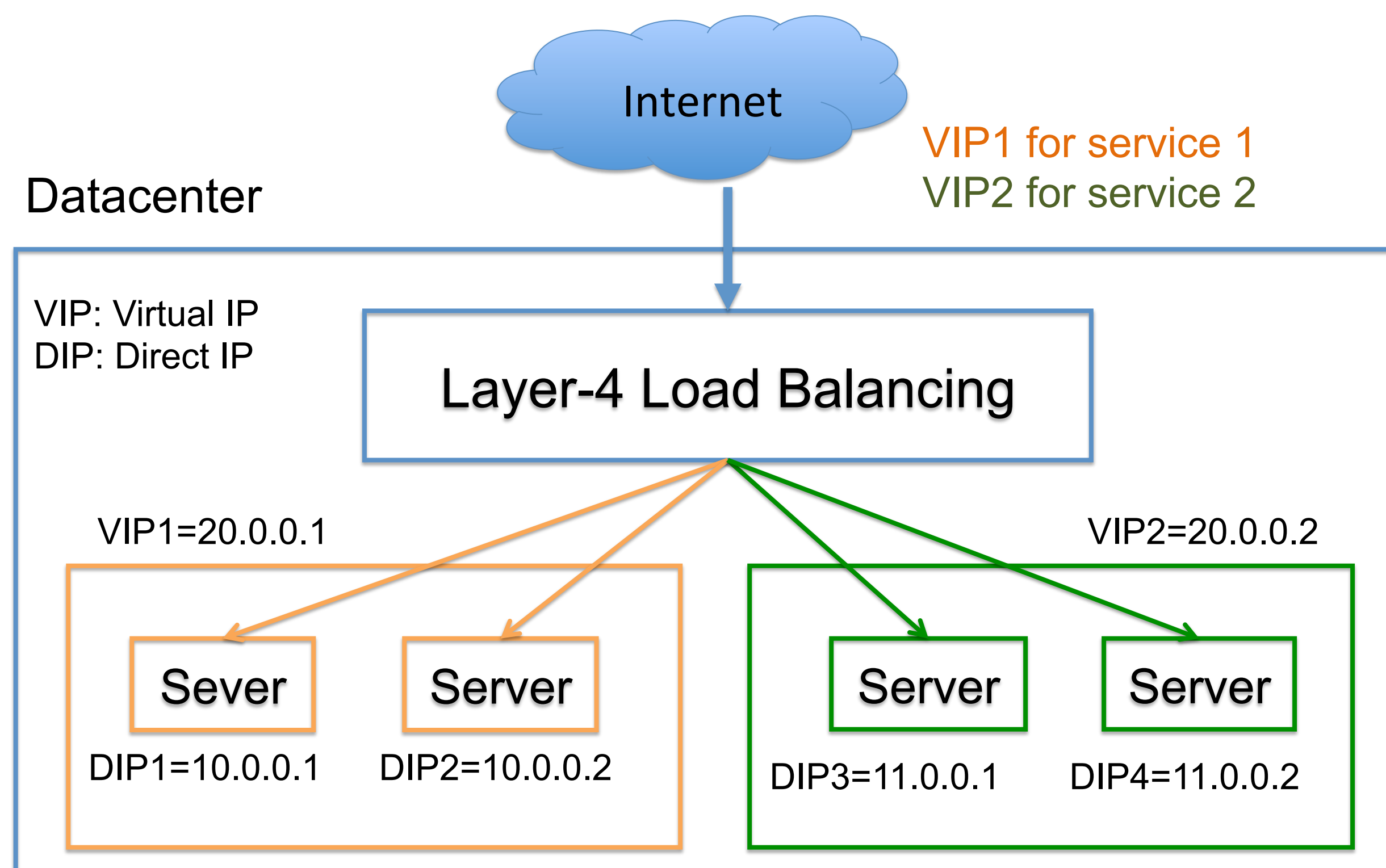


1. Layer-4 Load Balancing

Layer-4 load-balancing is widely used for scale and fault-tolerance in cloud services.



