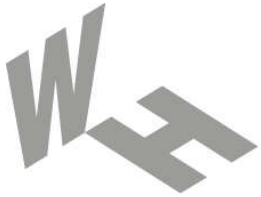


Westfälische Hochschule
Gelsenkirchen Bocholt Recklinghausen
University of Applied Sciences

IP-Telefonie, Voice over IP (VoIP)
- Session Initiation Protocol (SIP)
- Real-time Transport Protocol (RTP)

Prof. Dr. (TU NN)
Norbert Pohlmann

Institut für Internet-Sicherheit – if(is)
Westfälische Hochschule, Gelsenkirchen
http://www.internet-sicherheit.de

Inhalt

- Ziele und Einordnung
- Telefon und Datennetze
- SIP – Session Initiation Protocol
  - SIP – Nachrichten
  - SIP – Codierung
  - SIP – basierte Anwendungen
- RTP – Real-Time Transport Protocol
- VoIP
  - Codierung
  - Zusammenhänge

Inhalt

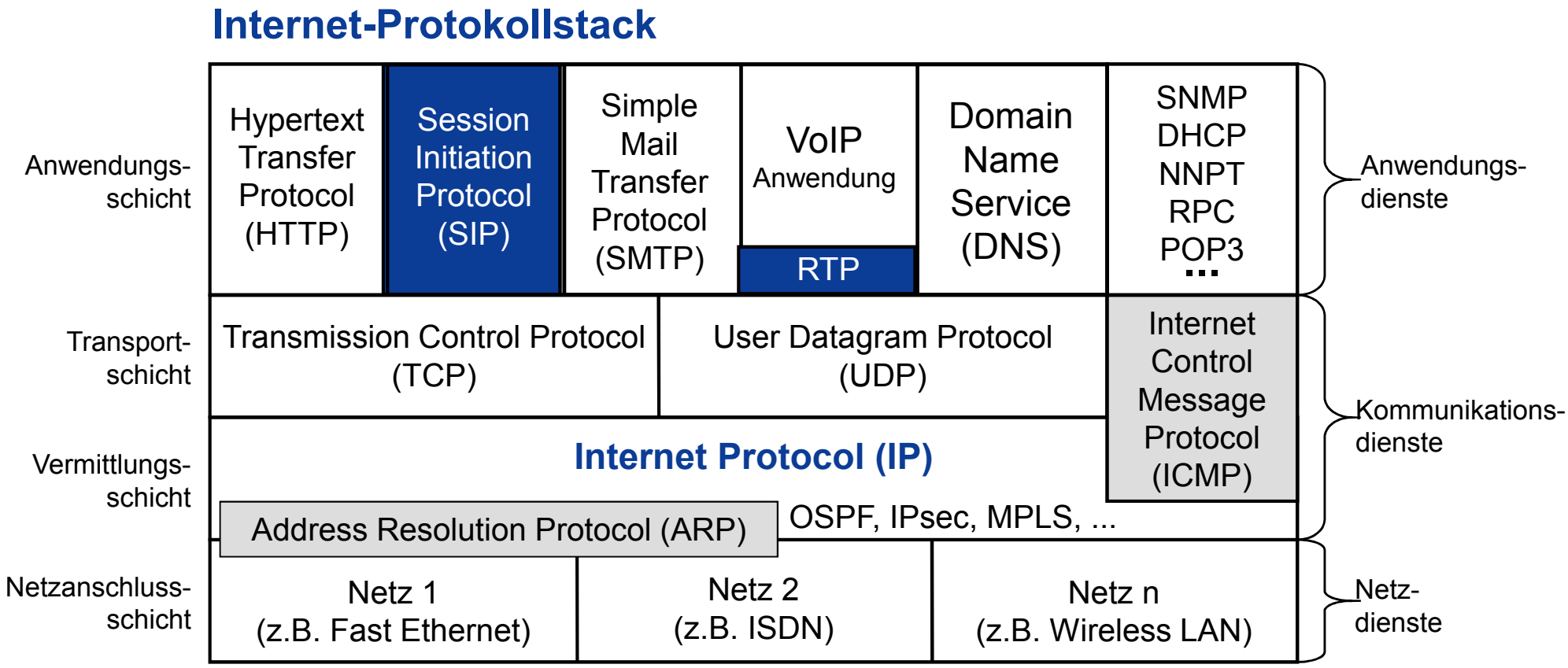
- **Ziele und Einordnung**

- Telefon und Datennetze
- SIP – Session Initiation Protocol
  - SIP – Nachrichten
  - SIP – Codierung
  - SIP – basierte Anwendungen
- RTP – Real-Time Transport Protocol
- VoIP
  - Codierung
  - Zusammenhänge

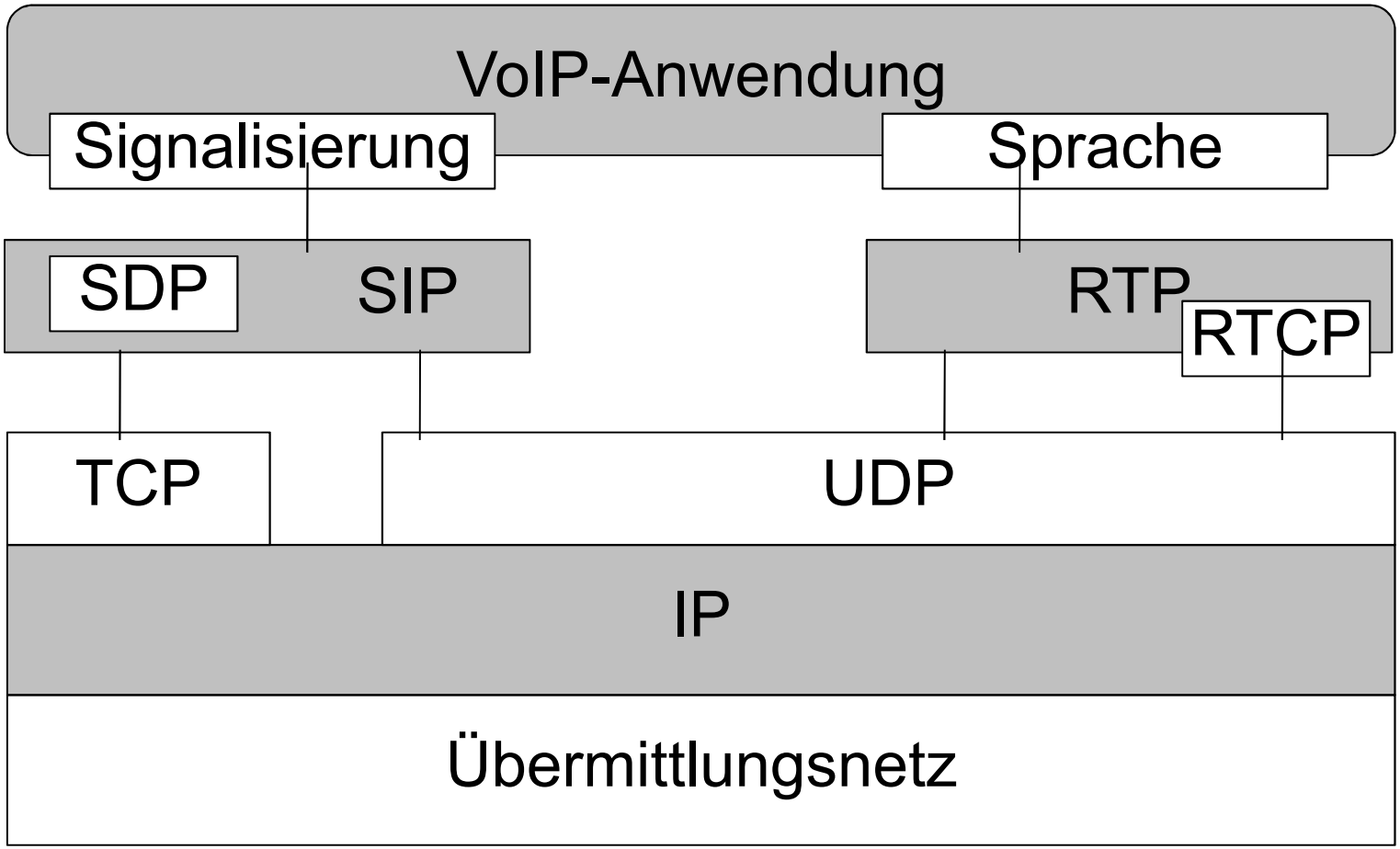
SIP – Session Initiation Protocol
→ Ziele

- Gutes Verständnis für eines der wichtigsten Anwendungsprotokolle der Internet-Telefonie, IP-Telefonie, Voice over IP (VoIP)
- Erlangen der Kenntnisse über die Aufgaben, Prinzipien und Mechanismen der VoIP-Protokolle (SIP, SDP, RTP, RTCP)
- Gewinnen von praktischen Erfahrungen über das SIP-Protokoll mit Hilfe von Protokollanalysen.

Die Anwendungsschicht
→ Session Inititation Protocol (SIP) – Einordnung (1/3)

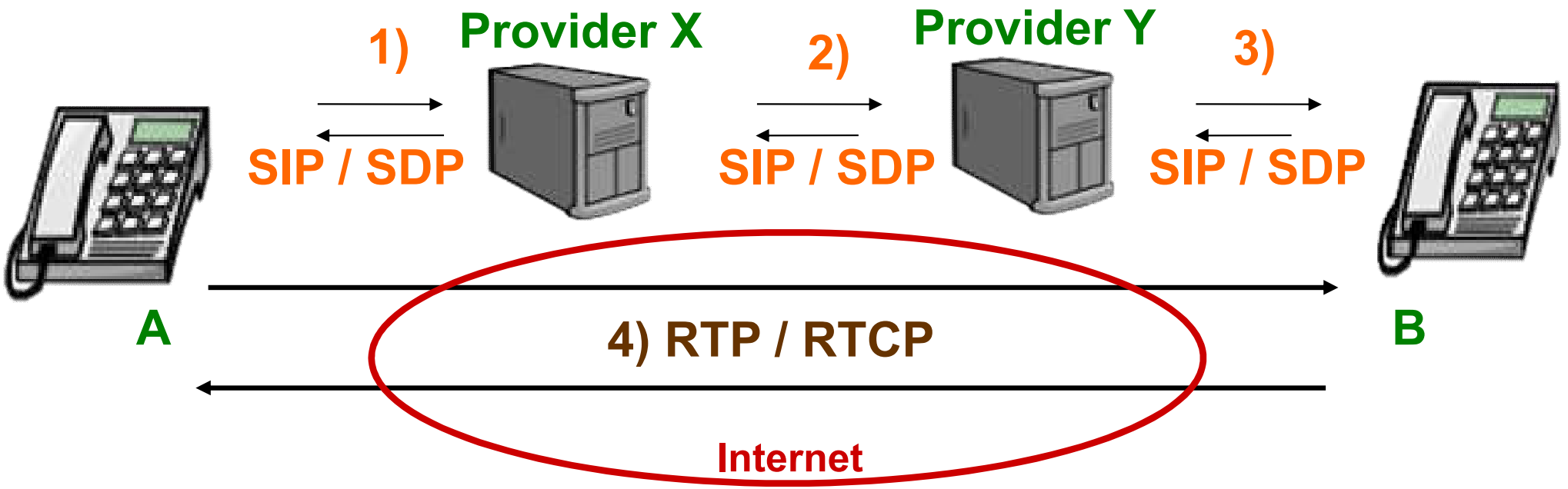


Die Anwendungsschicht
→ Session Initiation Protocol (SIP) – Einordnung (2/3)



Die Anwendungsschicht
→ Session Initiation Protocol (SIP) – Einordnung (2/3)

- 1) Signalisierung zwischen A und Provider X
- 2) Signalisierung zwischen Provider X und Provider Y
- 3) Signalisierung zwischen Provider Y und B
- 4) Mediendatenstrom zwischen A und B



Signalisierung (1 bis 3); **Media Transport (4);** **Quality of Service (QoS)**

Inhalt

- Ziele und Einordnung

- **Telefon und Datennetze**

- SIP – Session Initiation Protocol
  - SIP – Nachrichten
  - SIP – Codierung
  - SIP – basierte Anwendungen
- RTP – Real-Time Transport Protocol
- VoIP
  - Codierung
  - Zusammenhänge

Telefon und Datennetze
→ Einleitung (1/7)

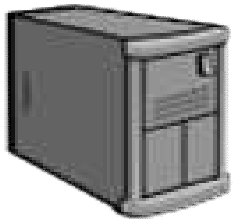
- Das Telefon existiert seit ca. 150 Jahren und ist inzwischen zum wichtigsten Kommunikationsmittel geworden.
- Heutzutage wird das World Wide Web auch als unverzichtbares Kommunikationsmittel betrachtet.
- Daten werden dort nicht analog, sondern digital und mit Hilfe des Internet Protokolls (IP) übertragen.
- Werden nun diese beiden Kommunikationsmittel kombiniert und die Sprache mit Hilfe des IP-Protokolls übertragen, wird die Anwendungen **Voice over IP (VoIP)** oder **IP-Telefonie** genannt.
- Die Folge davon ist ein Next Generation Network (NGN), ein konvergentes Netz, in dem die Übertragung von Daten-, Sprach-, und Videokommunikation angeboten wird.

Telefon und Datennetze
→ Einleitung (2/7)

- Einen wesentlichen Beitrag zum Next Generation Network wird das von der IETF spezifizierte Protokoll SIP (Session Initiation Protocol) leisten.
- SIP hat sich schon etabliert und ersetzt nach und nach die H.323 Protokollfamilie, die von der ITU-T entwickelt wurde.
- SIP ist ein transport unabhängiges Protokoll der Anwendungsschicht und dient zu Steuerung des Auf- und Abbaus und zur Modifikation der Sessions zwischen den beteiligten Kommunikationsteilnehmern.
- SIP stellt die Mechanismen zur Implementierung von multimedialen Anwendungen im Internet bereit.
- Mit VoIP werden neue Echtzeitanforderungen an das Internet gestellt, damit die Anwendungen funktionieren kann.

