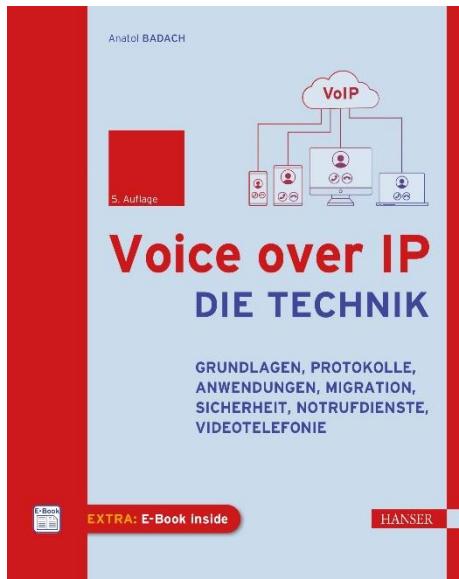


# HANSER



## Leseprobe

zu

## Voice over IP

von Anatol Badach

Print-ISBN: 978-3-446-46944-0

E-Book-ISBN: 978-3-446-47150-4

Weitere Informationen und Bestellungen unter

<https://www.hanser-kundencenter.de/fachbuch/artikel/9783446469440>

sowie im Buchhandel

© Carl Hanser Verlag, München

# Inhalt

<b>1</b>	<b>Vom einfachen Telefon bis zu Next Generation Networks.....</b>	<b>1</b>
1.1	Vom Telefon bis zum intelligenten Netz .....	2
1.1.1	Erfindung des Telefons.....	2
1.1.2	Vom analogen Telefonnetz zum ISDN .....	4
1.1.3	Vom ISDN zum Intelligenten Netz .....	6
1.2	Ansätze für VoIP .....	8
1.2.1	Allgemeines über Internet-Telefonie.....	9
1.2.2	Erweiterung von ISDN mit einem IP-Netz.....	11
1.2.3	IP-Netz als Backbone für PSTN/ISDN .....	13
1.2.4	Kleines IP-Netzwerk als IP-TK-Anlage .....	15
1.3	Evolution der Mobilfunknetze .....	19
1.3.1	Aufbau der Mobilfunknetze nach GSM .....	20
1.3.2	Aufbau von GPRS .....	22
1.3.3	Konzept von UMTS .....	23
	Vereinfachte Architektur von UMTS .....	24
	UMTS-Ausbau und IMS .....	25
1.4	VoIP und Konvergenz der Netze .....	26
1.4.1	Von Single-service-Netzen zum Multiservice-Netz .....	26
1.4.2	Integration von Internet mit Intelligent Network.....	29
	PINT .....	29
	SPIRITS .....	31
1.4.3	Gateway-Plattformen und Migration zu NGNs.....	32
1.4.4	Konzept von Parlay/OSA .....	35
1.4.5	Konzept von JAIN.....	39
1.5	IMS als Kern von Next Generation Networks .....	41
1.5.1	Allgemeines Konzept von IMS .....	42
1.5.2	Mobilität von Benutzern in NGNs.....	43
1.5.3	Registrierung der Lokation eines Benutzers.....	45
1.5.4	VoIP-Session zwischen Benutzern.....	47
1.6	VoIP-Aktivitäten bei Standardisierungsgremien, Organisationen und Foren....	48
1.6.1	IETF und Internet-Standards .....	48
	Organisation der IETF.....	48
	Working Groups mit VoIP-relevanten Themen .....	49
1.6.2	ITU-T und Telekommunikationsstandards.....	51

1.6.3	Organisation des ITU-T .....	51
1.6.3	VoIP-betreffende SGs beim ITU-T.....	52
1.6.4	ETSI und VoIP.....	53
1.6.4	Organisationen und Foren mit VoIP-Aktivitäten.....	54
1.7	Schlussbemerkungen .....	55
<b>2</b>	<b>Signalisierung in Telefonnetzen und ISDN.....</b>	<b>57</b>
2.1	Signalisierung in Telefonnetzen .....	58
2.2	ISDN-Konzept.....	60
2.2.1	ISDN-Schnittstellen .....	61
2.2.2	Protokollbereiche im ISDN.....	62
2.3	D-Kanal-Protokoll.....	63
2.3.1	Schicht 3 des D-Kanal-Protokolls.....	64
2.3.2	Auf- und Abbau einer ISDN-Verbindung.....	66
2.4	Signalisierungssystem Nr.7 .....	68
2.4.1	Funktionsteile von SS7 .....	70
2.4.2	Funktionelle Struktur von SS7.....	71
2.4.3	SS7-Verlauf beim Auf- und Abbau einer ISDN-Verbindung .....	73
2.5	Schlussbemerkungen .....	75
<b>3</b>	<b>TCP/IP- und VoIP-Protokolle .....</b>	<b>77</b>
3.1	Protokollfamilie TCP/IP .....	78
3.2	Prinzip der Kommunikation im Internet.....	80
3.2.1	Bildung von IP-Paketen .....	81
3.2.2	Prinzip der Kommunikation im Internet .....	82
3.2.3	Interpretation von IP-Adressen .....	83
3.2.4	Zweistufige Adressierung .....	84
3.3	Internet-Protokoll IP.....	85
3.4	Transportprotokolle in IP-Netzen.....	86
3.4.1	Verbindungsloses Transportprotokoll UDP .....	87
	Nachteil der UDP-Fehlerkontrolle bei VoIP .....	88
	UDP-Lite.....	89
3.4.2	Verbindungsorientiertes Transportprotokoll TCP.....	90
	TCP-Nutzung .....	91
	Aufbau und Abbau einer TCP-Verbindung .....	93
3.5	Einsatz von DNS .....	95
3.5.1	Aufbau des DNS-Namensraums .....	96
3.5.2	Resource Records.....	97
3.5.3	Beispiel für eine Namensauflösung .....	98

3.5.4	Ermittlung des SIP-Proxy in einer anderen Domain .....	99
3.6	Protokolle für VoIP – eine Übersicht.....	102
3.7	Bedeutung des Protokolls SCTP .....	105
3.7.1	SCTP versus UDP und TCP .....	105
3.7.2	SCTP-Assoziationen .....	106
3.8	ENUM – Konzept und Einsatz .....	108
3.8.1	Bildung von ENUM-Domainnamen und NAPTR-RRs.....	110
3.8.2	Beispiele für den ENUM-Einsatz .....	112
3.9	Schlussbemerkungen .....	114
<b>4</b>	<b>VoIP und QoS in IP-Netzen.....</b>	<b>115</b>
4.1	QoS-Anforderungen bei VoIP .....	116
4.1.1	Einflussfaktoren auf die VoIP-Qualität .....	116
4.1.2	Ende-zu-Ende-Verzögerung .....	117
4.1.3	Übermittlungszeit über ein IP-Netz .....	121
4.1.4	Jitter-Ausgleichpuffer und Paketverluste .....	123
4.2	Verfahren zur Garantie von QoS-Anforderungen.....	124
4.3	Priorisierung von MAC-Frames .....	125
4.4	Differentiated Services .....	126
4.4.1	Differenzierung der IP-Pakete.....	127
4.4.2	DiffServ-Domäne und -Region .....	128
4.5	Queue-Management.....	130
4.5.1	Priority Queueing .....	133
4.5.2	Custom Queueing .....	134
4.5.3	Fair Queueing .....	137
4.5.4	Weighted Fair Queueing.....	139
4.5.5	Class-based Weighted Fair Queueing.....	140
4.6	Einsatz von RSVP.....	142
4.7	Schlussbemerkungen .....	145
<b>5</b>	<b>Sprachcodierung und Echtzeitkommunikation mit RTP/RTCP ...</b>	<b>147</b>
5.1	Sprachcodierung bei VoIP .....	148
5.1.1	Abtastwert-orientierte Sprachcodierung.....	150
5.1.2	Prinzipien der Quantisierung.....	153
5.1.3	Nichtlineare Quantisierung bei PCM .....	154
5.1.4	Nachbildung der Spracherzeugung.....	157
5.1.5	Segment-orientierte Sprachcodierung .....	159
5.1.6	VoIP-relevante Sprachcodierungsverfahren .....	161

5.1.7	Sprachqualität nach MOS-Skala .....	163
5.2	Protokolle für Sprachübermittlung .....	164
5.2.1	Bedeutung einer Session .....	165
5.2.2	RTP/RTCP und Transportprotokolle der IP-Netze .....	168
5.3	Konzept und Funktionen von RTP .....	171
5.3.1	Aufbau von RTP-Paketen .....	172
5.3.2	Statische und dynamische Payload-Typen.....	174
5.3.3	Zeitstempel – Berechnung und Nutzung.....	176
	Berechnung von Zeitstempel für RTP-Pakete.....	177
	Nutzung von Zeitstempel in RTP-Paketen.....	178
5.4	Translator und Mixer.....	180
5.4.1	Translator-Einsatz .....	180
5.4.2	Mixer-Einsatz.....	181
5.5	Protokoll RTCP .....	182
5.5.1	Funktion von RTCP .....	183
5.5.2	Typen der RTCP-Pakete .....	184
5.5.3	Struktur der RTCP-Pakete .....	184
5.5.4	Sender-Report (SR).....	185
	Angaben im SR-Header .....	187
	Sender-Informationen .....	187
	Angaben in Report Blocks .....	188
5.5.5	Receiver Report (RR) .....	188
5.5.6	Einsatz von RTCP XR und VoIP-Metriken .....	189
5.6	Abschätzung von QoS-Parametern.....	191
5.6.1	Garantie der Isochromität.....	192
5.6.2	Abschätzung von Jitter.....	193
5.6.3	Abschätzung des Round-Trip Time .....	194
5.6.4	Aussage über die Häufung von Paketverlusten.....	196
5.6.5	E-Modell von der ITU-T .....	197
5.7	Secure Real-time Transport Protocol (SRTP) .....	198
5.7.1	Sicherheitsfunktionen von SRTP .....	199
5.7.2	Key-Management-Protokoll und SRTP .....	200
5.7.3	Gesicherte Kommunikation nach SRTP .....	202
5.7.4	Prinzip der Integritätsprüfung und Authentifizierung .....	204
5.7.5	SRTP- und SRTCP-Pakete.....	205
5.7.6	Session Keys bei SRTP .....	206
5.7.7	Vorbereitung eines RTP-Pakets zum Senden.....	208
5.7.8	Bearbeitung eines empfangenen RTP-Pakets .....	210
5.7.9	Schritte bei der Bearbeitung eines RTP-Pakets.....	211
5.8	Kompression des RTP/UDP/IP-Headers .....	212

5.8.1	Bedeutung von CRTP und ROHC .....	213
5.8.2	Konzept der Kompression des RTP/UDP/IP-Headers.....	214
5.8.3	Kompression und Dekompression nach CRTP .....	216
5.8.4	Besonderheiten von ROHC .....	220
5.9	Schlussbemerkungen .....	221
<b>6</b>	<b>VoIP nach dem Standard H.323 .....</b>	<b>223</b>
6.1	Systemkomponenten nach H.323.....	224
6.1.1	H.323-Domains .....	225
6.1.2	Protokollfamilie TCP/IP und H.323 .....	226
6.1.3	Sprach- und Videocodierung in H.323-Systemen .....	228
6.1.4	Arten von Kanälen bei der Multimedia-Kommunikation.....	229
6.2	Signalisierung nach H.323 .....	230
6.2.1	Schritte vor der Audio/Video-Übermittlung.....	231
6.2.2	Schritte nach der Audio/Video-Übermittlung .....	232
6.2.3	Fast Connect Procedure .....	233
6.3	Realisierung von RAS-Funktionen .....	236
6.3.1	Gatekeeper-Entdeckung .....	237
6.3.2	Registrierung und Deregistrierung beim Gatekeeper .....	238
6.3.3	Zulassung von Verbindungen .....	239
6.3.4	Abfrage der IP-Adresse eines Endpunktes .....	241
6.4	Signalisierung der Anrufe nach H.225.0.....	242
6.4.1	Struktur von Anruf-SIG-Nachrichten beim H.225.0 .....	243
6.4.2	Anrufsignalisierung ohne Gatekeeper .....	243
6.4.3	Direkte Anrufsignalisierung beim Gatekeeper-Einsatz .....	245
6.4.4	Über Gatekeeper geroutete Anrufsignalisierung .....	246
6.4.5	VoIP im Verbund mit ISDN .....	248
6.5	Einsatz des Protokolls H.245 .....	249
6.5.1	Beschreibung von Terminal-Fähigkeiten .....	250
6.5.2	Austausch von Terminal-Fähigkeiten.....	252
6.5.3	Master/Slave-Festlegung .....	252
6.5.4	Aufbau logischer Kanäle .....	253
6.5.5	Abbau logischer Kanäle .....	254
6.5.6	Änderung von Eigenschaften einer Verbindung.....	255
6.5.7	Beispiel für einen Verlauf des Protokolls H.245 .....	256
6.6	Supplementary Services nach H.450.x .....	257
6.6.1	H.450.1 als Basis für Supplementary Services .....	259
6.6.2	Beispiele für Supplementary Services .....	260
6.7	Roaming bei VoIP nach H.323 .....	262

6.7.1	Arten von Roaming .....	262
6.7.2	Registrierung eines Gast-Teilnehmers .....	264
6.7.3	Ankommender Anruf zu einem Gast-Teilnehmer .....	267
6.7.4	Abgehender Anruf aus einer Fremd-Domain .....	269
6.7.5	Deregistrierung eines Gast-Teilnehmers .....	270
6.8	Schlussbemerkungen .....	270
<b>7</b>	<b>VoIP mit SIP .....</b>	<b>273</b>
7.1	Verschiedene Aspekte des SIP-Einsatzes .....	274
7.1.1	SIP und verschiedene Transportprotokolle .....	274
7.1.2	Wichtige SIP-Besonderheiten .....	276
7.1.3	Struktur von SIP-Adressen .....	278
7.1.4	Funktion eines SIP-Proxy .....	280
7.1.5	Trapezoid-Modell von SIP .....	282
7.1.6	SIP-Verlauf im Trapezoid-Modell .....	284
7.1.7	Unterstützung von Benutzermobilität .....	285
7.1.8	Erweiterter SIP-Proxy als B2BUA .....	287
7.1.9	Typischer SIP-Verlauf .....	288
	Angaben in SIP- Nachrichten .....	290
	SIP-Verlauf innerhalb einer Domain .....	293
	SIP-Verlauf ohne Proxy .....	293
7.2	Beispiele für den Einsatz von SIP .....	294
7.2.1	Typischer Einsatz von SIP-Proxy-Servern .....	295
7.2.2	Umleitung einer Session mit Redirect-Server .....	296
7.2.3	Weiterleitung einer Session mit Proxy-Servern .....	298
7.2.4	Anrufverzweigung mit SIP .....	299
7.2.5	Einsatz eines Voice-Mail-Servers .....	301
7.3	SIP-Nachrichten – ihre Bedeutung und Struktur .....	303
7.3.1	Request-Typen .....	303
7.3.2	Response-Klassen .....	306
7.3.3	Aufbau von SIP-Nachrichten .....	307
	Struktur von SIP-Requests .....	307
	Struktur von SIP-Responses .....	309
	Wichtige Header-Felder .....	310
7.4	Beschreibung von Sessions mit SDP .....	313
7.4.1	Typischer Einsatz von SDP .....	314
7.4.2	Bestandteile der Beschreibung einer Session .....	316
7.4.3	Beschreibung auf dem Session-Level .....	320
7.4.4	Zeitspezifische Angaben .....	322
7.4.5	Beschreibung von Medien .....	323

7.5	Betriebsarten bei SIP .....	327
7.5.1	Proxy-Mode und Redirect-Mode.....	327
7.5.2	Einsatz von Proxy- und Redirect-Server .....	328
7.6	Registrierung der Lokation von Benutzern .....	330
7.7	Sessionbezogene Leistungsmerkmale mit SIP .....	332
7.7.1	Klassen der Leistungsmerkmale mit SIP .....	332
7.7.2	Call Hold/Retrieve – Anhalten/Wiederaufnahme.....	336
7.7.3	Consultation Hold – Anhalten mit Rückfrage .....	337
7.7.4	Call Park – Parken einer Session.....	338
7.7.5	Call Pickup – Übernahme einer Session.....	341
7.7.6	Call Forwarding – Weiterleitung einer Session.....	342
7.7.7	Unattended Call Transfer .....	343
7.7.8	Attended Call Transfer .....	344
7.7.9	SIP-Verlauf bei Rückruf.....	346
7.8	Response- und Request-Routing.....	348
7.9	Konvergenz der IP-Netze und ISDN .....	350
7.9.1	SIP und das D-Kanal-Protokoll .....	351
7.9.2	SIP und Signalisierungssystem Nr. 7 .....	352
7.10	Koexistenz von SIP und H.323 .....	353
7.11	Schlussbemerkungen .....	355
<b>8</b>	<b>VoIP-Gateways: Konzepte und Protokolle .....</b>	<b>357</b>
8.1	VoIP und klassische Systeme für Sprachkommunikation.....	358
8.2	Konzept von MGCP.....	360
8.2.1	Grundbegriffe bei MGCP .....	360
8.2.2	MGCP-Commands .....	362
8.2.3	MGCP-Responses .....	363
8.2.4	Auf- und Abbau einer VoIP-Session nach MGCP .....	364
8.3	Protokoll Megaco.....	368
8.3.1	Konzept von Megaco.....	369
8.3.2	Megaco-Commands.....	371
8.3.3	Auf- und Abbau einer VoIP-Session nach Megaco .....	372
8.3.4	Megaco und Integration von VoIP mit ISDN .....	374
8.4	Schlussbemerkungen .....	376
<b>9</b>	<b>IP-Telefonie-Routing und VoIP-Peering.....</b>	<b>377</b>
9.1	Typische Probleme bei VoIP .....	378
9.1.1	Routing ankommender Anrufe aus dem ISDN/PSTN .....	379
9.1.2	Routing abgehender Anrufe .....	381

