# Hierarchical SDN to Scale the DC/Cloud to Tens of Millions of Endpoints at Low Cost

Luyuan Fang
lufang@microsoft.com

**IETF 91, SDNRG**

**November 10, 2014**

# This Is the Hyper-Scale Cloud

- The cloud is growing at unprecedented rate
- We need to be able to scale to 10’s of millions of underlay network endpoints (servers) and 100’s of millions of VMs and virtualized network functions

A scale we have never seen before, and we are just at the beginning…

# Example: The Scale and Growth of Microsoft Cloud

## Microsoft Azure Numbers

**>10,000**
New Azure customers a week

**1,200,000**
SQL databases In Azure

**>30 Trillion**
Storage objects in In Azure

**350 Million**
Azure Active Directory Users

**>18 Billion**
Azure Active Directory authentications/week

**>2 Million**
Developers with Virtual Studio Online

**+10%**
Growth MoM

## Huge Infrastructure Scale

| Microsoft Cloud | Quantity |
|---|---|
| Data Centers | 100+ |
| Global Regions | 19 |
| Servers | 1,000,000+ |

## Azure Cloud Growth

| Azure Components | 2010 | 2014 |
|---|---|---|
| Azure Compute Instances | 100K | Millions |
| Azure Storage | 10s of PBs | Exabytes |
| Azure DC Network Capacity | 10s of Tbps | Pbps |

# Underlay Network Challenges

- Scale at low-cost, use commodity HW
  - Use small FIBs/LFIBs in all network nodes, avoid FIB explosion
- Achieve high resource utilization
  - Efficiently support ECMP and any-to-any, server-to-server TE
- Scale at low operational and computational complexity
  - Locally minimize complexity and network state, with no information loss
- Scale while achieving improved cloud elasticity and service velocity
  - Overcome today's challenges of NFV (e.g. SLB) scalability and VM/NFV mobility

This is a whole new game, no existing solutions for such a scale

# Choice of Technologies: MPLS + SDN

- MPLS
  - Unify forwarding (DC and core), no IP lookup other than at the edge/server
  - Flexibility of the label stack, naturally suitable for hierarchical decomposition
  - Ease of redirection, can be leveraged to increase elasticity
  - Security advantages
- SDN
  - Allow decoupling of control plane and data plane, make HW fungible
  - Take ownership of control plane development (short release cycle for bug-fixes and new features)
  - Reduce number of protocols
  - Make global optimization possible

# Hierarchical SDN (HSDN)

> *… SDN provides the discipline to extract simplicity... Abstractions are key…*
>
> ***SDN is all in decomposing problems into basic components…***
>
> SCOTT SHENKER

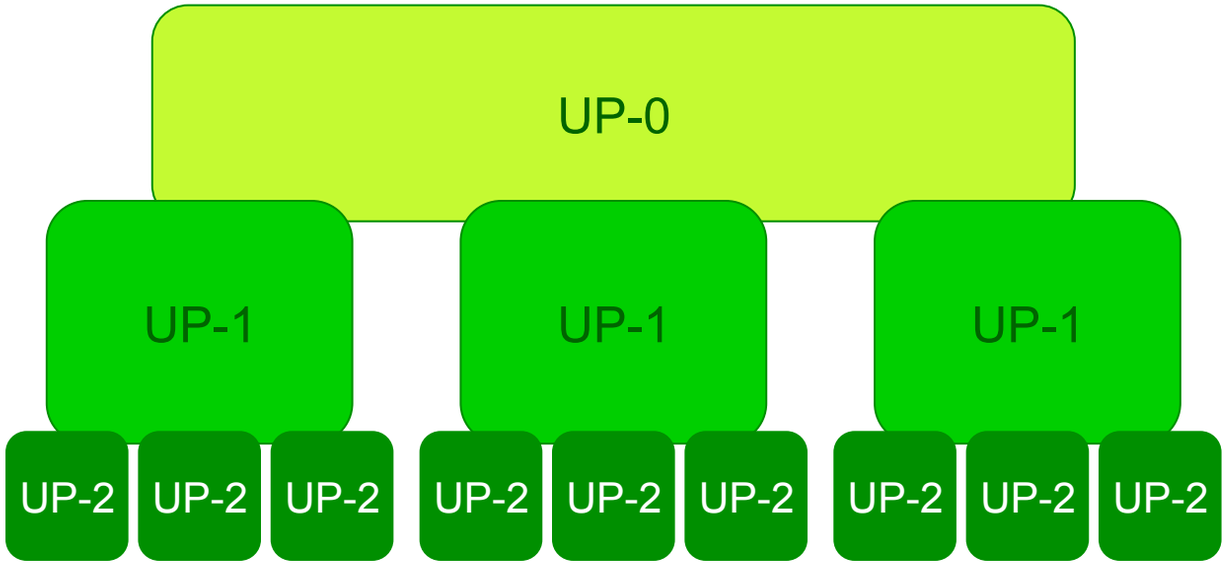
- HSDN is an industry-first architectural framework to consistently decompose many complex hyper-scale problems into manageable ones, so we scale in an optimal way


