

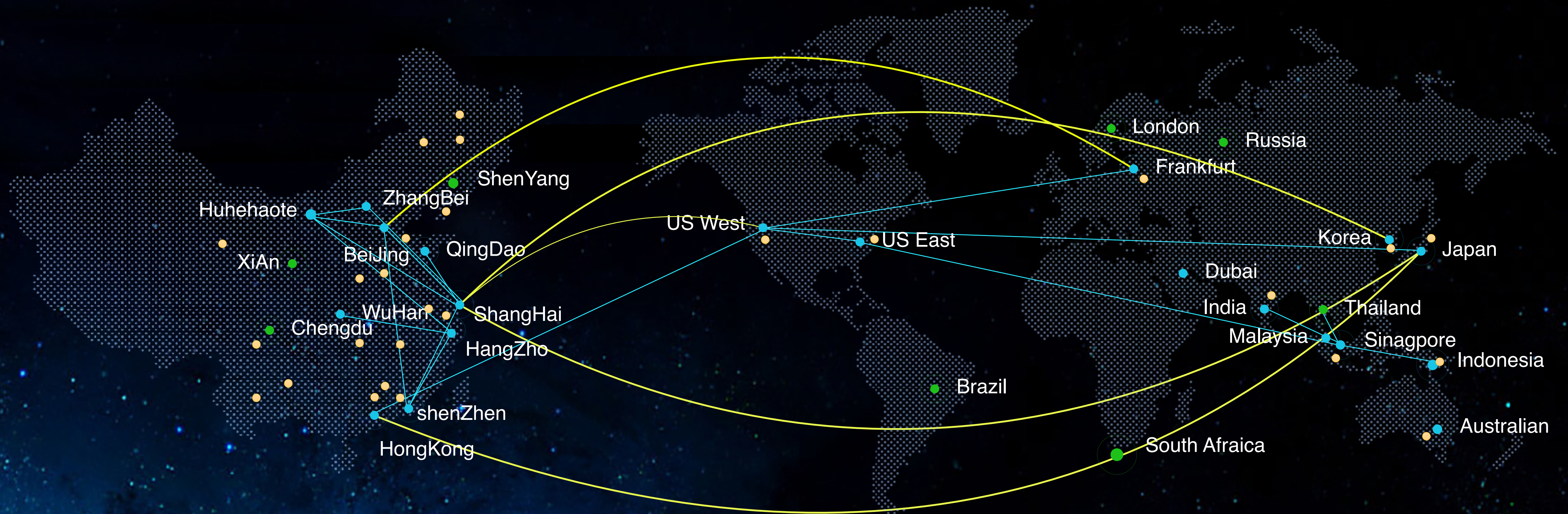
Simplify Distributed Rate-Limiting in Overlay Cloud Network with FDRL

Alibaba Cloud Senior Technical Expert, Stephen Xu

Agenda

- Alibaba cloud network infrastructure Introduction
- What's problem we meet in Cloud overlay network?
- Key idea of FDRL
- Experiment in VPP

