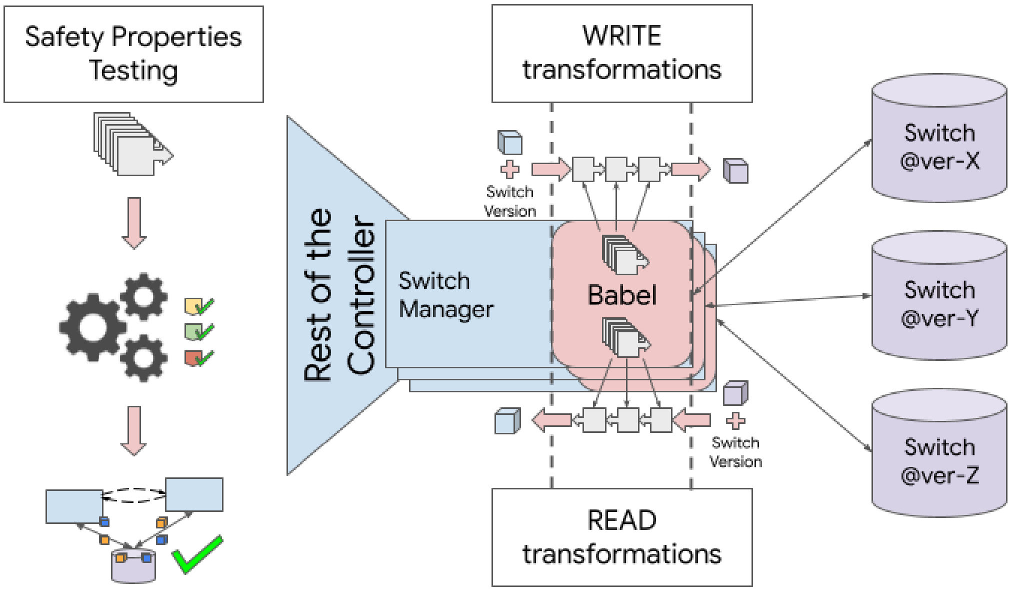
Babel: The tower so far

What is Babel

Babel is a migration framework that resolves compatibility issues with high confidence by enabling per-migration user-written transformations. High confidence is provided by safety properties that are enforced using simple-to-reason-about and easy-to-amend testing.

Within the context of SDN, Babel serves as a transparent layer between the controller and switches, providing the controller the illusion of a fixed switch version. This illusion is achieved by applying user-provided transformations based on switch version during reads and writes.

Babel is deployed at Google and has been successfully used to perform migrations.



Real-World Migrations

We undertook 5 migrations encompassing 3 non-breaking changes and 2 breaking changes. Out of those, 3 migrations are fully completed, and 2 are still underway. We will discuss the specifics of these migrations and how Babel helped us roll them out.

Observations and Impact

- Although pre-Babel migrations required much effort, roll-backs were a common occurrence. Babel migrations have required comparatively little effort (especially mental) and finished Babel migrations have so far never required roll-backs! In fact, a difficult pre-Babel migration was successfully rolled out using Babel.
- Coordination required for migrations was significantly reduced. Only requires an agreed upon representation and version number, no need for strict timeline dependencies and discussion of implementation details.
- Testing Babel migrations only requires new, formulaic test cases. There is no need to write any new tests from scratch, letting you avoid tests specific to a migration.
- Adding and removing Babel migrations is simple and easy, allowing us to monitor the progress of migrations with minimal effort.
- End-to-end effort (designing, implementing and rolling-out) reduced from years to weeks.

Key Takeaways

Babel is a migration framework that:

- resolves controller-switch compatibility issues due to version skew,
- ensures rollouts are automatically safe, no manual reasoning required,
- and greatly reduces effort for API migrations (from X months to X days).

Babel significantly increases the velocity and predictability of switch API changes. It makes API design decisions reversible, enabling quicker decision-making and easier tech-debt reduction.

Future Work

Babel support for Non-stop Forwarding (NSF). Babel allows a controller to read and write to a switch, but Babel does not provide the controller with any information regarding the version of entries on a switch. Babel cannot remove support for a migration until all entries written for a specific version have been removed. Previously, this was not a major concern as switch updates required removing switches from the dataplane, which allowed us to clear and reinstall all entries translated for the new version.

NSF allows non-breaking (with respect to the current entries on a switch) changes to be made without bringing down the dataplane, meaning Babel (and the switch) may have to support a migration indefinitely due to stale entries. Since we want to be able to get rid of unused portions of our switch API, Babel must find a way to ensure that these stale entries are updated.