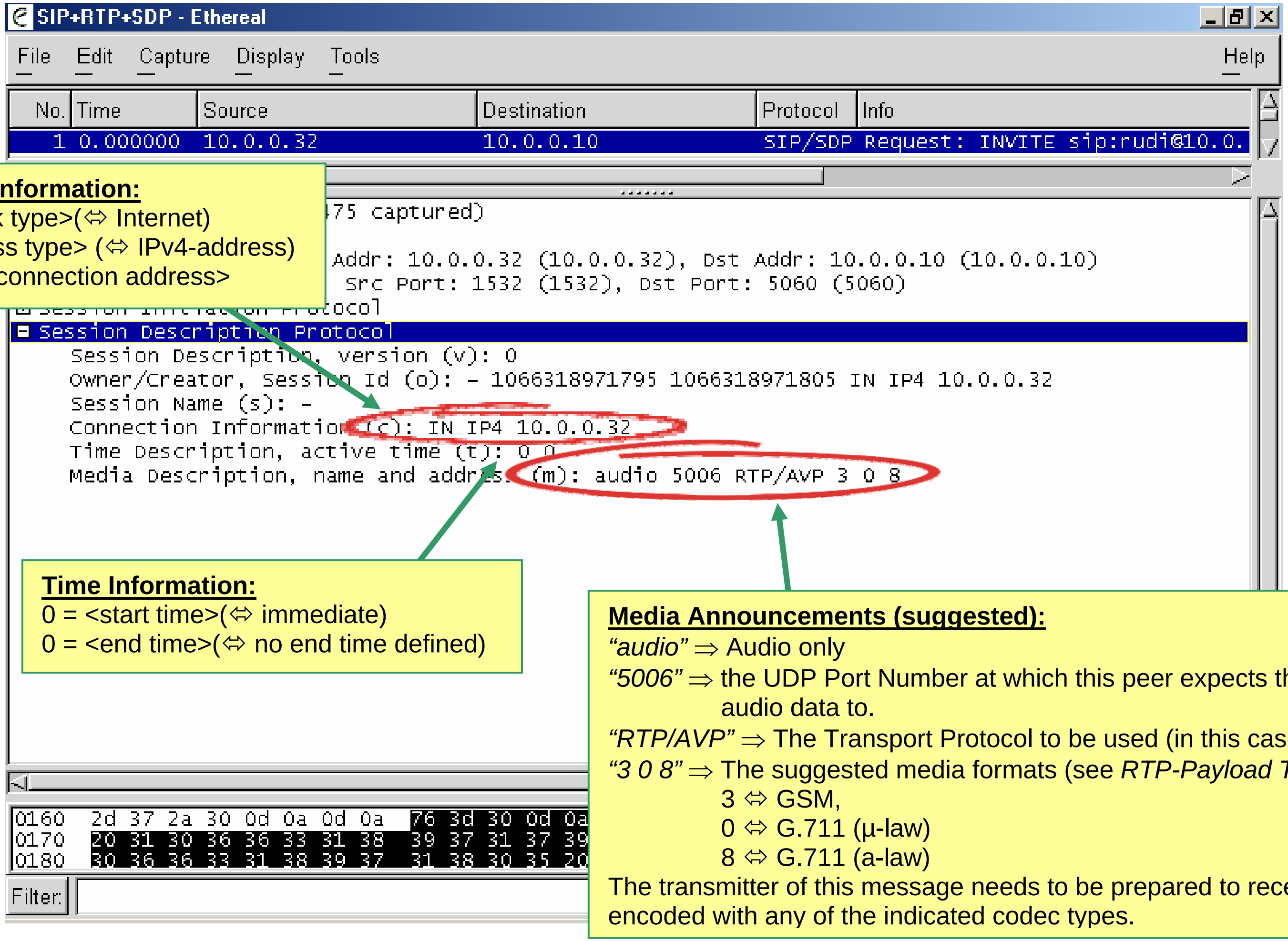
The parameter “expires” contains the registration timeout (the period after which re-registration is required) as desired by the user in REGISTER-request messages or the possibly downgraded timeout in the respective response message from the registrar. Note that the “contact-params” are provided per “Contact:”-parameter which means that different expiration times can be indicated per “Contact:”-address.

As an alternative, the dedicated “Expires:”-header field can be used which then applies globally [RFC 3261 (10.2.1.1)].