Global Alibaba Cloud Network Infrastructure

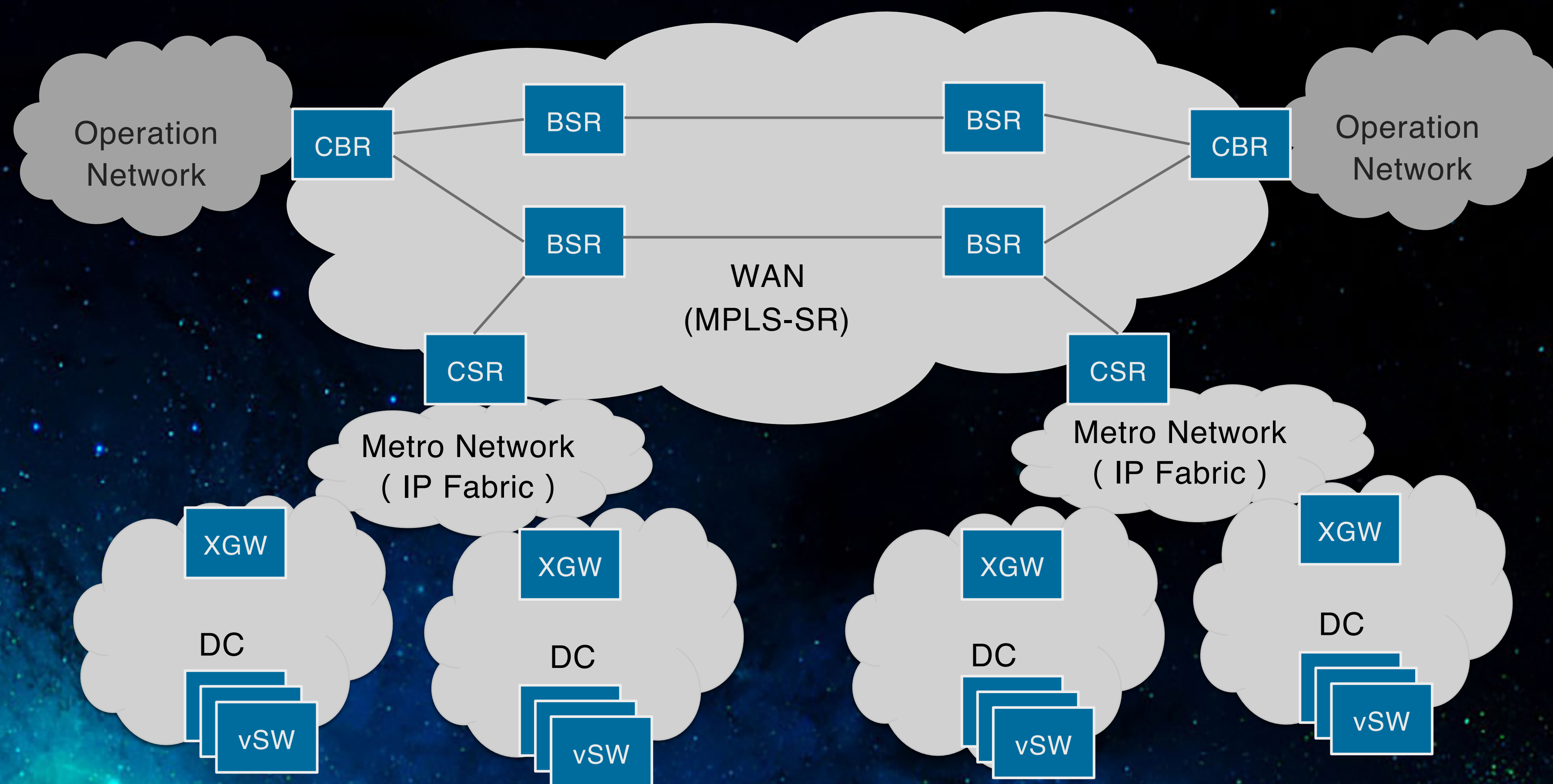


18 Region

110+ PoP

1500+ Edge

Alibaba Network Architecture



Apsara LuoShen, Alibaba Cloud SDN Architecture

Apsara LuoShen System

Control Plane

VPC Controller

CEN Controller

CCN Controller

NFV Controller

Management Plane

Intelligent
Maintenance System

Intelligent
Operation System

Data Analysis
System

Data Plane

Apsara vSwitch

DCN GW

Hybrid Cloud GW

Internet GW

SLB

CCN

LuoShen Evolving

Classic

First Gen

Network
connection



Classic

VPC

Second

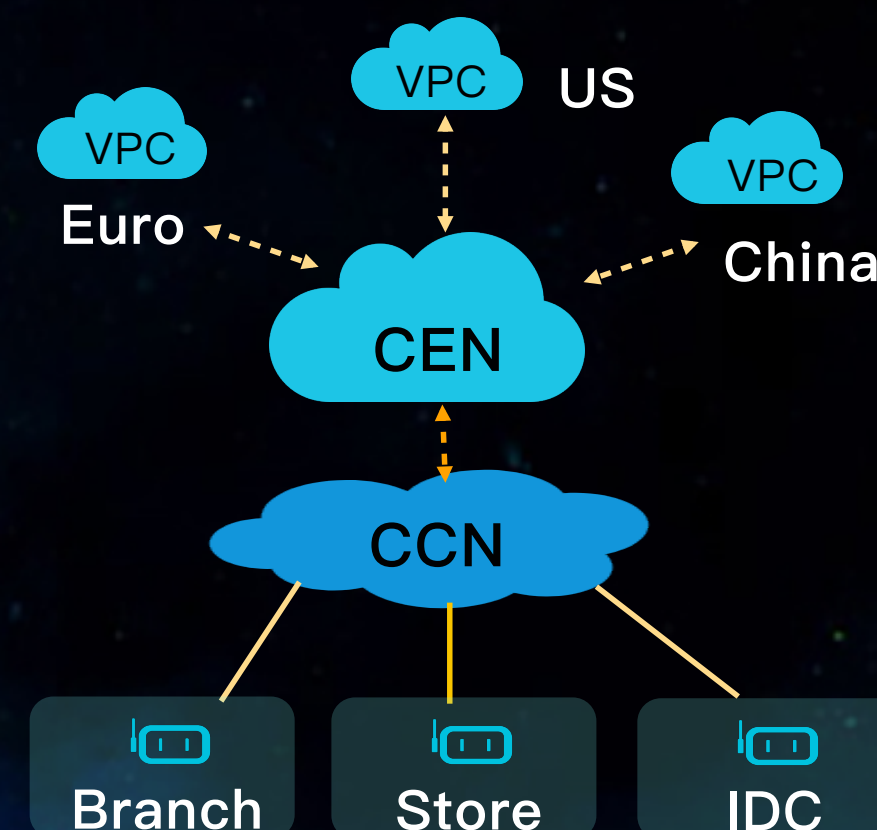
Tenant
Separation



VPC

Cross Connect

Third



Networkless

The Next..

Comprehensive Networking Product Family

12 products for 5 scenarios



Self-Defined Cloud
Networking Environment

VPC



Internet Access

SLB
NAT Gateway
EIP



Saving BGP Cost

Shared Flow Package
Shared Bandwidth Package



Building Hybrid Cloud

Express Connect
Cloud Hosting
VPN Gateway
Smart Access Gateway(CCN)