- **draft-fang-mpls-hsdn-for-hsdc-00**

# HSDN – One Fundamental Abstraction for Both Forwarding and Control

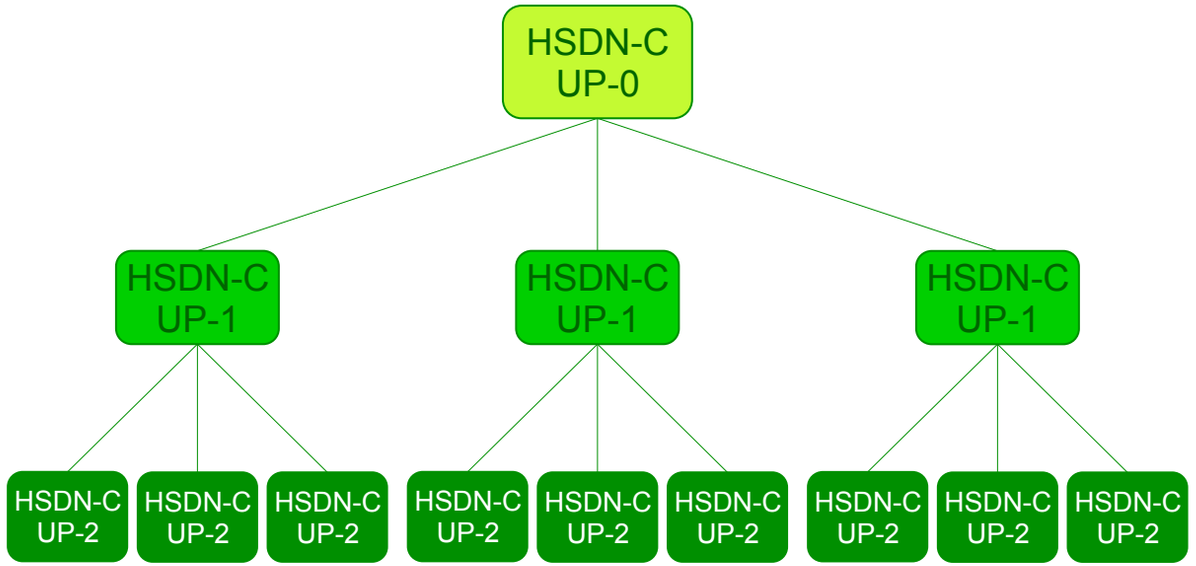
## Forwarding Plane

### HIERARCHICAL UNDERLAY PARTITION (UP)




## Control Plane

### HIERARCHICAL CONTROL




- **Keep number of destinations in all domains small**
- **Locally, hide destination explosion using hierarchical partitioning**

## One Consistent Abstraction Paradigm

- **Divide and conquer**
- **Keep all domains balanced and small**
- **Locally minimize network state**

