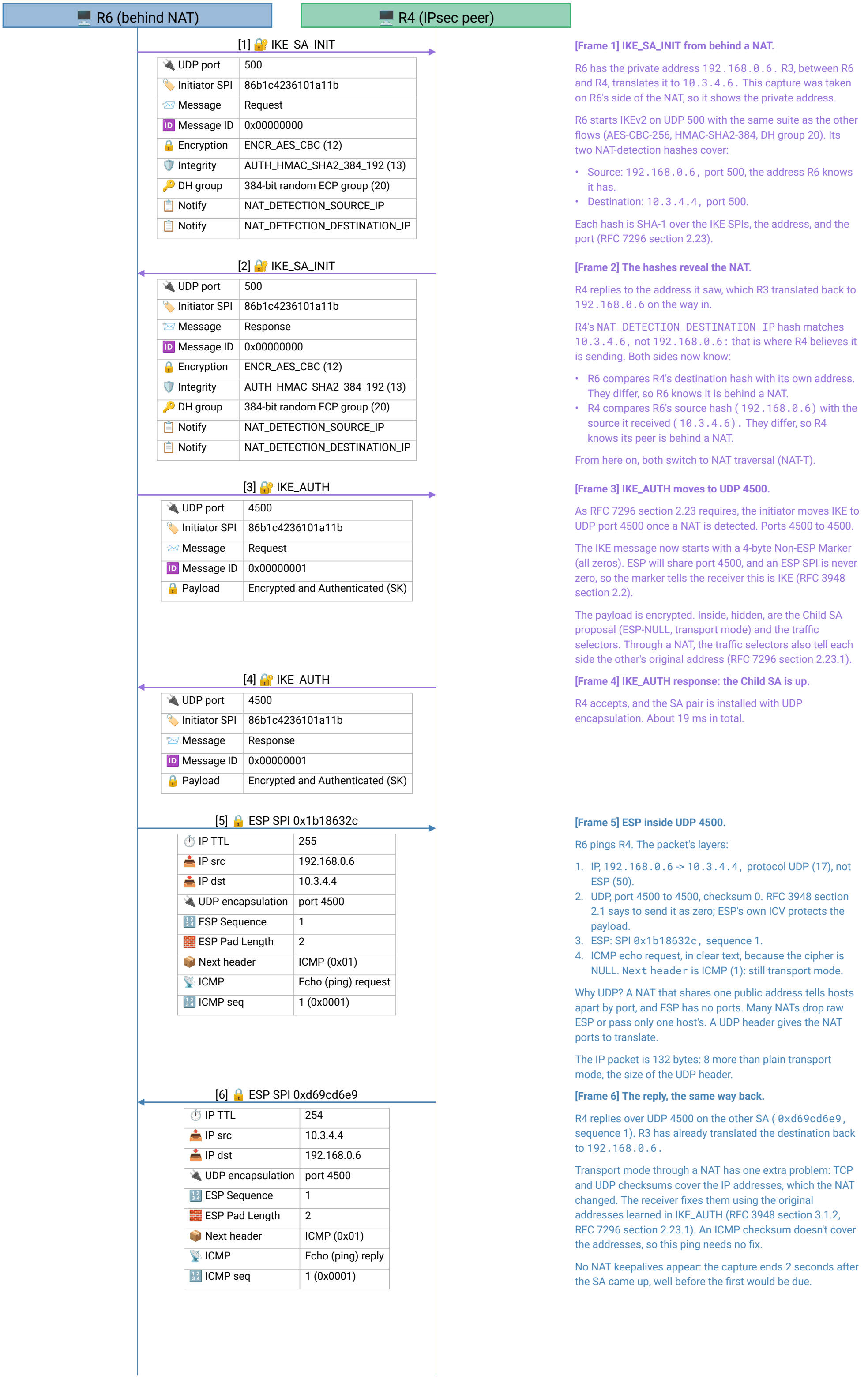
ipsec-esp-transport-nat-t.annotated



[Frame 1] IKE_SA_INIT from behind a NAT.

R6 has the private address `192.168.0.6`. R3, between R6 and R4, translates it to `10.3.4.6`. This capture was taken on R6's side of the NAT, so it shows the private address.

R6 starts IKEv2 on UDP 500 with the same suite as the other flows (AES-CBC-256, HMAC-SHA2-384, DH group 20). Its two NAT-detection hashes cover:

- Source: `192.168.0.6`, port 500, the address R6 knows it has.
- Destination: `10.3.4.4`, port 500.

Each hash is SHA-1 over the IKE SPIs, the address, and the port (RFC 7296 section 2.23).

[Frame 2] The hashes reveal the NAT.

R4 replies to the address it saw, which R3 translated back to `192.168.0.6` on the way in.

R4's `NAT_DETECTION_DESTINATION_IP` hash matches `10.3.4.6`, not `192.168.0.6`: that is where R4 believes it is sending. Both sides now know:

- R6 compares R4's destination hash with its own address. They differ, so R6 knows it is behind a NAT.
- R4 compares R6's source hash (`192.168.0.6`) with the source it received (`10.3.4.6`). They differ, so R4 knows its peer is behind a NAT.

From here on, both switch to NAT traversal (NAT-T).

[Frame 3] IKE_AUTH moves to UDP 4500.

As RFC 7296 section 2.23 requires, the initiator moves IKE to UDP port 4500 once a NAT is detected. Ports 4500 to 4500.

The IKE message now starts with a 4-byte Non-ESP Marker (all zeros). ESP will share port 4500, and an ESP SPI is never zero, so the marker tells the receiver this is IKE (RFC 3948 section 2.2).

The payload is encrypted. Inside, hidden, are the Child SA proposal (ESP-NULL, transport mode) and the traffic selectors. Through a NAT, the traffic selectors also tell each side the other's original address (RFC 7296 section 2.23.1).

[Frame 4] IKE_AUTH response: the Child SA is up.

R4 accepts, and the SA pair is installed with UDP encapsulation. About 19 ms in total.

[Frame 5] ESP inside UDP 4500.

R6 pings R4. The packet's layers:

1. IP, `192.168.0.6` -> `10.3.4.4`, protocol UDP (17), not ESP (50).
2. UDP, port 4500 to 4500, checksum 0. RFC 3948 section 2.1 says to send it as zero; ESP's own ICV protects the payload.
3. ESP: SPI `0x1b18632c`, sequence 1.
4. ICMP echo request, in clear text, because the cipher is NULL. `Next header` is ICMP (1): still transport mode.

Why UDP? A NAT that shares one public address tells hosts apart by port, and ESP has no ports. Many NATs drop raw ESP or pass only one host's. A UDP header gives the NAT ports to translate.

The IP packet is 132 bytes: 8 more than plain transport mode, the size of the UDP header.

[Frame 6] The reply, the same way back.

R4 replies over UDP 4500 on the other SA (`0xd69cd6e9`, sequence 1). R3 has already translated the destination back to `192.168.0.6`.

Transport mode through a NAT has one extra problem: TCP and UDP checksums cover the IP addresses, which the NAT changed. The receiver fixes them using the original addresses learned in IKE_AUTH (RFC 3948 section 3.1.2, RFC 7296 section 2.23.1). An ICMP checksum doesn't cover the addresses, so this ping needs no fix.

No NAT keepalives appear: the capture ends 2 seconds after the SA came up, well before the first would be due.