[RFC 3261 (8.1.1.8), (20.10)]

Note that in 3GPP-networks, the UA shall re-register 600 seconds before the “expires”-timer expires (if the timer value is larger than 1200 s) or after half of the time indicated by the “expires”-timer has elapsed (if the timer value is 1200 s or smaller) [3GTS 24.229 (5.1.1.4)]

Logfile Example: Session and Media Descriptors through SDP



```
No. Time      Source       Destination   Protocol Info
  1 0.000000  10.0.0.32    10.0.0.10     SIP/SDP Request: INVITE sip:rudi@10.0.

75 captured)

Addr: 10.0.0.32 (10.0.0.32), Dst Addr: 10.0.0.10 (10.0.0.10)
 Src Port: 1532 (1532), Dst Port: 5060 (5060)
Session Initiation Protocol
Session Description Protocol
    Session Description, version (v): 0
    Owner/Creator, Session Id (o): - 1066318971795 1066318971805 IN IP4 10.0.0.32
    Session Name (s): -
    Connection Information (c): IN IP4 10.0.0.32
    Time Description, active time (t): 0 0
    Media Description, name and address (m): audio 5006 RTP/AVP 3 0 8

0160  2d 37 2a 30 0d 0a 0d 0a  76 3d 30 0d 0a
0170  20 31 30 36 36 33 31 38  39 37 31 37 39
0180  30 36 36 33 31 38 39 37  31 38 30 35 20
```

<u>Connection Information:</u>
IN = <network type>(⇔ Internet)
IP4 = <address type> (⇔ IPv4-address)
10.0.0.32 = <connection address>

<u>Time Information:</u>
0 = <start time>(⇔ immediate)
0 = <end time>(⇔ no end time defined)

<u>Media Announcements (suggested):</u>
"audio" ⇒ Audio only
"5006" ⇒ the UDP Port Number at which this peer expects the other peer to send audio data to.
"RTP/AVP" ⇒ The Transport Protocol to be used (in this case RTP/UDP/IP)
"3 0 8" ⇒ The suggested media formats (see *RTP-Payload Type*) in a prioritized list
- 3 ⇔ GSM,
- 0 ⇔ G.711 (μ-law)
- 8 ⇔ G.711 (a-law)

The transmitter of this message needs to be prepared to receive audio information encoded with any of the indicated codec types.

Logfile Example: Session and Media Descriptors through SDP

Additional Information

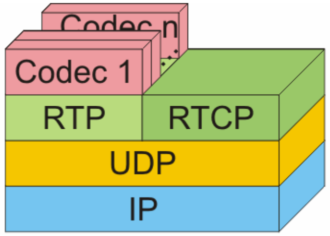
⇒ It is worth to mention that the statement in the yellow box “… encoded with any of the indicated codec types.” applies fully: The other peer may during the session at any time switch to another codec than what was used initially.

⇒ This obviously will cause an interruption until the new codec mode has resynchronized but it allows to switch to more error-resilient codecs, if there are means to determine that transmission quality dropped.

⇒ Note that the IETF recommends in the RFC 4504 (2.14 / Req. 72) that all SIP-devices support the iLBC-voice codec (Internet Low Bitrate Codec).The license-free iLBC-voice codec is a relatively new development (Dec 2004) with data rates of 13.33 kbit/s and 15.2 kbit/s. It has been published in RFC 3951 and its transfer through RTP has been described in RFC 3952.

(1) "m"-line / Details of the Transport Protocol Types

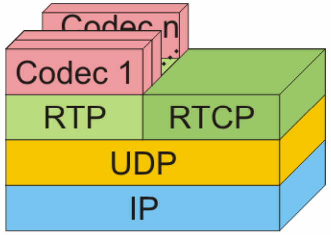
- **RTP/AVP**



Applications:
Standard Voice and Video Calls, PoC, VoD, AoD

m-line (Example):
`m = video 4500 RTP/AVP 31`

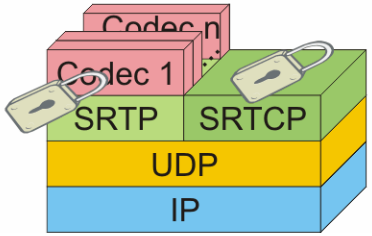
- **RTP/AVPF**



Applications:
Quality sensitive Voice and Video Calls or VoD, AoD sessions with the need for sophisticated RTCP-reporting.

m-line (Example):
`m = audio 6888 RTP/AVPF 8 0 3`

- **RTP/SAVP**



Applications:
Quality sensitive Voice and Video Calls or VoD, AoD sessions with the need for integrity protection and encryption.

m-line (Example):
`m = video 45002 RTP/SAVP 34 31`

(1) “m”-line / Details of the Transport Protocol Types

RTP/AVP

The related protocol stack is most frequently used as it is well suited for real-time user applications like voice calls or video conferencing. Note that the protocol stack always contains two parts: one for RTP and the voice or video codec on top and another one for RTCP.
RTP/AVP is also applicable in case of unidirectional VoD or AoD-sessions in which case a media stream only flows in one direction. The structure of the m-line is also indicated. The payload-type-list does contain one or more integer codec identifiers which are used within the payload type field of the related RTP-frames

[RFC 3551]

RTP/AVPF

The same applies which has been said about RTP/AVP. The only difference between RTP/AVP and RTP/AVPF is the use of advanced “Feedback” reports on RTCP-level. These advanced feedback reports provide information about lost RTP-frames and their numbers or encoder specific feedback information about lost graphics frames (which is important for high quality video conferencing and VoD).