Telefon und Datennetze
→ Einleitung (3/7) - Legende

User Agent (UA)

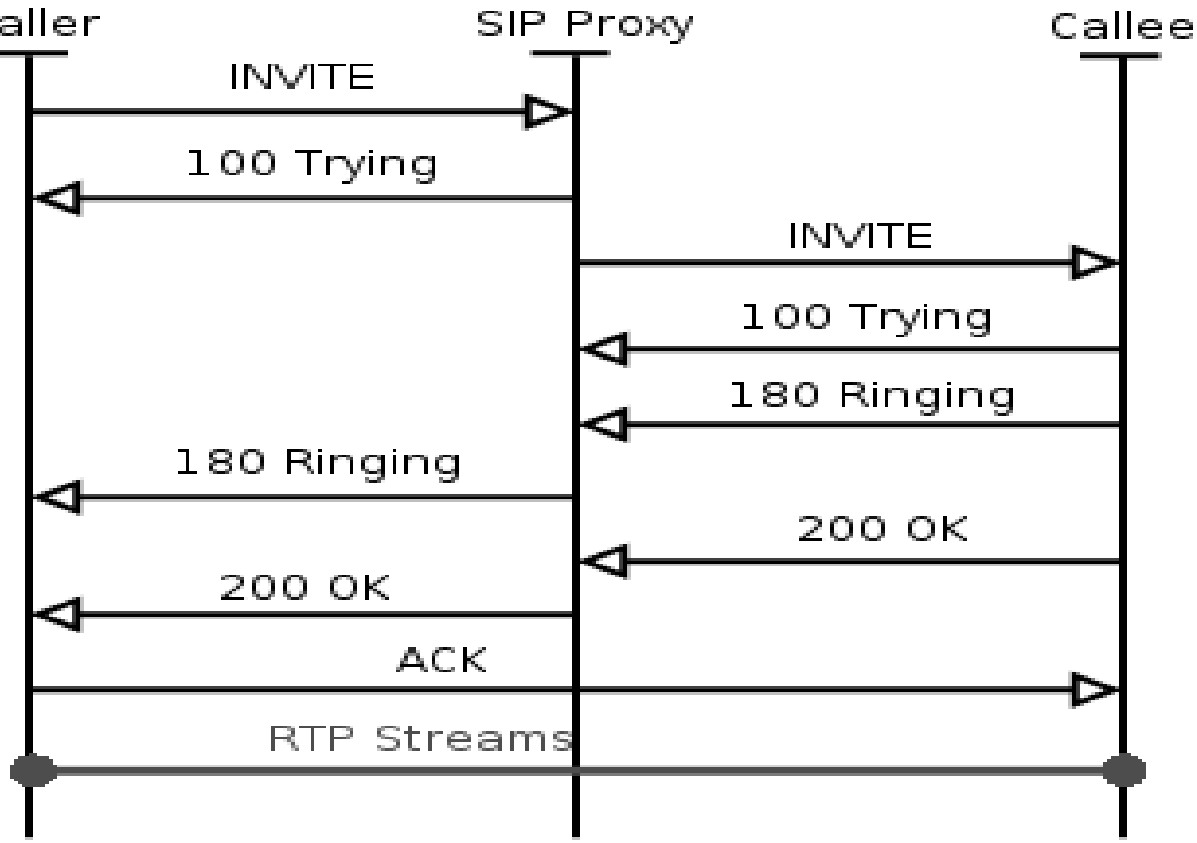


Proxy

Telefon und Datennetze
→ Einleitung (4/7)

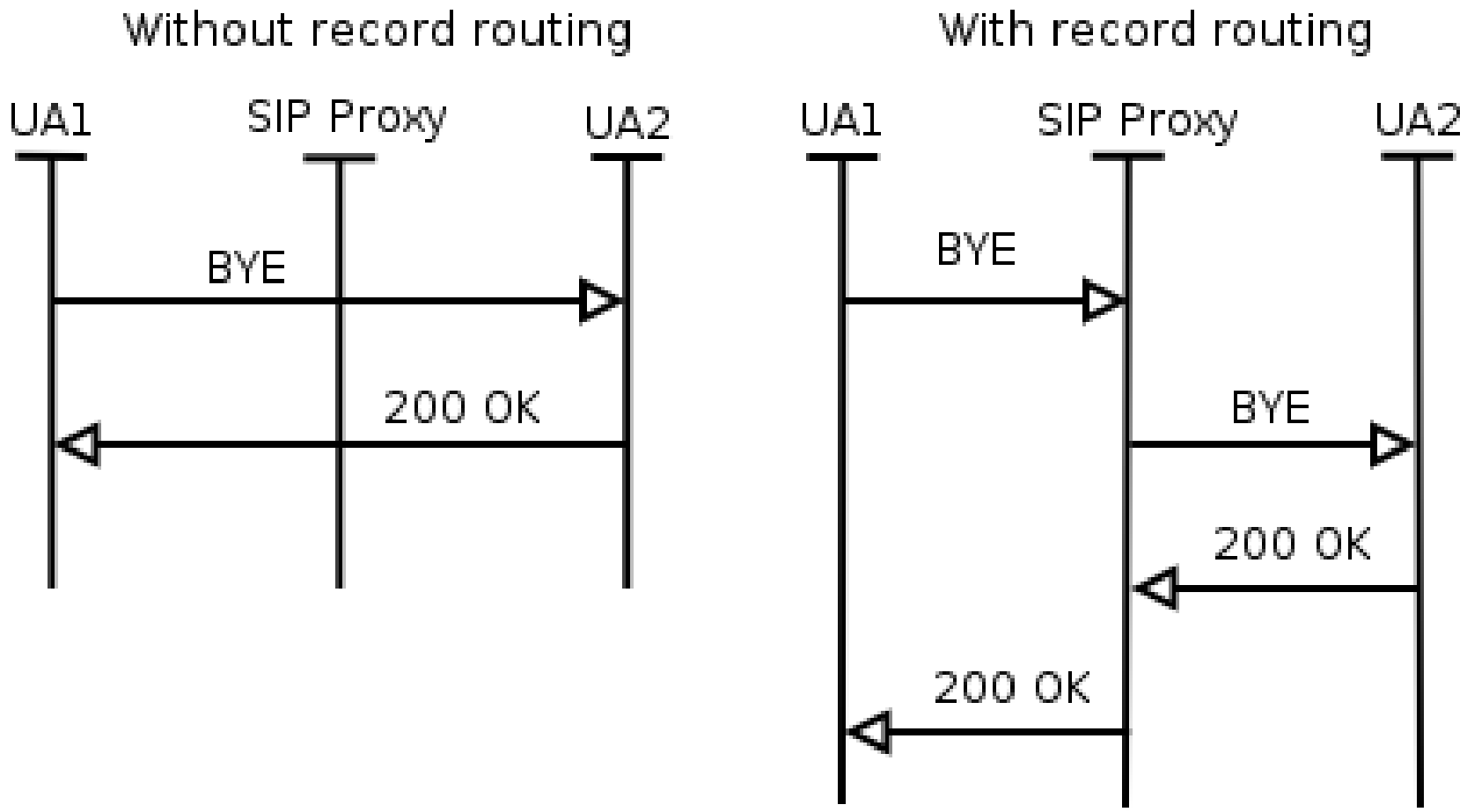
- SIP kann aber nicht nur Sprachübertragung im Internet realisieren, es ist auch dazu geeignet, digitale Fernsehnbilder und Videos im Internet oder aber auch in zellularen Netzen zu organisieren.
- **Der Unterschied zur herkömmlichen Telefonie besteht darin, dass die Telefonnummern nicht mehr ortsgebunden sind!**
- Ein Kommunikationsteilnehmer ist nach seiner Registrierung bei seinem Dienstanbieter weltweit, wo auch immer ein Internetanschluss ist, erreichbar.
- Jeder User Agent (UA), der an einer SIP Session teilnehmen möchte, wird über seine SIP-Identität angesprochen, der sogenannten SIP-URI (SIP Uniform Resource Identifier).
- Die logische Instanz, die die SIP-Nachrichten erzeugt, und die Antworten des User-Agent-Servers (UAS) bekommt, nennt man User-Agent-Client (UAC)
- Ein wichtiger Bestandteil der SIP-Kommunikation ist der SIP-Proxy, er muss beide logischen Instanzen repräsentieren.

Telefon und Datennetze
→ Einleitung (5/7)



- Ein SIP-Proxy vermittelt die Kommunikation, danach zieht er sich aus der eigentlichen Datenübertragung zurück.
- SIP ist dem Transaktionmodell von HTTP nachempfunden, wonach auf jeden Request ein Response folgt.
- Voraussetzung für eine funktionierende Kommunikation ist, dass beide Kommunikationsteilnehmer bei ihrem Service-Anbieter angemeldet sind.

Telefon und Datennetze
→ Einleitung (6/7)



- Optional kann der SIP-Proxy über das Feld “Record-Route” im SIP-Header den Endgeräten mitteilen, ob weitere Signalisierungen über ihn geroutet werden oder nicht.

Telefon und Datennetze
→ Einleitung (7/7)

- VoIP ist also keine neue Technologie, sondern es benutzt bestehende Technologien und Entwicklungen, die sich schon bewährt haben.
- Die Kommunikation basiert auf Nachrichtenaustausch, wie z.B. HTTP oder SMTP.
- SIP dient zur Signalisierung, RTP zum Transport der Nutzdaten in Echtzeit.
- Endpunkte können sowohl mit Telefonnummern (0209/123456) als auch mit einer SIP-URL (SIP:Pohlmann@ifis-voip.de) angesprochen werden.
- Dies hängt vom Anbieter und von der Implementierung der Systemkomponenten ab.

Inhalt

- Ziele und Einordnung
- Telefon und Datennetze
- **SIP – Session Initiation Protocol**
  - SIP – Nachrichten
  - SIP – Codierung
  - SIP – basierte Anwendungen
- RTP – Real-Time Transport Protocol
- VoIP
  - Codierung
  - Zusammenhänge

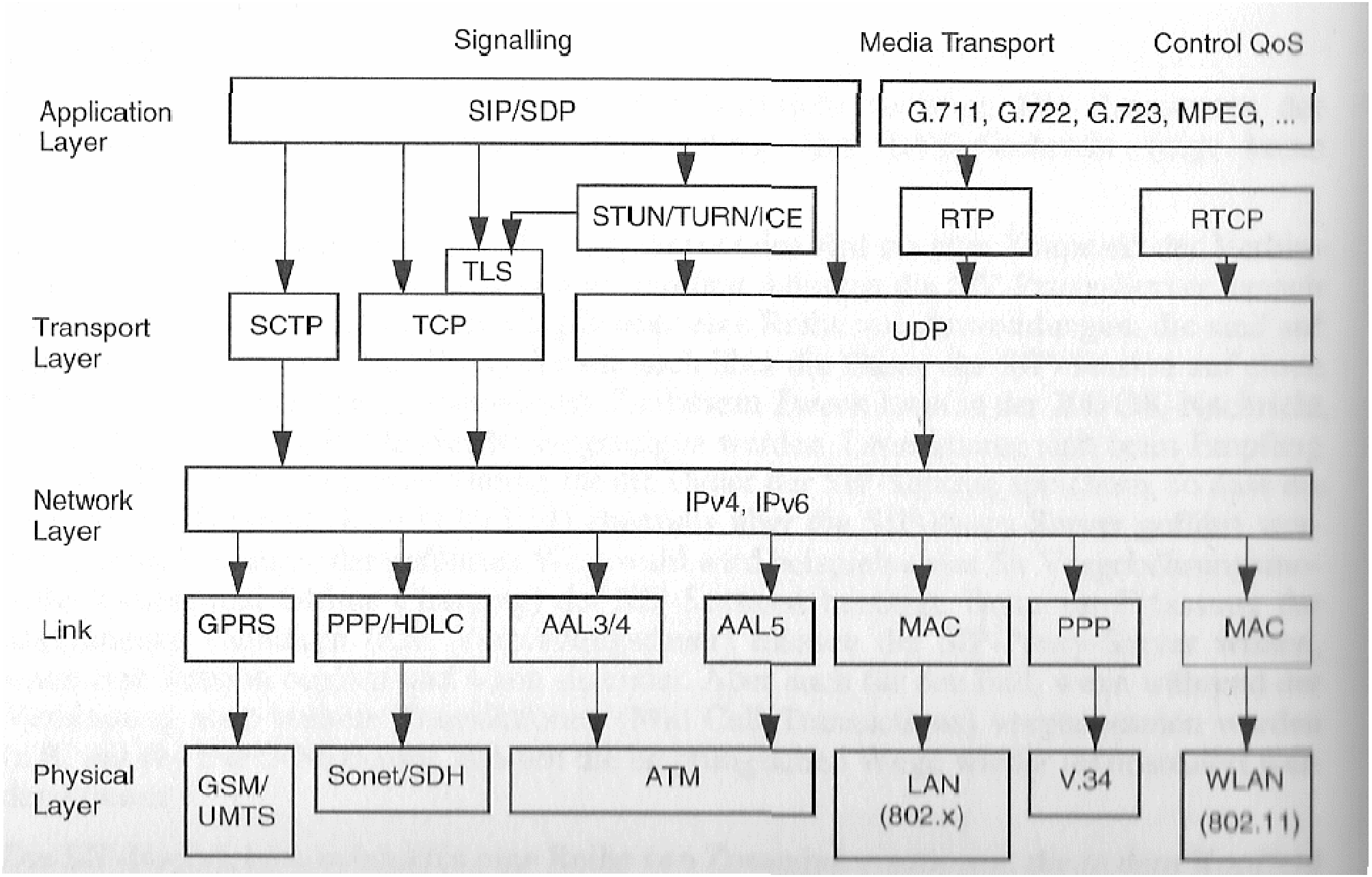
SIP – Session Initiation Protocol

- 1999 -> RFC 254
  - Das Protokoll erlangte schnell breite Akzeptanz, da es einfachen Protokollen wie z.B. http nachempfunden wurde.
  - Aus diesem Grund wurde SIP weiterentwickelt.
- 2002 -> RFC 3261
  - Allgemeine Beschreibung
- RFC 3262 bis RFC 3265
  - Dient der Darstellung von Zusatzfunktionen

SIP – Session Initiation Protocol
→ Besonderheiten

- SIP ist ein Protokoll von der IETF für die Signalisierung der Anrufe bei Multimediakommunikationen.
- Mit Hilfe von SIP werden RTP-Sessions ausgebaut, mit denen Sprache, Video oder auch Daten ausgetauscht werden.
- Parameter über die Art der zu übertragenen Medien sowie die gewählte Kodierung der RTP Sessions werden nach dem SDP Protokoll (Session Description Protcol) im “SIP-Body” ausgehandelt.
- SIP ist an HTTP angelehnt, es funktioniert nach dem Request/Response-Prinzip.
- Die Nachrichten werden nach der gleichen Syntax wie HTTP/1.1 codiert.
- SIP Kommunikationsteilnehmer werden über einen URL adressiert, die Auflösung der Adresse erfolgt via DNS (Domain Name System).
- Zum Schutz vor Missbrauch/Manipulation und Vertraulichkeit kann das SIP-Protokoll “SSL/TLS”-gesichert werden.
- Name: SIPS – Session Initiation Protocol Secure
  Default Port: 5061 (statt **Port 5060 bei SIP**)

SIP – Session Initiation Protocol
→ Protokollarchitektur (1/2)



SIP – Session Initiation Protocol
→ Protokollarchitektur (2/2)

- Die multimedialen Eigenschaften der UAs werden durch die SIP-Zeichengabe gesteuert.
- Es wird zwischen **Signalling, Media Transport** und **Control QoS** unterschieden.
- Je nach Netzarchitektur und Implementierung kann diese Unterteilung eine monolithische Einheit bilden oder auf verschiedene Netzelemente verteilt werden.
- Liegt die letztere Form vor, sind zusätzliche Protokolle für die Kommunikation zwischen **SIP-Zeichengabe, Control-QoS** und **Media Transport** notwendig.
- Den eigentlichen Transport des Payloads übernimmt das Realtime Transport Protocol (RTP).

Inhalt

- Ziele und Einordnung
- Telefon und Datennetze
- SIP – Session Initiation Protocol
  - **SIP – Nachrichten**
    - SIP – Codierung
    - SIP – basierte Anwendungen
- RTP – Real-Time Transport Protocol
- VoIP
  - Codierung
  - Zusammenhänge

SIP – Nachrichten und Adressen
→ Nachrichtenaufbau

SIP-Nachrichten sind textcodiert Nachrichten und benutzen den Zeichensatz UTF-8.