Software load-balancing (on x86 servers) is very common, but incurs **high cost and latency**

Key Idea: Add layer-4 load-balancing to every switch using P4.

2. Problem

All packets of a connection must be forwarded to the same server, even when server pool changes (aka “**per-connection consistency**”).

Why it is hard:

1. **Tracking 10M connections at line-rate**

2. **Consistent connection-to-DIP mapping**

DIP pool change shouldn't break on-going connections

3) **Robustness under scanning attacks (DDoS)**

Maintain only “legitimate” 10M connections

3. Our Approach

1. **Scale:** use hash-digest to save H/W memory

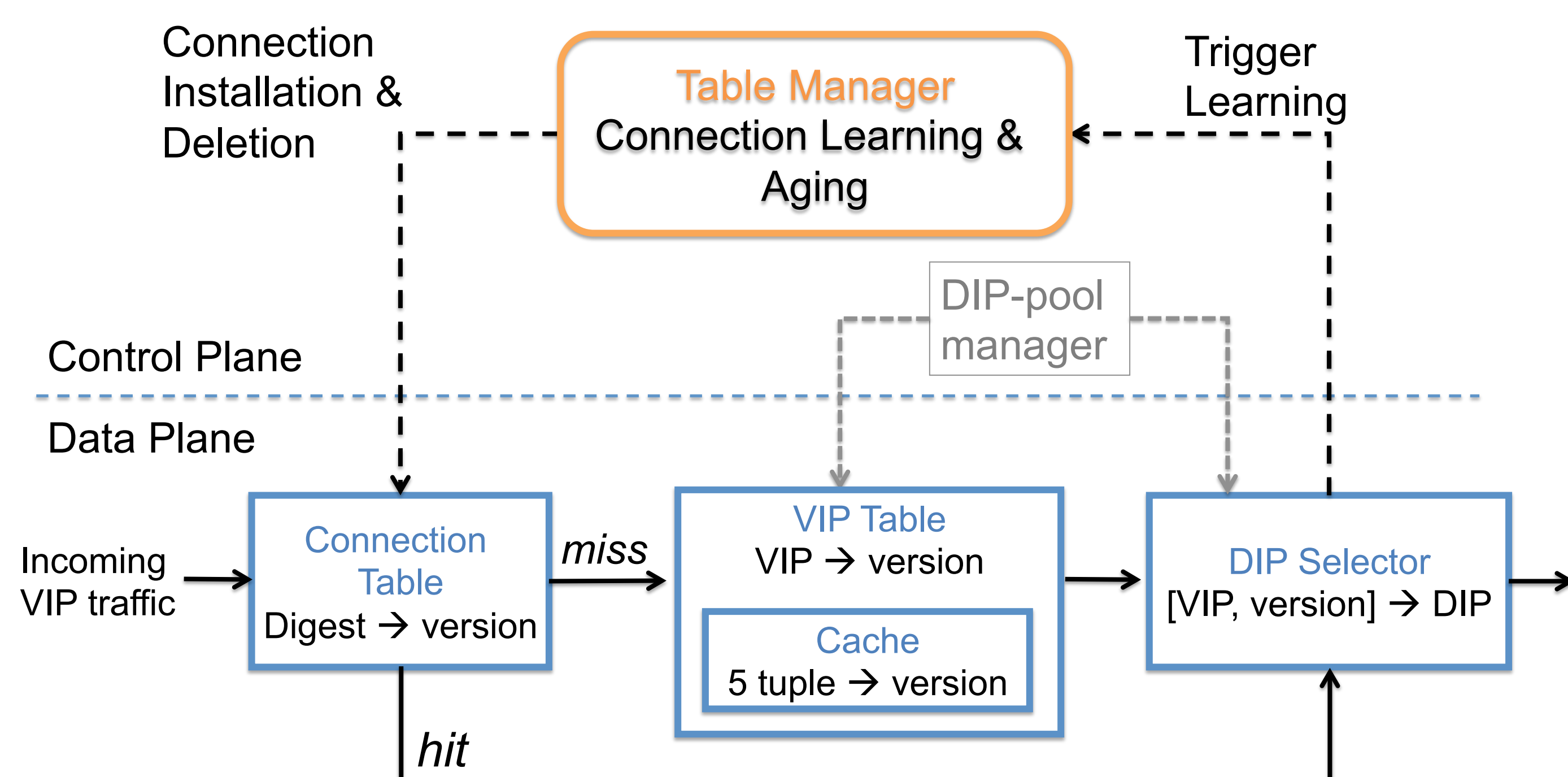
104 bits (5-tuple) \rightarrow 24 bits (16M digests)

