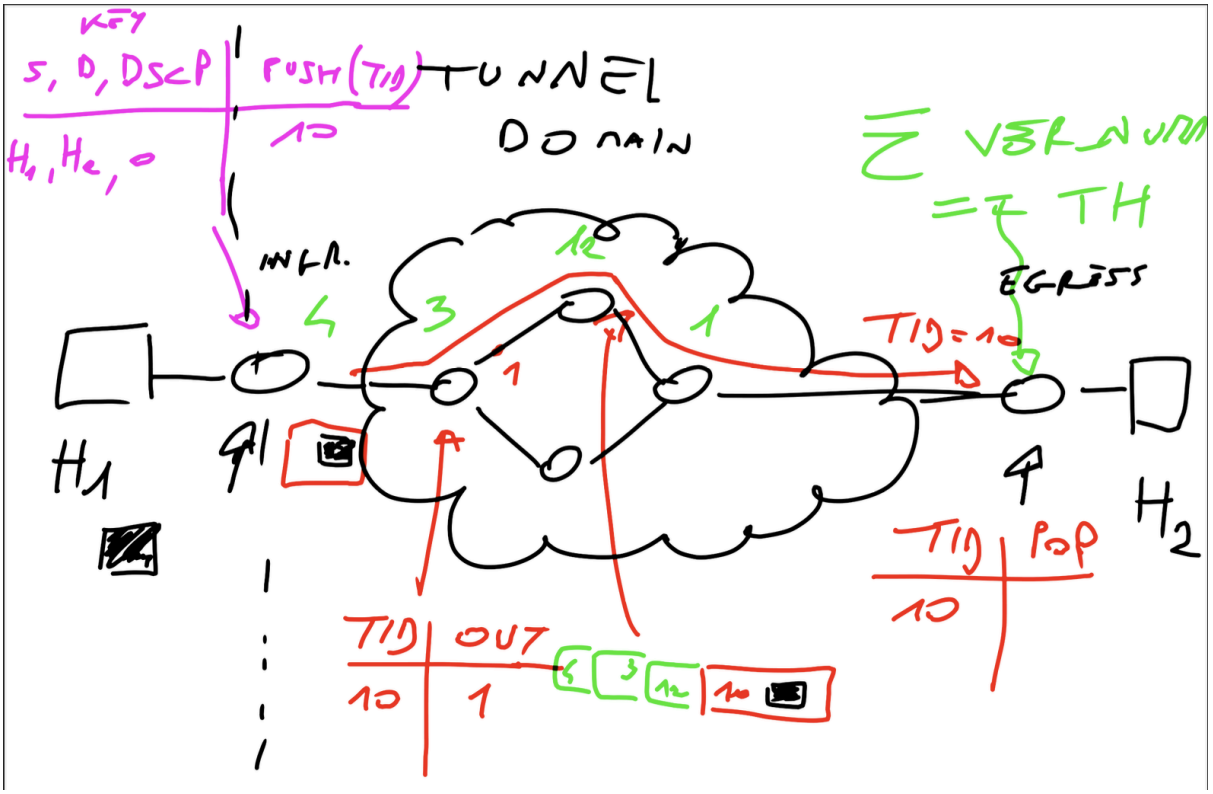
Project Assignment: Tunnel-Based Forwarding and Proof of Transit in P4

Objective

Design and implement in P4 a network pipeline where packets are forwarded based on a **custom tunnel identifier**, rather than destination IP. The ingress switch classifies packets and inserts a **tunnel header** with a chosen **tunnel ID**. Transit switches forward packets based on the tunnel ID, and contributes to a **path validation mechanism** by appending headers to a **validation stack**. At the egress, the validation stack is processed and the packet is either accepted or dropped based on a threshold.



System Overview

- A custom header `tunnel_h` is used for **forwarding**, and contains:
- A **validation stack** is built using `validation_h` headers, one per transit switch.
    - Each contains a hop-specific integer value.

- The **ingress switch**:
    - Chooses the tunnel ID based on `IPv4 src`, `dst`, and `DSCP`.
    - Pushes the tunnel header and initializes the stack.
- **Transit switches**:
    - Forward packets based on `tunnel_id`.
    - Append one `validation_h` header with a configurable value.
- The **egress switch**:
    - Processes the validation stack.
    - Sums all values.
    - Compares the result to a per-tunnel **threshold**.
    - Accepts or drops the packet accordingly.

Project Tasks

Ingress Switch

- Match on `IPv4 src`, `dst`, and `DSCP` to select a `tunnel_id`.
- Push `tunnel_h` with selected ID.
- Push an initial `validation_h`.

Transit Switches

- Match on `tunnel_id` to decide the next hop.
- Append a `validation_h` to the stack with a configured `hop_value`.

Egress Switch

- Match on `tunnel_id`.
- Iterate over the `validation_stack`:

    - Sum all `hop_value`s.

- Compare the sum to a **threshold** for that tunnel:
    - If `sum ≥ threshold`: forward to final destination (based on IPv4).
    - Else: drop the packet.