[draft-ietf-avt-rtcp-feedback-11.txt]

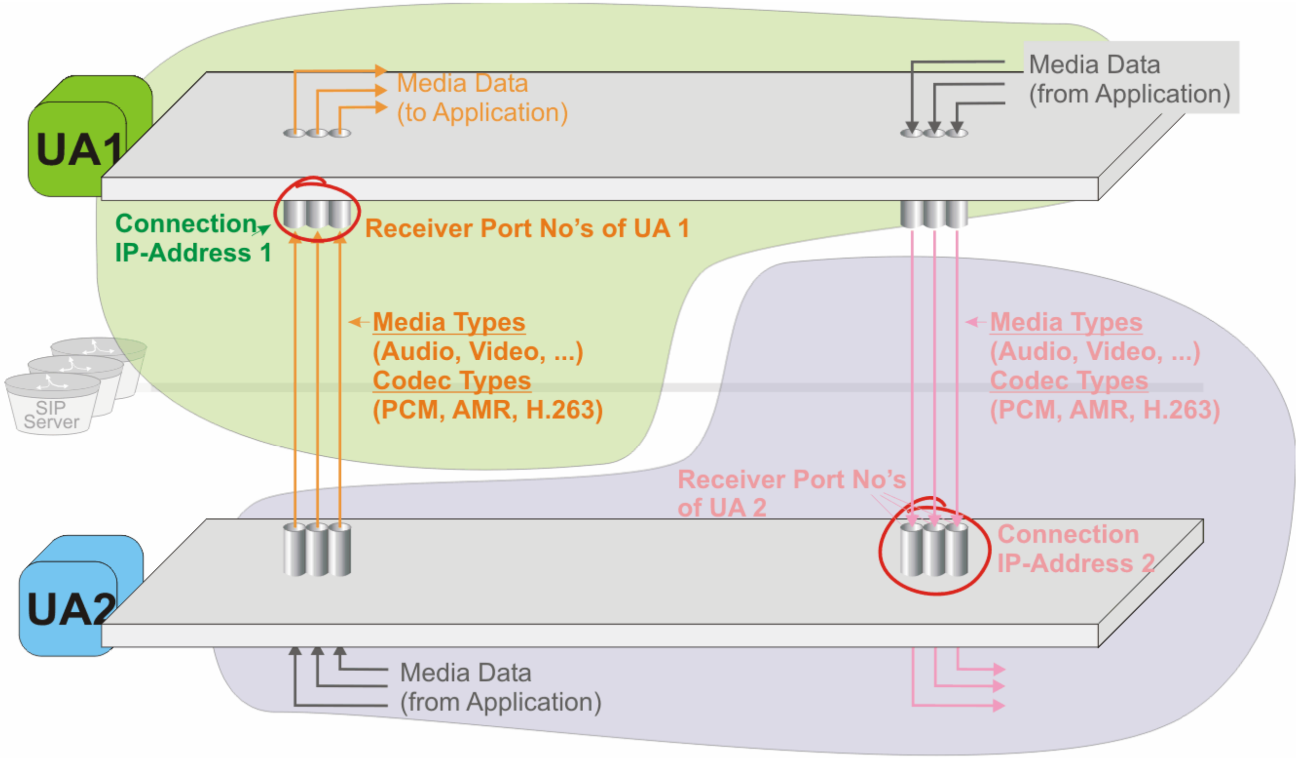
RTP/SAVP

The related protocol stack indicates the additional value of RTP/SAVP over RTP/AVP: It uses SRTP and SRTCP to provide privacy in the user plane. In that respect, privacy relates to RTP- / RTCP-data frame integrity protection, replay protection and encryption. The provision of the related keying material is out of the scope of RTP/SAVP and SRTP.

[RFC 3711]

Session Identification Parameters at both Peers

Announced by UA 1:
Session ID 1
(⇔ different value than Session ID 2)
Session Description Version 1
(⇔ different value than Session Description Version 2)
+
Media Descriptions



