Wireshark VoIP Analysis — Quick Reference

Network Analysis for VoIP Engineers | Sean Cheesman | Cheesman Press

SIP Display Filters

Filter	Shows
`sip`	All SIP traffic
`sip && udp`	SIP over UDP only
`sip && tcp`	SIP over TCP only
`sip.Method == "INVITE"`	INVITE (initial + re-INVITE)
`sip.Method == "REGISTER"`	REGISTER requests
`sip.Method == "BYE"`	BYE (teardowns)
`sip.Method == "OPTIONS"`	OPTIONS (keepalives)
`sip.Method == "REFER"`	REFER (call transfers)
`sip.Method == "NOTIFY"`	NOTIFY (transfer, MWI, BLF)
`sip.Status-Code == 200`	200 OK
`sip.Status-Code == 403`	403 Forbidden (auth fail)
`sip.Status-Code == 486`	486 Busy Here
`sip.Status-Code == 488`	488 Not Acceptable (codec)
`sip.Status-Code == 503`	503 Unavailable (overload)
`sip.Status-Code >= 400`	All error responses
`sip.Call-ID == "value"`	All messages in one dialog
`sip.from.user == "1001"`	From extension 1001
`sip.to.user == "+15551234567"`	To a specific number
`sip.CSeq.seq > 1`	re-INVITEs only
`sdp.media contains "image"`	T.38 fax SDP

RTP / RTCP / Media / DNS / Layer 2

Filter	Shows
`rtp`	All RTP
`rtp.ssrc == 0x4a2f1b3c`	One RTP stream (by SSRC)
`rtp.payload_type == 0`	G.711 mu-law (PCMU)
`rtp.payload_type == 8`	G.711 A-law (PCMA)
`rtp.payload_type == 18`	G.729
`rtp.payload_type == 101`	RFC 2833 DTMF events
`rtp.marker == 1`	Talk spurt starts
`rtp && ip.dsfield.dscp != 46`	RTP NOT marked EF (QoS)
`rtcp`	RTCP quality reports
`t38`	T.38 fax UDPTL
`stun`	STUN / ICE checks
`dtls`	DTLS-SRTP key exchange
`tls.alert_message`	TLS errors (cert failure)
`dns || sip`	DNS + SIP together
`dns.qry.type == 33`	DNS SRV queries
`dns.flags.rcode == 3`	DNS NXDOMAIN
`ip.flags.mf == 1`	Fragmented packets (MTU)
`tcp.flags.reset == 1`	TCP connection reset
`cdp || lldp`	Phone VLAN assignment

Compound Filters

Filter	Use For
`sip || rtp || rtcp`	Full VoIP view
`sip.Status-Code >= 400`	Failures only
`rtp.payload_type == 101 || sip.Method == "INFO"`	DTMF (all methods)
`sip.Method == "REFER" || sip.Method == "NOTIFY"`	Transfer flow
`sip.Method == "REGISTER" || sip.Status-Code == 401`	Registration flow

Timer Drop Reference

Drop Time	Timer	Check For
`~32s ringing`	`Timer B`	INVITE no 200 OK
`~32s answered`	`Timer H`	200 OK no ACK
`~32s BYE`	`Timer F`	BYE no 200 OK
`~30 min`	`Session-Expires`	re-INVITE blocked
`30-60s audio`	`Firewall UDP`	RTP pinhole expired

5-Step Analysis Workflow

#	Step
1	Apply filter: sip
2	Telephony > Flow Graph (limit to display filter)
3	Filter: sip.Call-ID == "value" to isolate one dialog
4	Right-click INVITE > Set/Unset Time Reference
5	Examine 200 OK SDP: check c= IP and m= codec
6	Telephony > RTP > RTP Streams for audio analysis

Display Filter Syntax Quick Reference

Network Analysis for VoIP Engineers · Sean Cheesman · Cheesman Press · 2026

== !=	equals / not equal
&&	AND (both true)
||	OR (either true)
!	NOT / negate
contains	substring match
matches	regex match

ip.addr ==	src or dst IP
ip.src ==	source IP only
ip.dst ==	destination IP only
udp.port ==	UDP port (src or dst)
tcp.port ==	TCP port (src or dst)
ip.dsfield.dscp ==	DSCP marking

sip || rtp	SIP or RTP
!sip && !rtp	exclude SIP and RTP
sip && ip.addr == 10.0.2.5	SIP to/from one host
frame contains "INVITE"	text search in frame
frame.time_delta > 0.5	inter-packet gap > 500ms
udp.length > 200	large UDP (likely RTP)