Multi-stage cuckoo hashing for high-utilization at low collision rate.

2. **Consistency:** update DIP-pool by 2-phase commit

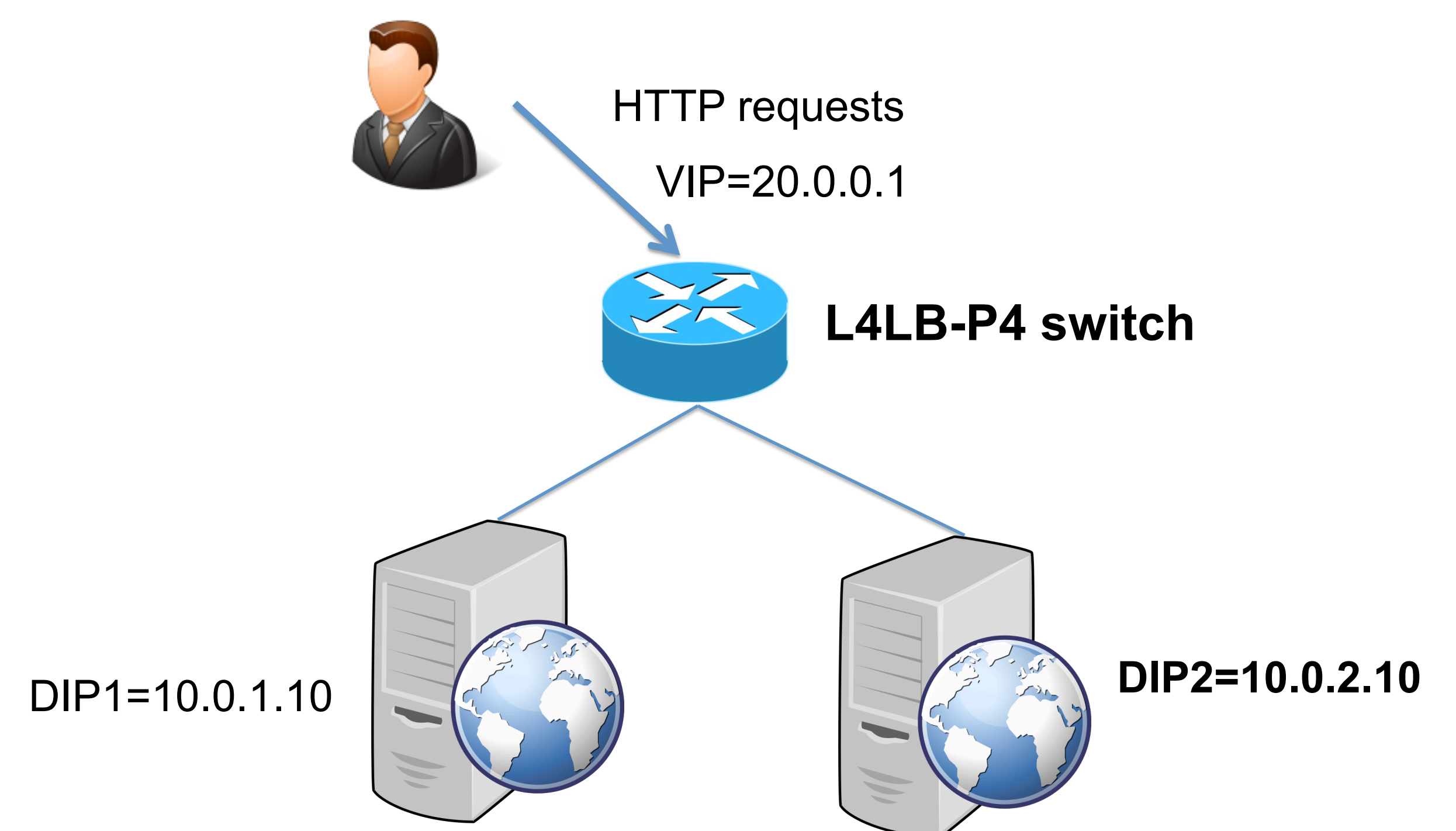
Differentiate connections before/after the update.