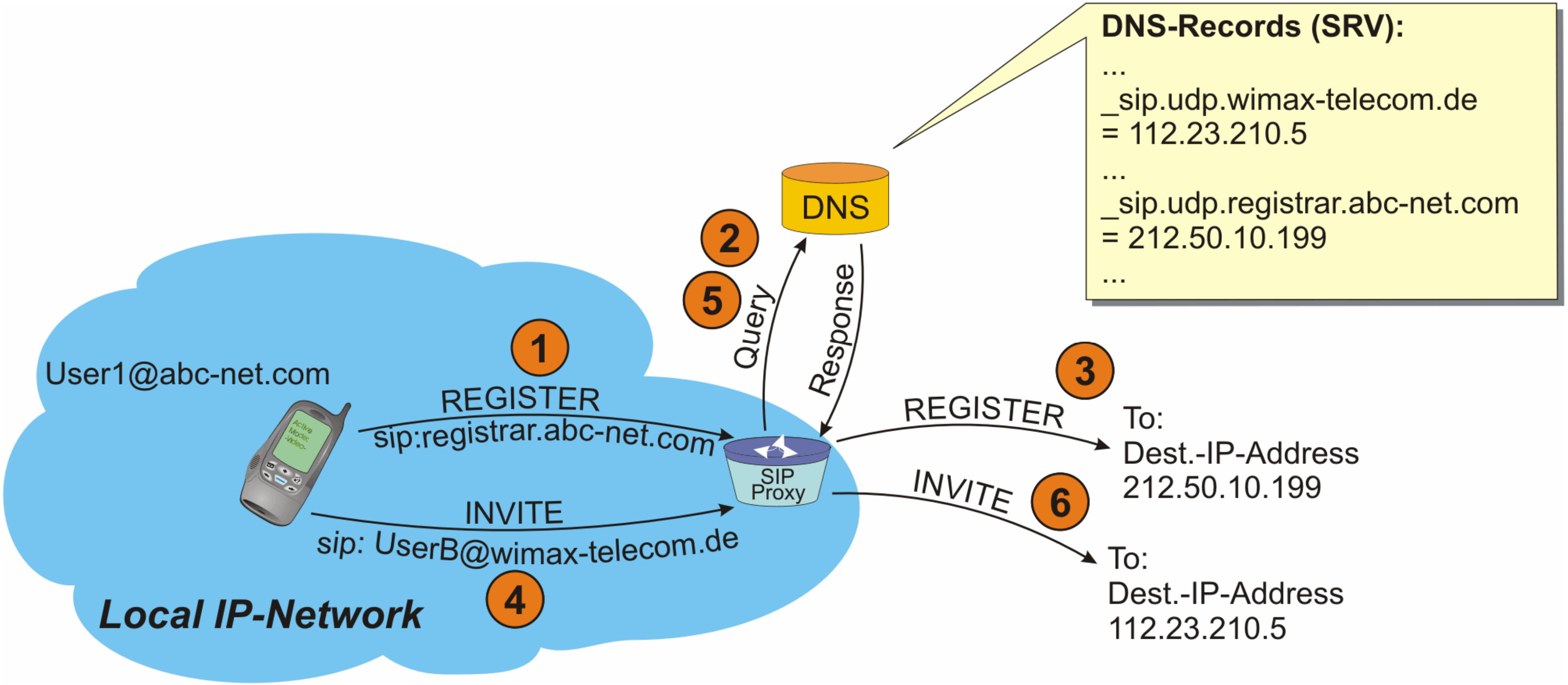
Announced by UA 2:
Session ID 2
(⇔ different value than Session ID 1)
Session Description Version 2
(⇔ different value than Session Description Version 1)
+
Media Descriptions

- SDP-Media Description related
- SDP-Media Description related
- SDP-Session Description related
- SDP-Session Description related

Session Identification Parameters at both Peers

- ⇒ The figure illustrates which parameters are used at each peer (UA) to identify a session uniquely. Note that session-ID's are in general different at both peers.
- ⇒ Intentionally, the connection IP-address of UA 1 is illustrated as being part of the session description items while the connection IP-address of UA 2 is illustrated as being part of the media description items. Both options are legitimate.
- ⇒ The codec type should be the same in both directions.

DNS-Queries with NAPTR- and SRV-Records



DNS-Records (SRV):
...
_sip.udp.wimax-telecom.de
= 112.23.210.5
...
_sip.udp.registrar.abc-net.com
= 212.50.10.199
...

DNS-Queries with NAPTR- and SRV-Records

- ⇒ The regular DNS-queries to resolve an FQDN into an IP-address can be expanded through service specific aspects.
- ⇒ That is the domain of the so called NAPTR- and SRV-records (Naming Authority Pointer / Service Location) that allow for the resolution of an FQDN and a service (e.g. SIP) into one or more IP-addresses within that domain that can be contacted.