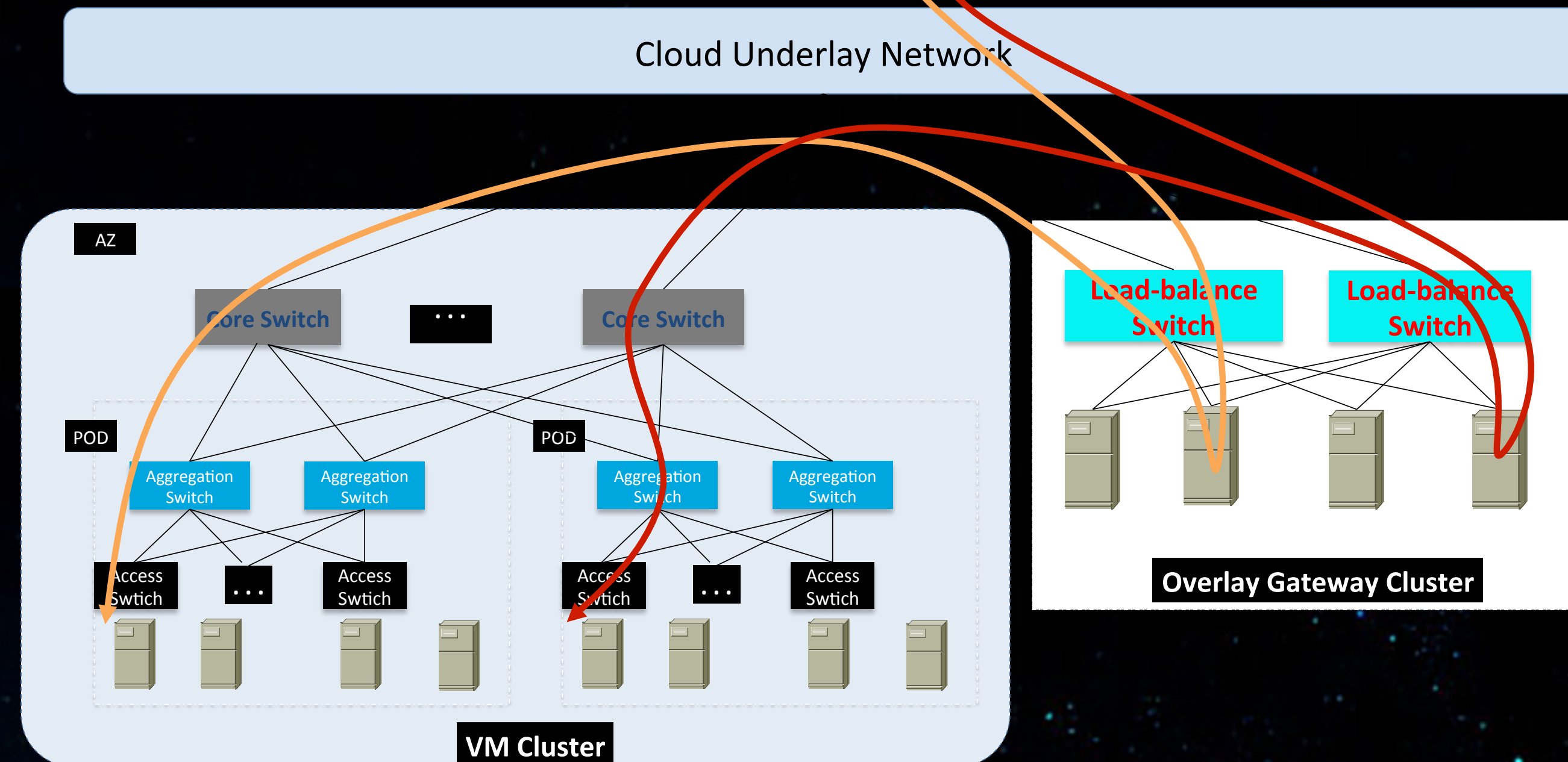
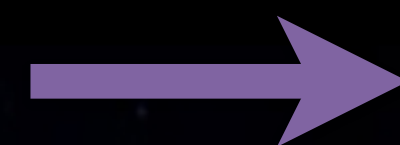
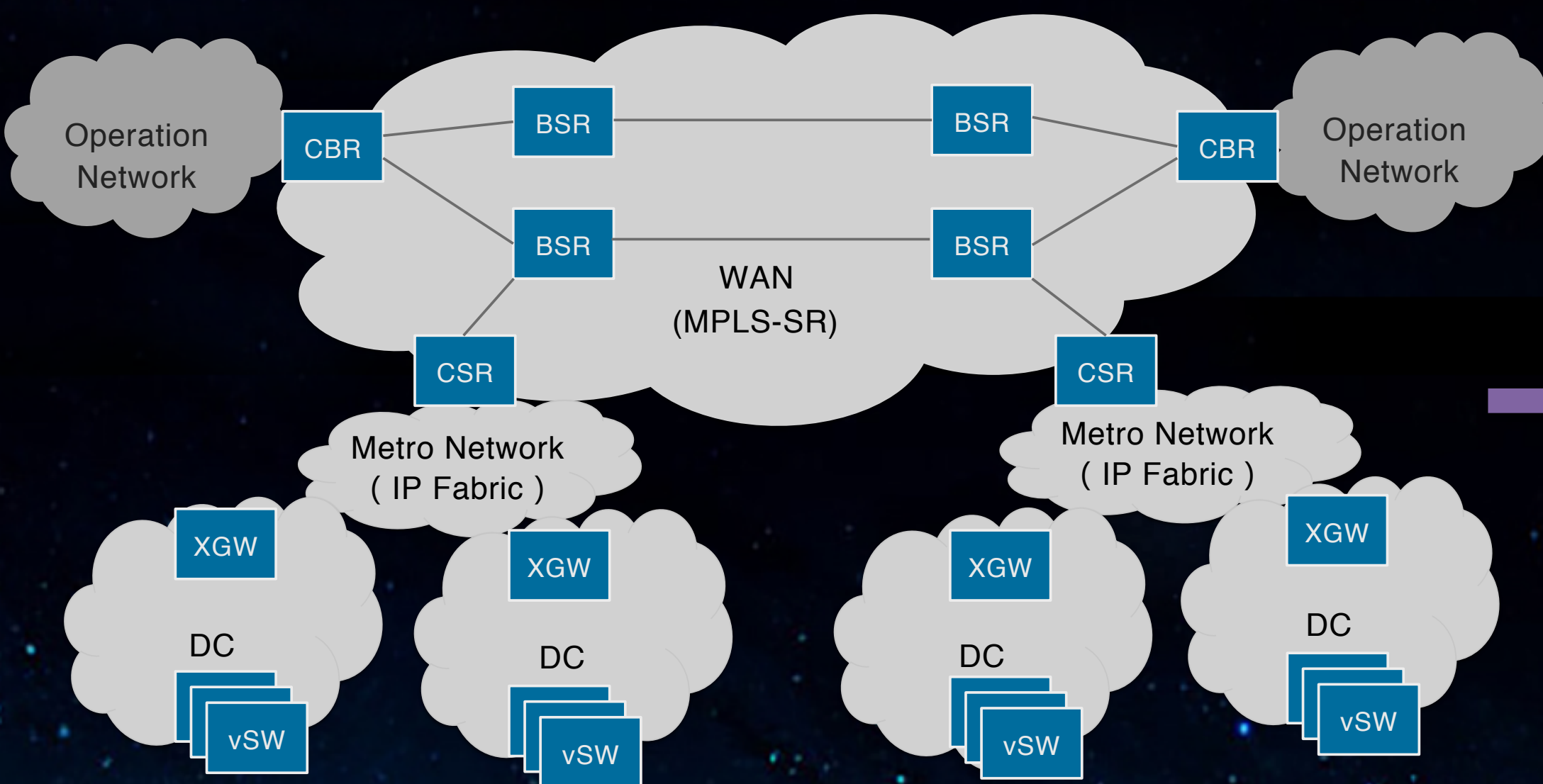


