WIRESHARK VoIP

LAB GUIDE

Hands-on exercises for
Network Analysis for VoIP Engineers

Sean Cheesman

Cheesman Press . 2026

How to Use This Guide

This lab guide provides structured exercises for every case file and baseline capture in the companion pcap library. Each lab has a pcap file, learning objectives, step-by-step instructions, and answer boxes. Work through labs in order.

Before You Begin: Install the VoIP Analysis Wireshark profile (see README_PROFILE.txt). It sets up column layout, color rules, and filter shortcuts used throughout these exercises.

Difficulty Levels

Beginner — open the file, apply the filter, answer direct questions. No prior Wireshark experience required.

Intermediate — requires SIP call flow understanding and navigation by Call-ID and time reference.

Advanced — requires multi-capture analysis, RTP stream analysis, or timing anomaly diagnosis.

Module 1 — Reading a Healthy Call

Work through these labs before the failure cases. Knowing what normal looks like is the prerequisite for recognising what abnormal looks like.

Lab 1.1 The Ground Truth — Healthy Outbound Call [Beginner]

File: `baseline_1_healthy_outbound_call.pcap`

Learning Objectives

>> Navigate a complete SIP call flow in Wireshark Flow Graph

>> Identify the five phases of a VoIP call from a capture

>> Describe the normal RTP packet signature (rate, sequence, timestamp)

>> Use Telephony > RTP > RTP Streams to inspect quality metrics

Setup

1. Open baseline_1_healthy_outbound_call.pcap
2. Apply filter: sip || rtp
3. Open Telephony > Flow Graph — check Limit to display filter
4. Right-click the INVITE > Set/Unset Time Reference

Exercises

1. Apply filter: sip. How many SIP messages? List methods and codes in order.

Your answer:

2. Find the 200 OK. Expand the SDP body. What IP is in the c= line? Private or public?

Your answer:

3. Apply filter: rtp. Select any RTP packet. Record: Payload Type, Sequence, Timestamp, SSRC.

Your answer:

4. Open Telephony > RTP > RTP Streams. How many streams? What does each represent? Max jitter?

Your answer:

5. With INVITE as time reference, when does the BYE occur? How long was the call?

Your answer:

Key Takeaway: A healthy call follows INVITE > 100 > 180 > 200 OK > ACK > [RTP] > BYE > 200 OK. The 200 OK SDP c= shows the SBC WAN IP (203.0.113.5). RTP: 50 pkts/sec, sequence +1, timestamp +160 per packet.

Lab 1.2 Audio Playback — Hearing the RTP Stream [Beginner]

File: `baseline_0_audio_playback_demo.pcap`

Learning Objectives

>> Play back audio from an RTP stream in Wireshark

>> Identify talk spurts using the RTP marker bit

>> Distinguish the two call directions by SSRC and tone

Setup

1. Open baseline_0_audio_playback_demo.pcap
2. Apply filter: rtp

Exercises

1. Open Telephony > RTP > RTP Streams. How many streams? Record source IP:port and SSRC for each.

Your answer:

2. Select first stream > Analyze > Play. Describe what you hear. Select second > Play. How does it differ?

Your answer:

3. Apply filter: rtp.marker == 1. How many packets? What does each represent? Time gap between markers for the same SSRC?

Your answer:

Key Takeaway: The RTP marker bit is set on the first packet of each talk spurt. Filtering rtp.marker == 1 shows the conversation rhythm. Wireshark plays G.711 audio directly.

Lab 1.3 Registration Cycle [Beginner]

File: `baseline_2_healthy_registration.pcap`

Learning Objectives

>> Identify all four messages in a normal SIP registration

>> Explain the purpose of the 401 challenge and Authorization header

>> Describe re-registration timing and why it matters

Setup

1. Open baseline_2_healthy_registration.pcap
2. Apply filter: sip.Method == REGISTER || sip.Status-Code == 401 || sip.Status-Code == 200

Exercises

1. How many complete registration cycles? At what timestamps does each start?

Your answer:

2. Click the 401 Unauthorized. Expand WWW-Authenticate. What is the realm? What is the nonce?

Your answer:

3. Click the second REGISTER. Expand Authorization. What algorithm? What field has the hash?

Your answer:

4. Click the 200 OK. What Expires value is in the Contact header? When should the phone re-register? (Hint: 50% of Expires.)