9.2	Konzept und Einsatz von TRIP .....	382
9.2.1	Bedeutung von TRIP.....	383
9.2.2	TRIP als Bruder von BGP.....	384
9.3	Vernetzung von VoIP-Zonen mit H.323 .....	385
9.3.1	Routing abgehender Anrufe zwischen H.323-Zonen.....	385
9.3.2	Routing der Anrufe aus dem ISDN zu einer H.323-Zone .....	387
9.4	Vernetzung von VoIP-Zonen mit SIP .....	388
9.4.1	Routing der Anrufe zwischen VoIP-Zonen mit SIP.....	388
9.4.2	Routing der ISDN-Anrufe zu VoIP-Zonen mit SIP .....	389
9.5	Peering bei VoIP mit SIP .....	390
9.5.1	Ziele und Arten von Peering .....	390
9.5.2	Prinzip von Basic Peering .....	392
9.5.3	Integrated Peering versus Decomposed Peering .....	393
9.5.4	Federation-based Peering .....	394
9.6	Schlussbemerkungen .....	396
<b>10</b>	<b>Migration zum VoIP-Einsatz.....</b>	<b>397</b>
10.1	Verschiedene Aspekte der Migration zu VoIP .....	398
10.1.1	Sanfte Migration zu VoIP .....	398
10.1.2	Harte Migration zu VoIP .....	398
10.1.3	Typische Fälle bei der Migration zu VoIP .....	399
10.1.4	Architekturmödelle der VoIP-Systeme .....	400
10.2	Hybride VoIP-Systemarchitekturen .....	402
10.2.1	Hybride VoIP-Systemarchitektur am Einzelstandort.....	402
10.2.2	Arten der Vernetzung von TK-Anlagen.....	403
	Vernetzung von TK-Anlagen mit zentraler Anrufsteuerung.....	403
	Vernetzung von TK-Anlagen mit verteilter Anrufsteuerung .....	404
10.2.3	Standortübergreifende hybride VoIP-Systemarchitekturen .....	404
	VoIP-Systemarchitekturen mit zentraler Anrufsteuerung .....	404
	VoIP-Systemarchitekturen mit verteilter Anrufsteuerung .....	405
10.3	Reine VoIP-Systemarchitekturen .....	406
10.3.1	Reine VoIP-Systemarchitektur am Einzelstandort.....	408
10.3.2	Verkabelung für die Unterstützung von VoIP.....	410
	Getrennte Sprach- und Datenverkabelung .....	410
	Gemeinsame Sprach- und Datenverkabelung .....	411
10.3.3	Standortübergreifende reine VoIP-Systemarchitekturen.....	412
	VoIP-Systemarchitektur mit zentraler Anrufsteuerung.....	412
	VoIP-Systemarchitektur mit verteilter Anrufsteuerung .....	415
10.4	Auswahl einer VoIP-Systemlösung.....	416

10.5	Hauptschritte bei der Migration zu VoIP .....	417
10.5.1	Ist-Analyse bei der Migration zu VoIP.....	419
	Organisatorische Aspekte der Ist-Analyse.....	420
	Technische Aspekte der Ist-Analyse .....	421
10.5.2	Anforderungen an VoIP-System .....	423
	Organisatorische Anforderungen.....	423
	Technische Anforderungen .....	424
10.5.3	Komponenten des VoIP-Systemkonzeptes.....	425
10.6	VoIP mit SIP in Netzwerken mit NAT .....	426
10.6.1	Prinzipien von NAT .....	427
10.6.2	Probleme mit SIP beim NAT-Einsatz .....	429
10.6.3	Symmetric Response – Hilfe bei der Signalisierung .....	432
10.6.4	Symmetric RTP/RTCP – Hilfe beim Medientransport.....	433
10.6.5	Einsatz von STUN.....	434
10.6.6	Nutzung von TURN .....	437
10.6.7	ICE als Lösung des NAT-Problems .....	439
10.7	Schlussbemerkungen .....	443
<b>11</b>	<b>VoIP-Sicherheit .....</b>	<b>445</b>
11.1	Probleme der VoIP-Sicherheit .....	446
11.1.1	Primäre Ziele der VoIP-Sicherheit .....	446
11.1.2	Verschiedene Aspekte der VoIP-Sicherheit .....	448
11.1.3	Sicherheitsproblembereiche im Netzwerk .....	449
11.1.4	Phasen des VoIP-Sicherheitsprozesses.....	451
11.1.5	Vorgehensweise bei der Planung der VoIP-Sicherheit .....	452
11.2	Bedrohungstypen und Angriffsarten bei VoIP.....	454
11.2.1	Typische Angriffe in Netzwerken .....	454
11.2.2	Typische Angriffe bei VoIP .....	456
	Angriffe auf dem Anwendungsniveau.....	456
	Angriffe auf dem Niveau der Transportschicht .....	458
	Angriffe auf IP-Niveau.....	458
	Angriffe auf MAC-Niveau .....	459
	Beispiele für einige Angriffe bei VoIP.....	459
	Klassen der Angriffe auf VoIP-Systeme .....	461
11.2.3	Lauschangriffe bei VoIP und Gegenmaßnahmen.....	462
11.2.4	Abfangen und Modifikation von VoIP-Anrufen .....	463
11.2.5	Beeinträchtigen des VoIP-Dienstes .....	466
11.2.6	Missbrauch des VoIP-Dienstes.....	466
11.3	Sicherheit bei VoIP mit SIP .....	468
11.3.1	Gefährdungen in VoIP-Systemen mit SIP .....	468

Registration Hijacking .....	470
Session Hijacking – Entführung einer Session .....	472
Imitation eines SIP-Proxy-Servers.....	473
11.3.2 SIP Digest Authentication – Einsatz und Konzept.....	474
Prinzip der Authentifizierung nach SIP-Digest.....	474
Authentifizierung bei Registrierung.....	476
Benutzer-Authentifizierung von einem Proxy .....	477
11.3.3 Einsatz von S/MIME bei SIP .....	478
Asymmetrische Kryptosysteme als Grundlage von S/MIME .....	478
Idee des S/MIME-Einsatzes bei SIP .....	479
Garantie der Vertraulichkeit bei SIP mit S/MIME.....	480
Signierung von SIP-Nachrichten .....	481
11.4 Ermittlung des Schutzbedarfs bei VoIP .....	482
11.4.1 Beschreibung der Sicherheitsschwachstelle.....	483
11.4.2 Vorgehensweise bei der Analyse von Bedrohungen.....	484
11.4.3 Aussage über den Schutzbedarf .....	487
11.4.4 Risikoanalyse .....	487
11.4.5 Erfassung des Schutzbedarfs.....	489
11.5 Festlegung von Sicherheitsanforderungen.....	490
11.5.1 Darstellung der Sicherheitsschwachstelle .....	490
11.5.2 Katalog von Sicherheitsanforderungen .....	490
11.6 Maßnahmen zur Erhöhung der VoIP-Sicherheit .....	491
11.6.1 Spezifikation von Sicherheitsmaßnahmen .....	491
11.6.2 Typische Sicherheitsschwachstellen .....	493
11.7 Schlussbemerkungen .....	495
<b>12    VoIP mit Peer-to-Peer SIP .....</b>	<b>497</b>
12.1 Besonderheiten der P2P-Netzarchitektur.....	498
12.1.1 Traditionelle Client-Server-Architektur .....	498
12.1.2 Arten von P2P-Netzarchitekturen.....	499
12.1.3 Bedeutung des Bootstrap-Servers.....	499
12.1.4 Overlay-Ringnetz für P2P-Kommunikation .....	500
12.1.5 Peer, Client und Benutzer.....	502
12.2 Funktionsweise des P2P-Overlay-Netzes .....	503
12.2.1 P2P-Overlay-Netz als Ringnetz.....	504
12.2.2 Bedeutung von Finger-Tabellen .....	506
12.2.3 Beitritt eines Peer zum Overlay-Ringnetz .....	507
12.2.4 Routing im Overlay-Ringnetz .....	508
12.3 Ziele und Bedeutung des P2PSIP .....	511

---

12.3.1	Allgemeines Prinzip von Instant Messaging .....	512
12.3.2	Informationsmodell von Presence Services .....	513
12.4	Gegenüberstellung von SIP und P2PSIP .....	514
12.5	Konzept von P2PSIP .....	517
12.5.1	Prinzip der Anrufsignalisierung bei P2PSIP .....	518
12.5.2	Funktionskomponenten von P2PSIP .....	519
12.5.3	Peer bei P2PSIP im Schichtenmodell .....	522
12.6	Peer-Protokoll bei P2PSIP .....	523
12.6.1	Funktionen des Peer-Protokolls bei P2PSIP .....	524
12.6.2	Beitritt eines Peer zum Overlay-Ringnetz .....	525
12.6.3	Registrierung eines Client-Knotens im Overlay-Ringnetz .....	527
12.6.4	Aufbau einer Session für VoIP-Kommunikation .....	530
12.7	Abschließende Bemerkungen .....	531
<b>13</b>	<b>VoIP-basierte Notrufdienste .....</b>	<b>533</b>
13.1	Wichtige Aspekte von Notrufdiensten .....	534
13.1.1	Notrufdienst aus der Sicht des Notrufenden .....	534
13.1.2	Probleme bei der Realisierung von Notrufdiensten .....	535
13.2	Grundlagen VoIP-basierter Notrufdienste .....	537
13.2.1	Schritte bei der Realisierung von VoIP-Notrufdiensten .....	537
13.2.2	Typische Struktur VoIP-basierter Notrufsysteme .....	539
13.2.3	Anforderungen an VoIP-Notrufdienste .....	541
13.2.4	Identifizierung eines Notrufes .....	542
13.3	Bestimmung der Lokation des Notrufenden .....	543
13.3.1	Bestimmung der Lokation in Netzwerken .....	544
13.3.2	Bestimmung der Lokation in Mobilfunknetzen .....	545
13.3.3	Bedeutung von Positionierungssystemen .....	546
13.4	Realisierung von VoIP-Notrufdiensten .....	546
13.4.1	Emergency Service Framework für VoIP-Notrufdienste .....	547
13.4.2	Migration zum VoIP-basierten Notrufdienst .....	548
13.5	Konzept und Einsatz von LoST .....	550
13.5.1	Typische Anwendungen von LoST .....	550
	LoST in VoIP-basierten Notrufsystemen .....	551
	LoST in Überwachungssystemen .....	553
	Bedeutung von LoST in Location Based Services .....	554
	LoST-Einsatz beim Katastrophenschutz .....	556
13.5.2	Logische Architektur von LoST .....	556
	Tree mit LoST-Servern .....	558
	Rekursive Ermittlung von SIP-URI .....	559

Nachricht <code>findService</code> .....	562
Nachricht <code>findServiceResponse</code> .....	563
Nachricht <code>listServiceByLocation</code> .....	565
Nachricht <code>listServiceByLocationResponse</code> .....	565
13.6 LoST in VoIP-basierten Notrufsystemen .....	566
13.7 Sicherheitsaspekte in VoIP-Notrufsystemen .....	568
13.8 Abschließende Bemerkungen .....	569
<b>14 WebRTC – Konzept und Einsatz .....</b>	<b>571</b>
14.1 Funktionale Komponenten von WebRTC .....	572
14.1.1 Webbrowser mit WebRTC-Unterstützung .....	572
14.1.2 WebRTC-Server und WebSocket-Protokoll .....	574
14.1.3 Signalisierungsprotokoll bei WebRTC .....	575
14.1.4 Arten der Kommunikation bei WebRTC .....	575
14.2 Modell der Kommunikation bei WebRTC .....	576
14.2.1 Dreiecksmodell von VoIP mit SIP .....	576
14.2.2 WebRTC-Dreiecksmodell – ohne Transcoder-Einsatz .....	578
14.2.3 WebRTC-Dreiecksmodell – mit Transcoder-Einsatz .....	580
14.3 Schritte vor und nach der WebRTC-Nutzung .....	581
14.4 Session zwischen WebRTC-Clients .....	584
14.5 Bedeutung von ENUM bei WebRTC .....	586
14.5.1 Ermittlung der IP-Adressen von WebRTC-Clients .....	587
14.5.2 Dreiecksmodell von WebRTC und ENUM-Einsatz .....	588
14.6 Nutzung von SIP bei WebRTC .....	589
14.6.1 Modell von WebRTC beim Einsatz von SIP .....	590
14.6.2 WebRTC mit SIP und privaten IPv4-Adressen .....	592
14.7 Kopplung von WebRTC mit VoIP-Systemen .....	592
14.8 Sicherheitsproblemberiche bei WebRTC .....	594
14.9 Standardisierung von WebRTC .....	598
14.10 Schlussbemerkungen .....	599
<b>Literatur, Standards, Webquellen .....</b>	<b>601</b>
<b>Abkürzungsverzeichnis .....</b>	<b>611</b>
<b>Index .....</b>	<b>619</b>

# Vorwort

Die heutige Gesellschaft ist ohne Telefon und Internet nicht mehr vorstellbar. Das Internet ist zum unabdingbaren Kommunikationsmedium geworden, über das jeder jederzeit Zeit Informationen über fast alles abrufen sowie Nachrichten senden und empfangen kann. Das Internet ist ein weltweites Rechnernetz, in dem die Daten nach dem sog. *Internet Protocol* (IP) übermittelt werden. Man kann es auch als Dienst für die Übermittlung von Informationen in Form von IP-Paketen ansehen. Vergleicht man diesen Dienst mit dem Briefdienst der Post, so entspricht ein IP-Paket einem Brief und die sog. *IP-Adresse* einer postalischen Adresse. Auch in anderen Netzen werden Daten als IP-Pakete übermittelt. Alle Rechnernetze mit dem Protokoll IP werden als *IP-Netze* bezeichnet. Sie dienen unter anderem zur Sprachkommunikation. Die Übermittlung von Sprache in IP-Paketen wird als *Sprache über IP* bzw. kurz *VoIP* (*Voice over IP*) bezeichnet.

VoIP bedeutet nicht nur zwei Telefone und IP dazwischen. Hinter diesem Begriff verbergen sich sehr komplexe Vorgänge. Hierzu gehören sog. *Signalisierungsprotokolle*, nach denen eine Verbindung zwischen Telefonen vor einem Telefongespräch aufgebaut und danach abgebaut werden kann. Die Signalisierungsprotokolle H.323 und SIP sind in der „IT-Welt“ geläufig. Ein Telefon für VoIP, d.h. ein *IP-Telefon*, ist nicht nur ein Telefon, sondern ein Rechner an einem IP-Netz, der eine IP-Adresse hat. Das IP-Telefon hat zusätzlich eine VoIP-spezifische Adresse, die eine Telefonnummer sein kann. Eine Telefonverbindung im Telefonnetz wird unter einer Telefonnummer aufgebaut. Bei VoIP wird zwar das Ziel der Verbindung mit einer VoIP-spezifischen Adresse – z.B. einer Telefonnummer – angegeben, aber diese Verbindung kann bei IP nur unter einer IP-Adresse aufgebaut werden. Das ist ein Beispiel für die vielen Probleme bei VoIP.

Dieses Buch stellt sowohl die Technik von VoIP als auch die Migration zu VoIP und die VoIP-Sicherheit fundiert dar. Hierfür geht es u.a. auf folgende Themen ein: die Perspektiven der Sprachkommunikation, die Signalisierung im Telefonnetz und im ISDN, die Internetprotokollfamilie, Quality of Service, wichtige Sprachcodierungsverfahren, die Prinzipien der Echtzeitkommunikation mit RTP/RTCP und mit Secure RTP, den Standard H.323 und das Protokoll SIP, VoIP-Gateways, Peering bei VoIP, SIP Security, VoIP-Notrufdienste, VoIP mit Peer-to-Peer SIP und WebRTC als Videotelefonie übers Internet. Dieses Werk vermittelt unabdingbare Informationen, um die Sprachkommunikation in IP-Netzen (z.B. im Internet) besser verstehen, diese nutzen und neue VoIP-Anwendungen konzipieren bzw. auch entwickeln zu können.

*VoIP:  
nicht nur  
zwei Telefone  
und IP*

*Ziel des  
Buches*

*An wen  
richtet sich  
das Buch?*

Das Buch ist so aufgebaut, dass jeweils zunächst die Grundlagen fundiert dargestellt und danach praktische Anwendungen diskutiert werden. Damit eignet es sich nicht nur als Lehrbuch für Studenten und Neueinsteiger, sondern auch als Nachschlagewerk für alle Experten, zu deren Aufgaben *die Entwicklung, Planung oder Betreuung* verschiedener Netzwerke oder Netzwerkapplikationen gehört. Die praxisorientierte und reichlich illustrierte Darstellung der Inhalte verfolgt das Ziel, allen „Netzwerk-Fans“ die Nutzung dieses Buches im Selbststudium zu ermöglichen.

*Kapitel 1*

Kapitel 1 enthält einen kompakten Überblick über klassische Netze zur Sprachkommunikation, Mobilfunknetze (GSM, UMTS), Ansätze für VoIP sowie eine Einführung in Next Generation Networks (NGN), die durch die Konvergenz von Netzen entstehen. Die VoIP-Aktivitäten der verschiedenen Standardisierungsgremien, Konsortien und Foren werden hier ebenfalls kurz dargestellt.

*Kapitel 2*

Kapitel 2 widmet sich den *Signalisierungsprinzipien*, also der Übermittlung der Steuerung beim Auf- und Abbau von Telefonverbindungen. Die Schwerpunkte liegen hier auf einer fundierten Darstellung des D-Kanal-Protokolls aus dem ISDN und des Signalisierungssystems Nr. 7. Diese Inhalte sind Basiswissen in Bezug auf VoIP.