Global Connection

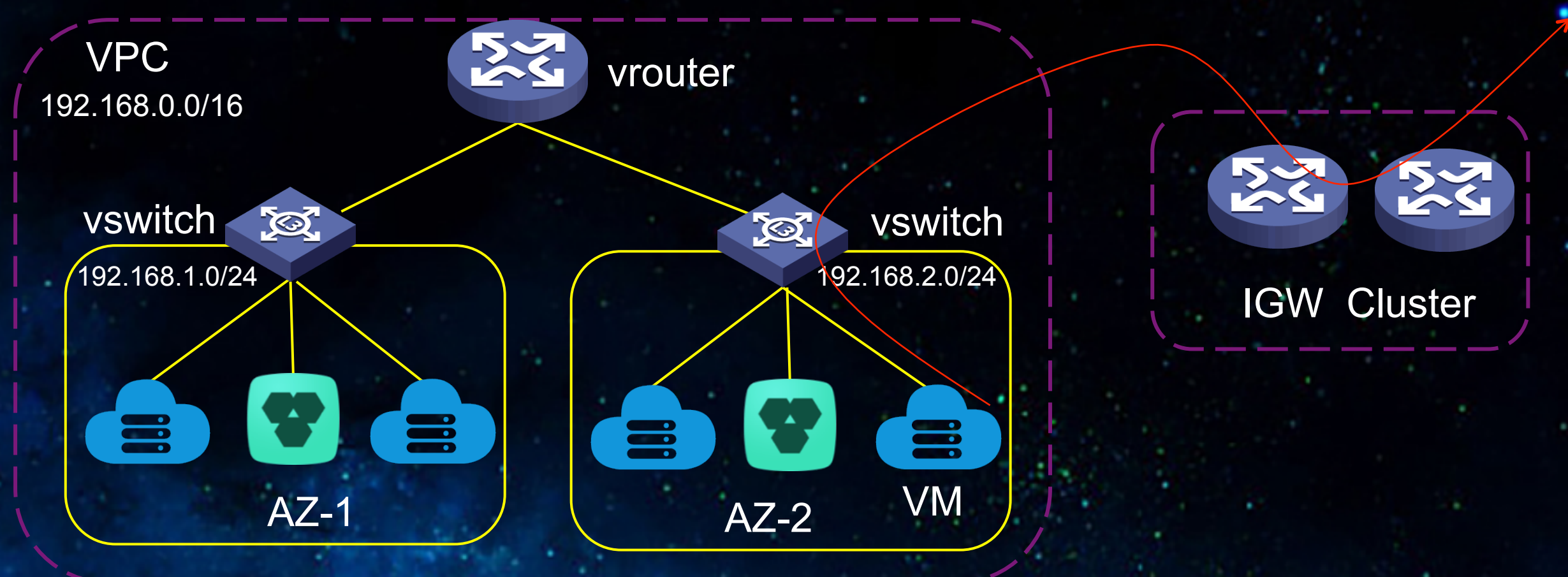
Cloud Enterprise Network(CEN)
Global Acceleration(GA)