Your answer:

Key Takeaway: Normal registration: REGISTER > 401 (challenge) > REGISTER+auth > 200 OK. The phone must re-register before the Expires value elapses — typically at 50% of the granted interval.

Module 2 — Diagnosing Call Failures

For each lab, a baseline is referenced for comparison — open both side by side when the lab instructs.

Lab 2.1 One-Way Audio — SDP Private IP Leak [Intermediate]

File: `case_3_1a_nat_private_ip_in_sdp.pcap`

Learning Objectives

>> Identify a private IP leaking into WAN-side SDP

>> Explain why this produces one-way audio

>> Distinguish one-way RTP from completely absent RTP

Setup

1. Open case_3_1a_nat_private_ip_in_sdp.pcap
2. Also open baseline_1_healthy_outbound_call.pcap in a second Wireshark window
3. Apply filter: sip on the failure capture
4. Set INVITE as time reference

Exercises

1. Find the 200 OK. Expand the SDP body. What IP is in the c= line? Is it routable from the carrier?

Your answer:

2. Compare with baseline_1 200 OK SDP. What IP appears there? Why is that correct and this wrong?

Your answer:

3. Apply filter: rtp && ip.dst == 192.168.10.45. Any packets? What does this tell you about where the carrier sends audio?

Your answer:

4. Apply filter: rtp && ip.src == 203.0.113.5. Do you see outbound RTP? Describe the user experience in one sentence.

Your answer:

Key Takeaway: The SDP c= line is where the far end sends RTP. A private IP in a WAN SDP means the carrier sends audio to an unreachable address. Always check the 200 OK SDP c= line first for one-way audio.

Lab 2.2 Timer B — INVITE Retransmission Pattern [Intermediate]

File: `case_3_3a_timer_b_no_final_response.pcap`

Learning Objectives

>> Identify the Timer B retransmission pattern from intervals alone

>> Calculate Timer T1 and Timer B from observed packet timing

>> Distinguish Timer B failure from asymmetric loss (Lab 2.3)

Setup

1. Open case_3_3a_timer_b_no_final_response.pcap
2. Apply filter: sip
3. Set first INVITE as time reference

Exercises

1. How many INVITE packets are in the capture? Record the timestamp of each.

Your answer:

2. Calculate the interval between consecutive INVITEs. What is the pattern? At what value does it cap? What timer governs that cap?

Your answer:

3. Is there any response to any INVITE? Does a 100 Trying appear? What does absence of 100 Trying tell you about where the fault is?

Your answer:

4. At what time does the capture end? What timer fired? What response code does the originating UAC generate locally?

Your answer:

Key Takeaway: Timer T1 = 500ms. Retransmissions double: 500ms, 1s, 2s, 4s, capped at T2=4s. Timer B = 64*T1 = 32s. No 100 Trying = INVITE never reached the far end. Compare with Lab 2.3.

Lab 2.3 Asymmetric Packet Loss [Advanced]

File: `case_3_3d_asymmetric_packet_loss.pcap`

Context: This file simulates a merged near-end + far-end capture. Packets from 192.168.10.45 = originating phone. Packets from 10.0.2.5 = SBC responses. In production: mergecap -w combined.pcap near.pcap far.pcap

Learning Objectives

>> Distinguish asymmetric loss from total loss using Flow Graph

>> Read a merged near/far capture to identify loss direction

>> Name three root causes and their diagnostic tests

Setup

1. Open case_3_3d_asymmetric_packet_loss.pcap alongside case_3_3a_timer_b_no_final_response.pcap
2. Apply filter: sip to both
3. Set INVITE as time reference in both

Exercises

1. Open Telephony > Flow Graph on the asymmetric capture. How does it differ from case_3_3a? Which direction of arrows is in 3.3d but not 3.3a?

Your answer:

2. Apply filter: ip.src == 192.168.10.45 (near-end only). How does this compare to case_3_3a?

Your answer:

3. Apply filter: ip.src == 10.0.2.5 (far-end only). What messages is the SBC sending? Why does it keep sending 100 Trying?

Your answer:

4. Name three root causes that produce this pattern and the diagnostic command or test for each.

Your answer:

Key Takeaway: Near-end-only view of asymmetric loss looks identical to Timer B. The difference is only visible with both sides captured. Far end responding = return path problem. Traceroute in both directions is the fastest diagnostic.