*Kapitel 3*

Kapitel 3 vermittelt die Grundlagen der bei VoIP benötigten Internetprotokollfamilie (IP, TCP, UDP, SCTP ...). Insbesondere wird hier auf die Bedeutung von DNS (*Domain Name System*) bei VoIP mit SIP eingegangen. In diesem Kapitel wird auch das Konzept ENUM präsentiert, nachdem Telefonnummern auch bei VoIP verwendet werden können.

*Kapitel 4*

Hinsichtlich der Qualität der Sprachübermittlung in IP-Netzen werden bestimmte Anforderungen an diese Netze gestellt, die als QoS-Anforderungen bezeichnet werden. Kapitel 4 zeigt, welche Konzepte zur Erfüllung dieser Anforderungen es gibt. Insbesondere werden die für VoIP wichtigen QoS-Parameter, Differentiated Services, Queue-Management und das Protokoll RSVP zur Reservierung der Bandbreite dargestellt.

*Kapitel 5*

Sprachkommunikation ist Echtzeitkommunikation. Sie wird mittels der Protokolle RTP und RTCP realisiert. Kapitel 5 zeigt zuerst, wie Sprache nach verschiedenen Verfahren codiert und mit Hilfe von RTP/RTCP übermittelt wird. Im Weiteren präsentiert dieses Kapitel Secure RTP sowie die Möglichkeiten der Kompression des RTP/UDP/IP-Headers und erläutert die Bedeutung von VoIP-Metriken.

*Kapitel 6*

Die ersten VoIP-Systemlösungen basierten auf dem Standard H.323. H.323 ist ein komplexes Rahmenwerk, das regelt, wie weitere Signalisierungsprotokolle wie H.225.0 und H.245 verwendet werden. Kapitel 6 ist dem VoIP-Konzept nach H.323 gewidmet. Hier werden auch die sog. *Supplementary Services* nach H.450.x und die Möglichkeiten zur Unterstützung der Mobilität von VoIP-Teilnehmern präsentiert.

Das Protokoll SIP (*Session Initiation Protokoll*), das bei VoIP als Signalisierungsprotokoll dient, gehört zu den wichtigsten Internetprotokollen. Kapitel 7 erläutert, wie SIP konzipiert wurde und zeigt mittels verschiedener SIP-Abläufe, wie es sich einsetzen lässt. Hierbei wird auf verschiedene SIP-Funktionen und Leistungsmerkmale von VoIP mit SIP eingegangen und auch darauf, wie SIP mit H.323 koexistieren kann. Kapitel 7

VoIP-Systeme entstehen nicht auf der „grünen Wiese“, sondern müssen in bereits vorhandene Systemkomponenten und Netze zur Sprachkommunikation integriert werden, um die Sprachkommunikation auch zwischen klassischen Telefonen und IP-Telefonen zu ermöglichen. Hierfür sind verschiedene VoIP-Gateways und Protokolle für die Steuerung dieser Gateways nötig. Auf diese Aspekte geht Kapitel 8 ein. Kapitel 8

Kapitel 9 zeigt die Prinzipien, nach denen das sog. *Telefonie-Routing* realisiert werden kann, um VoIP weltweit zwischen beliebigen administrativen Domänen (öffentlichen Verwaltungen, Unternehmen, ...) zu ermöglichen. Hierbei ist das Konzept TRIP von großer Bedeutung. Dieses Kapitel geht auch auf die Realisierung von Peering bei VoIP mit SIP ein. Kapitel 9

Der Einsatz von VoIP wird heute in keinem Netzwerkprojekt außer Acht gelassen. Die Migration zu VoIP in Unternehmen und anderen Institutionen ist ein komplexes Projekt, bei dem diverse Aspekte berücksichtigt werden müssen. Kapitel 10 widmet sich diesem Thema und erläutert technische Lösungen wie z.B. STUN, TURN und ICE, die eine Nutzung von VoIP mit SIP in Netzwerken mit privaten IP-Adressen ermöglichen. Kapitel 10

Um *VoIP-Sicherheit* zu gewährleisten und VoIP-Netzwerke gegen böswillige Angriffe zu schützen, sind bestimmte technische Lösungen und Maßnahmen nötig. Kapitel 11 vermittelt einen fundierten Überblick über Bedrohungen und Sicherheitsmechanismen bei VoIP – insbesondere bei VoIP mit SIP – sowie über die Planung der VoIP-Sicherheit. Kapitel 11

Eine spontane Rechnerkommunikation ermöglichen sog. *Peer-to-Peer-Netze* (*P2P-Netze*), welche auf der Idee serverloser Netzarchitekturen basieren. Für VoIP sind derartige Netze von enorm großer Bedeutung. Um die VoIP in P2P-Netzen zu ermöglichen, wurde eine als P2PSIP (*Peer-to-Peer SIP*) bezeichnete Ergänzung zum SIP entwickelt. Das P2PSIP ist ein komplexes Rahmenwerk, das zusätzliche Protokolle beschreibt und bestimmt, wie diese mit dem SIP kooperieren müssen. Kapitel 12 erläutert dies und vergleicht u.a. die Konzepte „VoIP mit SIP“ und „VoIP mit P2PSIP“. Kapitel 12

Eine wichtige Funktion öffentlicher Netze zur Sprachkommunikation ist es, Anrufe in Notsituationen zu ermöglichen. Uns allen sind die Notrufnummern 110 und 112 bekannt. Öffentliche Netze zur Sprachkommunikation bieten Notrufdienste an und öffentliche VoIP-Systeme müssen dies auch tun. VoIP-basierte Notrufdienste bringen mit Hilfe des Protokolls SIP neue Möglichkeiten Kapitel 13

der Hilfeleistung mit sich. So werden zwecks der Ermöglichung einer breiten Palette an Notrufdiensten spezielle Bezeichner in Form von URNs (*URN: Uniform Resource Name*) zur Identifikation verschiedener Notfälle verwendet – wie z.B. `urn:service:sos.police`. Eine besondere Aufgabe jedes VoIP-Notrufsystems ist die Abbildung eines URN auf die SIP-Adresse der zuständigen Notrufleitstelle. Hierfür wurde das Protokoll LoST (*Location-to-Service Translation*) entwickelt. Kapitel 12 stellt die Systemlösungen für die Realisierung von auf IP-Netzen – insbesondere auf dem Internet – basierender Notrufdienste vor.

#### *Kapitel 14*

Von großer praktischer Bedeutung ist die Integration multimedialer Echtzeitkommunikation, vor allem von VoIP, mit Webanwendungen im Internet. Diese Integration lässt sich weitgehend mit Hilfe des technischen Konzepts für WebRTC (*Web Real-Time Communication*) verwirklichen. Somit liegt die grundlegende WebRTC-Idee sämtlichen Systemlösungen zugrunde, die als technische Basis für die Einrichtung von Homeoffices anzusehen sind. Betrachtet man WebRTC aus rein technischer Sicht, stellt man fest, dass die Idee von VoIP bei WebRTC übernommen und um zusätzliche Funktionen erweitert wurde. Als Folge dieser WebRTC-Besonderheit können bei Bedarf verschiedene RTC-spezifische Funktionsmodule zur WebRTC-Realisierung von einem Webserver heruntergeladen und in den Webbrower direkt „eingebaut“ werden. Kapitel 14 erläutert das technische Konzept und die Einsatzmöglichkeiten von WebRTC.

Die ersten vier Auflagen dieses Buches entstanden größtenteils auf der Basis von Skripten meiner Vorlesungen zu den Themen *Integrierte Netze* und *VoIP – Technik und Anwendungen*, die ich über mehrere Jahre an der Hochschule Fulda, Fachbereich Angewandte Informatik im Studienschwerpunkt *Telekommunikation*, gehalten habe. Die letzten drei Kapitel, welche die Themen *VoIP mit P2PSIP*, *VoIP-basierte Notrufdienste* und *WebRTC* präsentieren, sind die Ergebnisse meiner privaten Forschung.

#### *Danksagung*

An dieser Stelle möchte allen Personen danken, die mich mit ihren Anregungen und Bemerkungen unterstützt haben. Für die sehr gute Zusammenarbeit mit dem Hanser Verlag möchte ich mich herzlich bei Frau Margarete Metzger, Frau Irene Weilhart, Frau Brigitte Bauer-Schiewek, Frau Sandra Gottmann, Frau Kristin Rothe und Frau Sylvia Hasselbach bedanken. Nicht zuletzt richte ich meinen Dank auch an meine Tochter Katarzyna für das fleißige Korrekturlesen und meine Frau Ingeborg für Unterstützung während des Schreibens dieses Buches.

Fulda, April 2022

*Anatol Badach*

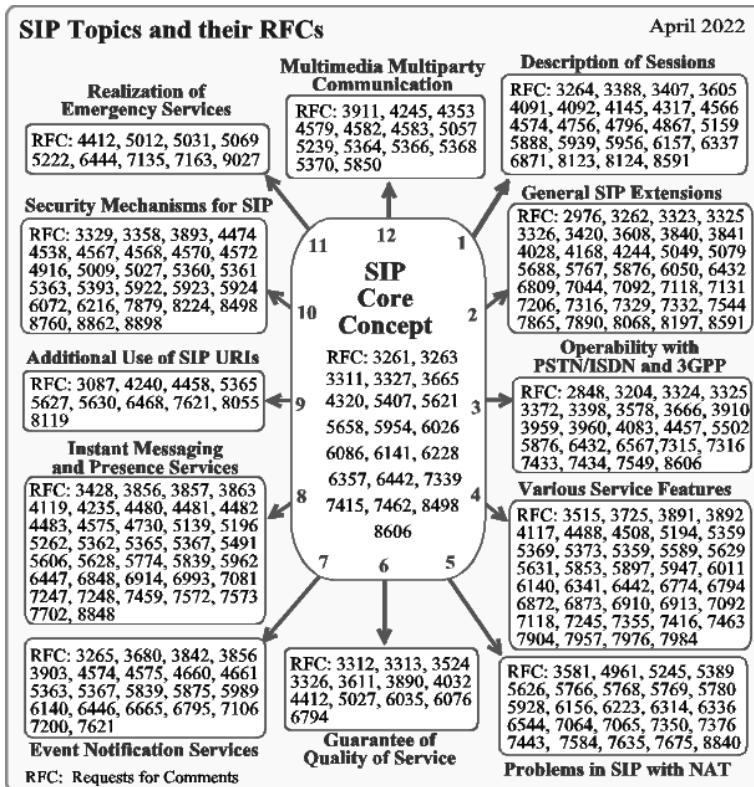
# SIP als Kern dieses Buches

Seit Erscheinen der vierten Auflage dieses Buches sind über 10 Jahre vergangen. Während dieser Zeit haben diverse neue VoIP-betreffende Themen an Bedeutung gewonnen. Da die vierte Auflage, technische Probleme, die zeitliche Entwicklung und typische Nutzungsaspekte von VoIP in fundierter Form präsentierte, wurde die fünfte nur um aktuelle Themen erweitert – nämlich um: VoIP mit Peer-to-Peer SIP, VoIP-basierte Notrufdienste und das WebRTC-Konzept und dessen Einsatz.

Ein wesentlicher Teil dieses Buches betrifft das Protokoll SIP. Dieses wurde in seiner ersten Version im März 1999 lediglich als Signalisierungsprotokoll für VoIP konzipiert. Damals hätten dessen Entwickler nicht davon zu träumen gewagt, dass das Konzept von SIP so einem breiten Spektrum VoIP-relevanten Themen, wie das folgende Bild illustriert, zugrunde liegen würde.

*Inhaltliche Erweiterungen in der 5. Auflage*

*Spektrum von SIP-relevanten Themen*



Für nähere Informationen zu den hier aufgelisteten Themen sei verwiesen auf

<https://www.researchgate.net/publication/288493389>

## Der Autor

**Prof. Dr.-Ing. Anatol Badach** arbeitet seit Beginn der 70er Jahre auf den Gebieten *Informatik* und *Telekommunikation*; Promotion (1975) auf dem Gebiet *Datenkommunikation*; Habilitation (1983) auf dem Gebiet *Rechnernetze*. Von 1985 bis 2012 war er Professor im Fachbereich *Angewandte Informatik* an der Hochschule Fulda. Seine Schwerpunkte in Lehre und Forschung waren: *Rechnerkommunikation*, *Netzwerktechnologien* und *Multiservice Networking*. Geforscht hat er im Bereich der Multimedia-Kommunikation über IP-Netze, dabei insbesondere in der Entwicklung intelligenter und multimedialer TK-Dienste.



Prof. Badach ist Autor zahlreicher Veröffentlichungen und mehrerer Fachbücher; dazu zählen u.a. beim Hanser-Verlag:

- *Technik der IP-Netze* (Mitautor),
- *Web-Technologien* (Mitautor),
- *Voice over IP – Die Technik*,
- *Netzwerkprojekte* (Mitautor).

und bei anderen Verlagen:

- *ISDN im Einsatz*,
- *High Speed Internetworking* (Mitautor),
- *Datenkommunikation mit ISDN*,
- *Integrierte Unternehmensnetze*.

Seine Erfahrung hat Prof. Badach auch als Leiter und Referent bei Fachkongressen und -seminaren vermittelt.

Abrufbar sind die Veröffentlichungen von Prof. Badach mit zahlreichen Abbildungen unter der Adresse:

<https://www.researchgate.net/profile/Anatol-Badach/research>

Ihre Kritik sowie Verbesserungsvorschläge und Korrekturen nimmt er gerne entgegen:

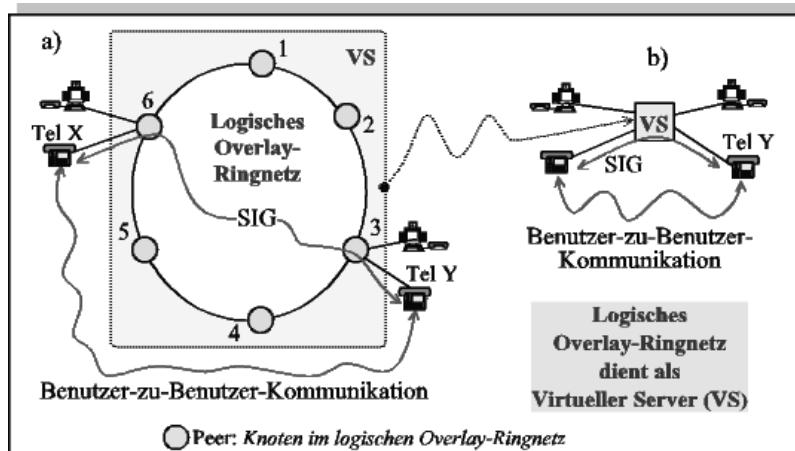
[Anatol.Badach@informatik.hs-fulda.de](mailto:Anatol.Badach@informatik.hs-fulda.de)

Auch stellt er Ihnen die Abbildungen gerne für Lehrzwecke zur Verfügung.

## 12.2 Funktionsweise des P2P-Overlay-Netzes

Ein Overlay-Ringnetz ermöglicht die Kommunikation zwischen jeweils zwei Peers durch die Übermittlung der hierfür notwendigen Signalisierung (SIG) zum Aufbau einer virtuellen Verbindung zwischen zwei IP-Telefonen. Abbildung 12.2-1a veranschaulicht dies am Beispiel der Verbindung zwischen den IP-Telefonen X und Y. Bei der Übermittlung der Signalisierung zwischen diesen IP-Telefonen sind nur die Peers 3 und 6 als Knoten auf dem logischen Ringnetz beteiligt. Diese Peers können an verschiedenen Stellen im Internet installiert werden. Anhand des hier gezeigten Beispiels soll insbesondere zum Ausdruck gebracht werden, dass die Übermittlung der Signalisierung dank einer sog. Finger-Tabelle (siehe Abb. 12.2-4) nur zwischen einigen Knoten des logischen Overlay-Ringnetzes verläuft. Die Übermittlung von Sprache – als Benutzer-zu-Benutzer-Kommunikation – verläuft „außerhalb“ des Overlay-Ringnetzes.

Overlay-Netz  
als virtueller  
Server



**Abb. 12.2-1:** Overlay-Ringnetz als virtueller Server für VoIP:  
a) Logisches Overlay-Netzwerk als Ringnetz, b) Virtueller Server  
SIG: Signalisierung, Tel: (IP-)Telefon

Vergleicht man die traditionelle Client-Server-Architektur (hierfür siehe Abb. 12.1-1) mit der in Abbildung 12.2-1a gezeigten logischen Vernetzungsstruktur, so könnte das ganze Overlay-Ringnetz – der Funktion nach – quasi als verteilter Server angesehen werden. Demzufolge ist das Overlay-Ringnetz als *virtueller (scheinbarer) Server* zu verstehen. Abbildung 12.2-1b illustriert diese Sichtweise.

Overlay-  
Ringnetz  
als virtueller  
Server

### 12.2.1 P2P-Overlay-Netz als Ringnetz

#### Overlay- Ringnetz als P2P-Chord

Im Weiteren wird nur auf Overlay-Netze mit einer Ringstruktur eingegangen. Diese Struktur ist sehr flexibel und alles deutet darauf hin, dass sie in allen zukünftigen Systemen eingesetzt werden wird. Die P2P-Netzarchitektur mit einem Overlay-Ringnetz ist dezentralisiert und so strukturiert, dass sie sich sehr gut zur Unterstützung der P2P-Kommunikation eignet. Die P2P-Netzarchitektur mit einem Overlay-Ringnetz wird oft als *P2P-Chord* bezeichnet. Abbildung 12.2-2 erläutert ihre Organisation.