Ctrl+R	set/unset time reference
Ctrl+E	expert info panel
Ctrl+F	find packet
Tel > Flow Graph	SIP call ladder
Tel > RTP Streams	RTP quality + playback
Tel > VoIP Calls	export single call

Wireshark VoIP Analysis — Protocol Reference

SIDE B · SIP Call Flow · SDP Anatomy · RTP Header

Network Analysis for VoIP Engineers · Cheesman Press

SIP Call Flow — Healthy Outbound Call (INVITE/BYE)

Phone A ←——— SBC / Carrier	Annotation
SETUP	
`INVITE ———————————>`	*Offer SDP: codec list, RTP IP:port*
`<——————— 100 Trying`	*Stops INVITE retransmits (UDP only)*
`<——————— 180 Ringing`	*Phone alerting; early media possible*
`200 OK <———————————`	*Answer SDP: selected codec, RTP port*
`ACK ———————————>`	*Completes 3-way handshake*
MEDIA	
`RTP ———————————>`	*PT=0 G.711 · 50 pkt/s · seq+1 · ts+160*
`<——————— RTP`	*Bidirectional · unique SSRC each stream*
TEARDOWN	
`BYE ———————————>`	*Either party initiates teardown*
`200 OK <———————————`	*Dialog fully terminated*

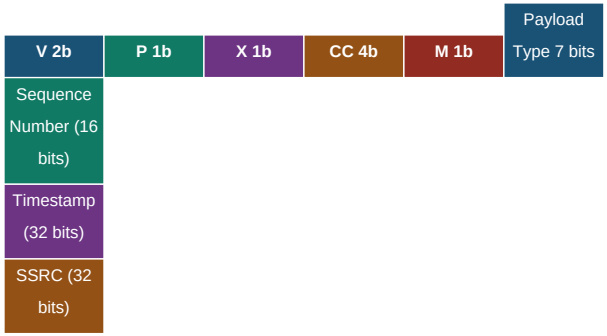
SIP Registration Flow (REGISTER)

Phone ←——— Registrar (Proxy)	Annotation
`REGISTER ———————————>`	*No credentials — first attempt*
`<——————— 401 Unauthorized`	*WWW-Authenticate: nonce + realm*
`REGISTER ———————————>`	*Authorization: Digest response hash*
`200 OK <———————————`	*Contact: expires=3600 granted*
`[repeat at ~T=1800s]`	*Re-register at 50% of Expires interval*
If 403 Forbidden after auth → wrong password or IP not whitelisted	
If no response at all → connectivity / DNS / firewall issue	

SDP Anatomy — What Each Line Means

SDP Line	Meaning / What to Check
`v=0`	Version — always 0
`o=alice 123 1 IN IP4 x`	Origin: sess-id, sess-ver, source address
`c=IN IP4 203.0.113.5`	WHERE to send RTP. Must be public IP on WAN side.
`t=0 0`	Timing: 0 0 = permanent session
`m=audio 16384 RTP/AVP 0 8 101`	type port profile payload-types (priority order)
`m=image 6000 udptl t38`	T.38 fax: image type, udptl transport (not RTP)
`a=rtpmap:0 PCMU/8000`	Maps PT number to codec name / clock rate
`a=rtpmap:101 telephone-event/8000`	Maps PT 101 to RFC 2833 DTMF
`a=fmtp:101 0-15`	DTMF digits 0-15 supported
`a=fmtp:18 annexb=no`	G.729: Annex B silence suppression disabled
`a=ptime:20`	20ms packets = 160 samples (G.711 8kHz clock)
`a=sendrecv`	sendrecv=normal · sendonly=hold · recvonly · inactive
`a=crypto:1 AES_CM...`	SDES key — media is SRTP encrypted
`c= private IP on WAN`	NAT failure — one-way audio
`m= changed in re-INVITE`	Hold (sendonly) · T.38 switch · codec renegotiation

RTP Header Anatomy (12 bytes fixed)



Field	Bits	What It Tells You
`V (Version)`	2	Always 2. If not 2, packet is not RTP.
`M (Marker)`	1	Talk spurt start / first DTMF event. Filter: rtp.marker == 1
`PT (Payload Type)`	7	Codec: 0=PCMU, 8=PCMA, 18=G.729, 101=telephone-event/DTMF
`Sequence Number`	16	Increments +1 per packet. Gaps = packet loss. Jumps = reorder.
`Timestamp`	32	Sample clock. G.711: +160 per 20ms packet (8000 Hz).
`SSRC`	32	Unique stream identifier. Each direction has its own SSRC.