From Data Center to Global Connection, for every enterprise networking scenarios

Cloud Network Topology

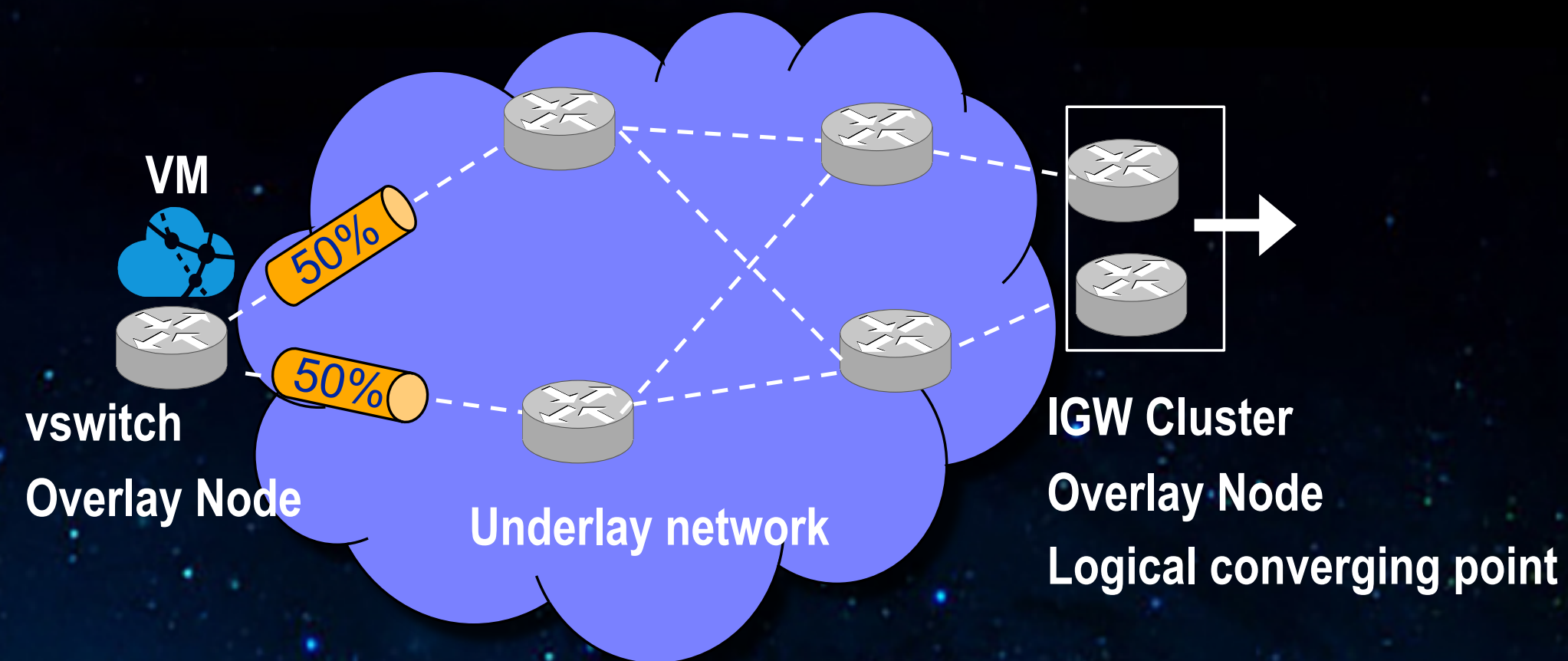


Overlay Logical Topology



Problems?

VM Want to Split Internet Traffic Across Multiple underlay Paths to multiple IGWs

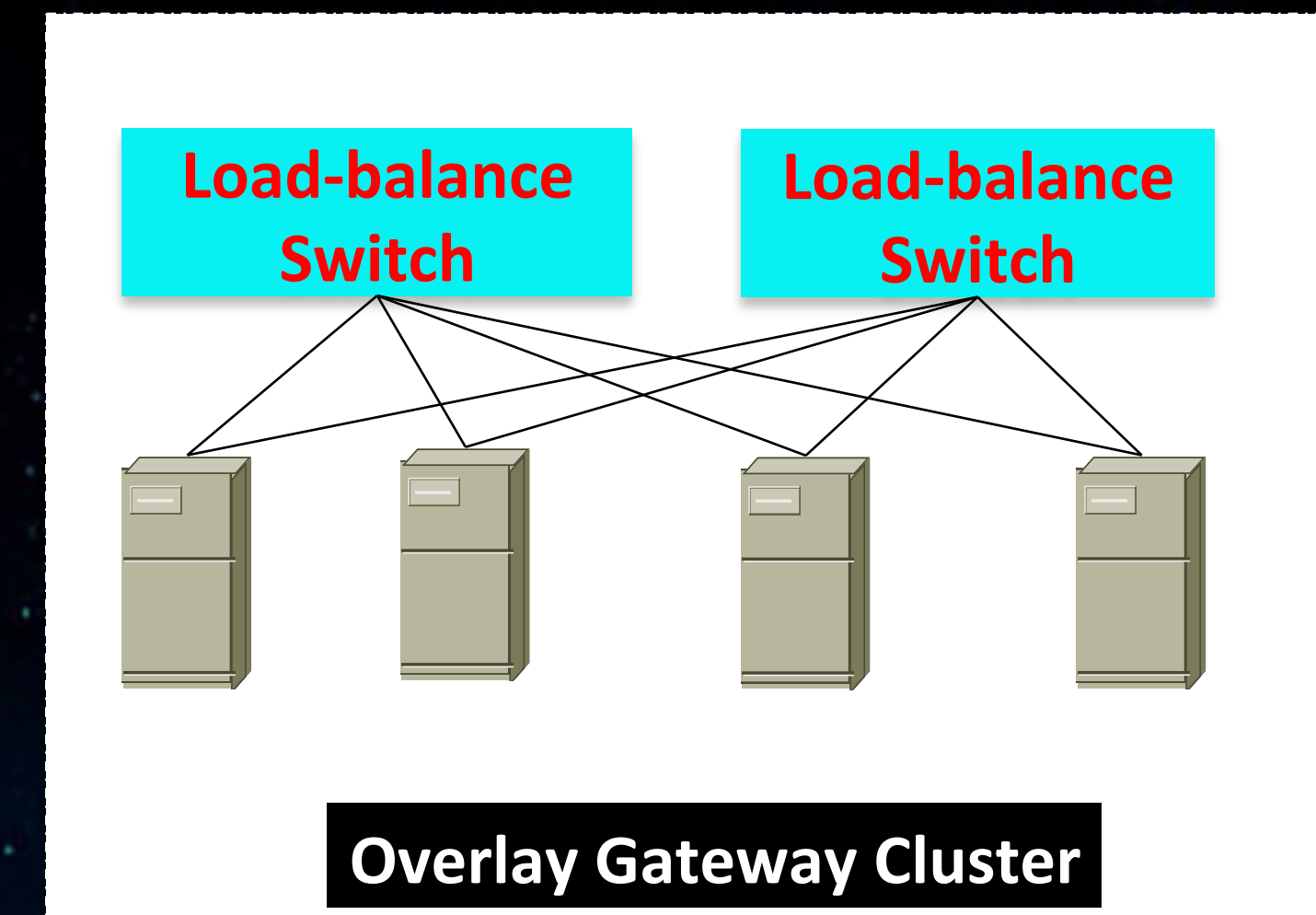


- Load balancing to different Path between VM and IGW cluster nodes.
- We focus on Overlay nodes, ignoring the underlay topology

- Underlay network is usually CLOS in DC deployment
- LB exactly do in **two Layers**:
 - Underlay network, ECMP is done btw different physical paths.
 - Overlay network, ECMP is done btw different IGWs within the cluster

Key challenges are in overlay :

1. How to do the ECMP
2. How to do the rate-limiting



How to Split traffic to different Paths?

How to Split Traffic?