Eine Nachricht setzt sich aus einer Startzeile, dem Header und dem Body zusammen, wobei der Body optional ist.

Mit der ersten Zeile wird erkannt, um welchen Typ von Nachricht es sich handelt.

Sie muss mit "**CRLF**" abgeschlossen werden.

In RFC 3261 wurde folgende Syntax definiert:

Request =

Method SP Request-URI SP SIP-Version CRLF

Response =

SIP-Version SP Status-Code SP Reason-Phrase CRLF

(SP: Leerzeichen , CRLF: Carriage-Return Line Feed)

SIP – Nachrichten und Adressen
→ Request-Typen (1/4)

- In SIP werden die PDUs als Methode bezeichnet.
- Die Basisspezifikation von SIP (RFC 3261) definiert sechs Request-Typen/Methoden.
- **Request** = **Method SP Request-URI SP SIP-Version CRLF**

1.) **Invite**

Ein Kommunikationsteilnehmer initiiert einen neuen Anruf mit INVITE und übermittelt dem System dabei beide SIP-Adressen, den Grund und die Priorität des Anrufs.

2.) **BYE**

Diese Nachricht initiiert den Abbau einer bestehende RTP-Verbindung zwischen den Kommunikationsteilnehmern.

3.) **ACK**

ACKnowledgement bestätigt die Annahme des Anrufs

SIP – Nachrichten und Adressen
→ Request-Typen (2/4)

4.) **OPTIONS**

Mit OPTIONS kann ein Endgerät nach seiner Fähigkeit befragt werden, z.B. Art der Medien oder Auswahl der Codierungen

5.) **CANCEL**

Im Gegensatz zu BYE wird hier der Aufbau einer initiierten Verbindung abgebrochen.

6.) **REGISTER**

Hiermit werden dem Registrar Informationen über die Lokalität (IP-Adresse) der einzelnen Kommunikationsteilnehmer bekannt gegeben.

SIP – Nachrichten und Adressen
→ Request-Typen (3/4)

- Erweiterungen von SIP (RFC 3261) sind :

1.) **INFO**

Austausch von Zusatzinformationen während einer bestehenden RTP-Session (RFC 2976).

2.) **PRACK**

Provisional Response Acknowledgement.
Bestätigung eines Response, sodass eine zuverlässige Übermittlung, auch bei UDP, garantiert werden kann (RFC 3262)

3.) **UPDATE**

Veränderung bestimmter Parameter bereits beim Aufbau einer RTP-Session (RFC 3311).

SIP – Nachrichten und Adressen
→ Request-Typen (4/4)

4.) **MESSAGE**

Gehört zur Realisierung der Instant Messaging Funktion (RFC 3311)

5.) **REFER**

Ermöglicht den Session Transfer (RFC 3515)

6.) **SUBSYRIBE & NOTIF**

Übermittelt bestimmte Ereignisse (RFC 3261 & RFC 3265)

7.) **PUBLISH**

Methode zur Veröffentlichung von Ereignissen

SIP – Nachrichten und Adressen
→ Response-Typen (1/3)

- Die Response Nachrichten haben eine ähnliche Struktur wie beim HTTP Protokoll.
- Es gibt sechs verschiedene Klassen
- **Response = SIP-Version SP Status-Code SP Reason-Phrase CRLF**

1xx: Provisional Response (”provisorische Statusinformationen“):

Hiermit werden Nachrichten beantwortet, deren Bearbeitung noch nicht abgeschlossen ist.

- 100 Trying,
- 180 Ringing

2xx: Succesful (erfolgreich): Kennzeichnet eine erfolgreich bearbeitete Nachricht, z.B. Annahme der Verbindung.

- 200 OK

SIP – Nachrichten und Adressen
→ Response-Typen (2/3)

3xx: Redirection (Umleitung): gibt Informationen darüber, wo ein Benutzer erreicht werden kann oder welche Alternativen es gibt.

- 301 Moved Permanently, - 302 Moved Temporarily

4xx: Client Error (Client Fehler): wenn der Request beim Server zu einem Fehler führte (User- Agent-Client muss seinen Request abändern).

- 400 Bad Request, 404 Not Found

5xx: Server Error (Servers Fehler): Antwort auf eine Nachricht, die aus Server-bedingten Gründen nicht bearbeitet werden konnte (User-Agent-Client kann den gleichen Request erneut stellen).

- 500 Internal Server Error, 502 Bad Gateway

6xx: Global Failure (Genereller Fehler): Die Bearbeitung wird durch den Server aus generellen Gründen abgelehnt.

- 600 Busy Everywhere, 606 Not Acceptable

SIP – Nachrichten und Adressen
→ Response-Typen (3/3)

Typ	Statuscode	Fehler
Informational	100	Trying
	180	Ringing
	181	Call Is Being Forwarded
	182	Queued
Success	200	OK
Redirection	300	Multiple Choices
	301	Moved Permanently
	302	Moved Temporarily
	303	See Other
	305	Use Proxy
	380	Alternative Service
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
	420	Bad Extension
	480	Temporarily not available
	481	Transaction Does Not Exist
	482	Loop Detected
	483	Too Many Hops

Client-Error	484	Address Incomplete
	485	Ambiguous
	486	Busy Here
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

SIP – Nachrichten und Adressen
→ Message Header (1/7)

- Nach der obligatorischen Start-Line folgt ein Message-Header.
- Jede Kopfzeile kann sich über mehrere Zeilen erstrecken und beginnt mit einem reservierten Feldnamen, dem ein Feldwert folgt.
- Feldname und -wert sind durch einen „:“ getrennt.
- Ein gültiger SIP-Request besteht aus mind. 6 Feldnamen, da diese zwingend vorgeschrieben sind (siehe Tabelle).

SIP – Nachrichten und Adressen
→ Message Header (2/7)

Reservierte Feldnamen:

Header field Long/short	Where	Proxy	ACK	BYE	CANCEL	INVITE	OPTIONS	REGISTER	REFER
Call-ID/i	c	r	m	m	m	m	m	m	m
CSeq	c	r	m	m	m	m	m	m	m
From/f	c	r	m	m	m	m	m	m	m
Max-Forwards	R	amr	m	m	m	m	m	m	m
To/t	c	r	m	m	m	m	m	m	c
Via/v	R	amr	m	m	m	m	m	m	c
Via/v		dr	m	m	m	m	m	m	c
Content-Type/c			m*	m*	-	m*	m*	m*	m*
Con-tent-Length/l		ar	t	t	t	t	t	t	t
Refer-To/r	R		-	-	-	-	-	-	m

(Method: m=mandotory, m*=required if message body is not empty, t=should be sent; Where: c=conditional, R=for Request, c=copied from the request to the response; Proxy: a=proxy can add if not present, r=proxy must be able to read the header field, m=proxy can modify the value, c=proxy can delete a value)

SIP – Nachrichten und Adressen
→ Message Header (3/7)

- TO-Feld:
  - Adressiert den gewünschten Kommunikationsteilnehmer mit einer Request-URI
  - Kann auch Display-Name enthalten -> URI muss dann in <> gesetzt werden
  - Zur Identifizierung des Dialogs kann ein eindeutiges “Tag” hinzugefügt werden
  - **To: Leonie <sip:1031@172.16.50.203>;tag=as38f6cfde**

- From-Feld:
  - Identifiziert den Urheber ein SIP-Kommunikation
  - Besteht aus optionalem Diaplay-Name, From-URI & From-Tag
  - From-Tag ist eine 32 Bit Zufallszahl zur eindeutigen Unterscheidung
  - **From: "Leonie" <sip:1000@voip>;tag=3213308c-6f8e-db11-92b0-000d61122a90**

SIP – Nachrichten und Adressen
→ Message Header (4/7)

- Cseq-Feld:
  - Sequenznummer & Name einer Methode
  - Nummer kann willkürlich gewählt werden, darf sich aber während eines Dialogs nicht verändern.
  - **CSeq: 1 INVITE**

- Call-ID-Feld:
  - Jede neue SIP-Anforderung bekommt eine neue Call-ID
  - Besteht aus Zufallszahlen und Komponeten aus Zeit und Raum.
  - **Call-ID: 020c308c-6f8e-db11-92b0-000d61122a90@172.16.50.xxx**

SIP – Nachrichten und Adressen
→ Message Header (5/7)

- Max-Forwards-Feld:
  - Ähnlich wie beim TTL-Feld (IPv4) wird auch hier der Wert nach jedem Durchlauf eines SIP-Proxy-Server dekrementiert.
  - Ist der Wert 0, wird die Nachricht verworfen und mit 483 (To Many Hops) geantwortet,
  - **Max-Forwards: 70**

- Via-Feld:
  - Enthält den Protokollnamen, die Version eine Branch-ID und weitere optionale Parameter
  - Die Adresse des nächsten SIP-Proxy wird hier festgelegt.
  - **Via: SIP/2.0/UDP 172.16.50.203:5060;branch=z9hG4bK41c250a0;rport**

SIP – Nachrichten und Adressen
→ Message Header (6/7)

- Content-Type-Feld:
  - Angabe der verwendeten Mediadaten
  - Können diskrete (image, audio, video...) oder auch zusammengesetzte (message, multipart...) Typen sein
  - Setzt sich aus einem Mediatyp und einem Media -Subtyp zusammen, die durch ein „/“ getrennt sind
  - **Content-Type: application/sdp**

SIP – Nachrichten und Adressen
→ Message Header (7/7)

- Content-Length-Feld:
  - Anzahl der Oktetts des Nachrichtenkörpers
  - **Content-Length: 391**

- Record-Route-Feld:
  - Gibt an, ob ein SIP-Proxy-Server für zukünftige Anforderungen im Dialog verbleiben will.
  - **Route: <sip:172.16.50.203:5060;lr>**

SIP – Nachrichten und Adressen
→ Message Body (1/5) - Session Description Protocol (SDP)

- Die Nutzlast einer SIP Nachricht ist der sogenannte Message Body.
- Der Body enthält die Parameter der RTP Session.
- **Die Beschreibung der Parameter legt das Session Description Protocol (SDP) fest.**
- SDP dient dazu, die zwischen den Endpunkten zu verwendenden Codecs, Transportprotokolle, usw. auszuhandeln.
- **SDP besitzt folgende Eigenschaften:**
  - Name und Zweck der Session
  - Zeitangaben, wann die Session aktiv sein soll
  - Mediatyp bezogene Empfanginformationen (Adresse, Formate ...)
  - Angabe der erforderlichen Bandbreite
  - **Kontakinformationen mit Transport Port**
  - Mediatypen
  - Transportprotokoll (RTP/UDP/IP, H.320 ...)
  - Mediaformat (H.261, video, MPEG video, ...)
  - Multicastadresse
  - Angabe des Transport Ports

SIP – Nachrichten und Adressen
→ Message Body (2/5) - Session Description Protocol (SDP)

- Das Session Description Protocol (SDP) ist eine reine textuelle Beschreibungssprache, welche zur Codierung der Mediaströme den UTF-8 nutzt.
- Jede Zeile beginnt mit einem Buchstaben, der den Mediatyp und einige Eigenschaften der Session kennzeichnet.
- Die Codierung der Mediaströme besteht aus der Vorgabe der Session-Description (SD) und optional der Vorgabe der Media-Description (MD).
- Die Kennzeichnung von Strömen beginnt immer mit der Session-Descritpion (SD), gefolgt von der optionalen Media-Descritpion (MD).
- Eine SD beginnt immer mit dem Buchstaben „v“ und endet mit dem Buchstaben „m“ am Anfang einer neuen Zeile.
- Eine MD beginnt mit einem „m“, endet mit einem „m“ oder erreicht das Ende einer SD.

SIP – Nachrichten und Adressen
→ Message Body (3/5) - Session Description Protocol (SDP)

- Die Reihenfolge der Buchstaben muss eingehalten werden, alle mit einem „*“ versehenen Einträge sind optional.
- Textzeilen sehen im Allgemeinen wie folgt aus:

<type>=<value>SP<value>SP.

SIP – Nachrichten und Adressen
→ Message Body (4/5) - Session Description Protocol (SDP)

- **Felder:**
  - v = Protokoll-Version
  - o = Ersteller der Session und Session-Identifizierung
  - s = Name der Session
  - i = zusätzliche Session-Informationen (optional)
  - u = URI mit der weiterführender Beschreibung (optional)
  - e = E-Mail-Adresse (optional)
  - p = Telefon-Nummer (optional)
  - c = Verbindungs-Information (wird nicht benötigt, wenn bei allen Medien angegeben, optional)
  - b = Information über die Bandbreite (optional)
  - Eine oder mehrere Zeit-Beschreibungen (s. u.)
  - z = Zeitzonen-Anpassungen (optional)
  - k = Verschlüsselungs-Schlüssel (encryption key) (optional)
  - a = ein oder mehrere Session-Attribute (optional)
  - Keine oder mehrere Medien-Beschreibungen (s.u.)

SIP – Nachrichten und Adressen
→ Message Body (5/5) - Session Description Protocol (SDP)

- **Zeit-Beschreibungen:**
  - t = Zeit, in der die Session aktiv ist
  - r = keine oder mehr Wiederholungen (optional)

- **Beschreibung der Medien:**
  - m = Medienname und Adresse für den Medientransport (i. d. R. IP/Adresse und Port)
  - i = Titel des Mediums (optional)
  - c = Verbindungs-Informationen, optional wenn diese nicht in der Session definiert sind (s. o, optional)
  - b = Information über die Bandbreite (optional)
  - k = Verschlüsselungs-Schlüssel ("encryption key", optional)
  - a = ein oder mehrere Session-Attribute (optional)

SIP – Nachrichten und Adressen
→ Struktur von SIP-Adressen

- **info@domain[;url-parameter]**
  - info: Username oder Nummer
  - domain: Domain oder Hostname oder IP-Adresse
  - Url-parameter sind optional, z.B.: transport = UDP
- SIP Adressen sind wie E-Mail-Adressen aufgebaut und können somit durch ein DNS System aufgelöst und in IP-Adressen umgesetzt werden.
- Die Architektur der SIP-Adresse bietet auch die Möglichkeit, diese Art der Kommunikation in bestehende Systeme leicht zu implementieren, dass sie wie E-Mail-Adressen aufgebaut sind.
- **Beispiele:**
  - 1000@hostname, 1000@172.16.50.230, pohlmann@voip.de, ...

Session Inititation Protocol (SIP)
→ Standards und Literatur

- **RFCs**
  - 1996 RFC 1889 (RTP)
  - 1999 RFC 2543 (SIP)
  - 2002 RFC 3261 (SIP-Erweiterung)

- **Literatur**
  - Kanbach, Andreas: **„SIP Die Technik“**; vieweg, 2005, BRD

Inhalt

- Ziele und Einordnung
- Telefon und Datennetze
- SIP – Session Initiation Protocol
  - SIP – Nachrichten
  - **SIP – Codierung**
  - SIP – basierte Anwendungen
- RTP – Real-Time Transport Protocol
- VoIP
  - Codierung
  - Zusammenhänge

SIP – Codierung
→ Registrierung (1/10)

- Damit ein Kommunikationsteilnehmer in einem Netz für andere erreichbar ist, muss dieser sich zunächst in seinem Heimnetz (Provider, SIP-Proxy) angemelden.
- Die sogenannte Registrierung wird zwischen dem anmeldenden Kommunikationsteilnehmer und einem SIP-Location-Server (SIP-LS) durchgeführt
- Nicht nur der Kommunikationsteilnehmer (UA), auch die SIP-Applikation-Server (SIP-AS) müssen innerhalb der Verwaltungseinheit einem SIP-LS bekannt gemacht werden.
- Ein SIP-Registrar ist ein Netzelement, welches Registrierungsanforderungen entgegennimmet und Kontakt zum SIP-LS hat.
- Ein Registrar kann eine eigene Einheit sein und über eine Schnittstelle angesprochen werden, oder aber zusammen mit einem SIP-Proxy-Server (SIP-P) eine physikalische Einheit bilden.