Lab 2.4 DTMF — Working vs. Missing [Intermediate]

File: `case_3_5a_dtmf_rfc2833_correct.pcap`

Learning Objectives

>> Identify RFC 2833 DTMF events in an RTP stream

>> Describe the start / continuation / end event structure

>> Compare a working DTMF capture against one without

Setup

1. Open case_3_5a_dtmf_rfc2833_correct.pcap
2. Apply filter: rtp.payload_type == 101
3. Also open case_3_5b_dtmf_pt101_absent.pcap in a second window with the same filter

Exercises

1. In case_3_5a, how many PT=101 packets? Expand the first. What digit? Is the marker bit set?

Your answer:

2. Find the end event packets (E bit set). How many end events per digit? Why?

Your answer:

3. Switch to case_3_5b with filter rtp.payload_type == 101. How many packets? Check SDP — does telephone-event appear?

Your answer:

4. What are two possible reasons PT=101 packets are absent despite telephone-event in the SDP?

Your answer:

Key Takeaway: RFC 2833 DTMF: PT=101, marker bit on first event, continuations at regular intervals, 3 redundant end events. Absent PT=101 despite telephone-event in SDP = SBC stripped them or QoS dropped them.

Lab 2.5 Session Timer — Failure vs. Success [Advanced]

File: `case_3_3b_session_timer_failure.pcap`

Learning Objectives

>> Identify session timer re-INVITEs by CSeq and timestamp

>> Describe the retransmission pattern when a re-INVITE gets no response

>> Compare the failure with the healthy session timer baseline

Setup

1. Open case_3_3b_session_timer_failure.pcap
2. Also open baseline_5_healthy_session_timer.pcap
3. Apply filter: sip.CSeq.method == INVITE to both
4. Set INVITE CSeq:1 as time reference in both

Exercises

1. In case_3_3b, at what timestamp does the CSeq:2 re-INVITE appear? What triggered it?

Your answer:

2. Count the re-INVITE retransmissions. Record each timestamp. When does BYE appear? How many seconds between first re-INVITE and BYE?

Your answer:

3. Open baseline_5. When does its CSeq:2 re-INVITE appear? How quickly does 200 OK arrive? What happens after?

Your answer:

4. In case_3_3b apply filter: sip.Status-Code == 200. Does a 200 OK exist for the re-INVITE? What does that tell you?

Your answer:

Key Takeaway: Session timer re-INVITEs fire at the Session-Expires interval (default 1800s). If no response, Timer B fires at 32s and the call drops. Timestamps: 1800 + 32 = 1832s. Re-INVITE disappearing = firewall expired dialog state.

Module 3 — Codec and Media Analysis

Lab 3.1 G.729 Codec Negotiation [Intermediate]

File: `baseline_4_healthy_g729_call.pcap`

Learning Objectives

>> Identify codec negotiation in SDP offer and answer

>> Describe how the answer selects a subset of the offer

>> Measure the payload size difference between G.711 and G.729

Setup

1. Open baseline_4_healthy_g729_call.pcap
2. Apply filter: sip || rtp

Exercises

1. Find the INVITE. Expand the SDP. List all PTs in the m= line. What do the a=rtpmap lines tell you?

Your answer:

2. Find the 200 OK. What PTs are in its m= line? Which codec was selected? Which rejected?

Your answer:

3. Apply filter: rtp. Select a G.729 packet. What is the payload length in bytes? Check baseline_1 for a G.711 packet.

Your answer:

4. Both calls send 50 pkts/sec. Calculate total bandwidth (payload + 40 bytes overhead) for G.729 vs G.711 at 20ms ptime.

Your answer:

Key Takeaway: SDP offer lists codecs in priority order; the answer selects a subset. G.729 = 20-byte payload vs G.711 = 160-byte. With 40-byte headers: 24 kbps (G.729) vs 80 kbps (G.711) on the wire.

Lab 3.2 Codec Mismatch — 488 Not Acceptable [Beginner]

File: `case_3_7a_488_codec_mismatch.pcap`

Learning Objectives

>> Identify a codec mismatch from SDP and the resulting 488 response

>> Explain why the call fails at codec negotiation stage

>> Read the Warning header for diagnostic detail

Setup

1. Open case_3_7a_488_codec_mismatch.pcap
2. Apply filter: sip

Exercises

1. Open Telephony > Flow Graph. At what stage does the call fail? What response code terminates it?

Your answer:

2. Find the INVITE. What PT(s) are offered in the m= line? Find the 488 and expand the Warning header. What reason is given?