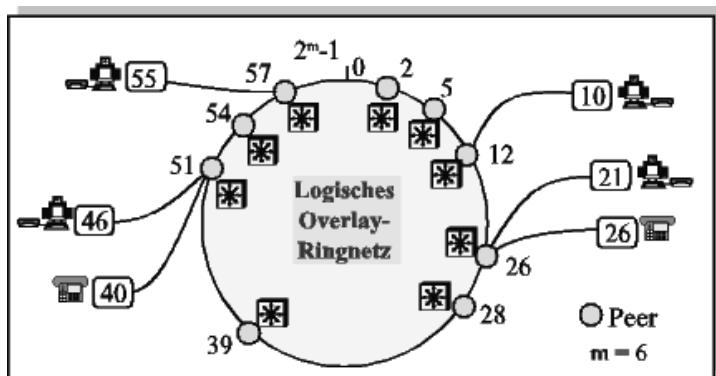


Abb. 12.2-2: Prinzip des Aufbaus eines Overlay-Ringnetzes mit maximal  $2^{m-1}$  ( $m = 5$ ) Knoten

#### ID als Wert einer Hash- Funktion

Jeder Rechner – als Peer bzw. Client – muss über eine eindeutige Identifikation (ID) verfügen. Diese Identifikation wird als Wert einer Hash-Funktion F berechnet. Als Argument dieser Funktion kann die IP-Adresse des Rechners genommen werden. Die Identifikation (ID) wird daher wie folgt bestimmt:

$$ID = F(\text{IP-Adresse})$$

Bei P2P können u.a. folgende Hash-Funktionen genutzt werden:

- *Message Digest (MD)*: MD4 oder MD5
- *Secure Hash Algorithm (SHA)*: SHA-1, SHA-256 oder SHA-512

**Bemerkung:** Jede Hash-Funktion hat folgende zwei Eigenschaften, die bei der P2P-Kommunikation von fundamentaler Bedeutung sind:

- *Sie ist kollisionsfrei*. Dies bedeutet, dass es keine zwei Argumente gibt, die zu dem gleichen Wert der Hash-Funktion führen. Nutzt man die IP-Adressen von Peers bzw. von Clients als Argumente einer Hash-Funktion, um ihre Identifikationen (IDs) zu bestimmen, kann man somit sicher sein, dass diese Identifikationen eindeutig sind.
- *Binäre Werte einer Hash-Funktion haben immer die gleiche Länge*. Sind die Argumente der Hash-Funktion – als binäre Zahlen – unterschiedlicher Länge, haben die binären Werte der Hash-Funktion aus diesen Argumenten immer die gleiche Länge – z.B. m. Daher können mit einem Overlay-Ring maximal  $2^{m-1}$  Peers logisch vernetzt werden.

Die Peers auf dem Overlay-Ring werden – *so wie Stunden auf dem Ziffernblatt einer Uhr* – fortlaufend nummeriert. Als Nummer jedes Peers auf dem Overlay-Ring wird seine ID (Identifikation) angenommen. Daher kann hier von einem *Uhrmodell* der Overlay-Ringnetze gesprochen werden. Da der Wert einer Hash-Funktion – als binäre Zahl – eine feste Länge hat, ist die Anzahl von Peers auf dem Ring begrenzt. Der Overlay-Ring in Abbildung 12.2-2 kann maximal  $2^{m-1}$  ( $m = 6$ ) Peers als Knoten beinhalten.

*Uhrmodell der Overlay-Ringnetze*

Ein Client kann sich an ein Overlay-Netz – z.B. im in Abbildung 12.2-2 gezeigten Beispiel an einen Overlay-Ring – nach bestimmten Regeln an einen Peer anschließen. Dieser Peer wird als sein *Successor* bezeichnet. Als Successor des Clients mit  $ID = k$  dient der erste Peer auf dem Ring, dessen ID gleich nach  $k$  folgt bzw. gleich  $k$  ist. Wie Abbildung 12.2-2 zeigt, werden beispielsweise der Client mit  $ID = 10$  an den Peer mit  $ID = 12$  und die Clients mit  $ID = 21$  und  $ID = 26$  an den Peer mit  $ID = 26$  angeschlossen. Benutzt man den Begriff Successor, können die Anschluss-Peers von Clients mit IDs 10, 21 und 26 wie folgt spezifiziert werden:

*Begriff: Successor eines Clients*

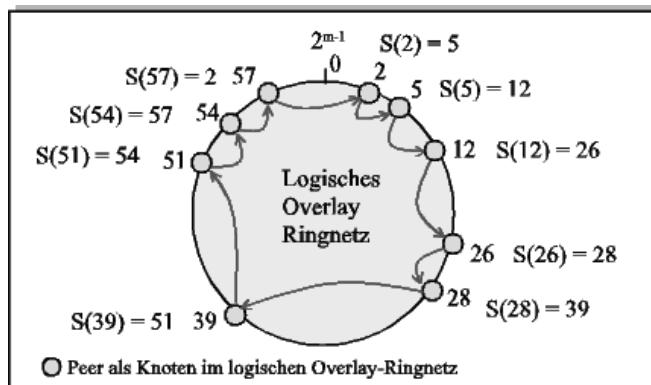
$$\text{Successor}(10) = 12, \text{Successor}(21) = 26, \text{Successor}(26) = 26$$

Das Prinzip, nach dem der Successor eines Clients bestimmt wird, ermöglicht eine eindeutige Ermittlung, an welchem Peer ein Client angebunden ist, vorausgesetzt, man kennt seine ID. Da diese ID aus der IP-Adresse des Clients durch eine Hash-Operation entsteht, ermöglicht dieses Prinzip eine eindeutige Ermittlung des Peers für den Anschluss des Clients, falls nur seine IP-Adresse bekannt ist. *Diese Tatsache ist beim P2P-Networking von zentraler Bedeutung.*

*Ermittlung: Woran ist der Client angeschlossen?*

Ein Overlay-Ringnetz funktioniert nämlich nur dann korrekt, wenn jeder Peer auf dem Ring seinen Nachfolger – will heißen seinen Successor – kennt. Abbildung 12.2-3 bringt diese Voraussetzung für eine korrekte Funktionsweise eines Overlay-Ringnetzes näher zum Ausdruck.

*Peer muss seinen Successor kennen.*



**Abb. 12.2-3:** In einem Overlay-Ring muss jeder Peer seinen Successor kennen.  
S: Successor

## *Redundante Wege im Overlay- Ring*

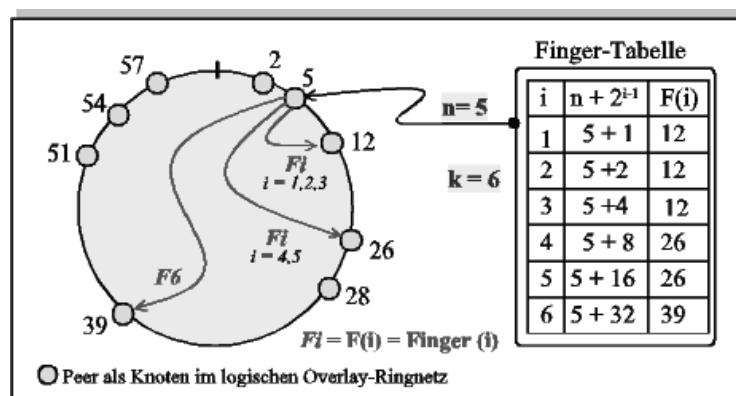
Dadurch, dass jeder Peer seinen Successor kennt, entsteht quasi eine geschlossene logische Schleife – das heißt ein Ring. Dies birgt jedoch ein Risiko in sich. Fällt ein Peer plötzlich aus, so gibt es auch keinen Ring mehr. Umgehen lässt sich dieses Problem mittels der Einführung *redundanter Wege im Overlay-Ringnetz*. Die Einführung und Erfassung der redundanten Wege erfolgen mit Hilfe sog. *Finger-Tabellen* und diese können als Kern des P2P-Networking auf der Basis von Overlay-Ringnetzen verstanden werden.

## *Einsatz von Finger- Tabellen*

Zwecks der Möglichkeit einer Einführung redundanter Wege im Overlay-Ringnetz enthält jeder Peer also eine sog. *Finger-Tabelle*. Jedem Peer werden mehrere *Finger* zugeordnet. Abbildung 12.2-4 zeigt eine Finger-Tabelle und veranschaulicht dabei ihre Bedeutung. Wie hier zum Ausdruck gebracht wurde, dient der *Finger* eines Peers als Verweis auf seinen *Successor*. Dadurch können einem Peer mehrere Successors zugeordnet werden.

## 12.2.2 Bedeutung von Finger-Tabellen

Die Tatsache, dass ein Peer auf dem logischen Overlay-Ringnetz dank seiner Finger-Tabelle mehrere Successors hat, ist in P2P-Netzen von enormer Bedeutung. Aber die Finger-Tabelle hat noch einen weiteren positiven Aspekt, den man beim Routing im Overlay-Netz nutzt. Auf diesen wird in Abschnitt 12.2.4 (siehe Abb. 12.2-6) eingegangen.



**Abb. 12.2-4:** Aufbau einer Finger-Tabelle und ihre Bedeutung

Im Ring mit  $2^{k-1}$  Peers können dem Peer mit ID = n mittels seiner Finger (F) folgende mögliche Successors auf dem Overlay-Ringnetz zugeordnet werden:

Finger (i) = Successor( $n+2^{i-1}$ ), wobei  $i = 1, 2, \dots, k$

Fällt z.B. der Peer mit  $ID = 12$  aus, so kann der Peer mit  $ID = 5$  den Peer mit  $ID = 26$  als seinen Successor nutzen. Folglich hat der Ausfall eines Peers

keine Auswirkung auf die weitere Funktionsweise des Overlay-Ringnetzes. Der Parameter  $k$  bestimmt die maximale Anzahl möglicher Successors auf dem Overlay-Ringnetz.

Die Vernetzungsstruktur innerhalb des Overlay-Ringnetzes und die Reihenfolge der Anbindung von Clients auf diesem können durch ihre, als Werte einer ausgewählten Hash-Funktion berechneten, Identifikationen in Form einer Tabelle beschrieben werden. Da die Inhalte dieser Tabelle verteilt und ihre Teile in einzelnen Peers und Clients abgespeichert sind, wird sie *Distributed Hash Table* (DHT) genannt. Die Abkürzung DHT ist bei P2P-Networking bereits üblich.

DHT als  
Beschreibung  
des Overlay-  
Ringnetzes

### 12.2.3 Beitritt eines Peers zum Overlay-Ringnetz

Das Overlay-Ringnetz kann als eine „offene Gruppe“ von als Peers bezeichneten Rechnern betrachtet werden. Daher kann jederzeit ein neuer Peer hinzukommen. Man spricht in diesem Zusammenhang von einer *Join-* bzw. *Beitrittsphase*. Abbildung 12.2-5 illustriert ihren Verlauf.

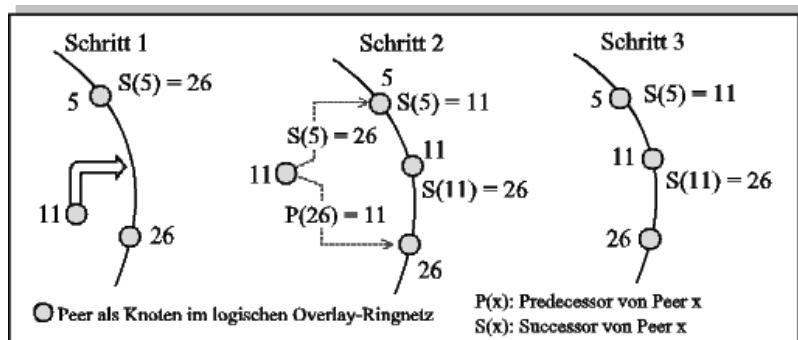


Abb. 12.2-5: Beispiel für den Verlauf einer Beitrittsphase

Während der Join-Phase sind folgende Schritte zu unterscheiden:

1. *Ermittlung des Successors*

Ein Rechner – hier mit  $ID = 11$  – möchte sich als Peer einem Overlay-Ringnetz anschließen. Hierfür muss er seinen Successor finden. Der Successor des Peers mit  $ID = 11$  ist der Peer mit  $ID = 26$ . Für die Ermittlung des Successors sind spezielle Nachrichten erforderlich. Diese werden beispielsweise im Protokoll RELOAD (siehe RFC 6940) definiert.

2. *Benachrichtigung des Vorgängers auf dem Overlay-Ringnetz*

Der neue Peer mit  $ID = 11$  muss dem Peer mit  $ID = 5$  – d.h. seinem Vorgänger (Predecessor) auf dem Overlay-Ringnetz – mitteilen, dass er

nun einen anderen Vorgänger (Successor) hat, denn der neue Peer ist jetzt sein Vorgänger.

### 3. Neue Konfiguration auf dem Overlay-Ringnetz

Nachdem der neue Peer bei sich seinen Successor eingetragen und dem Vorgänger auf dem Ring mitgeteilt hat, dass dieser ab jetzt als sein Successor fungiert, ist eine neue stabile Konfiguration auf dem Overlay-Ringnetz gegeben.

#### 12.2.4 Routing im Overlay-Ringnetz

Ein Overlay-Ringnetz enthält einerseits Knoten als Peers, d.h., jeder Peer kann mit jedem anderen Peer im gleichen Overlay-Ringnetz direkt kommunizieren. Andererseits kann das Overlay-Ringnetz der Funktion nach als verteilter – also als eine Art virtueller – Kommunikationsserver (siehe Abb. 12.2-1) verstanden werden. Dies wurde in Abbildung 12.2-1 verdeutlicht. Jeder Peer dient allen an ihn angeschlossenen Clients sozusagen als lokaler Kommunikationsserver. Abbildung 12.2-6 bringt dies zum Ausdruck.

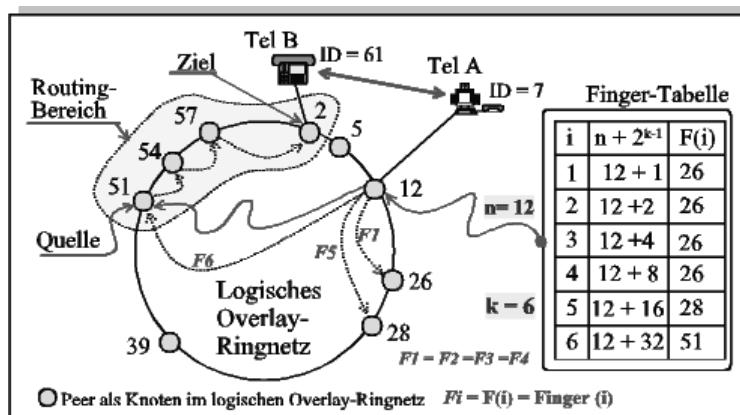


Abb. 12.2-6: Prinzip der Realisierung von Routing in einem Overlay-Ringnetz

Zur Ermöglichung der Kommunikation zwischen zwei an verschiedenen Peers auf dem Overlay-Ringnetz angebundenen Clients müssen diese Peers entsprechende Informationen austauschen – also direkt miteinander kommunizieren. Dies führt jedoch zu einem Routing-Problem, welches im Folgenden mit Hilfe der Abbildung 12.2-6 erläutert wird.

In dieser Abbildung soll beispielsweise mit Hilfe des Signalisierungsprotokolls SIP eine Verbindung zwischen den IP-Telefonen A und B aufgebaut werden. Die Telefone sind als Clients an die Peers 12 und 2 angebunden. Die Verbin-

dung wird hier vom Telefon A mit  $ID = 7$  initiiert. Das Telefon A übermittelt an den Quell-Peer mit  $ID = 12$  die SIP-Nachricht `INVITE` mit der Angabe von SIP-URI des Telefons B. Aus SIP-URI ermittelt der Quell-Peer zuerst – mit Hilfe von DNS – die IP-Adresse von Telefon B. Dann berechnet er anhand der angewandten Hash-Funktion die Identifikation  $ID = 61$  des Telefons B.<sup>2</sup>

Der Quell-Peer kennt nun den Ziel-Client. Genauer gesagt weiß er nur, dass dessen  $ID = 61$  ist und muss den Ziel-Peer ermitteln, an den der Ziel-Client (Tel B) angebunden ist und über den die Verbindung zum Ziel-Client aufgebaut werden kann. Um den Ziel-Peer – d.h. den Peer, an den der Ziel-Client angebunden ist – ermitteln zu können, muss ein Routing-Vorgang durchgeführt werden.

Der Quell-Peer stellt aufgrund der ID des Ziel-Clients und seiner Finger-Tabelle fest, dass der Ziel-Peer ein Peer mit einer ID zwischen 51 und 5 sein muss (siehe Abb. 12.2-3). Daher ist der Ziel-Peer in diesem ID-Bereich zu ermitteln. Dies geschieht mit Hilfe eines Routing-Prozesses. Sobald der Ziel-Peer mit  $ID = 2$ , an den der Ziel-Client angeschlossen ist, ermittelt wurde, kann die Verbindung zwischen den Clients – hier den zwei IP-Telefonen – aufgebaut werden. Abbildung 12.2-7 illustriert diesen Vorgang näher.

*Routing-Notwendigkeit*

*Quell-Peers als lokale Kommunikationsserver*

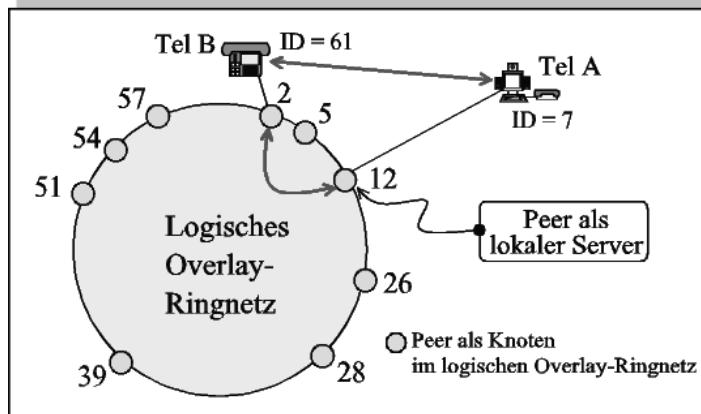


Abb. 12.2-7: Peers im Overlay-Ringnetz als lokale Kommunikationsserver

Aus dem hier dargestellten Beispiel geht hervor, dass das logische Overlay-Ringnetz als ein verteiltes System von Peers, einer Art Kommunikationsservern, dient. Mit zwei Peers wird die Kommunikation zwischen zwei Clients ermöglicht. Der Quell-Peer, der die Kommunikation initiiert, ist bekannt. Der

<sup>2</sup> Hierbei wurde die Nutzung von Hashwerten als Identifikationen stillschweigend vorausgesetzt. Diese Identifikationen von Peers und Clients werden aus ihren IP-Adressen als Hashwerte ermittelt.