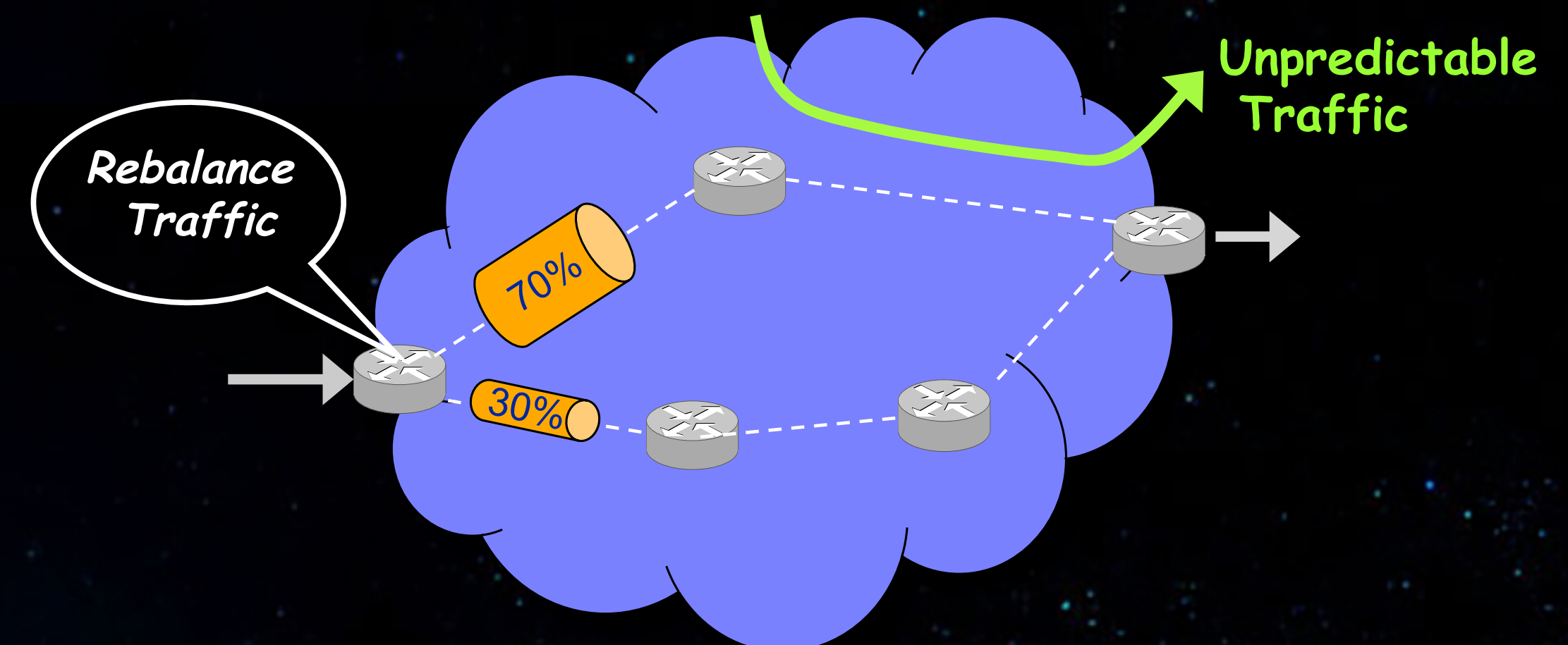
Packet-Based

- Accurate
- Reorders TCP packets
- Easily tracks dynamic ratios

Flow-Based

- Inaccurate
- No packet reordering
- Problem: Elephants Flow and Mice flows

Can we simply combine the best of the two approaches in overlay case?



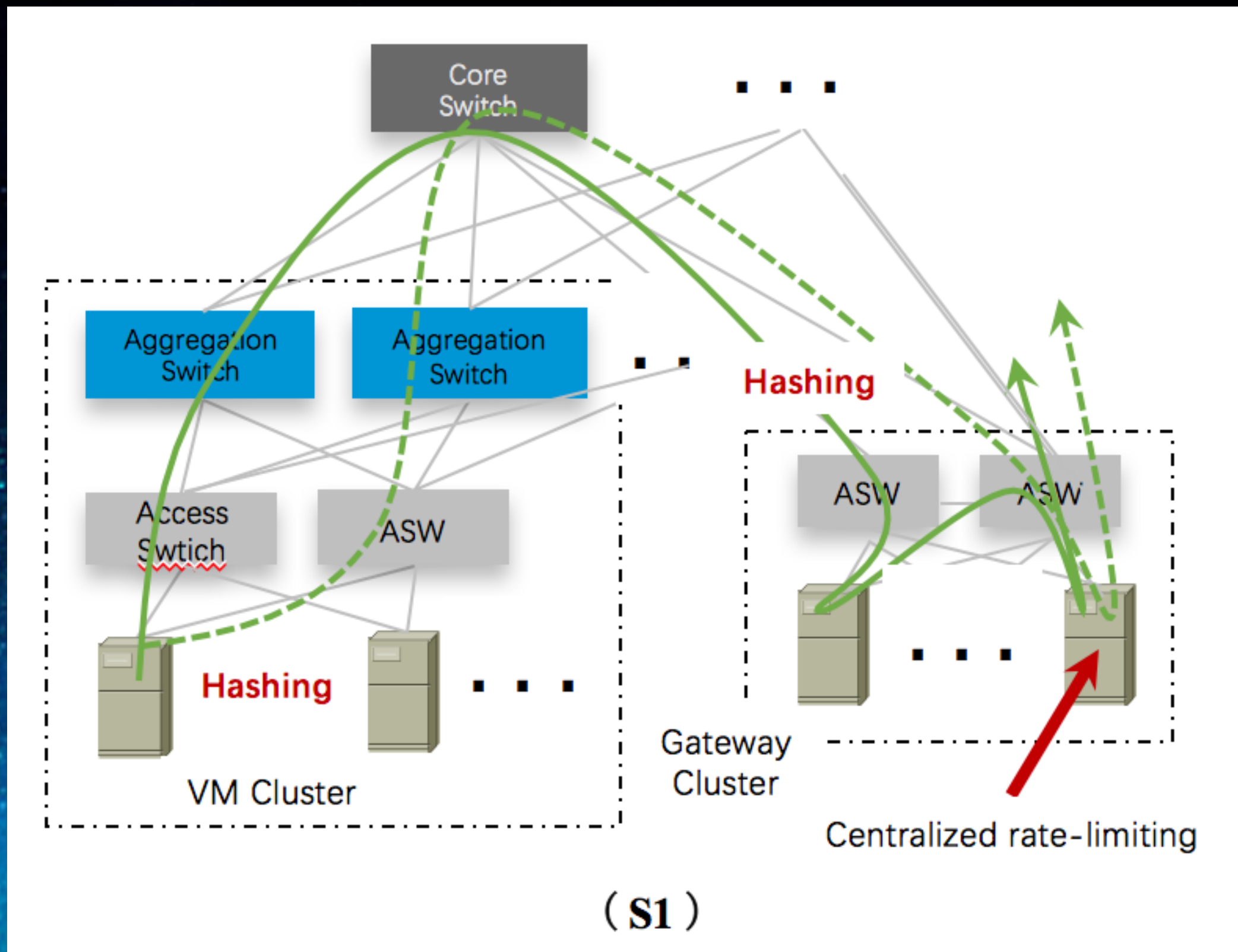
- Load balancing to remove hot spots
- Problem : Elephant flow and Mice flow
- Rebalance traffic when unpredictable events occur (Outages, DoS, BGP reroutes, Flash Crowds, ...)