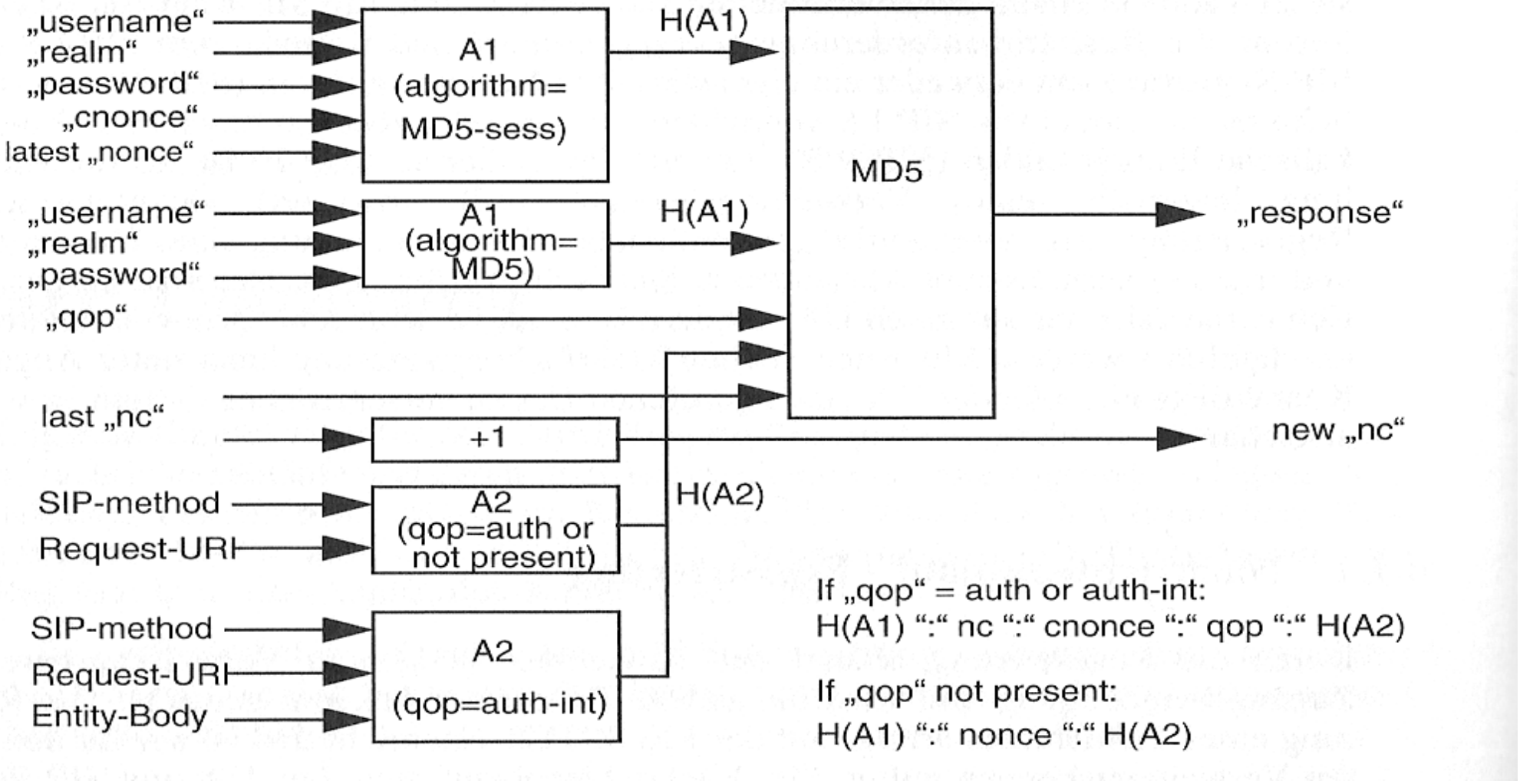
SIP – Codierung
→ Registrierung (2/10)

- Ein SIP-Proxy (SIP-P) trägt innerhalb eines Netzes die Verantwortung für die Rufbehandlung.
- Eine Registrierung muss in periodischen Abständen erneuert werden.
- Nach der Registrierung hat der Kommunikationsteilnehmer eine Zugangsberechtigung und ist anderen Kommunikationsteilnehmern bekannt.
- Die SIP-Nachricht beinhaltet die „REGISTER“ Anforderung und gilt nur für das Heimnetz.
- Echtheitsprüfungen können bei der Registrierung durchgeführt werden oder bei der Rufbehandlung vom SIP-P
- Authentizitätsprüfungen können via
  - Token-basiertes Verfahren
  - HTTP Digest
  - AKA-Verfahren erfolgen.

SIP – Codierung
→ Registrierung (3/10)

- Die Registrierung im Internet benutzt eine HTTP-Digest Authentizitätsprüfung
- Username und Passwort werden nicht im Klartext übertragen, sondern es wird zunächst vom SIP-Client eine Prüfsumme berechnet.
- Diese wird mithilfe des MD5-Algorithmus u.a. aus username, realm, passwort und dem nonce-Wert gebildet.
  - realm: Anmeldedomain (Heimnetz)
  - nonce: Zufallswert (vom SIP-P)
- Das Ergebnis wird mit dem Parameter „response“ als Berechtigungsnachweis dem SIP-P zur Verifikation übertragen
- Der SIP-P berechnet ebenfalls die Prüfsumme und vergleicht beide vorhandenen Prüfsummen.

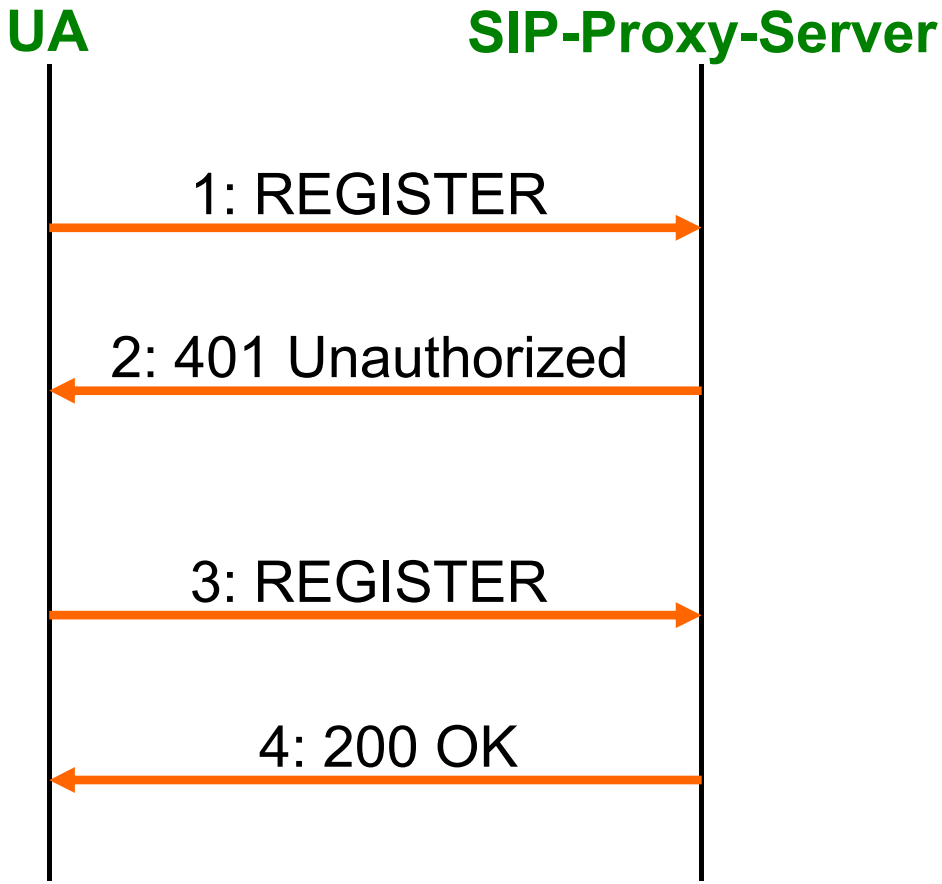
SIP – Codierung
→ Registrierung (4/10)



SIP – Codierung
→ Registrierung (5/10)

- Bei einer Registrierung muss noch unterschieden werden zwischen:
  - Erfolgreiche SIP-Registrierung
  - Aktualisierung der Kontaktliste
  - Anfrage der aktuell gültigen Kontaktliste
  - Beendigung der SIP-Registrierung
  - Nicht erfolgreiche SIP-Registrierung

SIP – Codierung
→ Registrierung (6/10): Übersicht des Ablaufes



SIP – Codierung
→ Registrierung (7/10): Register

UA ⟶ SIP-P

```
REGISTER sip:sipgate.de SIP/2.0
Via: SIP/2.0/UDP 172.16.50.37:5060;branch=z9hG4bK3f754b88;rport
From: <sip:123456@sipgate.de>;tag=as5f04045c
To: <sip:123456@sipgate.de>
Call-ID: 780eda9d59c919415d30536a052b32af@127.0.0.2
CSeq: 8928 REGISTER
Max-Forwards: 70
Contact: <sip:s@172.16.50.37>
Content-Length: 0
```

- Die Registrierung wird initiiert, indem die Nachricht an einen SIP-P gesendet wird.
- Steht in der Request-Line nach “REQUEST” anstelle von “sip“ das Schlüsselwort „sips“, werden die Nachrichten mithilfe von TLS über alle Netzelemente hinweg verschlüsselt und integritätsgesichert.
- Als Transportprotokoll kann sowohl UDP als auch TCP gewählt werden.

SIP – Codierung
→ Registrierung (8/10): 401 Unauthorized

UA ⟵ SIP-P

SIP/2.0 401 Unauthorized
Via: SIP/2.0/UDP
172.16.xxx.xxx:5062;branch=z9hG4bK0a6a30e0-cc6c-db11-9820-000d61122a90;received=172.16.xxx.xxx
From: <sip:1000@voip>;tag=c2442fe0-cc6c-db11-9820-000d61122a90
To: <sip:1000@voip>;tag=as721c61c7
Call-ID: 98ab2ee0-cc6c-db11-9820-000d61122a90@Homer
CSeq: 1 REGISTER
WWW-Authenticate: Digest algorithm=MD5, realm="asteriskPBX", nonce="25a12335"
Content-Length: 0

- Die erste Zeile zeigt dem Benuter 1000@voip mit **“401 Unauthorized”**, dass eine Authentizitätsprüfung für sein Heimatnetz erforderlich ist.
- Die Art der Prüfung wird in der Zeile „www-Authenticate:“ angegeben.
- Hier wird der MD5 Algorithmus verlangt.
- Zur Berechnung eines Wertes zur Prüfung der Authenzität liefert der Server die Parameter “realm” und “nonce”.

SIP – Codierung
→ Registrierung (9/10): REGISTER

UA → SIP-P

```
REGISTER sip:voip SIP/2.0
Via: SIP/2.0/UDP 172.16.xxx.xxx:5061;branch=z9hG4bK2acae54d-cf6c-
db11-934e-000d61122a90;
Max-Forwards: 70
From: <sip:1000@voip>;tag=5ab4e44d-cf6c-db11-934e-000d61122a90
To: <sip:1000@voip>
Call-ID: 081be44d-cf6c-db11-934e-000d61122a90@Homer
CSeq: 2 REGISTER
Contact: <sip:1000@172.16.xxx.xxx:5061;transport=udp>
Authorization: Digest username="1000", realm="asteriskPBX",
nonce="2e5e6a49", uri="sip:voip", algorithm=md5,
response="49e0ef3078b93b788b3598e853fe98f8"
Content-Length: 0
```

- Authentizitätsprüfung von user „1000“ durch Berechnung des Parameter „response“.
- “response” wird gebildet durch:
  username, passwort, realm, nonce HTTP-Methode und Request-uri.
- Parameter „response“ ist eine 128-bit Zeichenkette einer MD5-Prüfsumme.
- “nonce” ist eine willkürlich gewählte Zufallszahl
- “CSeq” muss inkrementiert werden

SIP – Codierung
→ Registrierung (10/10): 200 OK

UA ← SIP-P

SIP/2.0 200 OK
Via: SIP/2.0/UDP 172.16.xxx.xxx:5061;branch=z9hG4bK2acae54d-cf6c-db11-934e-000d61122a90;received=172.16.xxx.xxx
From: <sip:1000@voip>;tag=5ab4e44d-cf6c-db11-934e-000d61122a90
To: <sip:1000@voip>;tag=as179d4a64
Call-ID: 081be44d-cf6c-db11-934e-000d61122a90@Homer
CSeq: 2 REGISTER
Contact: <sip:1000@172.16.xxx.xxx:5061;transport=udp>;expires=3600
Content-Length: 0

