



A note from the student behind this lab

This is an **independent, free, learning-only student project**. It respects every company and vendor, and every trademark belongs to its owner. It is **not affiliated with or endorsed by** any of them.

If anyone has a concern, the content will be removed or corrected promptly: please write via <https://cyphermonkeylabs.com/#contact>. Nothing here is sold, and I am happy to volunteer for any organization.

Transactions and dialogs

A transaction is identified by the branch, and a retransmission repeats that branch instead of starting a new one.

One branch, one transaction

The branch parameter identifies the transaction. An implementation of RFC 3261 starts it with z9hG4bK and makes the rest unique across space and time. A retransmission sends the same request again, including that branch.

CANCEL is an exception: it uses the same branch as the INVITE it cancels. The ACK for a non-2xx final response uses that same branch too. The ACK for a 2xx does not. It is a different transaction, so it gets a new branch.

INVITE is the special state machine

On UDP the client starts in Calling, retransmits at T1 doubling, and starts Timer B at 64 times T1. A provisional response moves it to Proceeding and stops those retransmissions. If Timer B fires, the client must not invent an ACK.

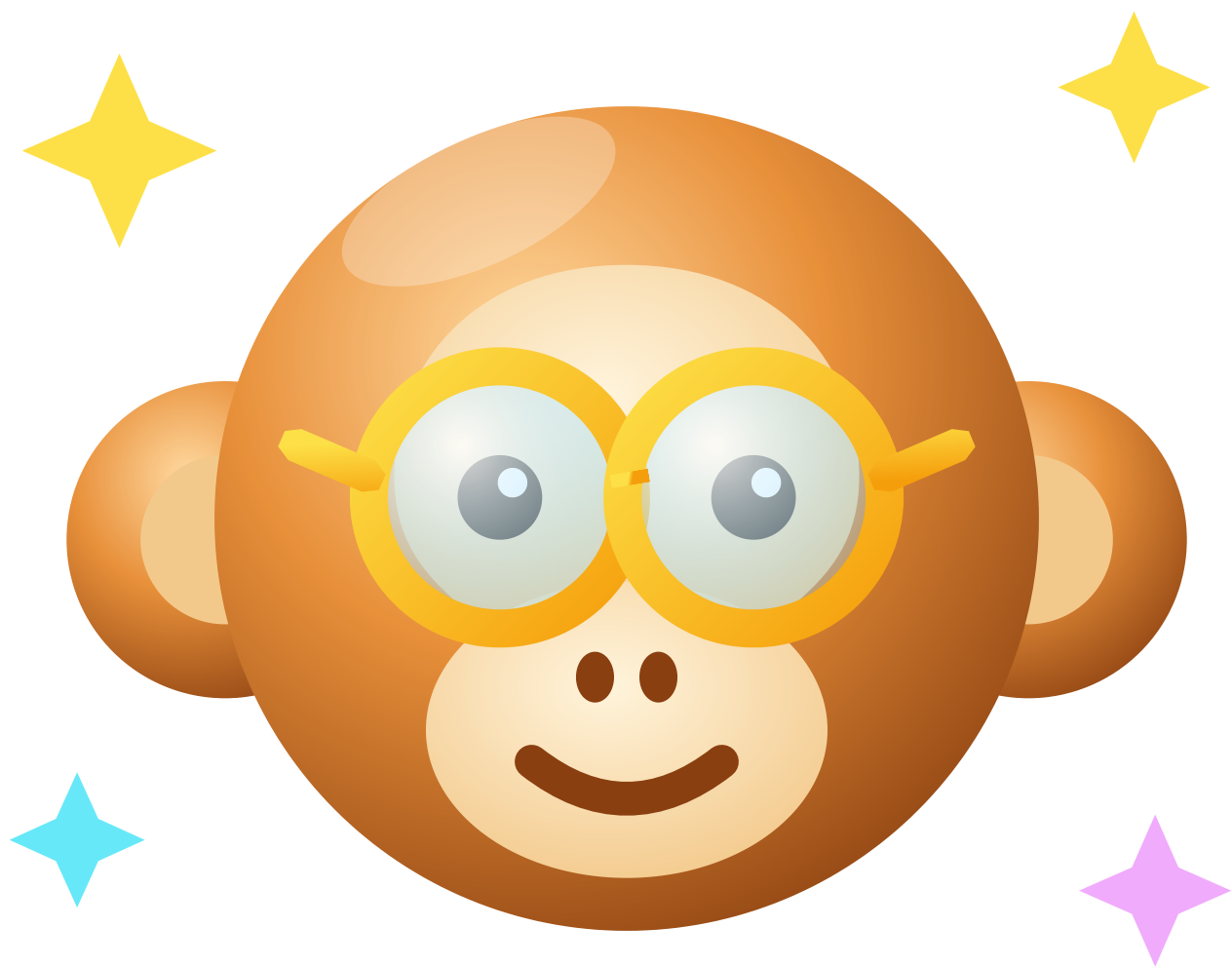
The overview in section 17 calls INVITE a three-way handshake. RFC 6026 corrects the 2xx path: the transaction user sends the end-to-end ACK, and the transaction layer absorbs late 2xx responses. The non-2xx ACK is still generated hop by hop inside the INVITE transaction.