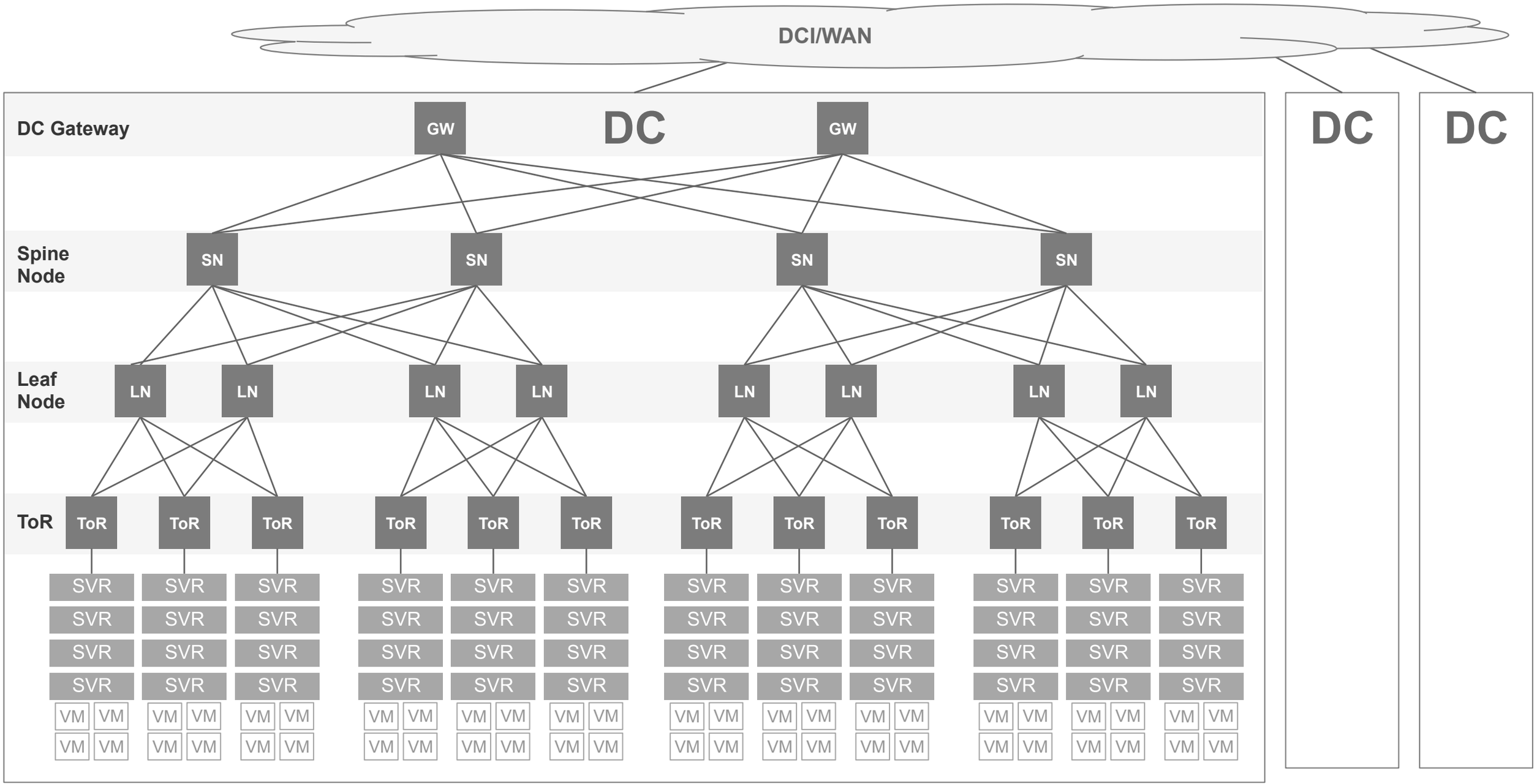
**→ “Infinite” Horizontal Scaling**

- **Keep number of paths per domain manageable**
- **Keep computational complexity per domain small**