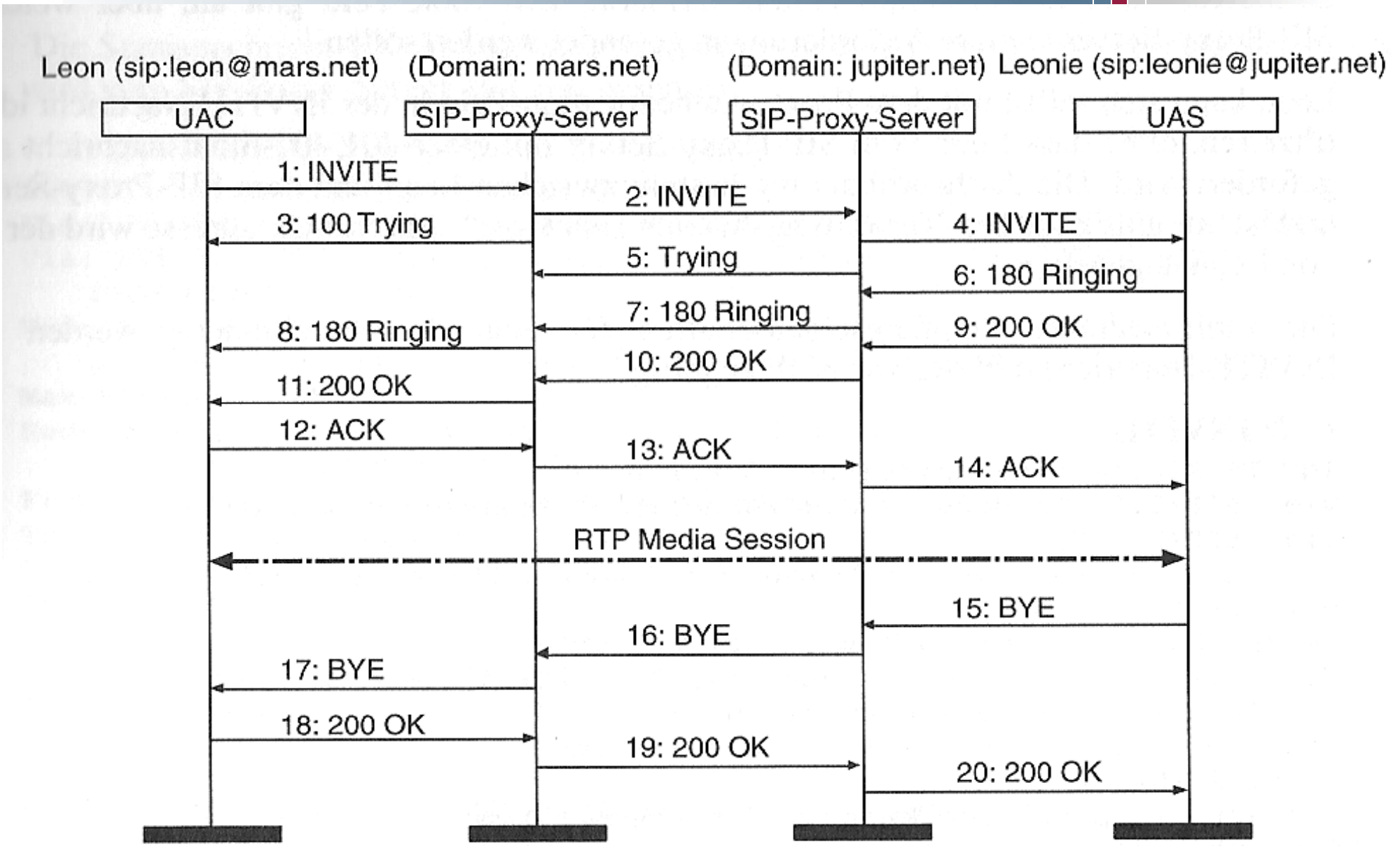
- Mit **"200 OK"** wird sowohl die Registrierungsanforderung als auch die erfolgreiche Authentizitätsprüfung bestätigt.
- **„expires"** zeigt die Gültigkeitsdauer von 3.600 s (1 h) an und kann/muss innerhalb dieser Zeit aufgefrischt werden.
- Eine Gültigkeitsdauer von 0 s entspricht einer Deregistrierung

SIP – Codierung
→ Normalruf (1/12)

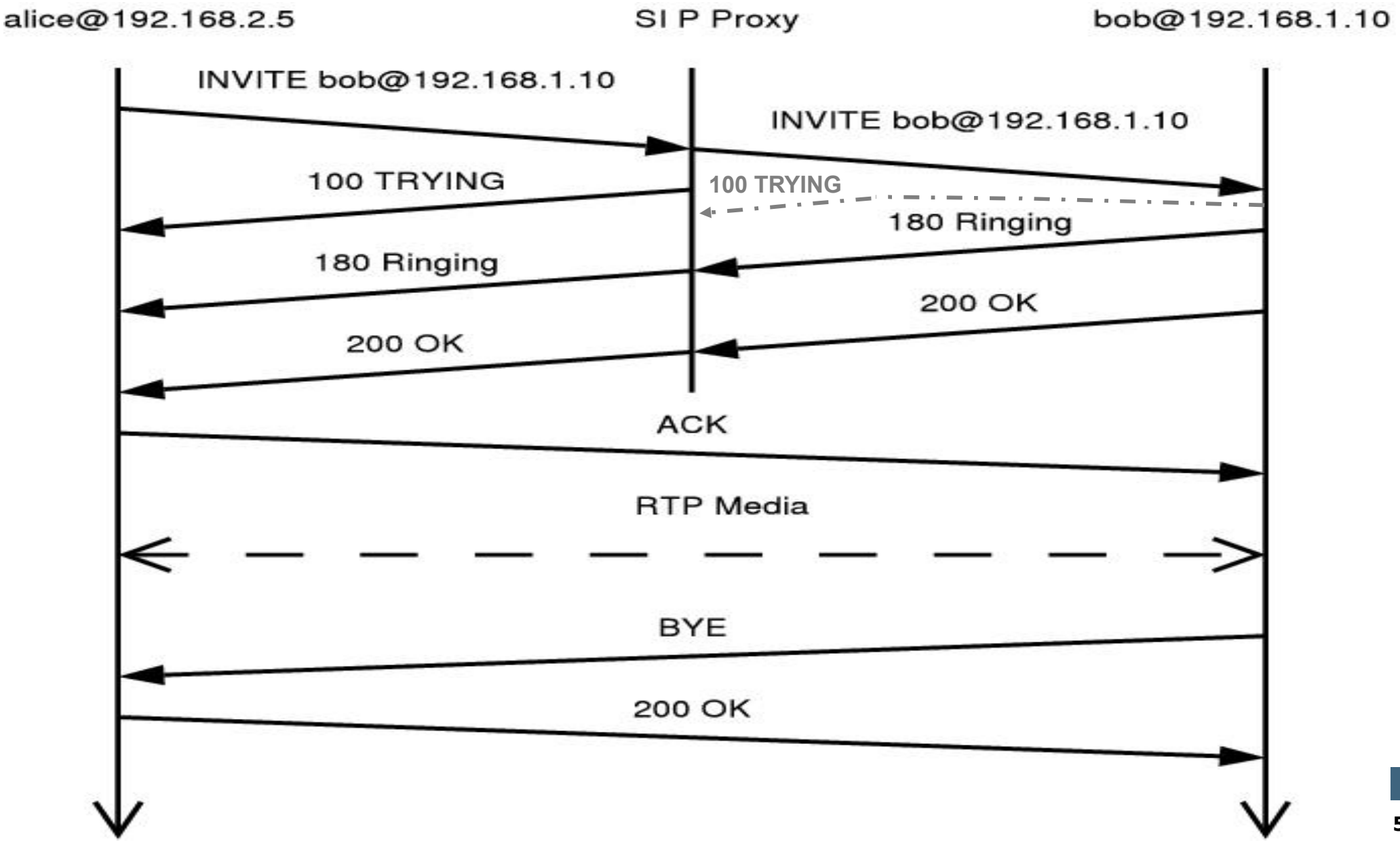
- Die folgende Grafik zeigt einen Normalruf
- Der Aufbau erfolgt zwischen Leon und Leonie, die jeweils einem eigenen Verwaltungsbereich zugehören.
- Eine erfolgreiche Registrierung ist hier vorausgesetzt.

- ***Abarbeitung des Rufs:***
  - **1 – 14** Verbindungsaufbau
  - **1, 2, 4** Übermittlung von Mediainformationen mit SDP (offer)
  - **3, 5, 6, 7, 8** Übermittlung von Statusinformationen
  - **9, 10, 11** Übermittlung von Mediainformationen mit SDP (answer)
  - zwischen **14 – 15** RTP Media Session
  - **15 – 20** Abbau der Verbindung

SIP – Codierung
→ Normalruf (2/12) – With record routing



SIP – Codierung
→ Normalruf (3/12) - Without record routing



SIP – Codierung
→ Normalruf (4/12)

- Als erste Nachricht wird eine **INVITE-Nachricht** mit der Zieladresse von “Leonie” verschickt
- In der **Startzeile** befindet sich die Anforderung (request) „INVITE“, die **Zieladresse und die Protokollversion**.
- Im „route-Feld“ steht der SIP-Proxy-Server, über den weitere Anforderungen gesendet werden sollen.
- Die Authentifizierung zwischen Leon und dem SIP-Proxy-Server ist nur für den Verwaltungsbereich „mars.net“ gültig
- Die multimedialen Eigenschaften werden mithilfe von SDP als Nutzlast an die SIP-Nachricht angehängt

SIP – Codierung
→ Normalruf (5/12)

- **1: INVITE**

```
INVITE sip:leonie@jupiter.net SIP/2.0
Via: SIP/2.0/TCP pc1-leon.mars.net:5060;branch=z9hG4bK74bf9
Max-Forwards: 70
Route: <sip:node1-sps.mars.net;lr>
From: leon <sip:leon@mars.net>;tag=9fxced76sl
To: leonie <sip:leonie@jupiter.net>
Call-ID: 3848276298220188511@leon.mars.net
CSeq: 2 INVITE
Contact: <sip:pc1-leon@mars.net;transport=tcp>
Proxy-Authorization: Digest username="leon",realm="leon.mars.net",
    nonce="wf84f1ceczx41ae6cbe5aea9c8e88d359",opaque="",
    uri="sip:leonie@jupiter.net",response="42ce3cef44b22f50c6a6071bc8"
Content-Type: application/sdp
Content-Length: 151

v=0
o=leon 2890844526 2890844526 IN IP4 pc1-leon.mars.net
s=-
c=IN IP4 197.0.24.1
t=0 0
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

SIP-Signaling

SDP-Signaling

SIP – Codierung
→ Normalruf (6/12)

- Die „INVITE-Nachricht“ wird von Proxy 1 nach Proxy 2 gesendet.
- Proxy 1 muss eine DNS Anfrage starten, da die Zieladresse sich in einem anderen Verwaltungsbereich befindet.
- Proxy 1 fügt in der “INVITE-Nachricht“ eine weitere via Zeile mit seinem eigenen Stationsnamen ein
- Zur besseren Rückverfolgung von folgenden SIP-Nachrichten innerhalb eines Session wird die Empfangsadresse der vorherigen „INVITE-Nachricht“ angehängt
- Als Antwort auf die “INVITE-Nachricht“ wird die Statusnachricht **„100 Trying“** an Leon gesendet.
- der „branch“ Parameter sorgt dafür, dass von den Proxy Servern Ringschlüsse erkannt werden.

SIP – Codierung
→ Normalruf (7/12)

- **2: INVITE**

```
INVITE sip:leonie@jupiter.net SIP/2.0
Via: SIP/2.0/TCP node1-sps.mars.net:5060;branch=z9hG4bK2d4790.1
Via: SIP/2.0/TCP
    pc1-leon.mars.net:5060;branch=z9hG4bK74bf9;received=197.0.24.1
Max-Forwards: 69
Record-Route: <sip:node1-sps.mars.net;lr>
From: leon <sip:leon@mars.net>;tag=9fxced76sl
To: leonie <sip:leonie@jupiter.net>
Call-ID: 384827629822018851l@leon.mars.net
CSeq: 2 INVITE
Contact: <sip:pc1-leon@mars.net;transport=tcp>
Content-Type: application/sdp
Content-Length: 143

v=0
o=leon 2890844526 2890844526 IN IP4 pc1-leon.mars.net
s=-
c=IN IP4 197.0.24.1
t=0 0
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

IP-Adresse, die für RTP verwendet werden soll.

UDP-Port-Nr., die für RTP verwendet werden soll.

SIP – Codierung
→ Normalruf (8/12)

- **3: 100 Trying**

```
SIP/2.0 100 Trying
Via: SIP/2.0/TCP
    pc1-leon.mars.net:5060;branch=z9hG4bK74bf9;received=197.0.24.1
From: leon <sip:leon@mars.net>;tag=9fxced76sl
To: leonie <sip:leonie@jupiter.net>
Call-ID: 384827629822018851l@leon.mars.net
CSeq: 2 INVITE
Content-Length: 0
```

- **„180 Ringing“** unterscheidet sich von der **„INVITE-Nachricht“** nur dadurch, dass die Statusmeldung keine SDP-Parameter enthält
- **„180 Ringing“** ist die Bestätigung des Rufes!
  (es schellt beim Empfänger)
- **„200 OK“** ist die finale Statusmeldung und signalisiert die **Verbindungsannahme**!
- Danach ist die SIP-Session aufgebaut und der UA-Client kann einen RTP-Transfer zur angegebenen IP-Adresse/UDP-Port durchführen.

SIP – Codierung
→ Normalruf (9/12)

- **9: 200 OK**

```
SIP/2.0 200 OK
Via: SIP/2.0/TCP node1-sps.jupiter.net:5060;branch=z9hG4bK721e418c4.1;
    received=198.0.16.4
Via: SIP/2.0/TCP
    node1-sps.mars.net:5060;branch=z9hG4bK2d4790.1;received=197.0.155.1
Via: SIP/2.0/TCP
    pc1-leon.mars.net:5060;branch=z9hG4bK74bf9;received=197.0.24.1
Record-Route: <sip:node1-sps.jupiter.net;lr>,
    <sip:node1-sps.mars.net;lr>
From: leon <sip:leon@mars.net>;tag=9fxced76s1
To: leonie <sip:leonie@jupiter.net>;tag=314159
Call-ID: 3848276298220188511@leon.mars.net
CSeq: 2 INVITE
Contact: <sip:pc1-leonie@jupiter.net;transport=tcp>
Content-Type: application/sdp
Content-Length: 147

v=0
o=leonie 2890844527 2890844527 IN IP4 pc1-leonie.jupiter.net
s=-
c=IN IP4 198.0.20.1
t=0 0
m=audio 3456 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

IP-Adresse, die für RTP verwendet werden soll.

UDP-Port-Nr., die für RTP verwendet werden soll.

SIP – Codierung
→ Normalruf (10/12)

- Die **Anforderung „ACK“** wird an den Proxy gesendet und beendet den Verbindungsaufbau.
- Die zeitkritischen Datenströme werden ab jetzt mithilfe von RTP zwischen den beiden Kommunikationsteilnehmern ausgetauscht!

- Beendet einer das Gespräch, wird eine **„BYE“-Nachricht** gesendet.
- „BYE“ wird mit „200 OK“ bestätigt, danach ist der Verbindungsabbau beendet.

SIP – Codierung
→ Normalruf (11/12)

- **13: ACK**

```
ACK sip:pc1-leonie@jupiter.net SIP/2.0
Via: SIP/2.0/TCP node1-sps.mars.net:5060;branch=z9hG4bK2d4790.1
Via: SIP/2.0/TCP
    pc1-leon.mars.net:5060;branch=z9hG4bK74b76;received=197.0.24.1
Max-Forwards: 69
Route: <sip:node1-sps.jupiter.net;lr>
From: leon <sip:leon@mars.net>;tag=9fxced76s1
To: leonie <sip:leonie@jupiter.net>;tag=314159
Call-ID: 3848276298220188511@leon.mars.net
CSeq: 2 ACK
Content-Length: 0
```

- **16: BYE**

```
BYE sip:pc1-leon@mars.net SIP/2.0
Via: SIP/2.0/TCP node1-sps.jupiter.net:5060;branch=z9hG4bK721e418c4.1
Via: SIP/2.0/TCP
    leonie.jupiter.net:5060;branch=z9hG4bKnashds7;received=198.0.20.1
Max-Forwards: 69
Route: <sip:node1-sps.mars.net;lr>
From: leonie <sip:leonie@jupiter.net>;tag=314159
To: leon <sip:leon@mars.net>;tag=9fxced76s1
Call-ID: 3848276298220188511@leon.mars.net
CSeq: 1 BYE
Content-Length: 0
```

SIP – Codierung
→ Normalruf (12/12)

- **18: 200 OK**