There is good idea of quickly bypassing the failure point through changing the overlay src_port (failure recovery)

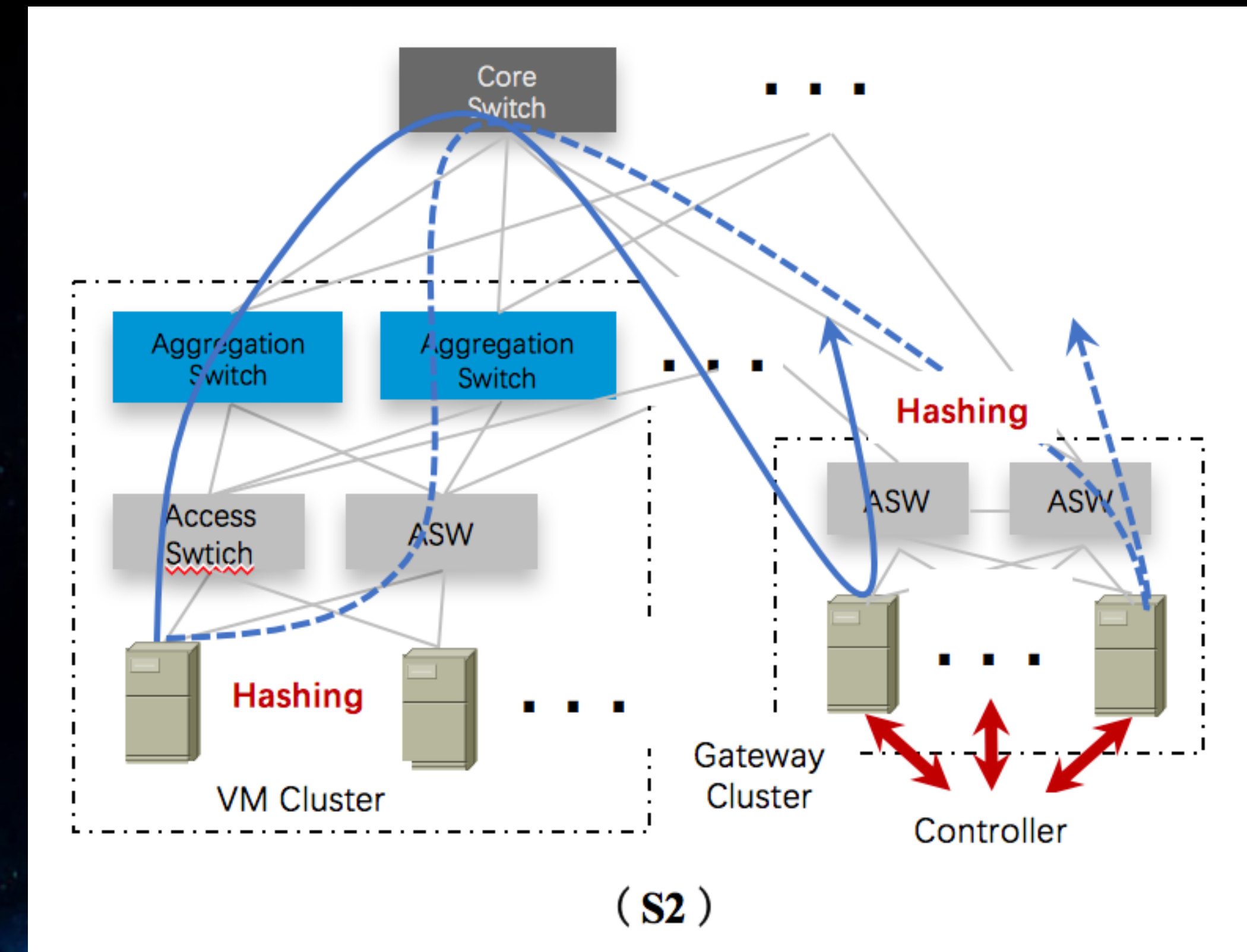
How to make the rate-limiting?