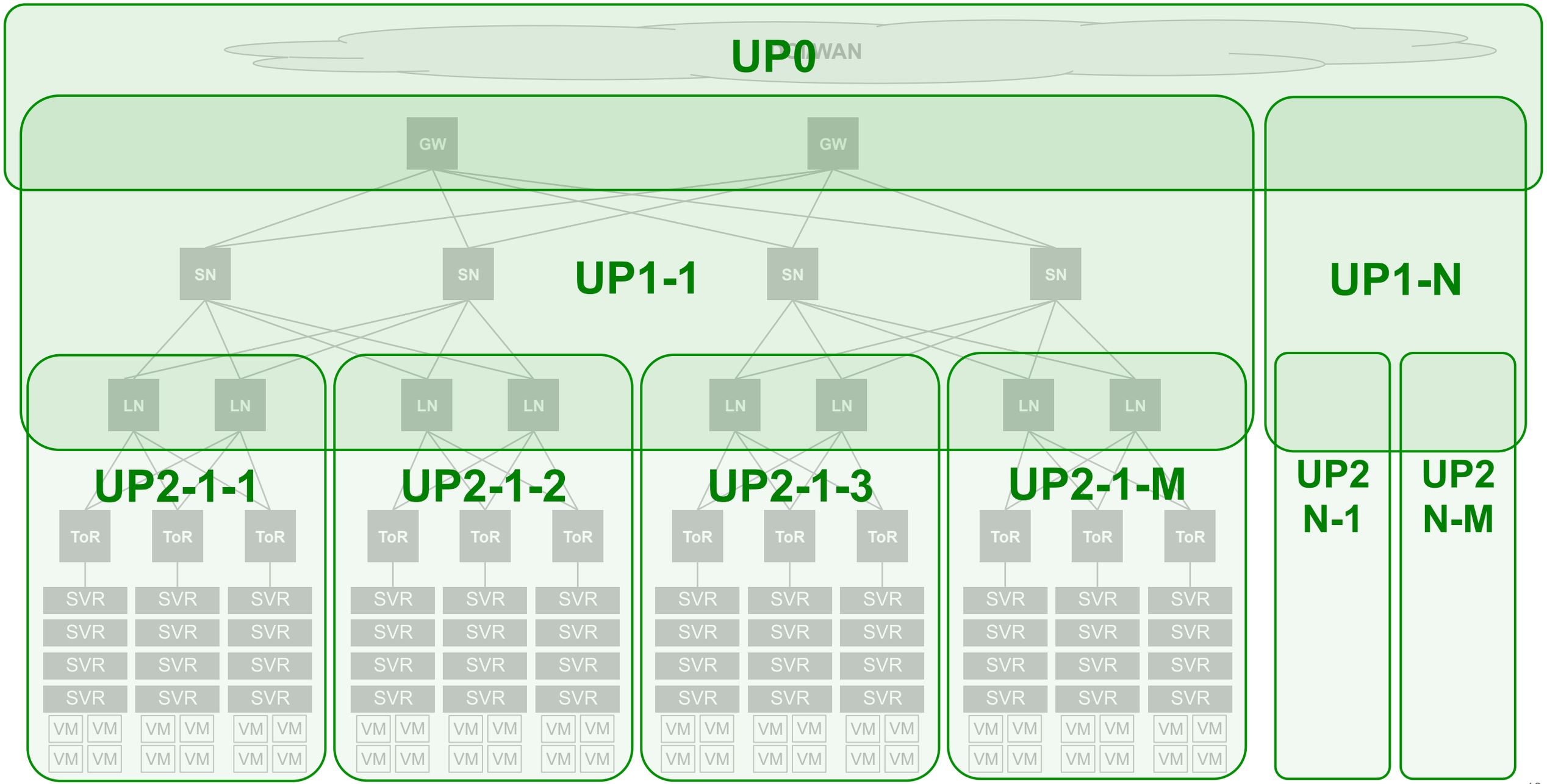
# HSDN Forwarding Plane

- Divide the DC and DCI/WAN in a hierarchically-partitioned structure
- Assign groups of Underlay Partition Border Nodes (UPBNs) in charge of forwarding within each partition
- Construct HSDN MPLS label stacks to identify the end points according to the HSDN structure
- Forward using the HSDN MPLS labels