Ziel-Peer, an den der andere Client angeschlossen ist, muss bestimmt werden, was zur Realisierung von *Routing im Overlay-Ringnetz* führt.

#### Routing-Arten

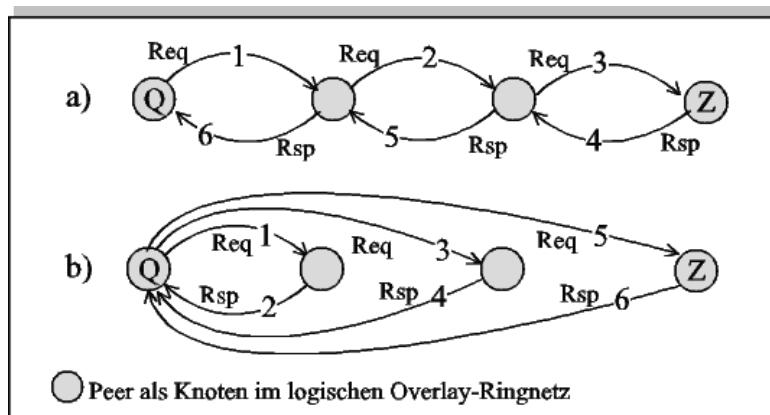
Im Allgemeinen ist zwischen den folgenden Arten von Routing zu unterscheiden:

- *rekursives Routing* und
- *iteratives Routing*.

Abbildung 12.2-8 zeigt die Unterschiede zwischen diesen Arten von Routing. Es sei angemerkt, dass das hier betrachtete Routing in einem *Overlay-Ringnetz* bei P2P-Kommunikation nur zur Ermittlung des Ziel-Peers führt.

#### Rekursives Routing

Wie aus der Abbildung 12.2-8a hervorgeht, leitet beim rekursiven Routing jeder Peer unterwegs zum Ziel-Peer die Nachricht des Quell-Peers – als *Request* (Req) – weiter. Hat dieser Request den Ziel-Peer erreicht, so sendet er eine Antwort – d.h. *Response* (Rsp) – an den Quell-Peer zurück. Diese wird an den Quell-Peer über die gleichen „Zwischen-Peers“ übermittelt.



**Abb. 12.2-8:** Routing-Arten bei P2P: a) rekursives Routing, b) iteratives Routing  
Q: Quell-Peer, Z: Ziel-Peer, Req: Request, Rsp: Response

#### Iteratives Routing

Wie die Abbildung 12.2-8b zeigt, gibt ein Peer beim iterativen Routing in der Response, die er an den Quell-Peer auf dessen Request zurücksendet, seinen Successor an. Im nächsten Schritt sendet der Quell-Peer einen Request (Req) an diesen Successor – an den nächsten Peer – und erhält von diesem eine Response (Rsp).

Die in Abbildung 12.2-8 gezeigten Routing-Arten entsprechen weitgehend den Betriebsarten von SIP-Servern. Das heißt:

- das rekursive Routing entspricht dem Einsatz von *Proxy-Servern* (siehe Abschnitt 7.1.4) und

- das iterative Routing entspricht dem Einsatz von *Redirect-Servern* (siehe Abschnitt 7.2.2).

## 12.3 Ziele und Bedeutung des P2PSIP

Eine Auflistung der wichtigsten mittels des Protokolls P2PSIP (*Peer-to-Peer SIP*) umzusetzenden Ziele zeigt Abbildung 12.3-1.

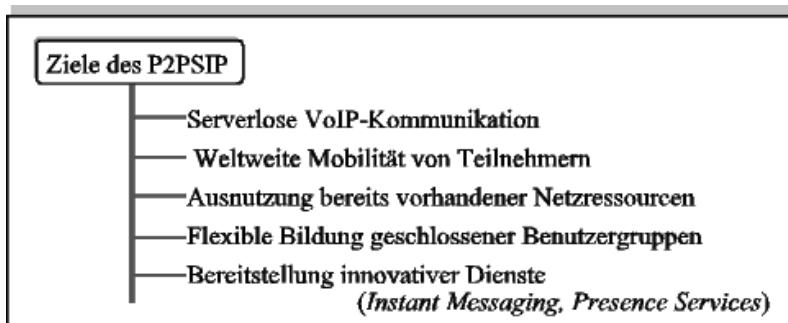


Abb. 12.3-1: Wichtigste Ziele des P2PSIP

Ein wesentliches Ziel bei der Entwicklung von P2PSIP war die Schaffung einer Netzarchitektur, mit deren Hilfe eine Gestaltung der Kommunikation zwischen allen Rechnern ohne den Einsatz von Servern – wie z.B. SIP-Proxies – möglich ist. Bei der serverlosen VoIP-Kommunikation mit P2PSIP entfällt also sowohl die aufwendige Konfiguration als auch die Verwaltung verschiedener Server. In diesem Zusammenhang ist insbesondere die Bildung serverloser IP-TK-Anlagen – auch IP-PBX (*Private Branch eXchange*) genannt – hervorzuheben.

Serverlose  
VoIP-  
Kommuni-  
kation

Ein anderes wichtiges Ziel bei der Entwicklung von P2PSIP war die Gewährleistung einer weltweiten Benutzermobilität. Dies wird bei P2PSIP dadurch erreicht, dass das logische P2P-Overlay-Ringnetz ein selbstorganisierendes Netz ist – also ein sog. *Ad-hoc-Netz*, in dem jeder Rechner mit jedem anderen spontan kommunizieren kann. P2PSIP ermöglicht so die VoIP-Kommunikation in mobilen Ad-hoc-Netzen. Als Beispiel für ein mobiles Ad-hoc-Netz kann eine Vernetzung von Autos – also ein *Car-to-Car-Network* – dienen.

Weltweite  
Mobilität  
von Nutzern

Für die Bildung eines P2P-Overlay-Ringnetzes zur Unterstützung der VoIP-Kommunikation – und somit auch zur Realisierung anderer Formen der Echtzeitkommunikation über IP-Netze – können bereits vorhandene Netzressourcen weitgehend weiterverwendet werden. Die Schaffung dieser Möglichkeit war ein wichtiges Anliegen bei der Konzeption von P2PSIP.

Ausnutzung  
vorhandener  
Netz-  
ressourcen

# 14 WebRTC – Konzept und Einsatz

WebRTC (*Web Real-Time Communication*) ist eine richtungsweisende Idee, die eine Realisierung multimedialer Echtzeitkommunikation mit Hilfe von Webbrowsern – also eine *Web-Echtzeitkommunikation* – ermöglicht, für die keine besonders komplexen zusätzlichen Software-Module installiert werden müssen. Zur Unterstützung von WebRTC seitens eines Webbrowsers können bei Bedarf von einem Webserver verschiedene RTC-spezifische Funktionsmodule heruntergeladen und in den Webbrowser direkt, quasi automatisch, „eingebaut“ werden. Welche Module hierfür nötig sind, wird später näher erläutert.

Betrachtet man WebRTC rein vom technischen Standpunkt her, so stellt man fest, dass weitgehend das Konzept von VoIP übernommen und um zusätzliche Funktionen erweitert wurde. Daher könnte man einige Systemlösungen für WebRTC auch als *Extended VoIP over Web* bezeichnen. Dies wird im Weiteren zum Ausdruck gebracht und dabei gezeigt, dass WebRTC auch als eine besondere Weiterentwicklung von VoIP angesehen werden kann. Demzufolge wird die Integration von WebRTC in bestehende VoIP-Systeme, vor allem im Hinblick auf eine breite Verwendung VoIP-fähiger Smartphones und deren Einsatz zur Nutzung webbasierter Internet-Dienste, von großer Bedeutung sein.

Das Ziel dieses Kapitels ist es, eine Übersicht über die allgemeinen Ideen, Konzepte und Anwendungsmöglichkeiten von WebRTC zu geben. Nach der Darstellung funktionaler Komponenten in Abschnitt 14.1 wird in Abschnitt 14.2 ein Kommunikationsmodell präsentiert. Welche Schritte vor und nach der WebRTC-Nutzung nötig sind und wie Sessions zwischen Clients verlaufen, zeigen die Abschnitte 14.3 und 14.4. Die Bedeutung von ENUM (*(t)Elephone Number URI Mapping*) zeigt Abschnitt 14.5. Der Einsatz von SIP und die Kopplung von WebRTC mit VoIP-Systemen wird in den Abschnitten 14.6 und 14.7 präsentiert. Auf Sicherheitsaspekte und die Standardisierung von WebRTC gehen die Abschnitte 14.8 und 14.9 ein.

Dieses Kapitel geht u.a. auf folgende Fragestellungen ein:

- Welche Kommunikationsmöglichkeiten sind bei WebRTC denkbar?
- Welche Ideen liegen WebRTC-basierten Homeoffices zugrunde?
- Wie können Kommunikationssysteme mit WebRTC aufgebaut werden?
- Welche Schritte sind beim Verlauf von WebRTC-Anwendungen nötig?
- Welche Bedeutung hat das VoIP-Protokoll SIP bei WebRTC?
- Wie kann WebRTC in VoIP-Systeme integriert werden?
- Welche Sicherheitsprobleme sind bei WebRTC relevant?

Multimediale  
Echtzeit-  
kommuni-  
kation mit  
WebRTC

WebRTC als  
Extended  
VoIP over  
Web

Überblick  
über das  
Kapitel

Ziel dieses  
Kapitels

## 14.1 Funktionale Komponenten von WebRTC

### WebRTC als Basis für Homeoffices

Mit WebRTC wird eine Idee zur Integration multimedialer Kommunikation, insbesondere von VoIP, mit Webanwendungen realisiert. Diese Integration kann als die wichtigste Besonderheit von WebRTC angesehen werden. Die WebRTC-Idee liegt allen Systemlösungen zugrunde, die als technische Basis für die Einrichtung von Homeoffices anzusehen sind. Von welcher, fast existenzieller, Bedeutung Homeoffices sein können, hat sich während der COVID-19-Pandemie gezeigt.

### 14.1.1 Webbrowser mit WebRTC-Unterstützung

#### RTC-fähiger Webbrowser

Bei WebRTC kann ein Webbrowser – auch kurz *Browser* genannt – nicht nur als Software-Telefon (Softphone), sondern auch als multimediale Kommunikationsinstanz zur Unterstützung von Sprach-, Video- und Datenkommunikation zwischen über das Internet kommunizierenden Personen dienen – d.h. als eine Art *Web-Videotelefon* (Web Video Phone) genutzt werden. Damit ein Webbrowser diese Funktionen bereitstellen kann, muss er um eine RTC-spezifische Software erweitert werden, also RTC-fähig sein. Abbildung 14.1-1 illustriert dies näher und zeigt, welche Komponenten nötig sind, um webspezifische Internet-Anwendungen mit *Multimedia over IP* (MoIP) integrieren zu können.

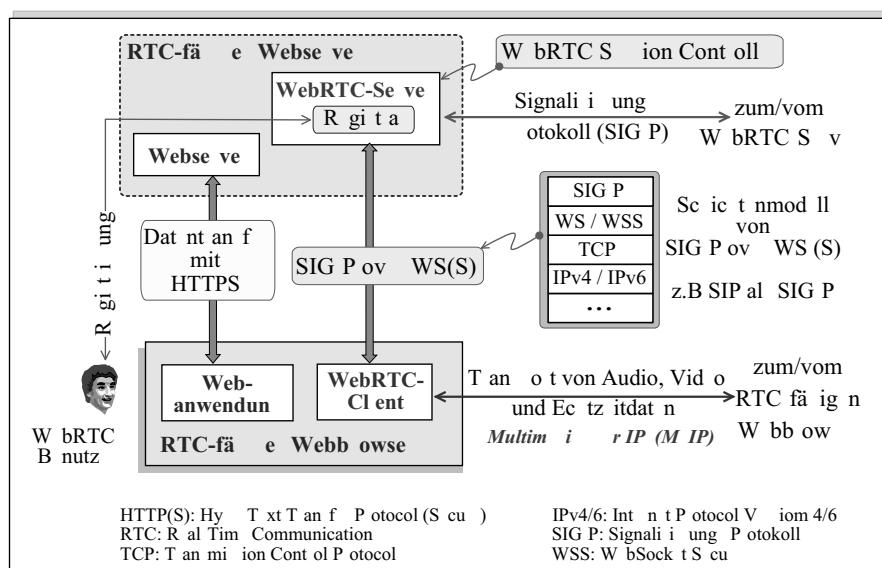


Abb. 14.1-1: Vereinfachtes Modell der Kommunikation zwischen WebRTC-Client und -Server

Sollte der Webbrowser in einem Rechner noch nicht RTC-fähig sein, so wird der Internetnutzer in die Lage versetzt, unter Verwendung eines Links auf dem Webbrowser zu veranlassen, dass eine RTC-spezifische Software mit Hilfe des Protokolls HTTPS (*HyperText Transfer Protocol Secure*) von einem Webserver heruntergeladen und in den Webbrowser automatisch integriert wird. Auf diese einfache Weise kann der Webbrowser RTC-fähig gemacht werden und als eine Art webfähiges Videotelefon – kurz Web-Videotelefon – dienen.

In allen privaten Systemen, welche Telefonie ermöglichen, werden für die Dauer der Kommunikation (des Telefonats) Telefonapparate als Endgeräte miteinander unter Mitwirkung zentraler Vermittlungsknoten verbunden. Als Vermittlungsknoten dient bei VoIP mit dem Protokoll SIP (*Session Initiation Protocol*) in einer Organisation (Unternehmen, Institution ...) ein spezieller SIP-Proxy-Server (s. Abb.7.2-1). Bei WebRTC wird ein funktionell ähnlicher Server, im Weiteren *WebRTC-Server* genannt, benötigt, der als Vermittlungsknoten zwischen Web-Videotelefonen fungiert. Eine RTC-spezifische, die Funktion eines Web-Videotelefons erbringende Software-Instanz im Webbrowser kann somit als Client des WebRTC-Servers angesehen werden. Aus diesem Grund wird diese in den Webbrowser integrierte Software-Instanz hier als *WebRTC-Client* bezeichnet.

Die grundlegende Idee von WebRTC (s. auch die Abbildungen 14.1-1, 14.2-2, 14.3-1 und 14.4-1) besteht darin, dass ein WebRTC-Server quasi als Manager multimedialer Verbindungen (Sessions) zwischen WebRTC-Clients fungiert. Über einen WebRTC-Server können zahlreiche, als Web-Videotelefone dienende WebRTC-Clients logisch/virtuell so miteinander verbunden werden, dass sie untereinander über das Internet in IP-Paketen Sprache, Video und Daten übermitteln können. Jeder WebRTC-Server dient somit als WebRTC Session Controller.

Ein WebRTC-Server kann die Kommunikation zwischen Benutzern nur dann ermöglichen, wenn sie zur Initiierung der Kommunikation berechtigt und dem WebRTC-Server ihre WebRTC-spezifischen Adressen bekannt sind. Hierfür müssen die Benutzer beim WebRTC-Server entsprechend registriert werden. Folglich muss im WebRTC-Server eine spezielle – oft als *Registrar* bezeichnete – Funktion enthalten sein bzw. eine solche Funktion an ihn angebunden werden.

Um welche RTC-Funktionen Webbrowser und Webserver erweitert werden müssen, wird aus den im Weiteren gezeigten Systemlösungen ersichtlich. Eine Grundlage für WebRTC bildet z.B. das im RFC 6455 spezifizierte *WebSocket Protocol* (WS-Protokoll bzw. kurz WSP oder WS). Dieses wird im Weiteren detaillierter erläutert.

*WebRTC-Server als Vermittlungsknoten*

*Idee von WebRTC*

*Registrar-Funktion im WebRTC-Server*

*Bedeutung von WebSocket Protocol*

### 14.1.2 WebRTC-Server und WebSocket-Protokoll

*Wo wird die Funktion vom WebRTC-Server erbracht?*

Da das WS-Protokoll normalerweise zwischen Webbrowsern und Webserver zum Einsatz kommt, wird in diversen Publikationen zum Thema WebRTC erläutert, dass die Funktion des WebRTC-Servers von einem Webserver erbracht wird. Aus der Sicht eines RTC-fähigen Webbrowsers kann der WebRTC-Server als eine funktionelle Erweiterung eines Webservers angesehen werden. Für einen RTC-fähigen Webbrowser bildet ein aus einem Webserver und einem WebRTC-Server bestehendes Paar quasi einen RTC-fähigen Webserver. In diesem Kapitel wird aber keine Festlegung getroffen, wo und wie die Funktion des WebRTC-Servers erbracht wird – also ob in einem physischen Webserver oder in einem anderen dedizierten Server, der das WebSocket-Protokoll unterstützt.

*Bedeutung des Web-Socket-Protokolls*

Klassischen Webanwendungen liegt eine request/response-spezifische Kommunikation zwischen Webbrowser und Webserver zugrunde. Diese besteht darin, dass der Webbrowser – immer zuerst – einen Request an den Webserver schickt und dieser dem Webbrowser darauf mit einer Response antwortet. Dies bedeutet also, dass in klassischen Webanwendungen die Kommunikation zum Webserver immer seitens des Webbrowsers initiiert wird. Der Webserver bei klassischen Webanwendungen war nicht dazu befähigt, seinerseits den Aufbau von Verbindungen zu Webbrowsern zu veranlassen.