NAPTR-Records

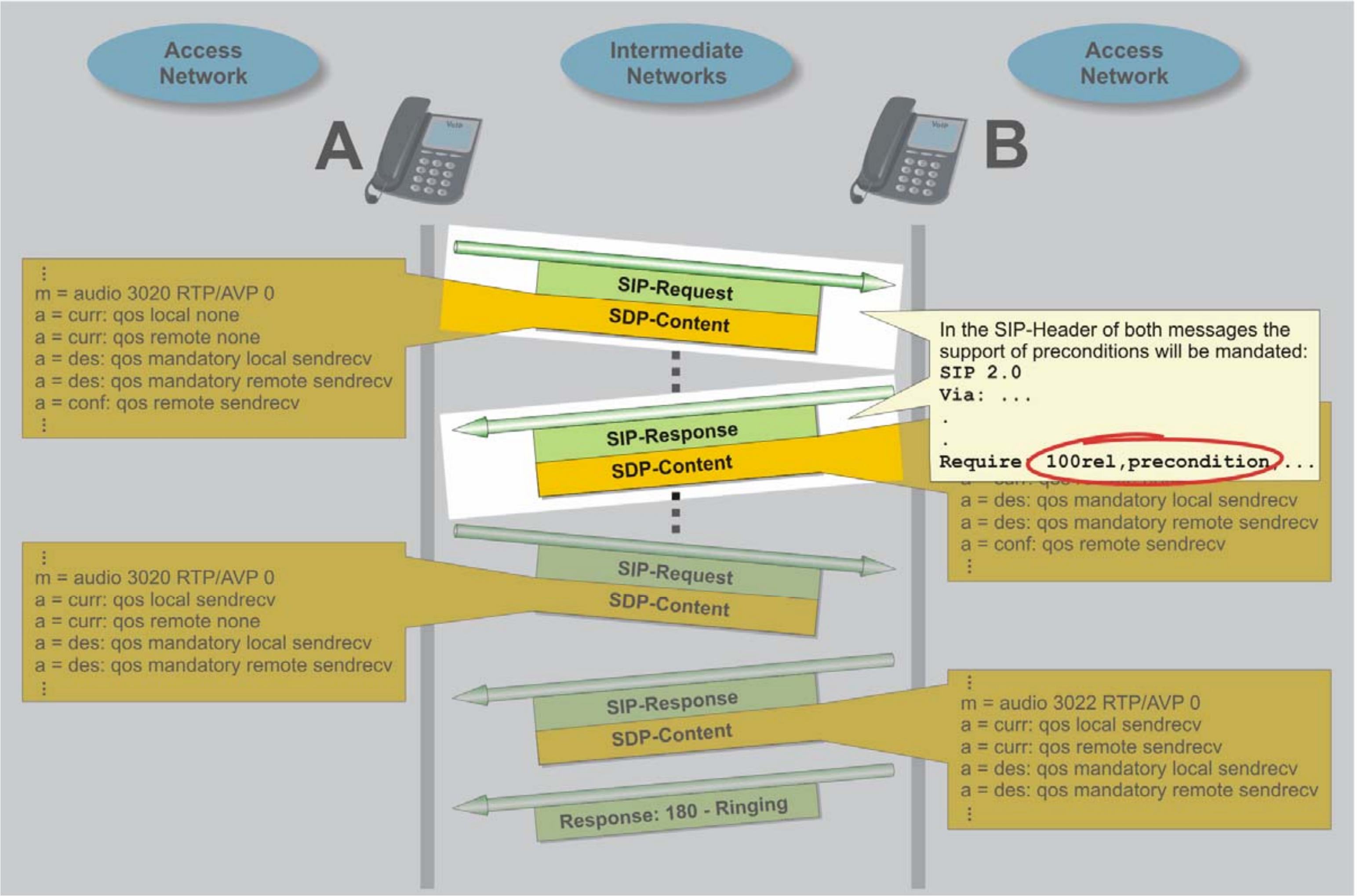
- ⇒ Usually, a query starts with the request for the NAPTR-records of a domain for a given service. Consider for instance that a DNS-query is received by a DNS server for sip:Miriam@inacon.com. The client performs a NAPTR query for that domain, and the DNS-server responds with the following NAPTR-records:
  “SIP+D2T” _sip.tcp.inacon.com
  “SIP+D2U” _sip.udp.inacon.com

SRV-Records

- ⇒ In a second step, the requesting client will decide to use UDP to contact the incoming SIP-server for inacon.com and will send a second DNS-query which resolves the SRV-record _sip.udp.inacon.com into an FQDN or directly into an IP-address which is then used as destination IP-address for the SIP-message to be sent to sip: Miriam@inacon.com

[RFC 2782 (SRV), RFC 2915 (NAPTR), RFC 3263]

The new Option Tag “precondition”



```
⋮
m = audio 3020 RTP/AVP 0
a = curr: qos local none
a = curr: qos remote none
a = des: qos mandatory local sendrecv
a = des: qos mandatory remote sendrecv
a = conf: qos remote sendrecv
⋮
```

In the SIP-Header of both messages the support of preconditions will be mandated:

```
SIP 2.0
Via: ...
.
.
Require 100rel,precondition,...
```

```
a = des: qos mandatory local sendrecv
a = des: qos mandatory remote sendrecv
a = conf: qos remote sendrecv
⋮
```

```
⋮
m = audio 3020 RTP/AVP 0
a = curr: qos local sendrecv
a = curr: qos remote none
a = des: qos mandatory local sendrecv
a = des: qos mandatory remote sendrecv
⋮
```