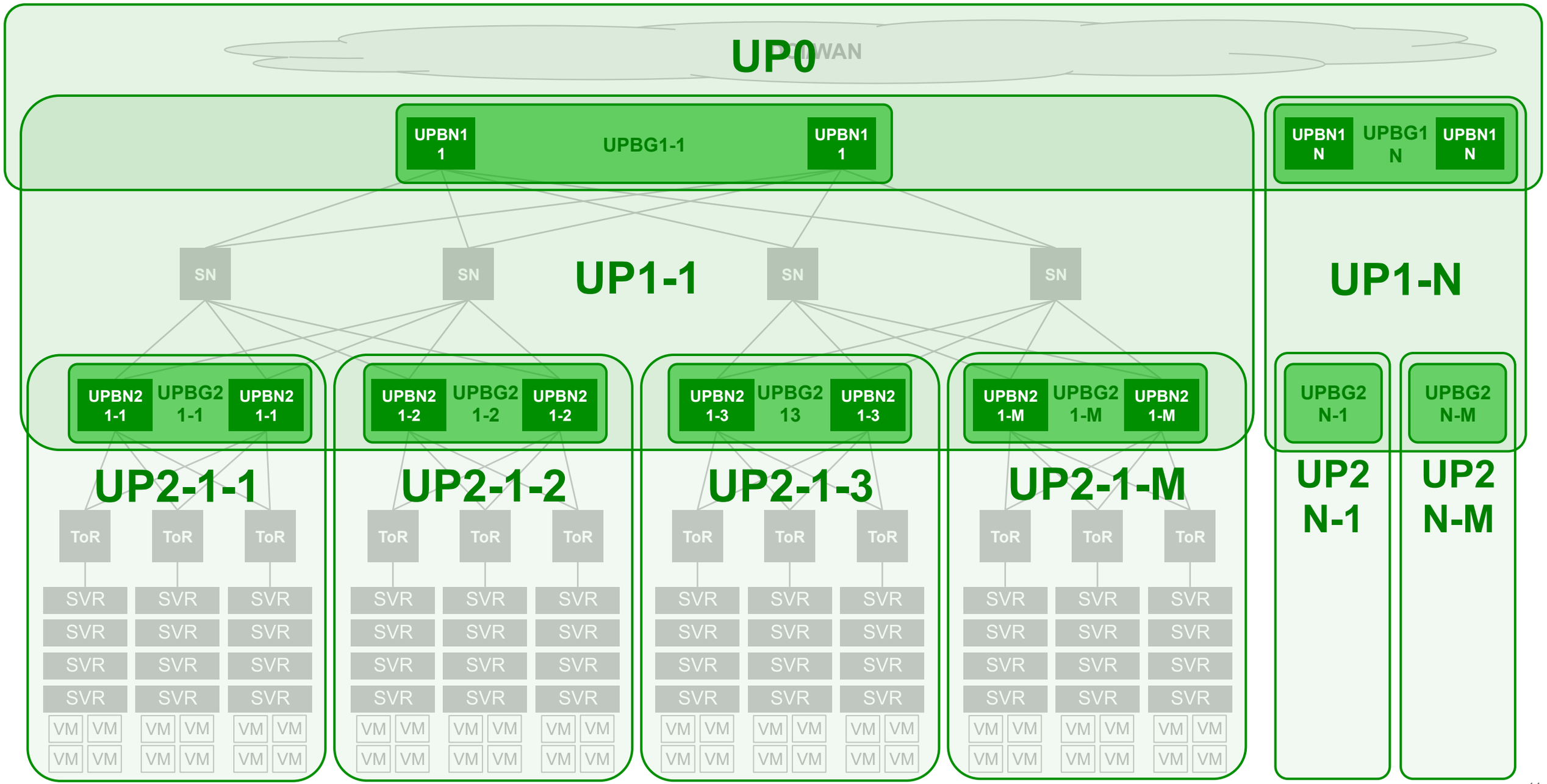
# Typical Clos-Based DC Topology, Spine and Leaf Architecture