*WebRTC-Server im Webserver*

WebRTC ist aber keine klassische Webanwendung mehr. Bei WebRTC muss jeder Webserver, der zusätzlich u.a. die Funktion eines WebRTC-Servers erbringt, in der Lage sein, eine Verbindung zum Webbrowser zu initiieren. Genauso betrachtet, muss jeder WebRTC-Server fähig sein, eine Verbindung zum WebRTC-Client im Webbrowser zu initiieren (vgl. Abb. 14.1-1). Ein wichtiges Ziel der Entwicklung des WS-Protokolls war es, die Webserver dazu zu befähigen. Dank der Nutzung des WS-Protokolls können Webserver daher diese – zusätzliche – Funktion von WebRTC-Servern erbringen und dadurch zur Unterstützung von WebRTC verwendet werden, als Verbindungsmanager dienen und u.a. den Aufbau multimedialer Verbindungen zu Webbrowsern initiieren.

**Bemerkung:** Theoretisch gesehen kann jeder Webserver so um RTC-Funktionen erweitert werden, dass er RTC-fähig wird und als Verbindungsmanager dienen kann. Soll ein „normaler“ Webserver aber eine große Anzahl von WebRTC-Benutzern unterstützen, also einer großen Anzahl von Webbrowsern ermöglichen, untereinander zu kommunizieren, dann ist es sinnvoller, zu diesem Zweck einen separaten Server mit Unterstützung des WS-Protokolls als dedizierten WebRTC-Server zu verwenden.

### 14.1.3 Signalisierungsprotokoll bei WebRTC

Bei WebRTC handelt es sich um eine multimediale Kommunikation – d.h. um eine Art Videotelefonie. Daher bedarf es eines Protokolls, welches nicht nur dazu dient, virtuelle Verbindungen zwischen WebRTC-Clients in Webbrowsersn auf- und abzubauen, sondern Benutzern ankommende Videotelefonieanrufe auch optisch anzuzeigen und bei Bedarf auch akustisch zu signalisieren; hierfür wird also ein Signalisierungsprotokoll benötigt. Als solches eignet sich besonders gut das Protokoll SIP. Es wurde hierfür bereits *SIP in JavaScript* (SIP-JS) implementiert – siehe <https://sipjs.com> für Näheres darüber.

*SIP als  
Signalisie-  
rungs-  
protokoll bei  
WebRTC*

Wie die Abbildungen 14.1-1 und 14.4-1 zeigen, wird SIP zwischen einem WebRTC-Client und einem -Server verwendet. Damit ein im Webserver zusätzlich installierter WebRTC-Server die zu einem WebRTC-Client im Webbrower führenden Verbindungen initiieren kann, muss zwischen Webserver und Webbrower das *WS-Protokoll* (WS: WebSocket) verwendet werden. Nach dem Aufbau einer WS-Verbindung zwischen Webserver und Webbrower (s. Abb. 14.3-1) werden über diese WS-Verbindung die SIP-Nachrichten in WS-Frames übermittelt. Dies kann als Realisierung von *SIP over WS* verstanden werden – siehe hierzu die Abbildung 14.4-1.

*SIP  
over WS*

Das WS-Protokoll nutzt das verbindungsorientierte Transportprotokoll TCP und kann zur Absicherung der Kommunikation zwischen WebRTC-Client und -Server auch das Sicherheitsprotokoll TLS (*Transport Layer Security*) nutzen. Ist dies der Fall, spricht man vom *Secure WebSocket Protocol* bzw. vom *WebSocket Security Protocol* (kurz *WSS Protocol*). Um Sicherheit bei der Übermittlung von SIP-Nachrichten zwischen WebRTC-Client und -Server zu garantieren, kann außerdem *SIP over WSS* eingesetzt werden.

*SIP  
over WSS*

Um die Realisierung WebRTC-basierter Systemlösungen in Netzwerken mit privaten IPv4-Adressen zu ermöglichen, wird das Protokoll ICE (*Interactive Connectivity Establishment*) verwendet. Wie bereits in Abschnitt 10.6. gezeigt, können mit dem Einsatz des Protokolls SIP bei der Nutzung privater IPv4-Adressen entstehende Probleme mit ICE-Hilfe gelöst werden.

*NAT und  
ICE bei  
WebRTC*

### 14.1.4 Arten der Kommunikation bei WebRTC

Es sei hervorgehoben, dass bei WebRTC alle Formen der Kommunikation zwischen Webbrowsersn realisiert werden können (s. Abb. 14.1-1) – und zwar:

- *Videotelefonie* – d.h. die gleichzeitige Sprach- und Videokommunikation über eine multimediale Session. Hier kommen die Echtzeittransportprotokolle RTP (*Real-time Transport Protocol*) und RTCP (*RTP Control Protocol*) zum Einsatz (s. Abschnitte 5.3 und 5.5). Für alle sicherheitsrelevanten Applikationen ist aber die Nutzung von SRTP (*Secure RTP*) erforderlich (s.

*Video-  
telefonie  
mit Hilfe  
von SRTP*

Abschnitt 5.7). Mit SRTP können die drei wichtigen Sicherheitsziele – Vertraulichkeit der Kommunikation, Authentifizierung von Nachrichten und Anti-Replay-Schutz – erreicht werden. Für weitere Anforderungen an die Echtzeitkommunikation bei WebRTC sei auf den Internetstandard RFC 8834 (*Media Transport and Use of RTP in WebRTC*) verwiesen.

Datenkommunikation mit Hilfe von SCTP

- *Datenkommunikation* über einen speziellen Datenkanal – als *Data Channel* bezeichnet. Über diesen Datenkanal können mit Hilfe des Protokolls SCTP (*Stream Control Transport Protocol*) mehrere Datenströme in beide Richtungen übermittelt werden. Um die Sicherheit des Datentransports zu gewährleisten, soll hier das Sicherheitsprotokoll DTLS (*Datagram Transport Layer Security*) zum Einsatz kommen; also *STCP over DTLS* realisiert werden. Für weitere Details darüber sei verwiesen auf die Internetstandards RFC 8831, RFC 8832 und RFC 8835.

Damit Kommunikation zwischen Benutzern, die bei verschiedenen WebRTC-Servern registriert sind, stattfinden kann, müssen diese Server untereinander kommunizieren und dabei entsprechende Signalisierungsnachrichten übermitteln. Als Signalisierungsprotokoll zwischen ihnen wird mit Sicherheit SIP oder SIPS (*SIP Secure*) zum Einsatz kommen. Bei einigen Anwendungen kann auch XMPP (*Extensible Messaging and Presence Protocol*) oder die als *Jingle* bezeichnete XMPP Extension verwendet werden.<sup>1</sup>

## 14.2 Modell der Kommunikation bei WebRTC

Zur Ermöglichung der WebRTC-spezifischen multimedialen Kommunikation im Internet zwischen RTC-fähigen Webbrowern (s. Abb. 14.1-1) müssen zwischen ihnen besondere virtuelle Verbindungen eingerichtet werden. Hierfür müssen zwischen den Webbrowern spezielle, als *WebRTC-Signalisierung* bezeichnete Steuerungsvorgänge stattfinden. In folgendem Abschnitt wird die WebRTC-Signalisierung mittels eines Dreiecksmodells veranschaulicht.

Ähnlichkeit von Ideen  
VoIP mit SIP und WebRTC

### 14.2.1 Dreiecksmodell von VoIP mit SIP

Mit dem Ziel einer fundierten Erläuterung der Idee von WebRTC zeigt Abbildung 14.2-1 eine entsprechend an die WebRTC-Idee angepasste Darstellung des allgemeinen Konzeptes von VoIP mit SIP innerhalb der als Beispiel dienenden DNS-Domain (*Domain Name System*) abc.de. Für eine detaillierte Erläuterung des Konzeptes von VoIP mit SIP sei auf das Kapitel 7 verwiesen. Ba-

---

<sup>1</sup> Für detaillierte Informationen darüber siehe die Spezifikation „XEP-0166: Jingle“ unter der Adresse: <https://xmpp.org/extensions/xep-0166.html>

sierend auf der in Abbildung 14.2-1 dargestellten Idee von VoIP mit SIP wird anschließend in Abbildung 14.2-2 die Idee von WebRTC erklärt und dabei zum Ausdruck gebracht, dass es sich bei WebRTC de facto um ein dem VoIP mit SIP ähnliches Konzept handelt.

Wie aus der Abbildung 14.1-1 ersichtlich ist, wird bei VoIP mit SIP als Vermittlungsknoten zwischen als VoIP-Clients bezeichneten Telefonen ein spezieller VoIP-Server, oft auch VoIP-Proxy genannt, eingesetzt. Demzufolge verläuft das Protokoll SIP beim Aufbau einer Session – für Transport von Voice und Video nach den Protokollen RTP/RTCP (*Real-time Transport Protocol* / *RTP Control Protocol*) – zwischen zwei VoIP-Clients entlang eines Dreiecks. Deshalb kann in diesem Zusammenhang auch vom *Signalisierungsdreieck* bzw. *SIP-Dreieck* gesprochen werden.

*Dreiecksmodell für den Verlauf der Signalisierung*

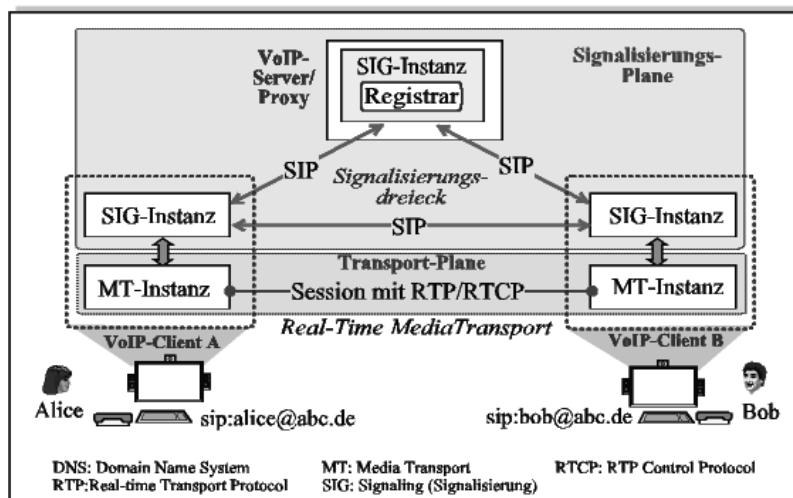


Abb. 14.2-1: Dreiecksmodell für den Verlauf der Signalisierung bei VoIP mit SIP – innerhalb einer DNS-Domäne

Im hier gezeigten Dreiecksmodell sind insbesondere folgende zwei Planes hervorzuheben:

- *Signalisierungs-Plane*, in der das Signalisierungsprotokoll SIP verläuft, um zwischen den MT-Instanzen in beiden VoIP-Clients eine Session mit den Protokollen RTP/RTCP für den Transport von Voice und Video einzurichten.
- *Transport-Plane*, innerhalb der die Echtzeitmedien Voice und Video über die zwischen den MT-Instanzen eingerichtete Session transportiert werden.

### SIP-Server als zentrale Signalisierungsinstanz

Eine neue, als Session bezeichnete virtuelle Verbindung wird bei VoIP mit SIP mit der SIP-Nachricht `INVITE` initiiert, in der die als SIP-URI bezeichnete Ziel-VoIP-Adresse `sip:bob@xyz.de` enthalten ist. In Abbildung 14.2-1 ist der als VoIP-Client A bezeichnete Rechner von Alice der Initiator einer Session zu Bob. Beim Initiiieren einer neuen Session wird `INVITE` an den SIP-Server übergeben. Dieser muss `INVITE` an das Ziel – hier zum VoIP-Client B – weiterleiten.

Hat `INVITE` den Rechner von Bob erreicht und die gewünschte Session kann zustande kommen, wird dies Bob akustisch durch Klingeln (Ringing) signalisiert und auch dem Rechner von Alice mittels der SIP-Nachricht `180 Ringing` mitgeteilt. Hat Bob den Anruf entgegengenommen, wird dies dem Rechner von Alice mittels der SIP-Nachricht `200 OK` angezeigt. Die beiden Nachrichten `180 Ringing` und `200 OK` werden über den SIP-Server übermittelt. Nach dem Empfang dieser Nachrichten kennt der Rechner von Alice bereits die IP-Adresse von Bobs Rechner und sendet daher die SIP-Nachricht `ACK` als Bestätigung des Empfangs von `200 OK` direkt an diesen – ohne den VoIP-Server nutzen zu müssen. Dadurch entsteht ein SIP-Dreieck innerhalb der Signalisierungs-Plane.

Nach dem Eintreffen von `200 OK` bei Bobs Rechner wird der Aufbau der Session beendet, und die beiden Medien Voice und Video können mit Hilfe von RTP/RTCP in IP-Paketen zwischen MT-Instanzen über diese Session übermittelt werden.

### 14.2.2 WebRTC-Dreiecksmodell – ohne Transcoder-Einsatz

#### Ähnlichkeit zwischen Dreiecksmodellen von VoIP und von WebRTC

Auf der Grundlage des in Abbildung 14.1-2 gezeigten Dreiecksmodells der Signalisierung bei VoIP mit SIP kann die grundlegende Idee von WebRTC anhand eines ähnlichen Dreiecksmodells dargestellt werden. Abbildung 14.2-2 zeigt ein solches Dreiecksmodell von WebRTC und verdeutlicht dabei, dass die beiden Konzepte VoIP mit SIP und WebRTC sehr ähnlich sind. Es gibt aber auch einige Unterschiede, auf die im Folgenden näher eingegangen wird.

Bei WebRTC soll in jedem Rechner, beispielsweise durch Anklicken auf einen Link bzw. auf ein Icon, die Möglichkeit bestehen, mittels des Protokolls `HTTPS (HTTP Secure)` aus einem Webserver einen WebRTC-Client herunterzuladen und diesen in den Webbrower quasi automatisch zu integrieren.

Damit jeder Webserver auch als Vermittlungsknoten bei WebRTC, genauer gesagt als WebRTC-Server, dienen kann, wurde das verbindungsorientierte *WebSocket Protocol* (WS-Protokoll) zur Übermittlung von Signalisierungsnachrichten zwischen WebRTC-Client und -Server entwickelt. Das WS-Protokoll ist ein verbindungsorientiertes Protokoll oberhalb des Transportprotokolls TCP. Bei

WS kann auch das Protokoll TLS<sup>2</sup> (*Transport Layer Security*) verwendet werden; ist dies der Fall, so spricht man von *Secure WS* (WSS).

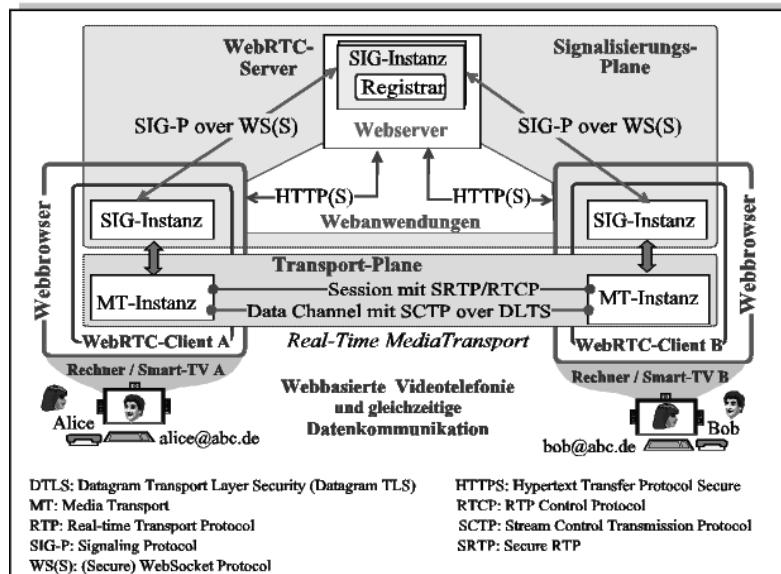


Abb. 14.2-2: Modell für den Verlauf der Signalisierung bei WebRTC – vergleiche dieses mit dem in Abb. 14.2-1 gezeigten Modell für VoIP mit SIP.

Mit Hilfe des WS-Protokolls ist der WebRTC-Server, welcher auf einem „normalen“ Webserver oder auf einem speziellen Server mit Unterstützung des WS-Protokolls installiert ist, in der Lage, eine Verbindung zum WebRTC-Client zu initiieren. Bei WebRTC verläuft das Signalisierungsprotokoll SIG-P (wie z.B. SIP) somit über WS oder WSS. Dies wird kurz als *SIG-P over WS(S)* bezeichnet. Die Nachrichten von SIG-P werden zwischen WebRTC-Client und -Server in WS-Frames (s. RFC 6455 – *The WebSocket Protocol*) transportiert:

- bei *SIG-P over WS* – über eine normale, also ungesicherte TCP-Verbindung,
- bei *SIG-P over WSS* – über eine mit Hilfe des Sicherheitsprotokolls TLS gesicherte TCP-Verbindung.

Da das WS-Protokoll nur zwischen WebRTC-Client und -Server verwendet wird, können die Nachrichten eines Signalisierungsprotokolls bei WebRTC nur über WebRTC-Server übermittelt werden und nicht, so wie es bei VoIP mit SIP der Fall war, auch direkt zwischen SIP-Instanzen in Webbrowsern. Demzufol-

*SIG-P over WS bzw. over WSS*

*Kein Signalisierungs-dreieck bei WebRTC*

<sup>2</sup> Für allgemeine Informationen über TLS sei verwiesen auf die Adresse:  
<https://www.researchgate.net/publication/292995131>

ge entsteht bei WebRTC, anders als bei VoIP, kein *Signalisierungsdreieck* – vergleiche hierfür die Abbildungen 14.2-1 und 14.2-2.

*Datenkanal  
bei WebRTC  
mit SCTP  
over DTLS*

Im Gegensatz zu VoIP mit SIP sind bei WebRTC die MT-Instanzen (*Media Transport*) in der Lage, untereinander einen gesicherten Datenkanal (*Data Channel*) zum Datentransport einzurichten. Um verteilte Datenanwendungen und verschiedene Arten multimedialer Kommunikation zu ermöglichen, kommt hierfür das Protokoll SCTP over DTLS zum Einsatz.