```
SIP/2.0 200 OK
Via: SIP/2.0/TCP
    node1-sps.mars.net:5060;branch=z9hG4bK2d4790.1;received=197.0.155.1
Via: SIP/2.0/TCP node1-sps.jupiter.net:5060;branch=z9hG4bK721e418c4.1;
    received=198.0.16.4
Via: SIP/2.0/TCP
    leonie.jupiter.net:5060;branch=z9hG4bKnashds7;received=198.0.20.1
From: leonie <sip:leonie@jupiter.net>;tag=314159
To: leon <sip:leon@mars.net>;tag=9fxced76sl
Call-ID: 3848276298220188511@leon.mars.net
CSeq: 1 BYE
Content-Length: 0
```

- Der Ablauf beschreibt nur die wichtigsten Nachrichten bei einem Anruf.
- Das Prinzip bleibt immer gleich, nur die Anzahl der zu sendenden Nachrichten erhöht sich und die Adresszeilen ändern sich.

Inhalt

- Ziele und Einordnung
- Telefon und Datennetze
- SIP – Session Initiation Protocol
  - SIP – Nachrichten
  - SIP – Codierung
  - **SIP – basierte Anwendungen**
- RTP – Real-Time Transport Protocol
- VoIP
  - Codierung
  - Zusammenhänge

SIP – basierte Anwendungen

	Anwendungen			
	mediale		multimediale	
	E2E	SIP-AS	E2E	SIP-AS
zeitkritische (real-time)	Streaming Service			
	IP-Telefonie (VoIP)		IP-Videotelefonie (Videophone)	IP-Videotelefo-nie-Konferenzen
	IP-Texttelefonie			
	Audiokonferenz (Focus)	Audiokonferenz		
	IP-Video	Push-to-Talk (PTT)		IP-Karaoke
	IP-Radio (Webcaster)			IP-Television (IP-TV)
nicht-zeitkritische (non real-time)	Instant Messaging (IM)			
	Presence			
	IP-Text (ToIP)	E-mail		E-mail

- Ausschnitt aus möglichen SIP-basierten Anwendungen
- Aufteilung unter Berücksichtigung ihrer Eigenschaften.

Inhalt

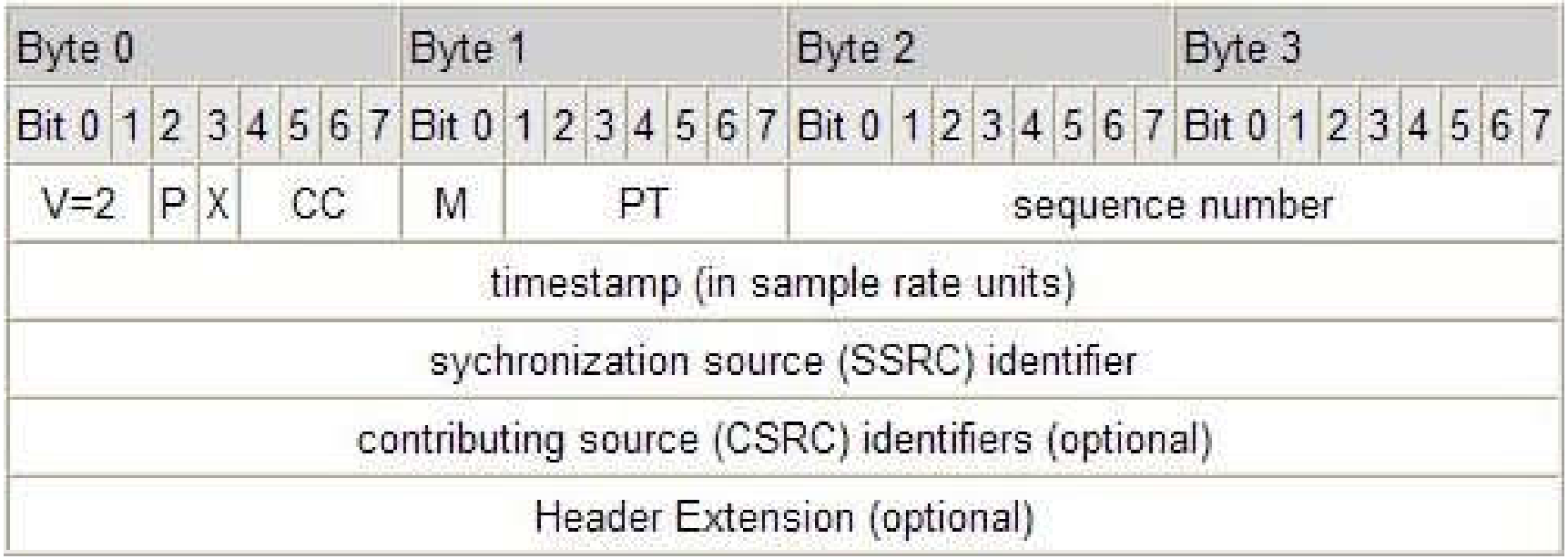
- Ziele und Einordnung
- Telefon und Datennetze
- SIP – Session Initiation Protocol
  - SIP – Nachrichten
  - SIP – Codierung
  - SIP – basierte Anwendungen
- **RTP – Real-Time Transport Protocol**
- VoIP
  - Codierung
  - Zusammenhänge

RTP – Real-time Transport Protocol

- RFC 3550 seit 2003
- RTP ist ein Protokoll zur kontinuierlichen Übertragung von Mediadaten.
- Es dient dazu, Multimedia-Datenströme über Netzwerke zu transportieren, d.h. Daten kodieren, paketieren und versenden.
- Es ist ein paket-basiertes Protokoll und stützt sich normalerweise auf UDP, kann aber auch TCP benutzen.

- **Realtime-time Control Protocol (RTCP) arbeitet mit RTP zusammen und dient dazu, QoS Parameter auszuhandeln und einzuhalten.**

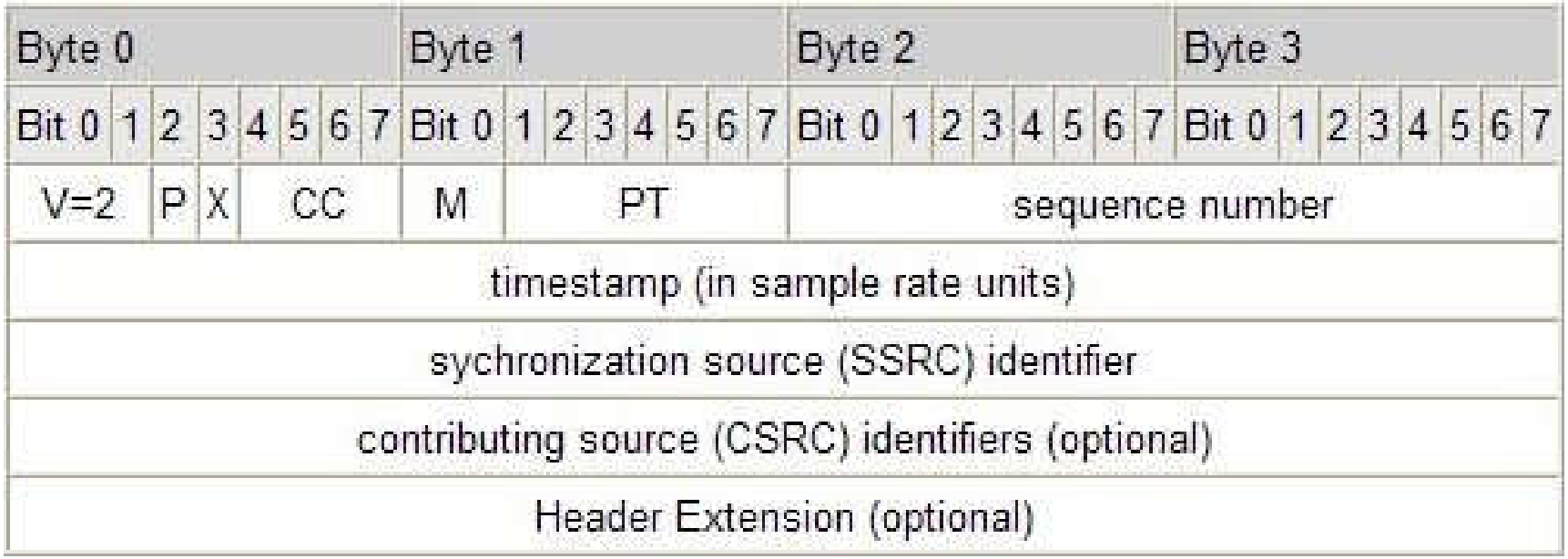
RTP – Real-time Transport Protocol
→ Header (1/12)



Byte 0								Byte 1								Byte 2								Byte 3							
Bit 0	1	2	3	4	5	6	7	Bit 0	1	2	3	4	5	6	7	Bit 0	1	2	3	4	5	6	7	Bit 0	1	2	3	4	5	6	7
V=2		P	X	CC				M	PT							sequence number															
timestamp (in sample rate units)																															
sychronization source (SSRC) identifier																															
contributing source (CSRC) identifiers (optional)																															
Header Extension (optional)																															

- **Version (V), 2 Bit**
  - Versionsstand des RTP-Protokolls
- **Padding (P), 1 Bit**
  - Das Füll-Bit ist gesetzt, wenn ein oder mehrere Füll-Oktets am Ende des Pakets angehängt sind, die nicht zum eigentlichen Dateninhalt (Payload) gehören.
  - Das letzte Füll-Oktet gibt die Anzahl der hinzugefügten Füll-Oktets an.
  - Füll-Oktets werden nur dann benötigt, wenn nachfolgende Protokolle eine vorgegebene Blockgröße benötigen, z.B. Verschlüsselungsalgorithmen.

RTP – Real-time Transport Protocol
→ Header (2/12)



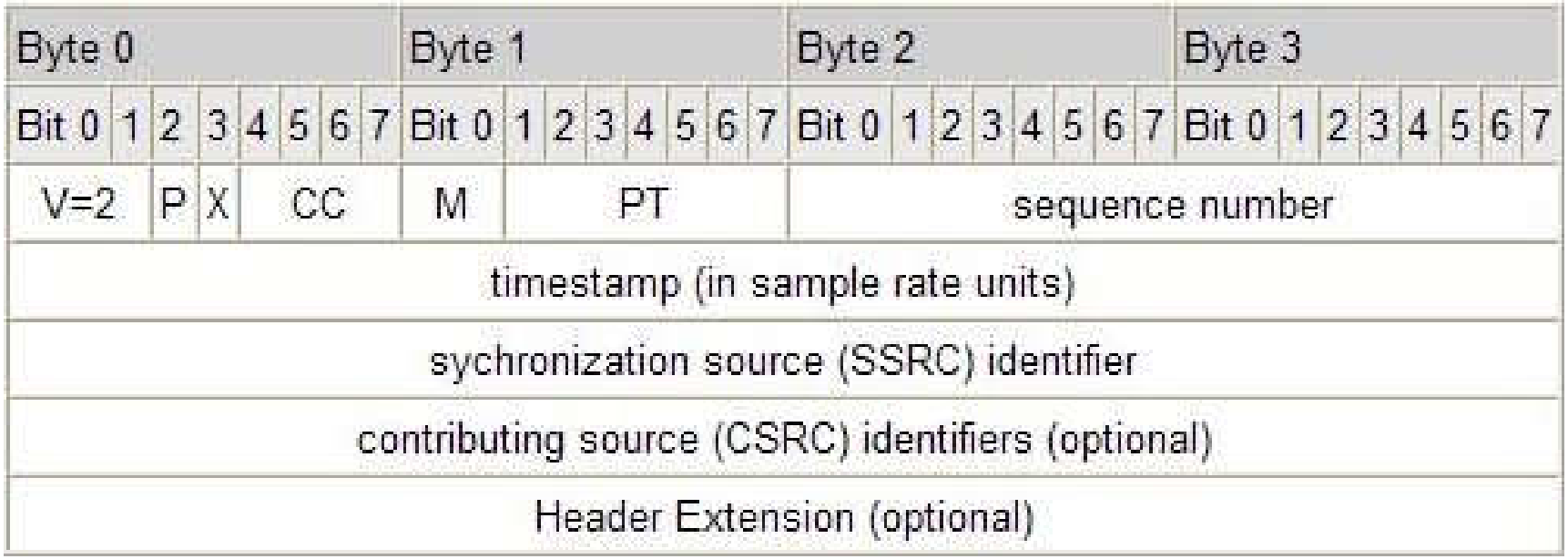
- **Extension (X), 1 Bit**
  - Das Erweiterungs-Bit ist gesetzt, wenn der Header um genau einen Erweiterungs-Header ergänzt wird.
- **CRSC Count (CC), 4 Bit**
  - Der CSRC-Zähler gibt die Anzahl der CSRC-Identifier an.
- **Marker (M), 1 Bit**
  - Das Marker-Bit ist für anwendungsspezifische Verwendungen reserviert.
- **Payload Type (PT), 7 Bit**
  - Dieses Feld beschreibt das Format des zu transportierenden RTP-Inhalts.

RTP – Real-time Transport Protocol
→ Header (3/12)

Byte 0								Byte 1								Byte 2								Byte 3							
Bit 0	1	2	3	4	5	6	7	Bit 0	1	2	3	4	5	6	7	Bit 0	1	2	3	4	5	6	7	Bit 0	1	2	3	4	5	6	7
V=2		P	X	CC				M	PT							sequence number															
timestamp (in sample rate units)																															
sychronization source (SSRC) identifier																															
contributing source (CSRC) identifiers (optional)																															
Header Extension (optional)																															

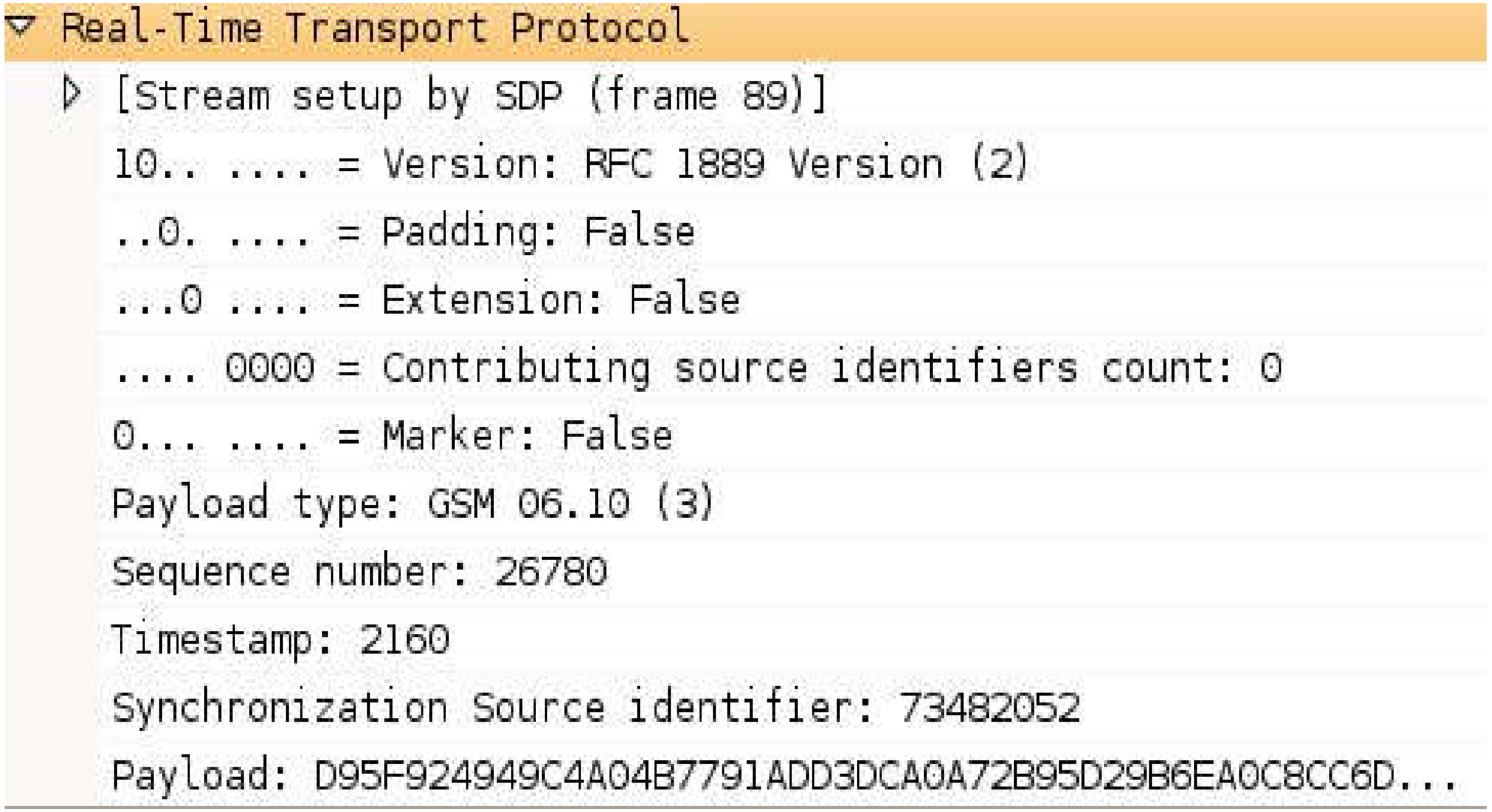
- **Sequence Number, 16 Bit**
  - Die Sequenznummer wird für jedes weitere RTP-Datenpaket erhöht.
  - Die Startnummer wird zufällig ausgewählt und ist nicht vorherbestimmbar.
  - Der Empfänger kann mit Hilfe der Sequenznummer die Paketreihenfolge wiederherstellen und den Verlust von Paketen erkennen.
- **Timestamp, 32 Bit**
  - Der Zeitstempel gibt den Zeitpunkt des ersten Oktets des RTP-Datenpakets an.
  - Der Zeitpunkt muss sich an einem Takt orientieren, der nicht kontinuierlich und linear, sondern als diskreter Wert zur Verfügung steht, damit die Synchronität des Streams sichergestellt und die Laufzeitunterschiede der Übertragungsstrecke (Jitter) ermittelt werden können.

RTP – Real-time Transport Protocol
→ Header (4/12)



- **SSRC, 32 Bit**
  - Dieses Feld dient zur Identifikation der Synchronisationsquelle.
  - Der Wert wird zufällig ermittelt, damit nicht zwei Quellen innerhalb der RTP-Session die gleiche Identifikationsnummer besitzen.
- **CSRC List, 0 bis 15 Felder je 32 Bit** (Optional)
  - Die CSRC-Liste dient zur Identifikation der Quellen, die im RTP-Payload enthalten sind.
  - Die Anzahl der Listenfelder wird im CC-Feld angegeben.
  - Falls mehr als 15 Quellen vorkommen, werden nur 15 identifiziert.
  - Die Liste wird von Mixern eingefügt, die dazu den Inhalt des SSRC-Feldes der beteiligten Quellen einsetzen.

RTP – Real-time Transport Protocol



RTCP – Real-time Control Protocol

- Das Real-time Control Protocol (RTCP) sendet periodisch Statusinformationen der Kommunikationsteilnehmer.
- RTCP sorgt für die Kontrolle der Signallaufzeiten und misst die Paketverlustraten.
- Die Übertragung kann so an die Verbindung angepasst werden.
- Die Übertragung dieser Nachrichten ist nicht gesichert.

- **Vier Grundfunktionen von RTCP:**
  - Rückmeldung über die Qualität der Datenverteilung
  - Beinhaltet Transport Level Identifizierung zur Synchronisation bei Konflikten
  - Datenrate in Abhängigkeiten der Kommunikationsteilnehmer kontrollieren
  - Beförderung von Sitzungssteuerungsinformationen, z.B. Kommunikationsteilnehmeridentifizierung

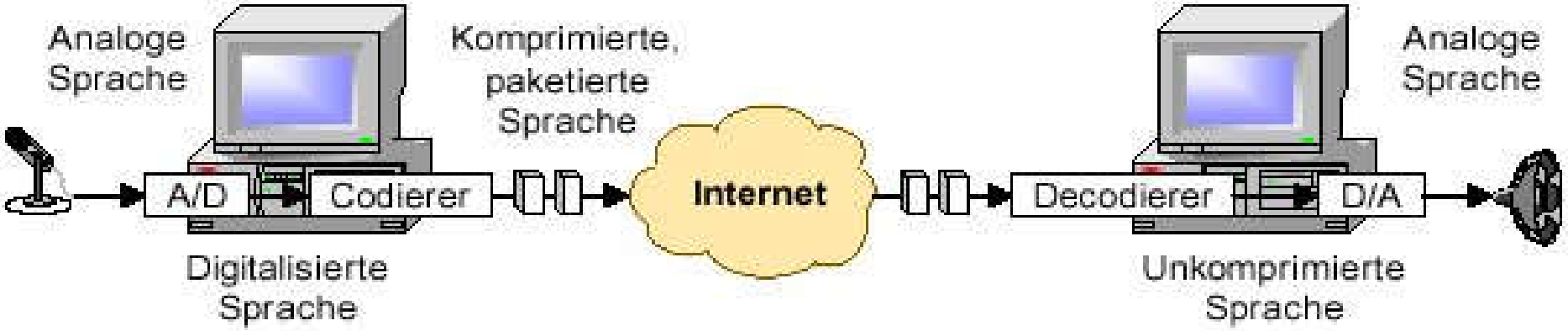
RTCP – Real-time Control Protocol