# HSDN: Hierarchical Underlay Partitioning

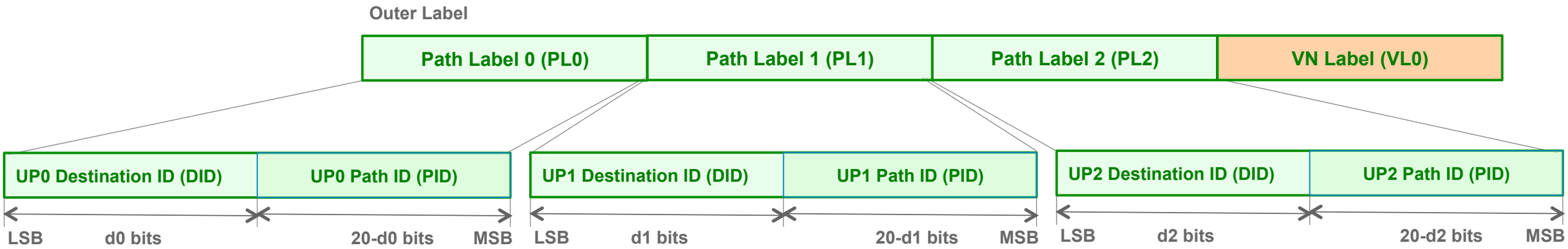


# HSDN: Assign UPBNs and UPBGs

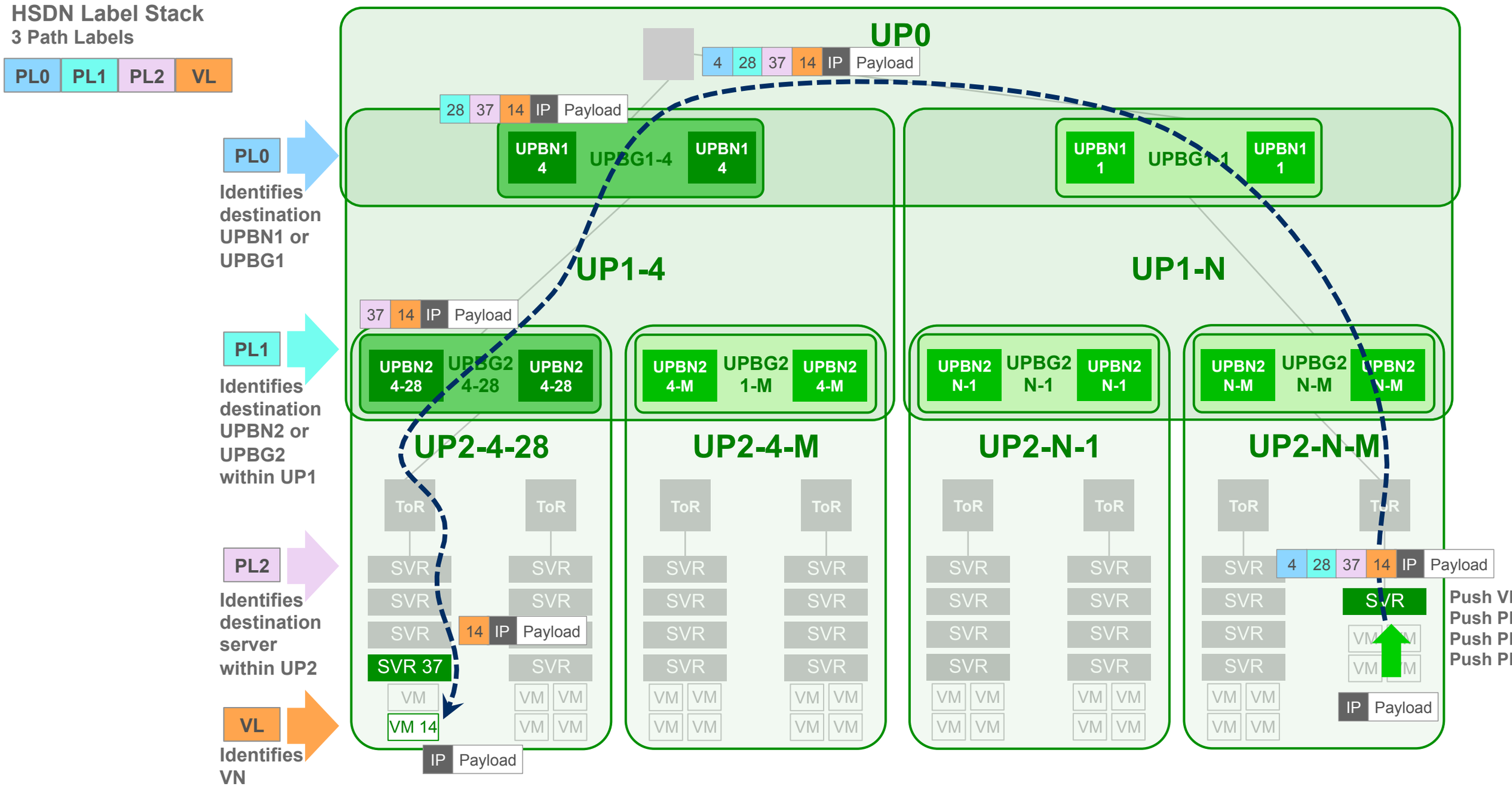


# HSDN Label Stack

- Stack of path labels, plus one VN label
- One path label per level of underlay partition



# HSDN Forwarding: Life of a Packet



# HSDN Control Plane

- HSDN Controller (HSDN-C) is horizontally scalable
  - Implemented as a set of local partition controllers HSDN-C-UP, following the HSDN hierarchy
  - Each HSDN-C-UP operates largely independently
  - Locally-reduced computational complexity for many functions, including TE
- Network state also distributed according to the HSDN hierarchy
  - Forwarding state is still in the network nodes, and is locally minimized
- HSDN supports both controller-centric SDN approach and traditional distributed routing/label distribution protocol approach
  - Useful during migration from legacy to full SDN (e.g., use BGP-LU for label distribution, RFC 3107)

# HSDN Scaling Examples

**HSDN scales to tens of millions of underlay network endpoints with small LFIBs**

- Assumptions
  - N hyper-scale DCs interconnected through DCI/WAN
  - DC fabrics are S-stage, asymmetrical, fat-Clos-based
- Support any-to-any, server-to-server
  - non-TE traffic with ECMP load balancing
  - TE traffic
- Max LFIB size (the largest LFIB size among all Tiers of switches) is as follows:

| Number of Server endpoints | Max LFIB size<br>ECMP only (No TE) | Max LFIB size<br>ECMP and TE Concurrently |
|---|---|---|
| 3 M | ~ 1K | < 14K |
| 10 M | < 2K | < 24K |
| 40 M | < 3K | < 36K |

# Conclusions

- Hyper-scale cloud growing at rate never seen before, need new solutions to scale
- HSDN is a novel paradigm to scale forwarding and control plane to unprecedented levels, at low cost
  - Scales to tens of millions of servers with very small LFIBs
  - Supports ECMP and any-to-any end-to-end TE concurrently with ease
  - Makes centralized SDN control and optimization practical at scale
  - Minimizes operation complexity, network state, and computation
  - Improves elasticity by simplifying and scaling NFV and overlay network mobility
- For more details, see draft-fang-mpls-hsdn-for-hsdc-00