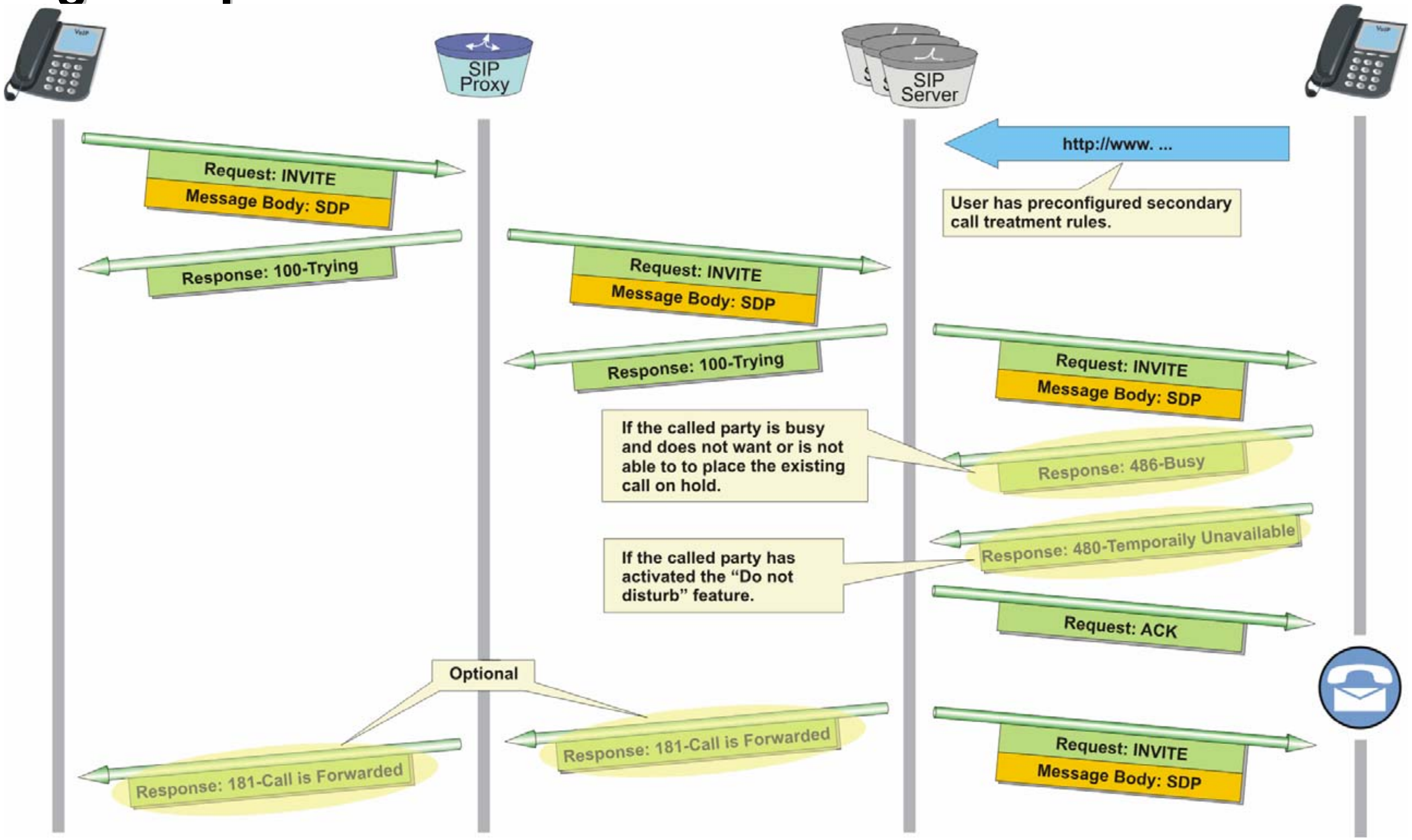
```
⋮
m = audio 3022 RTP/AVP 0
a = curr: qos local sendrecv
a = curr: qos remote sendrecv
a = des: qos mandatory local sendrecv
a = des: qos mandatory remote sendrecv
⋮
```

The new Option Tag “precondition”

- ⇒ The figure illustrates it: Preconditions within SDP-bodies cannot just be applied or used. The Request: INVITE and the related Response-messages will mandate the use of preconditions by including the option tag “precondition” in the “Require:”-header field [RFC 3312 (11)].
- ⇒ This header field shall be used, if preconditions are mandatory. Alternatively, the option tag “precondition” may be added to the “Supported:”-header field, if the preconditions are not mandatory.

Note that the support of “preconditions” also requires the support of “100rel” (acknowledged reliable provisional responses as per RFC 3262). Accordingly, this option tag shall also be present [RFC 3312 (11)].

(1) User Busy and "Do not Disturb" Feature – Detailed Message Sequence Chart



SIP Proxy

SIP Server

Request: INVITE
Message Body: SDP

Response: 100-Trying

Request: INVITE
Message Body: SDP

Response: 100-Trying

http://www. ...