During update, cache per-connection mappings in hw.

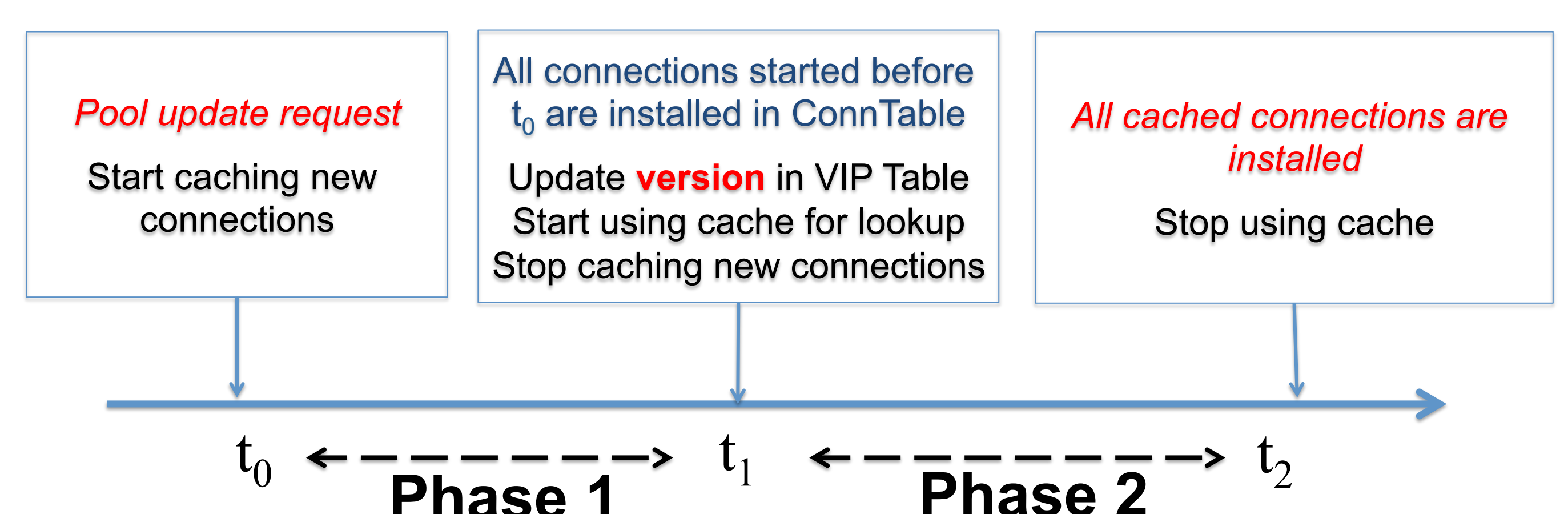


3. **Robustness:** TCP SYN authentication

4. P4 Demo Scenario



DIP Pool Version 1		DIP Pool Version 2	
VIP	DIP	VIP	DIP
20.0.0.1:80	10.0.1.10:8080	20.0.0.1:80	10.0.1.10:8080
			10.0.2.10:8080



Connection-to-version mapping kept consistent during DIP table update $[t_1, t_2]$