```
▽ Real-time Transport Control Protocol (Source description)
  ▷ [Stream setup by SDP (frame 57)]
    10.. .... = Version: RFC 1889 Version (2)
    ..0. .... = Padding: False
    ...0 0001 = Source count: 1
    Packet type: Source description (202)
    Length: 9
  ▽ Chunk 1, SSRC/CSRC 69259521
      Identifier: 69259521
    ▽ SDES items
        Type: CNAME (user and domain) (1)
        Length: 16
        Text: firstclaas@Homer
        Type: TOOL (name/version of source app) (6)
        Length: 5
        Text: ekiga
        Type: END (0)
  [RTCP frame length check: OK - 92 bytes]
```

Inhalt

- Ziele und Einordnung
- Telefon und Datennetze
- SIP – Session Initiation Protocol
  - SIP – Nachrichten
  - SIP – Codierung
  - SIP – basierte Anwendungen
- RTP – Real-Time Transport Protocol
- **VoIP**
  - **Codierung**
  - **Zusammenhänge**

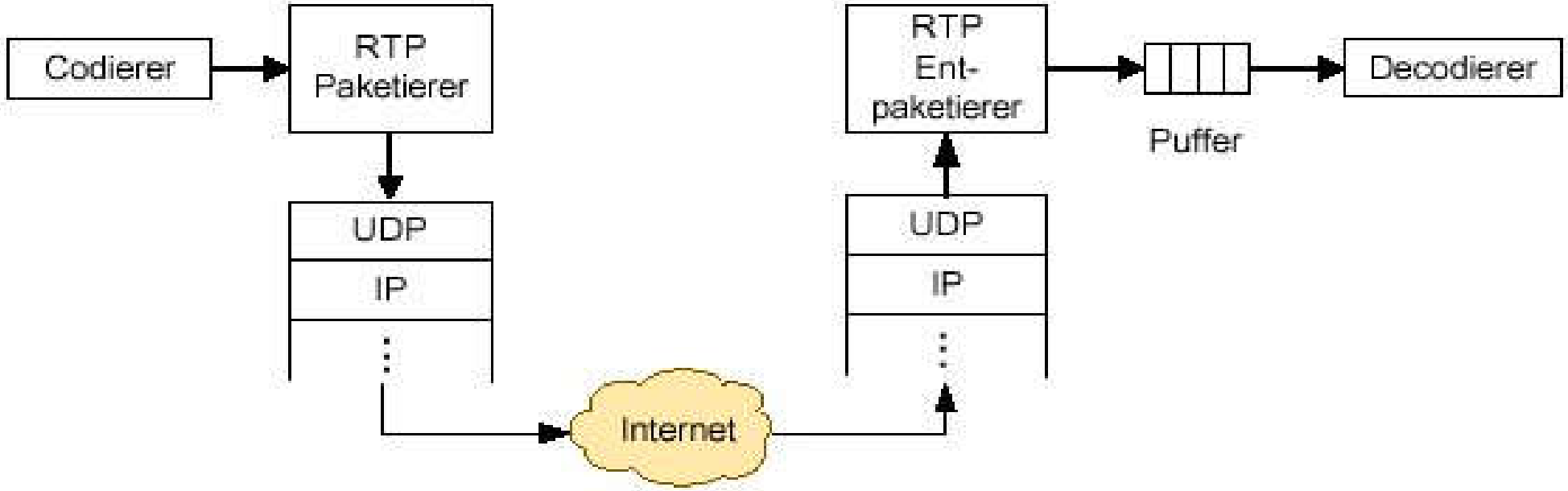
Voice over IP
→ Prinzip der IP-Telefonie



Ablauf:

- Konvertierung von analog zu digital
- Codierung, Kompression

Voice over IP
→ RTP als Paketierungseinheit



Dienstgüte (QoS)
→ Übersicht

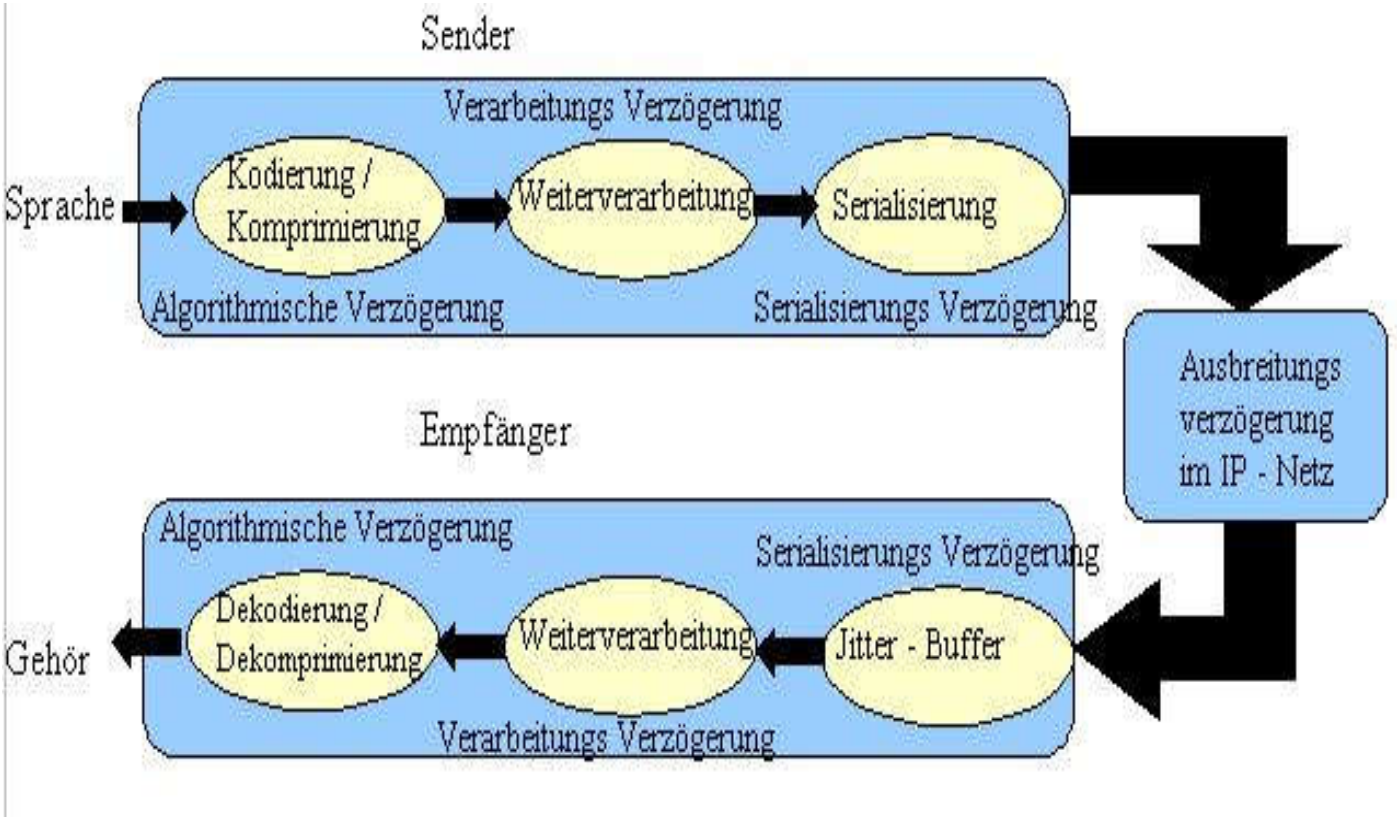
- **IP-Telefonie (VoIP) ist eine zeitkritische Realzeitanwendung.**
- Bei VoIP werden kontinuierliche Medieninformationen übertragen, bei denen sich die Werte über die Zeit verändern und **nur zu einem bestimmten Zeitpunkt Gültigkeit haben.**
- **Aus diesem Grund müssen Dienstgüteparameter eingehalten werden, damit das Sprach-Paket genutzt werden kann.**
- Sprach-Pakete, die nicht rechtzeitig kommen, werden verworfen.

Dienstgüte (QoS)
→ Wichtigste Parameter

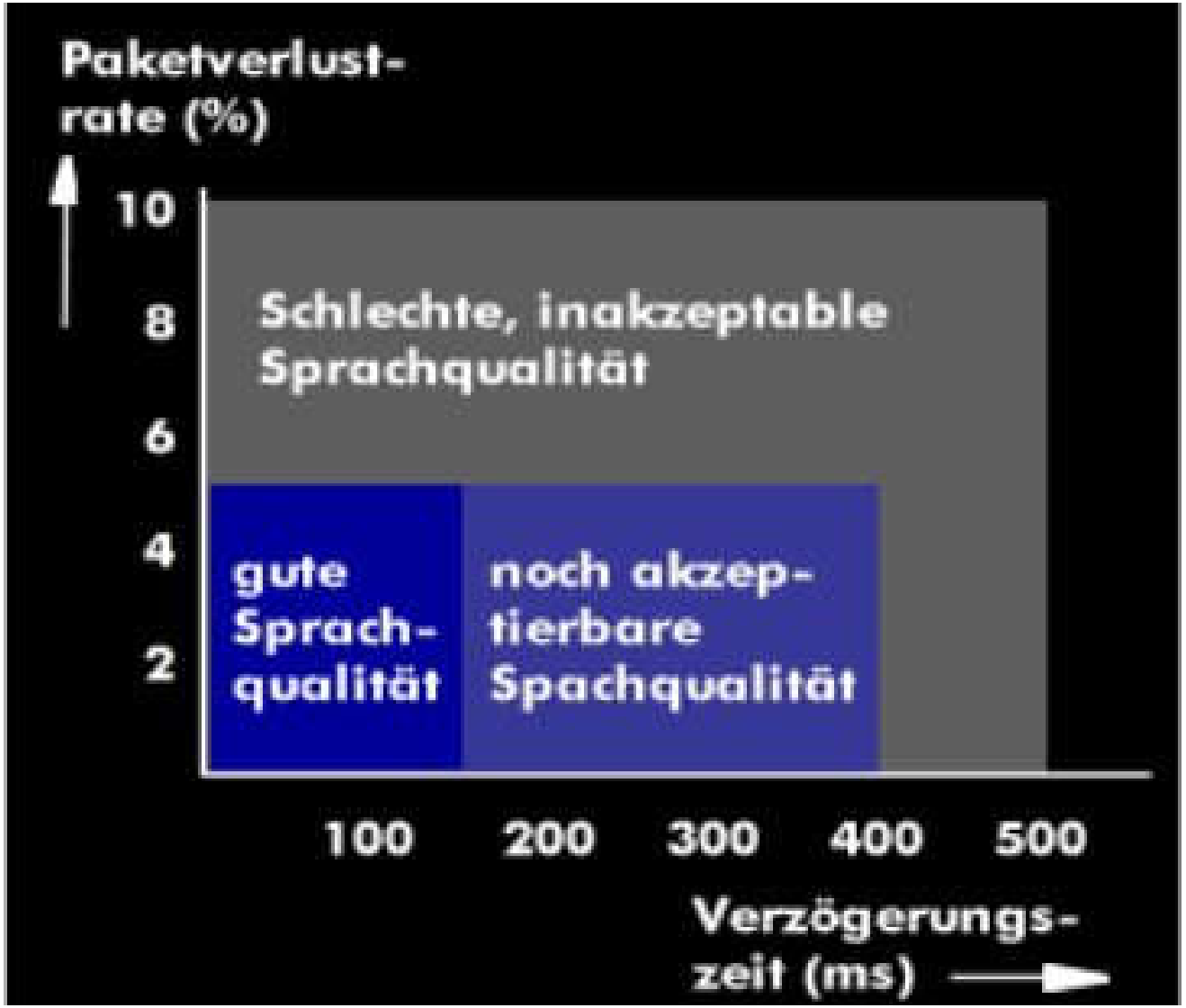
- **Paketverlustrate** (Packet Loss) in %
- **Verzögerung** (Delay) in ms
- **Schwankungen der Verzögerung** (Jitter) in ms
- **Bandbreite** in KBit/s

Dienstgüte (QoS)
→ Vier Klassen der Verzögerung

- Algorithmische Verzögerung
- Verarbeitungsverzögerung
- Serialisierungsverzögerung
- End-to-End Übertragungsverzögerung (Ausbreitungsverzögerung in Netz)



Dienstgüte (QoS)
→ Übersicht



Dienstgüte (QoS)
→ Bandbreite

- **Berechnung der Paketgröße**
  - Ethernet-Header + IP-Header + UDP-Header + RTP-Header
  - 14 Bytes + 20 Bytes + 8 Bytes + 12 Bytes = 54 Bytes

    Summe der Header = 54 Byte

  - **Payload: G. 711 (64 KBit/s) = 160 Bytes**

  - **Größe eines Sprachpaketes**
    54 Bytes Header + 160 Bytes Payload = **214 Bytes**

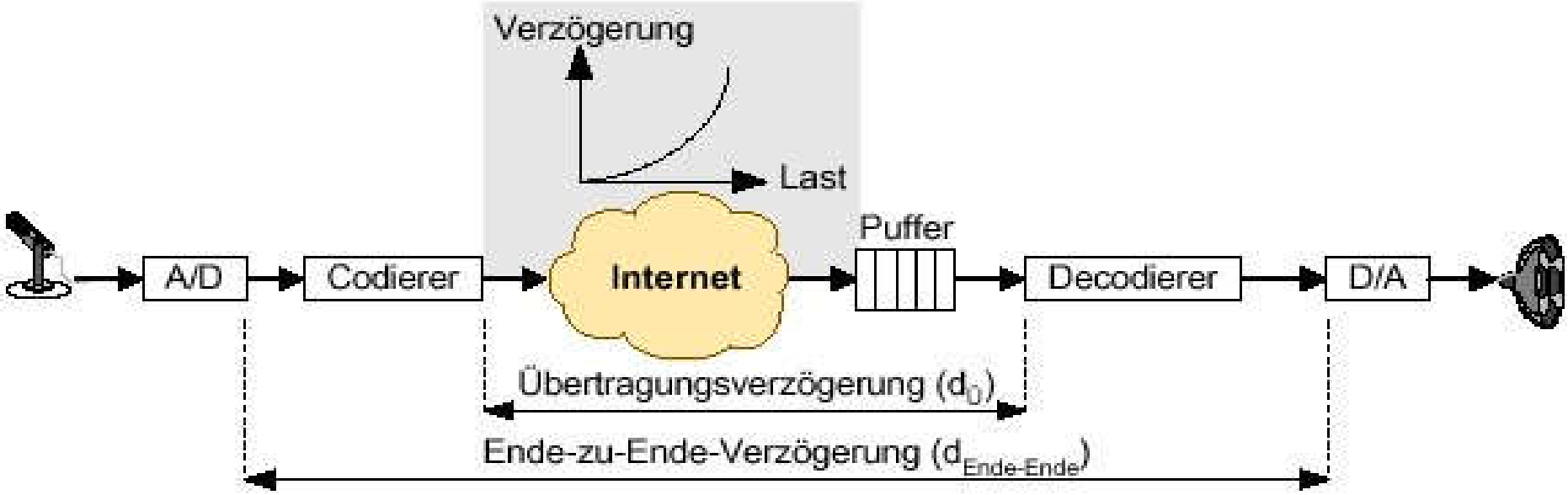
  - 214 Byte -> 85,6 Kbit/s
  - 85,6 KBit/s + 5% (RTCP) = **89,9 KBit/s notwendige Bandbreite**

Dienstgüte (QoS)
→ Verzögerung (1/2)

Bandbreite bei G. 723.1: 8 KBit/s

Paketverlustrate: ≤ 1%

Ende-zu-Ende-Verzögerung: < 300 ms



$d_{Ende-Ende} = d_{Codes} + d_Ü$ (≤ 300 ms für eine akzeptable Sprachqualität)

d_{Codes}(G.723.1) = 67.5 ms

$d_Ü = d_{Netz} + j_{Netz}$ (hängt von verwendetem Netz ab; ≤ 182,5 ms)

Dienstgüte (QoS)
→ Verzögerung (2/2)

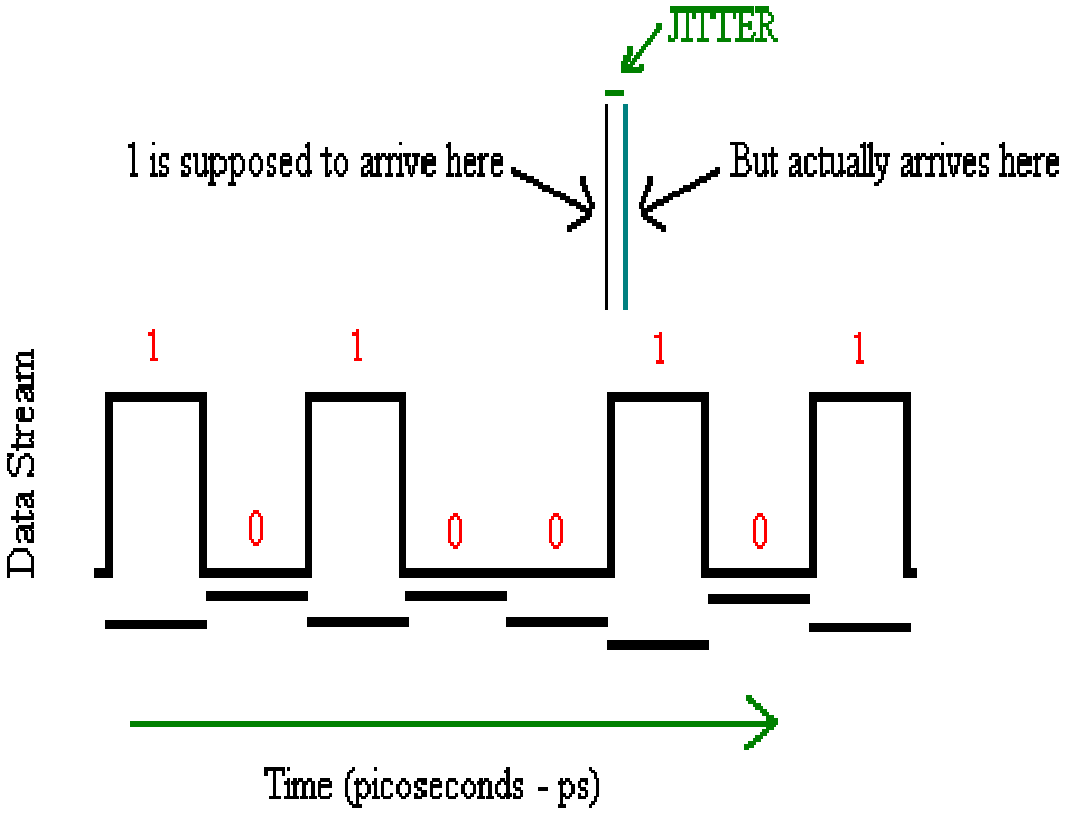
- Signallaufzeit: **5 ms / 1000 Km**
- Verzögerung in einem Router: **0.1 bis 2 ms**
- Kommunikationsverzögerung: **13,4 ms**
  (214 Byte (RTP), 128 KBit/s (DSL))

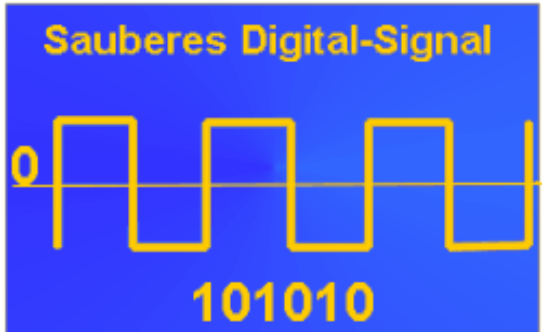
  Annahme: 500 Km, 13 Router, 128 KBit/s: **45 ms**
  (Signallaufzeit 5 ms + 13*2 ms + 14 ms)

- **RTT: 100 ms +- 100 ms**

Dienstgüte (QoS)
→ Jitter

- Verzögerung zwischen 250 ms und 300 ms im Grenzbereich
- Lösung: Jitter - Buffer





Codec Übersicht

Codec	Name/Bezeichnung	Übertragungsrate	MOS	MIPS	Delay	Audiofrequenz	Sprachqualität