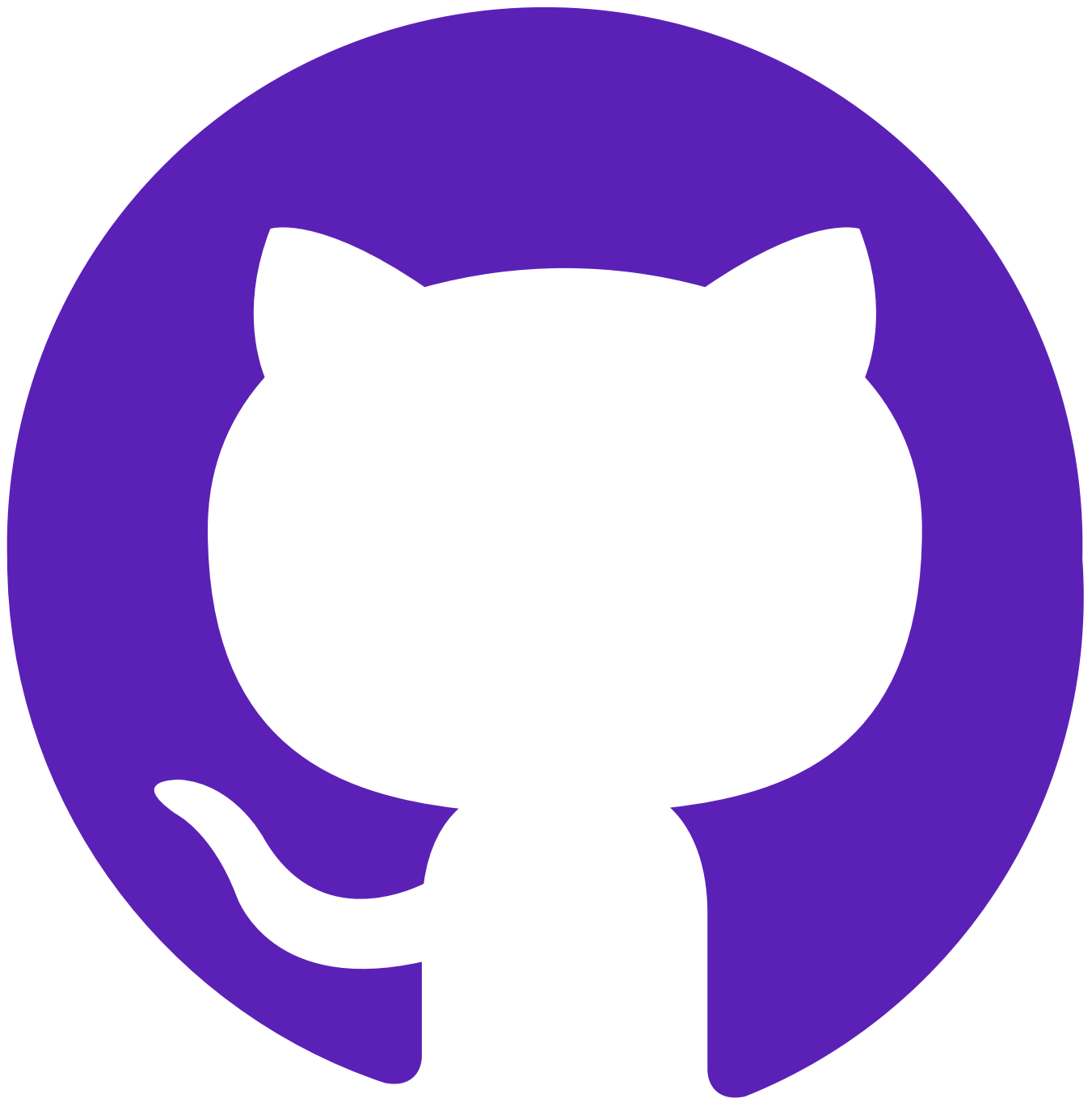
Room: [Transactions and the branch cookie](#)

Sources and go deeper

- [RFC 3261 §8.1.1.7](#) — last checked 2026-10
- [RFC 3261 §17](#) — last checked 2026-10
- [RFC 6026](#) — last checked 2026-10

CypherMonkey Labs







© 2026 CypherMonkeyLabs 🐵

[Open app](#) · [View repo](#)