Your answer:

3. What would the caller need to add to the SDP offer to make this call succeed? Which SDP line would change?

Your answer:

Key Takeaway: 488 Not Acceptable = no codec intersection between offer and answer. The Warning header often gives a machine-readable reason. Fix: always include G.711 (PT=0 or PT=8) as a fallback in every SDP offer.

Module 4 — Transfer, Fax, and Load

Lab 4.1 Successful Blind Transfer [Intermediate]

File: `case_3_10b_transfer_successful.pcap`

Learning Objectives

>> Trace the complete REFER/NOTIFY chain for a blind transfer

>> Identify when the transfer is confirmed and the original call released

>> Describe the role of each NOTIFY in transfer supervision

Setup

1. Open case_3_10b_transfer_successful.pcap
2. Apply filter: sip.Method == REFER || sip.Method == NOTIFY

Exercises

1. List all REFER and NOTIFY messages in order with their timestamps.

Your answer:

2. Click the 202 Accepted. Does 202 mean the transfer succeeded? What message actually confirms success?

Your answer:

3. Expand the body of each NOTIFY. What SIP status is reported in each? When does the original caller send BYE?

Your answer:

Key Takeaway: REFER > 202 means transfer initiated, not completed. Success is confirmed by NOTIFY body SIP/2.0 200 OK. Only after that NOTIFY should the transferring party send BYE.

Lab 4.2 T.38 Fax — Successful Negotiation [Intermediate]

File: `case_3_9a_t38_successful.pcap`

Learning Objectives

>> Identify the T.38 re-INVITE switching from audio to fax mode

>> Read T.38 SDP parameters (m=image, udptl)

>> Recognise UDPTL fax data in the capture

Setup

1. Open case_3_9a_t38_successful.pcap
2. Apply filter: sip.CSeq.method == INVITE

Exercises

1. How many INVITEs? What is different about the CSeq:2 INVITE compared to CSeq:1?

Your answer:

2. Click the CSeq:2 INVITE. Expand the SDP. What does the m= line show? How does it differ from CSeq:1 SDP?

Your answer:

3. Apply filter: t38. How many T.38 UDPTL packets? On which ports are they flowing?

Your answer:

Key Takeaway: T.38 is negotiated via a re-INVITE changing m=audio to m=image with udptl. If this re-INVITE is rejected (see case_3_9b), fax falls back to G.711 and almost always fails.

Lab 4.3 CPS Limit — 503 Storm Under Load [Intermediate]

File: `case_3_6a_503_cps_limit.pcap`

Learning Objectives

>> Identify a CPS limit from the pattern of 503 responses

>> Explain the significance of the Retry-After header

>> Calculate the call attempt rate from packet timestamps

Setup

1. Open case_3_6a_503_cps_limit.pcap
2. Apply filter: sip

Exercises

1. Apply filter: sip.Method == INVITE. How many INVITEs? Calculate time span. What is the call rate in calls/second?

Your answer:

2. Apply filter: sip.Status-Code == 503. How many 503 responses? Expand one and check the Retry-After header value.

Your answer:

3. Apply filter: sip.Status-Code == 100. How many 100 Trying? What does the ratio of 100s to 503s tell you about the CPS limit?

Your answer:

Key Takeaway: Some INVITEs getting 100 Trying while others get 503 = CPS rate limit at the carrier. Retry-After tells you how long to wait. Fix: spread call attempts over time, or negotiate a higher CPS allowance.

Quick Reference

Task	How
Set time reference	Right-click any packet > Set/Unset Time Reference
View call ladder	Telephony > Flow Graph > Limit to display filter
Analyse RTP quality	Telephony > RTP > RTP Streams > select > Analyze
Play back audio	Telephony > RTP > RTP Streams > select > Play
Isolate one call	sip.Call-ID == value (copy from any SIP packet)
Export call as pcap	Telephony > VoIP Calls > select call > Export
Merge two captures	mergecap -w combined.pcap near.pcap far.pcap
Extract SIP	tshark -r big.pcap -w sip.pcap -Y 'sip'
Find fragmented SIP	ip.flags.mf == 1 || ip.frag_offset > 0
Audit RTP QoS	rtp && ip.dsfield.dscp != 46
Add Delta column	Right-click column header > Column Prefs > Add: frame.time_delta_displayed