*Gemein-  
sames Socket  
für RTP und  
RTCP bei  
WebRTC*

Es sei hervorgehoben, dass es einen wesentlichen Unterschied zwischen einer multimedialen Session bei VoIP und einer multimedialen Session bei WebRTC gibt. Zwar werden die Protokolle RTP (*Real-time Transport Protocol*) und RTCP<sup>3</sup> (*RTP Control Protocol*) bei beiden Arten von Sessions eingesetzt, aber bei WebRTC ist es anders als bei VoIP. Im Unterschied zu VoIP wird bei WebRTC ein gemeinsames Socket für RTP und RTCP verwendet. Diese Lösung verfolgt das Ziel, die Nutzung privater IPv4-Adressen bei WebRTC zu erleichtern. Um private IPv4-Adressen bei WebRTC nutzen zu können, soll das Protokoll ICE zur Lösung des Problems NAT (*Network Address Translation*) zum Einsatz kommen (s. Abschnitt 10.6.7).

### 14.2.3 WebRTC-Dreiecksmodell – mit Transcoder-Einsatz

*Mehrere  
Audio- und  
Video-Codecs*

Die Verwirklichung der WebRTC-Idee soll Benutzern während einer Echtzeitkommunikation die Übermittlung aller Informationsformen – also von Audio und Video sowie Daten – ermöglichen. Aber damit sie erfolgreich kommunizieren und die gemeinsam genutzten Medien auch präsentieren können, müssen sich die kommunizierenden Webbrower auf einzusetzende Codecs, insbesondere Audio- und Video-Codecs, verstündigen.

Bei WebRTC soll die Anzahl der Audio- und Video-Codecs sehr eingeschränkt sein. Es werden die folgenden Codecs verbindlich vorgesehen:<sup>4</sup>

- für Audio der Codec nach dem ITU-T-Standard G.711 und der im RFC 6716 spezifizierte *Opus* Codec,
- für Video der Codec nach dem ITU-T-Standard H.264 und der lizenzgebührenfreie Codec VP8. Im Hinblick darauf sei verwiesen auf RFC 7741 mit der Spezifikation von „RTP Payload Format for VP8 Video“.

---

<sup>3</sup> Das Protokoll RTCP wird auch als *Real Time Control Protocol* bezeichnet.

<sup>4</sup> Für weitere Informationen über „Codecs used by WebRTC“ sei verwiesen auf die Adresse: [https://developer.mozilla.org/en-US/docs/Web/Media/Formats/WebRTC\\_codecs](https://developer.mozilla.org/en-US/docs/Web/Media/Formats/WebRTC_codecs)

# Index

180 Ringing 586  
181 Call Is Being Forwarded 304  
200 OK 578, 585  
202 Accepted 346  
301 Moved Permanently 462  
302 Moved Temporarily 298, 304, 397, 462, 475, 476  
3GPP 24, 25, 26, 38, 44, 57  
3PCC 9, 359  
468 Here Busy 346  
 $\mu$ -Kennlinie ( $\mu$ -Law) 158

## A

A-Kennlinie (*A-Law*) 156  
A-Query 100, 104, 275  
A-Record (A-RR) 100, 101, 283  
a=sendonly 339, 349  
Abbruch einer Session 472  
Abtasttheorem 149  
AC (*Authentication Center*) 22  
ACELP 121, 163  
Access Gateway 35  
ACK 94, 305  
ACM 59, 68, 76  
ACT 337  
Adaptive WFQ 141  
Address Resolution Protocol s. *ARP*  
Admission 241, 242  
ADPCM 150, 152, 163  
AES 201  
ALERT 66  
Algorithmic Delay 120

Allow (*SIP Header*) 312  
Alternate Gatekeeper 239  
analoger Sprachkanal 4  
Analyse des Schutzbedarfs 491  
Analysis-by-Synthesis 160  
Angriffsarten 490  
Angriffsklassen 462  
Anrufentföhrung 466  
Anruf-SIG-Kanal 230, 231, 233, 234, 235, 250  
Anrufweiterleitung 286  
Anti-Replay-Schutz 202, 210  
AoR (*Address of Record*) 332, 473, 583  
API 28, 37, 38  
ARP 78, 79, 85, 460, 498  
ARP-Spoofing 460, 498  
ASN.1 238, 251  
Assigned Gatekeeper 239  
Asterisk 445  
asymmetrisches Kryptosystem 481  
Attended CT 346, 348  
Authentifizierung 478, 479  
Authentizität 481, 489, 491, 495, 496, 497

## B

B2BUA 288  
Backdoor(programm) 455, 464, 466  
Bandbreite 4  
Basic Peering 392, 393  
Basisanschluss 61  
Basisstation 21

- Bedrohung(en) 487, 488, 489, 490  
Bedrohungsanalyse 486  
Bedrohungsmatrix 489  
Bedrohungstyp 489, 493  
Benutzer-Proxy-Authentifizierung 480  
BGP-4 116, 226, 385  
Bill&Keep (B&K) 392, 395  
Body-Teil 483, 485  
Border Element (BE) 226  
Boundary 486  
Branch 292  
BSC 21  
Burst 198  
BYE-Spoofing 470
- C**
- Call Agent (CA) 367  
Call Bombing 471  
Call Completion 260  
Call Control 227  
Call Control Plane 406, 407  
Call Controller 407  
Call Deflection 259  
Call Detail Record 291  
Call Diversion 259  
Call Features 334  
Call Flooding 471  
Call Forking 300  
Call Forwarding (CF) 259, 286, 336, 345  
Call Fraud 458, 461, 468  
Call Hold 259, 335, 340  
Call Hold/Retrieve 338  
Call-ID 291, 312, 315, 341, 342, 344  
Call Intrusion 260  
Call Park 260, 263, 336, 341  
Call Pickup 260, 263, 336, 344  
Call-Redirection Attack 472, 476  
Call Reference 65  
Call Session Control Function s. *CSCF*  
Call Splitting 300  
Call Transfer (CT) 259, 262, 306, 337, 346  
CANCEL 302, 345, 458, 466  
CANCEL-Spoofing 470  
Candidate 442, 443  
Capability Exchange 252  
CBQ 133, 135  
CBWFQ 134, 142  
CCBS 260, 338, 350  
CCITT 53  
CCNR 260, 38, 350  
CDN (Content Delivery Network) 600  
CELP 150, 163  
CFB 259, 303, 304, 337  
CFNA/R 337  
CFNR 03  
CFU 259, 303, 337  
Chunk 109  
Chunk DATA 110  
Class-based FQ 139, 140  
Click-to-Dial 9, 32, 355  
Click-to-Fax 32  
clock rate 179  
CNAME 101, 179, 185  
Computervirus 455  
Congestion Control 91  
CONN 67

- Consultation Hold 335, 340  
Contact (*SIP Header*) 291, 293, 312, 334, 462, 474, 475  
Content-Length (*SIP Header*) 291  
Content-Type (*SIP Header*) 484, 485  
COPS 148  
CORBA 42  
Core Network 21, 24  
CQ 133, 135, 138  
CPNP 392, 395  
CRTP 215, 216, 219, 222, 224, 225  
Cryptographic Context 214  
CSCF 45  
CS-Domain 23  
CSeq 312, 315  
CSRC 175, 183  
CT s. *Call Transfer*  
Custom Queueing 135
- D**
- D-Kanal 6, 61, 68, 230  
D-Kanal-Protokoll 6, 13, 18, 20, 57, 58, 63, 77, 105, 227, 233, 250, 355, 381  
Datagram Congestion Control Protocol s. *DCCP*  
Datagram TLS s. *DTLS*  
Datagramm 83  
Datagramm-Prinzip 126  
Data-Link-Adresse 85  
Data-Link-Frame 81  
DCCP 79, 88, 170, 173, 275  
DCPU 336  
Decomposed Peering 392  
Delay 116  
Denial of Service s. *DoS*  
Deregistrierung 240, 272  
DHCP 78, 80  
DHCP Starvation 460  
DHCP Spoofing 476  
Dialog-ID 291, 318  
Dialog-Information 344  
Diameter 46, 49  
Dienstgüte 116  
Differentiated Services 86, 115, 125, 127  
Diffie-Hellman 203, 204  
DiffServ s. *Differentiated Services*  
DiffServ-Domäne 130  
digitale Telefone 4  
digitaler Sprachkanal 5  
Digitalisierung von Sprache 11, 148  
DISC 67, 69  
DL-Frame 81  
DMZ 452  
DNS 15, 17, 18, 77, 80, 97, 111, 274, 277, 281, 291, 394, 396, 576  
DNS-Abfrage 103  
DNS-Domäne 18, 391, 590  
DNS-Poisoning 471  
DNS-Server 97, 279  
DNS-Spoofing 471, 475, 501  
Domain (Domäne) 97  
Domainname 282  
Domain Name System s. *DNS*  
DoS 107, 458, 459, 461, 466, 468, 498, 499  
DoS-Angriff 456, 457, 471  
DPCM 152  
Dropping 130  
DSS1 63, 78  
DTLS 276, 576, 591, 596

Dynamic Host Configuration Protocol  
s. *DHCP*  
dynamische PT-Nummer 176

**E**

E.164-Rufnummer 587  
Eavesdropping 463  
Echo 147  
Echtzeitkommunikation 147  
ECMA 77  
ECN 94, 128  
E-Modell 117, 199  
Ende-zu-Ende-QoS 131  
Ende-zu-Ende-Sicherheit 481  
Ende-zu-Ende-Signalisierung 71  
Ende-zu-Ende-Verzögerung 115, 116  
ENUM 6, 77, 110, 114, 393, 394, 501, 571, 586  
ENUM-DNS 111, 112, 587  
ENUM-DNS-Abfrage 115  
ENUM-Domain 111, 113  
ENUM-Domainname 112  
ENUM-Lookup 115  
ENUM-Registrar 111  
ENUM-Resolver 588  
ENUM-URI 587  
enveloped-data 484  
Ermittlung des Schutzbedarfs 486, 493  
Error Concealment 120  
ETSI 38, 50, 56  
Explicit Congestion Notification  
s. *ECN*  
Exploit 455  
Extended VoIP over Web 571

**F**

Fair Queueing 138  
Faktor R 200  
Fast Connect Procedure 232, 235, 245, 249, 358  
Federation SIP-Server 396  
Fehlerkontrolle 87, 89  
Fernvermittlungsstelle 5  
Firewall(s) 428, 487, 497  
Flusskontrolle (Flow Control) 87, 91  
Follow-me (Find-me) 301  
Forking-Funktion 300  
Fremd-Domain 266, 268, 272  
Fremd-IMS 48, 49  
Fremd-NGN 46  
FQ (*Fair Queueing*) 133, 138  
FQDN 98, 280, 291  
Fremd-Domain 287, 295  
From (*SIP Header*) 291, 313, 315, 483  
Full Cone NAT 428, 429  
Full Qualified Domain Name  
s. *FQDN*

**G**

G.107 117, 199  
G.114 118  
G.711 11, 580  
Gap 198  
Gatekeeper 16, 223, 234, 237, 240, 246, 248, 265, 267, 269, 377, 409  
Gatekeeper Discovery 238  
Gatekeeper-Proxy 380, 381, 388  
Gateway 13, 58, 358  
Gateway-Plattform 30, 35, 44, 360, 399  
Gathering 442

GCPU 336  
Gebührenbetrug 469, 499  
Glue 98  
Global System for Mobile Communications s. *GSM*  
Gmin 198  
GPRS 21, 23, 25  
GSM 20, 24, 25, 30, 32, 34, 45, 56, 69, 77, 106  
GSM RAN 24

## H

H.225.0 58, 105, 167, 227, 233, 243, 264, 270, 272, 273, 275, 357  
H.225.0-Kanal 234  
H.235 275, 499, 501  
H.245 58, 105, 251, 253, 254, 256, 258, 273  
H.245-Steuerungskanal 227, 230, 231  
H.245-Tunneling 236  
H.264 580  
H.323 12, 58, 78, 104, 105, 223, 224, 239, 273, 357, 408  
H.323-Domain 225, 265  
H.323-Forum 57, 273  
H.323-SIG 80, 105, 167, 170, 224, 244  
H.323 URL 239, 274  
H.323-Zone 224, 378  
H.450.x 259  
H.460.x 275  
Handover 22, 25  
harte Migration 398  
Hashfunktion 477, 478, 479  
HDLC 64  
Hear-Web-Content 32  
Heimat-Domain 266, 268, 286

Heimat-IMS 48  
Heimat-NGN 46  
herkunftsabhängiges Routing 8  
hierarchische Multiplane-Struktur 7  
Hijacking 456, 466  
HLF 265, 267, 268, 270, 273  
HLR 21, 22, 45, 266  
HMAC 206, 212  
HMAC-SHA1 207, 212  
Home Location Register s. *HLR*  
Homeoffice 592  
Host Candidate 441  
Hosted IP-TK-Anlage 445  
Hostname 18, 96, 282, 431, 467  
HSCSD 20  
HTTP 78, 80, 92, 276, 359  
HTTP-Digest 470  
HTTP Digest Authentication 477  
HTTPS 573, 578, 597, 600  
hybride Systemlösung 403

## I

IAM 59, 68, 76  
IANA 52, 167, 176, 275  
IAX 106  
ICE 428, 432, 440, 445, 580, 592  
ICMP 78, 79  
I-CSCF s. Interrogating-CSCF  
Identitäts-Spoofing 202  
IETF 50, 104, 598  
iLBC (*internet Low Bitrate Codec*) 175  
IMS 1, 26, 43, 56, 57, 58, 59  
IMTC 57  
IN 2, 6, 14, 15, 28, 32, 36, 45  
IN-Dienst 7, 15

- INAP 7, 31, 32, 74  
Inband-Signalisierung 5  
INFO 306  
Informationselement 66  
Infrastructure ENUM 398  
Integrated Peering 392, 394  
Integrated Services Digital Network s. *ISDN*  
Integrität 447, 481, 485, 489, 491, 495, 496  
Integrity 447  
Intelligent Network s. *IN*  
Inter-Asterisk exchange s. *IAX*  
Intermedia-Synchronisation 174, 181  
Internet 9  
Internet Assigned Numbers Authority s. *IANA*  
Internet Call Forwarding 33  
Internet Call Waiting 33  
Internet Protocol 1  
Internet-Telefonie 9  
Interrogating-CSCF 45, 47  
INVITE 19, 103, 177, 283, 287, 289, 298, 301, 305, 317, 318, 329, 461, 471, 578, 585  
INVITE-Flooding 471  
INVITE-Spoofing 470  
IP (*Internet Protocol*) 1, 84  
IP-Adresse 12, 84, 85, 100, 242, 379  
IP-basierte TK-Anlage 16  
IP-Centrex 445  
IP-Header 11, 86, 175  
IP Multimedia Subsystem s. *IMS*  
IP-Netz 9, 82  
IP-Netzknoten 132  
IP-Paket 1, 78, 86  
IP-Paketübermittlungsdienst 84, 87  
IP-PBX 16, 398, 501  
IP-Pseudo-Header 89  
IP Spoofing 456, 460, 498  
IP-Telefon 9  
IP Telephony Routing 377  
IP-TK-Anlage 16, 17, 398, 446  
IPsec (*IP Security*) 501  
ISDN 2, 5, 12, 18, 28, 30, 57, 60, 114, 379  
ISDN-TK-Anlage 354, 375  
ISDN-Verbindung 13, 60, 68, 76, 251  
ISN 93, 94  
Isochronität 12, 116, 173, 180, 193  
ISUP 72, 75, 356  
ITAD 380, 382, 384  
ITU-T 50, 54, 64, 69, 104, 199, 223
- J**
- JAIN 30, 41  
JavaScript Injection 596  
JavaScript Object Notation 596  
Jitter 83, 116, 117, 124, 174, 178, 180, 193, 195  
Jitter-Ausgleichpuffer 117, 119, 124, 194  
JSON 596
- K**
- Katalog von Sicherheitsanforderungen 495  
Klassifizierung 133  
KMP 202  
Kommunikationsprotokoll 77  
Kompondierungskennlinie 156  
Konvergenz der Netze 27, 45

**L**

LBS 34  
LAPD 64, 284, 380, 591  
Lauschangriff(e) 463, 464, 472  
Leistungsmerkmal 4  
Leitungsvermittlung 57, 60  
lineare Quantisierung 153  
Link-by-Link-Signalisierung 71  
LLQ 148  
Location-Server 295, 297, 331, 384  
logisches Netzwerkmodell 450  
LoST 59  
Low Cost Routing 16  
LPC 150, 163  
LPC-Filterung 150  
LPC-Vocoder 158

**M**

MAC 207  
MAC-Adresse 85  
MAC-Floating 498  
MAC-Frame 125, 126  
MAC-Funktion 81  
MAC-Spoofing 456, 460, 498  
Malformed JSON 596  
Malformed Request/Response 472  
Malware 455  
Man in the Middle s. *MitM*  
Man-in-the-Middle-Angriff(e) 459, 460, 597  
Master Key 201, 204, 205, 208, 209, 214  
MD5 (*Message Digest*) 477  
Media Access Control s. *MAC*  
Media Channel 10, 166  
Media Control Channel 167