User has preconfigured secondary call treatment rules.

Request: INVITE
Message Body: SDP

If the called party is busy and does not want or is not able to to place the existing call on hold.

Response: 486-Busy

Response: 480-Temporaily Unavailable

If the called party has activated the "Do not disturb" feature.

Request: ACK

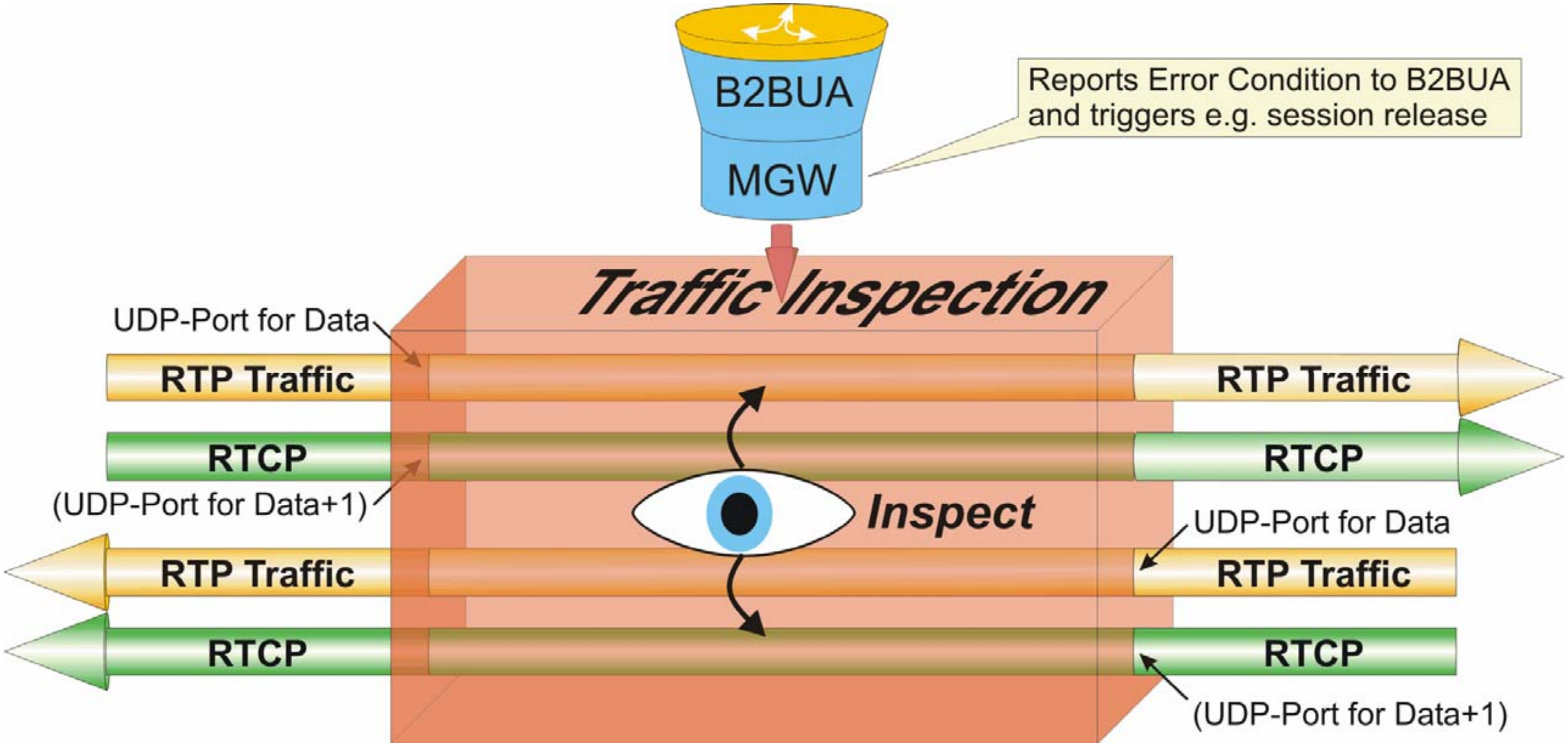
Optional

Response: 181-Call is Forwarded

Response: 181-Call is Forwarded

Request: INVITE
Message Body: SDP

Option 2 / Intro: the Idea is Media Stream Observation

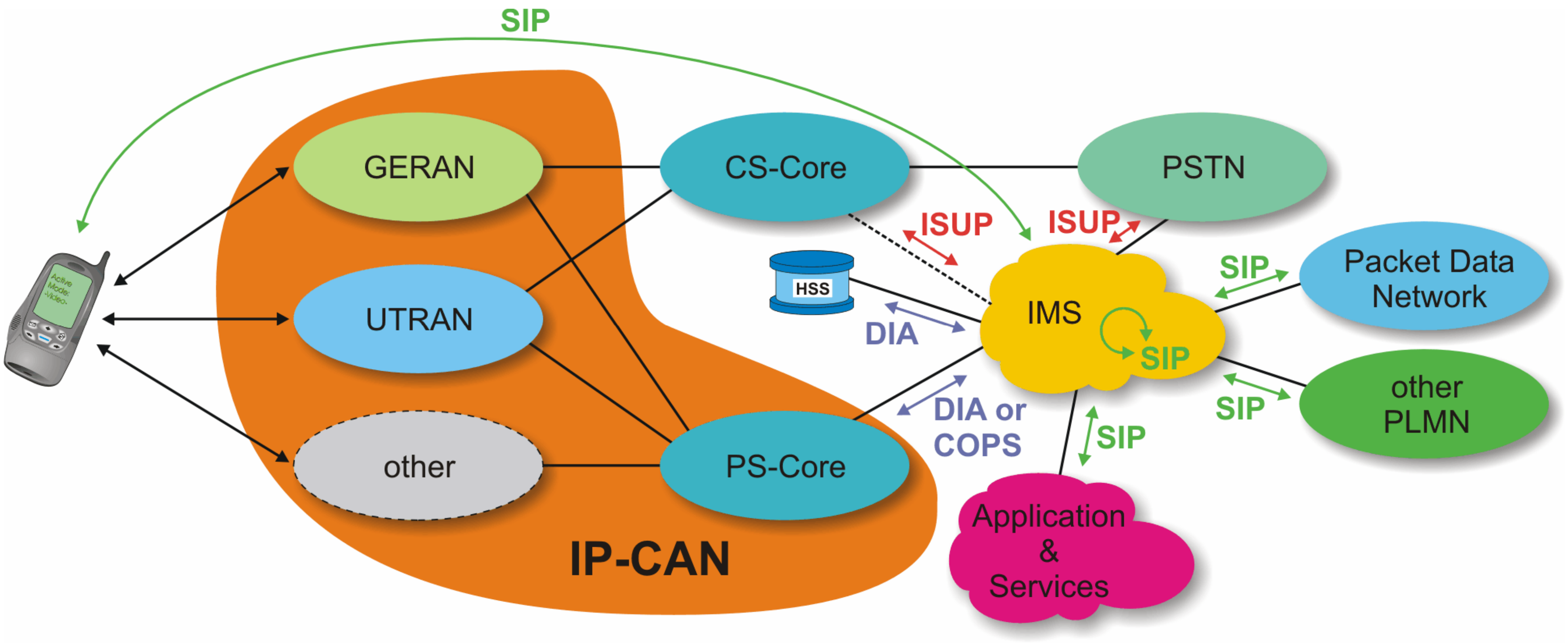


Option 2 / Intro: the Idea is Media Stream Observation

- **The SBC (combination of B2BUA and MGW) is best suited for this task**
- **Media Stream Observation may relate to both: Inspection of the actual media stream and the associated RTCP-Reports**
  RTCP has become more sophisticated and more powerful through the introduction of extended RTCP-reports (RTCP XR ⇔ RFC 3611) and through more sophisticated reporting options according to the internet draft draft-ietf-avt-rtcp-feedback-XX.txt.
- **If the Media Stream Observation reports back a fatal error of the media stream then it is the decision of the associated SIP-proxy server to terminate the session towards both peers**
  Note that there are currently no rules as what needs to be considered a fatal error. It may for instance be that the MGW does not receive data packets for some O&M-configurable time.