G.711	Pulse Code Modulation (PCM)	56 oder 64 kbit/s (80 kBit/s mit Header)	4,4	1	0,25 ms	300 bis 3400 Hz	ISDN
G.726	Adaptive Differential Pulse Code Modulation (ADPCM)	16-40 kbit/s	4,2	-	-	-	Mobilfunk
G.728	Low Delay Code Excited Linear Prediction (LD-CELP)	16 kbit/s	4,2	30	1,25 ms	300 bis 3400 Hz	ungefähr ISDN
G.729/ G.729A	Conjugate Structure Algebraic Code Excited Linear Prediction (CS-ACELP)	8 kbit/s	4,2	20/11	25 ms	300 bis 3400 Hz	besser als G.723.1
G.723.1	Multiple Maximum Likelihood Quantization (MPMLQ)	6,3 kbit/s	3,9	18	67,5 ms	300 bis 3400 Hz	Gut
G.723	Algebraic Code Excited Linear Prediction (ACELP)	5,3 kbit/s	3,5	-	-	-	-

- Hier eine Übersicht der gängigsten Codecs.
- Es gibt andere, aber diese hier bilden einen guten Kompromiss zwischen erzeugtem Payload und Qualität.

MOS

- Der Begriff Mean Opinion Score (MOS) bezeichnet in der Telekommunikation ein Verfahren zur subjektiven Beurteilung der Qualität von Sprach- und Bildübertragungen.
- Der **Mean Opinion Score** ist das Ergebnis eines festgelegten Ablaufs mehrerer Tests, bei dem die empfundene Qualität der Sprache beziehungsweise der Bilder durch eine Gruppe von Versuchspersonen beurteilt wird.
- Das Ergebnis der Testreihe wird in eine fünfstufige Qualitätsskala eingeordnet.

MOS

MOS-Qualitätsskala

Wert	quality	Qualität	Bedeutung
5	excellent	ausgezeichnet	Es ist keine Anstrengung nötig, um die Sprache zu verstehen.
4	good	gut	Durch aufmerksames Hören kann die Sprache ohne Anstrengung wahrgenommen werden.
3	fair	ordentlich	Die Sprache kann mit leichter Anstrengung wahrgenommen werden.
2	poor	mäßig	Es bedarf großer Konzentration und Anstrengung, um die übermittelte Sprache zu verstehen.
1	bad	mangelhaft	Trotz großer Anstrengung kann man sich nicht verständigen.

Probleme und Lösungen

Fehlerbild \ Ursache	Delay	Jitter	Packet Loss	Echo Compensation	Voice Activity Detection	DTMF Detection	Loudness (Lautstärke)
Knacken		X	X				
Rauschen	X			X			X
Silbenverlust	X	X	X		X		
Echo, Hall	X			X		X	X
Verzerrungen („Mickey Mouse“ Sprache)	X			X			X

Die Art von Problemen treten merklich nur in IP-basierten Netzen auf

Angriffsübersicht

Nr.	Beschreibung	Verletzung der			Angriff auf		
		Vertr.	Verf.	Int.	Clt.	Net.	Pxy.
1	SIP Call Setup Forking	x			x	x	x
2	SIP Flooding (DoS)		x		x		x
3	SIP Call Hijacking / Impersonating	x			x		x
4	**SIP Call termination**		**x**		**x**		**x**
5	SIP Identity Spoofing – Telefonieren auf fremde Rechnung			x	x		x
6	RTP Paket Injection – Datastream verändern		x	x	x	x	x
7	RTP Stream aufzeichnen/mithören	x			x	x	x
8	RTCP Paket Injection – Codec wechsel		x		x		x
9	Multicast Forking	x				x	
10	Verfälschte IP/TCP/SIP/SDP Pakete, Instabilität der Systeme		x		x		x
11	**Voice Spam – autom.Telefonanrufe**				**x**		
12	Asterisk OS Angriff	x [1)]	x [1)]	x [1)]			x
13	Asterisk SW schwäche	x [1)]	x [1)]	x [1)]			x
14	Asterisk Angriff auf Konfigurationsinterface WebGUI / Telnet / SSH	x [1)]	x [1)]	x [1)]			x
15	CDR verändern			x			x

1) Verletzung ist abhängig von der ausgenützten Schwachstelle und kann daher u.U. In allen Bereichen eine Verletzung bedeuten.

Westfälische Hochschule
Gelsenkirchen Bocholt Recklinghausen
University of Applied Sciences

IP-Telefonie, Voice over IP (VoIP)
- Session Initiation Protocol (SIP)
- Real-time Transport Protocol (RTP)

Vielen Dank für Ihre Aufmerksamkeit
Fragen ?

Prof. Dr. (TU NN)
Norbert Pohlmann

Institut für Internet-Sicherheit – if(is)
Westfälische Hochschule, Gelsenkirchen
http://www.internet-sicherheit.de

if(is) internet-sicherheit.