Media Gateway 35, 36, 384, 408  
Media Gateway Controller s. *MGC*  
Media Gateway Control Protocol s. *MGCP*  
Media-Kanal 10  
Media-Socket 434  
Media Transport Plane 406  
Medienkanal s. *Media Channel*  
Megaco 80, 106, 357, 359, 369  
Megaco-Commands 372  
MESSAGE 306  
Message Body 309, 316  
Message-Body Tampering 472  
Message Waiting Indication 260  
MG 35, 36, 359, 408  
MGC 36, 358, 359, 374, 408  
MGCP 36, 43, 80, 106, 357, 359, 360  
MGCP-Commands 362, 363  
MGCP-Responses 364  
MIKEY 202, 205  
MIME 306, 481, 482, 485  
MIME-Objekt 481, 483  
MitM 462, 463, 465, 466, 472, 490  
Mixer 173, 181, 83  
Mobilität von Benutzern 282  
MoIP 572  
MOS 164  
MPLS 126  
MSC 26  
MSF 58  
MT-Instanz 593  
MTP 72, 73  
Multilaterales Peering 392  
Multimedia-Session 168

multipart 485  
multipart/signed 481  
Multiplexer 132, 134, 138, 183  
Multipurpose Internet Mail Extension  
s. *MIME*  
Multiservice-Netz 29  
Music on Hold 335

## N

Nameserver 97  
Naming Authority Pointer s. *NAPTR*  
NAPT 428  
NAPTR 99, 102, 111  
NAPTR-Query 102, 104, 283  
NAPTR-RR 102, 112, 113, 588  
NAT 275, 427, 433, 580, 592  
Network Address Translation s. *NAT*  
Netzwerkprotokoll 78  
Next Generation Network s. *NGN*  
Next Generation Services 34  
NGN 1, 34, 35, 44, 46, 5759  
nichtlineare Quantisierung 151, 153  
nomadische Nutzung 49, 277  
NOTIFY 307, 345, 347, 349, 351  
Notrufdienste 59  
NTP 186, 322

## O

öffentlicher Schlüssel 482  
Offer (SDP-Offer) 316, 328  
Offer-Answer-Modell 317  
offizieller Socket 428  
OPTIONS 305  
OSPF 116  
OMA 41  
Opus Codec 580

Ortsvermittlungsstelle 5  
OSA 38, 56  
OSP 394, 397  
overlap sending 67

## P

P2PSIP 360  
Paketverlustrate 116, 117, 124  
Paketverlustwahrscheinlichkeit 124  
Park-Server 341, 343  
Parlay 40  
Parlay Group 38  
Parlay/OSA 30, 37  
Parlay-X 41  
Payload (Nutzlast) 176  
Payload-Typ (PT) 228  
P-CSCF s. *Proxy-CSCF*  
PCL 120  
PCM 11, 150, 163  
PDCA-Zyklus 452  
Peering-Punkt 392  
PESQ 165  
Pharming 458, 466, 467, 499, 501  
Pharming bei H.323 467  
Pharming bei SIP 467  
Phishing 456, 458, 466  
Phreaking 459, 463, 468, 469  
physikalische Adresse 85  
PINT 31, 307  
PINT-Client 31  
PINT-Gateway 31  
PINT-Server 31  
PKCS 481  
PKCS#7 484, 485  
PKI 275, 482, 499

Planung der VoIP-Sicherheit  
452, 454, 486

Port 10

Port Scanning 459, 498

PPP 81

PQ (*Priority Queueing*) 133, 134

PRACK 306

pre-granted Admission 227, 232, 248

Presence Service 307

Primärmultiplexanschluss 61

Priority Queueing 134

privater Schlüssel 482

privater Socket 428

PROGRESS 66

Protokoll 77

Proxy-CSCF 45, 46, 47

Proxy-Ebene 282

Proxy-Server 295, 329, 331

Proxy/Server-Imitation 471

PS-Domain 24, 26

PSTN 5, 6, 12, 28, 58, 67, 115, 358,  
379

PT-Nummer 172, 176

Public Switched Telephone Network  
s. *PSTN*

PUBLISH 307

## Q

Q.930 64, 244

Q.931 64, 227, 244

QoS s. *Quality of Service*

QoS-Abuse 472

QoS-Anforderungen 12, 58, 115

QoS-Missbrauch 472

QSIG (*Q-Signalling*) 77

QSIG über IP 78

Quality of Service 12, 58, 105, 115,  
116, 117

Quantisierung 150

Quantisierungsfehler 150, 154

Quell-Port 84

Queue-Management 115, 123, 125,  
132

Queueing 131, 133

## R

RAN 21

RAS 226, 227, 232, 237, 500

RAS-Kanal 230

Real-time Transport Protocol s. *RTP*

Rechnername 282

Record-Route 314, 352, 353, 433

Redirect-Mode 330

Redirect-Server 295, 297, 329

REFER (*SIP Header*) 307, 342, 346,  
349

Refer-To (*SIP Header*) 343, 347, 349

Referenzmodell 450

Referred-By (*SIP Header*) 307, 343,  
347, 349

REGISTER 49, 305, 333, 473, 479,  
583

Registrar 48, 295, 331, 332, 479, 498

Registration, Admission, Status  
s. *RAS*

Registration Hijacking 462, 466, 470,  
473, 476, 479, 500

Registrierung 48, 240, 266, 332, 473,  
479

Registrierungsführung 466

re-INVITE 318, 328

Relayed Candidate 442

Request to Call 32

- Request for Comments s. *RFC*  
Request-Routing 353  
Resolver 100  
Response-Klassen 308  
Response-Routing 351  
Resource Record (RR) 97, 98, 274, 281, 475  
Restricted Cone NAT 429  
R-Faktor 200  
RFC 51  
RFC Editor 51  
RIP 116  
Risiko 491  
Risikoabschätzung 492  
Risikoanalyse 454, 486, 492  
Roaming 46, 48, 223, 264  
ROHC 215, 216, 223, 225  
Rootkit 455  
Root-Server 100  
Round Robin 135, 139  
Round Trip Delay 196  
Round Trip Time (RTT) 193, 196  
Route (*SIP Header*) 314, 353  
Route Injection 460  
Routing von Telefonverbindungen 8  
RPNP 92, 395  
rport 33  
RR s. *Resource Record*  
RSVP 115, 123, 125, 144, 145, 226  
RSVP-TE 117  
RTCP 11, 58, 80, 104, 147, 165, 171, 184, 191, 220, 225, 228, 255, 256, 327, 328, 500, 575, 580  
RTCP Insertion 465  
RTCP-Kanal 11, 231, 258  
RTCP-Paket 210  
RTCP-Port 318  
RTCWEB 59  
RTCP XR 184, 191, 197, 200  
RTP 10, 80, 104, 147, 165, 172, 203, 220, 225, 274, 328, 365, 500, 575, 580  
RTP/AVP 178, 321, 326, 327  
RTP Control Protocol s. *RTCP*  
RTP-Header 11, 173  
RTP Insertion 459, 465  
RTP-KanaL 231, 255, 257, 290  
RTP-Paket 104, 170, 179, 195, 210, 211  
RTP-Port 318, 321  
RTP Profile 225  
RTP/SAVP 326  
RTT s. *Round Trip Time*
- S**  
Salting Key 209  
sanfte Migration 398  
SBC 392, 394  
SCCP 72, 75  
Schadensprogramm(e) 454, 466, 468, 487, 498  
Scheduling 133  
Schnittstelle S0 6, 61, 62, 63  
Schnittstelle S<sub>2M</sub> 62  
Schnittstelle U<sub>K0</sub> 62  
Schutzbedarf 495  
Schutzbedarfsstufen 492  
Schweizer-Käse-Modell 445, 453  
SCP 7, 74  
S-CSCF 45  
SCTP 79, 87, 101, 106, 107, 275, 280, 576, 586, 591  
SCTP-Assoziation 108

- SCTP-Endpunkt 108  
SCTP over DTLS 580, 594  
SCTP-Paket 109  
SDP 167, 169, 177, 273, 276, 277, 309, 310, 315, 317, 328, 339, 364, 434, 437, 596  
SDP-Offer 171, 585  
Seamless Mobility 47  
Second-Level-Domain 98  
Secure MIME s. *S/MIME*  
Secure RTP s. *SRTP*  
Secure WebSocket 582  
Serialisierung 119  
Sequence Number 93  
Server Reflexive Candidate 441  
Service-Diebstahl 469  
Service-Plane 7  
Service-Theft 468, 469  
Serving-CSCF 45, 47  
Session 10, 11, 49, 166, 274, 342, 365, 373  
Session Description Protocol s. *SDP*  
Session Forwarding 287  
Session Hijacking 470, 472, 474, 476, 499  
Session Initiation Protocol s. *SIP*  
Session Key 204, 214  
Session-Kontext 218  
Session Spoofing 470, 476  
Session-Tear-Down 472  
Session Traversal Utilities for NAT s. *STUN*  
Session-Umleitungsinstanz 299, 329  
Session-Weiterleitung 287, 329  
SETUP 20, 66, 67, 68, 250, 379, 388  
Shaping 130  
Shared Secret 477  
Sicherheitsanforderungen 495  
Sicherheitsanforderungskatalog 496  
Sicherheitsmaßnahmenkatalog 497  
Sicherheitsplanung 445  
Sicherheitsrisiken 489  
Sicherheitsschwachstelle 450, 453, 486, 494, 496, 497  
Sicherheitsziel(e) 496  
Singleservice-Netz 28  
Signalisierung 5, 57, 61  
Signalisierungsdreieck 580  
Signalisierungskanal 61  
Signalisierungsnetz 70  
Signalisierungsprotokoll 57, 167, 579  
Signalisierungs-Plane 7, 45  
Signalisierungssystem Nr. 7 s. *SS7*  
Signallaufzeit 123  
Signalling System No. 7 s. *SS7*  
Signatur 485  
Signierung 485  
SIG-P s. *Signalisierungsprotokoll*  
SIG-P over WS(S) 579  
Singlemedia-Session 166  
SIP 11, 12, 27, 31, 33, 36, 43, 46, 57, 58, 80, 96, 101, 104, 105, 114, 167, 169, 170, 274, 273, 274, 354, 355, 356, 359, 367, 378, 389, 391, 571, 573  
SIP in JavaScript (SIP-J) 575  
SIP over DCCP 276  
SIP over TLS 500  
SIP over UDP 428  
SIP over WS 584, 585  
SIP over WSS 575, 584, 585  
SIP über TCP 102, 103

- SIP über TLS 102, 103, 275  
SIP über UDP 102, 103  
SIP-Adresse 277  
SIP Bombing 461, 471  
SIP-Client 277  
SIP Digest 476, 480, 500  
SIP-Digest Authentication 470  
SIP-Digest-Authentifizierung 478  
SIP-Forum 58  
SIP-Message Tampering 472  
SIP-Nachricht 305  
SIP-Proxy 19, 101, 103, 239, 279, 281, 283, 288, 393  
SIP-Proxy Hijacking 475  
SIP-Server 46, 277, 281, 390  
SIP-Registrar 473, 479  
SIP-Request 277, 289, 297  
SIP-Request-Spoofing 470  
SIP-Response 277, 289, 297  
SIP-Response-Spoofing 470  
SIP Service Provider s. SSP  
SIP-Trapezoid 282, 284, 433  
SIP-Tunneling 486  
SIP-URI 17, 19, 96, 111, 277, 278, 279, 280, 301, 332, 394, 473  
SIPS 102, 275, 280, 500, 576  
SIPS over UDP 276  
SIPS über SCTP 275  
SIPS über TCP 275  
SIPS-URI 280, 583  
SLA 265  
S/MIME 470, 481, 482, 483, 484, 500  
SMTP 80  
Socket 428  
Soft-IP-Telefon 9, 282, 294  
Softswitch 30, 36, 42, 357, 399, 408  
Spezifikation des Schutzbedarfs 493  
Spezifikation von Sicherheitsmaßnahmen 494  
SPEERMINT 391, 395, 398  
SPIM 472  
SPIRITS 31, 32, 34, 307  
SPIRITS-Client 33  
SPIRITS-Server 33  
SPIT 458, 463, 468, 469, 489, 500  
Spoofing 466, 470, 499  
Sprachkanal 4  
Spracherzeugung 159  
Sprach-VLAN 410  
SRTP 147, 199, 200, 201, 206, 209, 210, 464, 596, 575, 576, 596  
SRV (*SeRVer*) s. *SRV-RR*  
SRV-Query 102, 104, 394  
SRV-RR 99, 102, 274, 281  
SS7 6, 13, 23, 31, 45, 58, 67, 68, 69, 71, 106, 356  
SSL 275  
SSP (*Service Switching Point*) 7, 74  
SSP (*SIP Service Provider*) 391, 392, 398  
SSRC 175, 183, 188  
SSRC-Kollision 465  
STCP over DTLS 576  
Stream Control Transmission Protocol s. *SCTP*  
Stromverschlüsselungsverfahren 209  
STUN 427, 432, 435, 592  
STUN over TCP 436  
STUN over TLS 436  
STUN-Client 437  
STUN-Server 437

SUBSCRIBE 307, 344, 351  
Supplementary Services 259, 278  
Symmetric NAT 429, 430, 438, 439, 441  
Symmetric Response 293, 427, 431, 433  
Symmetric RTCP 435  
Symmetric RTP 434  
Symmetric RTP/RTCP 427  
SYN 94  
SYN Flood 456  
Systemverfügbarkeit 448

## T

TCP 9, 78, 79, 82, 84, 87, 91, 107, 171, 227, 228, 275, 386  
TCP-Datensegment 82  
TCP-Header 91  
TCP-Hijacking 457  
TCP-Paket 92  
TCP/RTP/AVP 172  
TCP-Verbindung 91, 93, 94, 95, 110, 171, 236, 244, 247, 250, 270, 389, 461  
Teilnehmer-Roaming 264  
Teilnehmersignalisierung 58, 67  
Telefon-Hub 417  
Telefonie-Routing 377  
Telephony Routing over IP s. *TRIP*  
Telefonnetz 4, 6  
Telefonverbindung 57  
TGREP 397  
Third Party Call Control s. *3PCC*  
Timestamp (Zeitstempel) 174, 178, 179, 181  
TIPHON 56  
TISPAN 44

TK-Anlage 15, 16  
TLS 101, 275, 280, 464, 575, 579, 582, 585, 596  
To (*SIP Header*) 291, 314, 315  
Token-Bucket-Modell 144  
Toll Fraud 458, 468  
Top-Level-Domain 98, 100  
TPKT 244, 252  
Transcoding-Service 581  
Translator 173, 181, 182  
Transmission Control Protocol s. *TCP*  
Transport Layer Security s. *TLS*  
Transport-Plane 45  
Transportschicht 87  
Trapezoid-Modell 590, 591  
TRIP 116, 226, 265, 377, 379, 383, 385, 389  
Trojaner 455, 464, 466, 468, 487  
Trust Center 482  
TTL (*Time To Live*) 83, 87  
TTL-Wert 83  
TURN 428, 432, 438, 592  
TURN-Server 439, 441, 444

## U

UA 282, 583  
UAC 276, 283, 288  
UAS 276, 283, 288  
Überflutung mit INVITE 471  
Überlastkontrolle 87, 91  
UCT 337  
UDP 8, 79, 81, 87, 88, 107, 170, 237, 256, 275, 275, 326, 360, 365  
UDP-Header 11, 175  
UDP-Lite 79, 90

- UDP-Paket 81, 88  
UE 49  
Umleitung bei Besetzt 301  
Umleitungsinstanz 330  
UMTS 21, 24, 30, 32, 34, 43, 45, 56, 57, 77, 106, 115, 212  
Unattended CT 346  
Uniform Resource Identifier s. *URI*  
Uniform Resource Locator s. *URL*  
Uniform Resource Name s. *URN*  
UNSUBSCRIBE 307  
UPDATE 310  
URI 278, 281  
URI-Spoofing 456, 458  
URL 282, 467  
URL-Spoofing 455, 457, 464, 468, 469  
URN 59  
User Agent s. *UA*  
User Agent Client s. *UAC*  
User Agent Server s. *UAS*  
User Datagram Protocol s. *UDP*  
User Part 72, 74  
UTRAN 24
- V**
- V-Bombing 468  
Vernetzung von TK-Anlagen 403  
Verfügbarkeit 489, 491, 495, 496, 497  
Vertraulichkeit 447, 481, 483, 491, 495, 496  
Via (*SIP Header*) 291, 292, 314, 315, 351, 431, 434  
virtuelle Verbindung 79  
Vishing 469  
Visitor Location Register s. *VLR*
- VLAN 126, 409  
VLF 266, 267, 268, 270, 273  
VLR 21, 22, 266  
Voice-Bombing 468  
Voice-Mail-Server 302, 475  
Voice over NGN 59  
VoIP (*Voice over IP*) 1, 9  
VoIP-Adresse 17  
VoIP-Client 593  
VoIP-Forum 57  
VoIP-Gateway 13, 58, 110, 147, 357, 378, 402, 497  
VoIP-Metrik(en) 117, 186, 191, 192, 199  
VoIP-Notrufdienste 59  
VoIP-Peering 377, 391  
VoIP Phishing 468  
VoIP-Server 16, 407  
VoIP-Session 166, 365  
VoIP-Sicherheit 445, 450, 490  
VoIP-Sicherheitsprozess 452  
VoIP-Sicherheitsziele 446  
vollständiger Rechnername 98  
VPN 405, 406  
VSP 288
- W**
- W3C 598  
WAP 41  
Web-Echtzeitkommunikation 571  
WebRTC 571, 577  
WebRTC-Client 573, 574, 581, 582, 585, 593  
WebRTC-HTTP Ingestion Protocol 600

WebRTC-Server 573, 574, 581, 582, 593

WebRTC-Sicherheit 594, 595

WebRTC-Signalisierung 576

WebRTC/VoIP-Gateway 593

Web Services 30, 40

WebSocket Protocol 573

Web-Videotelefon 572, 573

Weighted Fair Queueing 140

Weiterleitungsinstanz 330

Well Known Port 89, 95, 75

WFQ 134, 139, 140, 141

WISH 600

WRR 135

WS (WebSocket) 573, 575

WSP (WebSocket Protocol) 573, 575, 579, 581

WSS (WebSocket Secure) 582

WS-Verbindung 582

Wurm 455

XMPP 576

## X

XR-Paket 120

## Z

Zelle 21

Zeitstempel s. *Timestamp*

Zertifikat 485

Zertifizierungsstelle 482

Zone 224, 378, 388

ZRTP 203