  [RFC 3611, RFC 4083 (14.1), draft-ietf-avt-rtcp-feedback-XX.txt]

Relationship between SIP, the IMS and 3GPP-Networks



Relationship between SIP, the IMS and 3GPP-Networks

⇒ The figure illustrates how the IMS is interconnected to the various entities and network clouds that a 3GPP-network consists of or that a 3GPP-network is connected to. For more details about the different logical entities within the IMS please refer to chapter 2 and to the course book “IMS – Architecture Details & System Engineering”.

⇒ More importantly, the figure illustrates how the IMS communicates with these different entities and network clouds. Intentionally, we did not include any notion of user data transfers to keep the illustration simple. Hence, the focus is purely on signaling.

⇒ Note that between the PS-Core (⇔ or more precisely the GGSN) and the IMS (⇔ or more precisely the P-CSCF) there may be a dedicated PDF (Policy Decision Function) or the PDF may be integrated into the P-CSCF or it may be integrated into the GGSN. Depending on this configuration, either COPS or DBP is used between the GGSN and the IMS for QoS-Policing [3GTS 23.228 (4.6.1), 3GTS 23.207 (5.1), 3GTS 29.209 (4.2)].

[3GTS 23.228]

???? Question Section 23 ????

⇒ In chapter 1 we stated explicitly that the IMS provides its services exclusively through the packet-switched domain. Why did we still need to insert the dotted line between the IMS and the circuit-switched domain?