Reroute and aggregate the same flow to **one GW** to do the centralized rate-limiting

- Flow-based traffic Load balance;
- dynamically estimate the rate of each TCP flow
- automatically change the rate-limiting rates

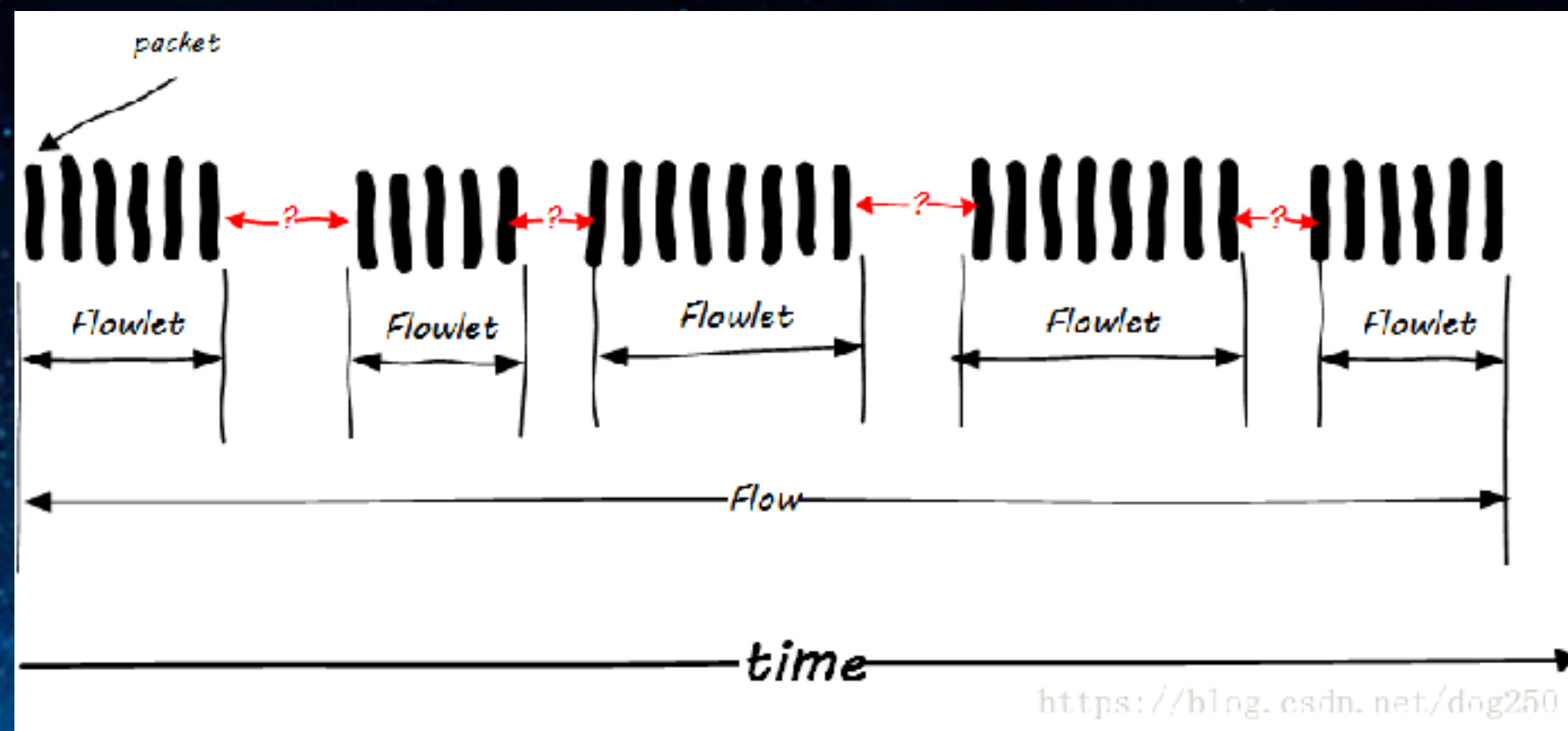
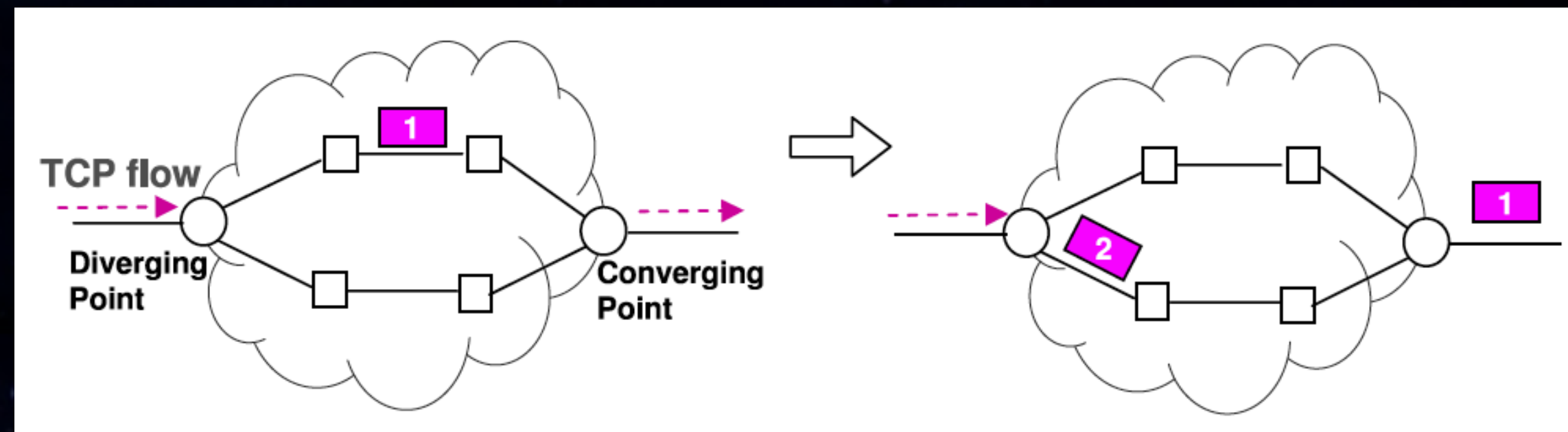


Simple but not efficient



Too complex, may be not accurate

What is the FlowLet?



Flowlets exist because TCP is burst:

- TCP usually sends a window in one or a few **bursts** and waits for acks
- Slow-start
- Ack compression
- Window is much smaller than delay-BW product

Most flowlets have inter-arrivals less than an RTT

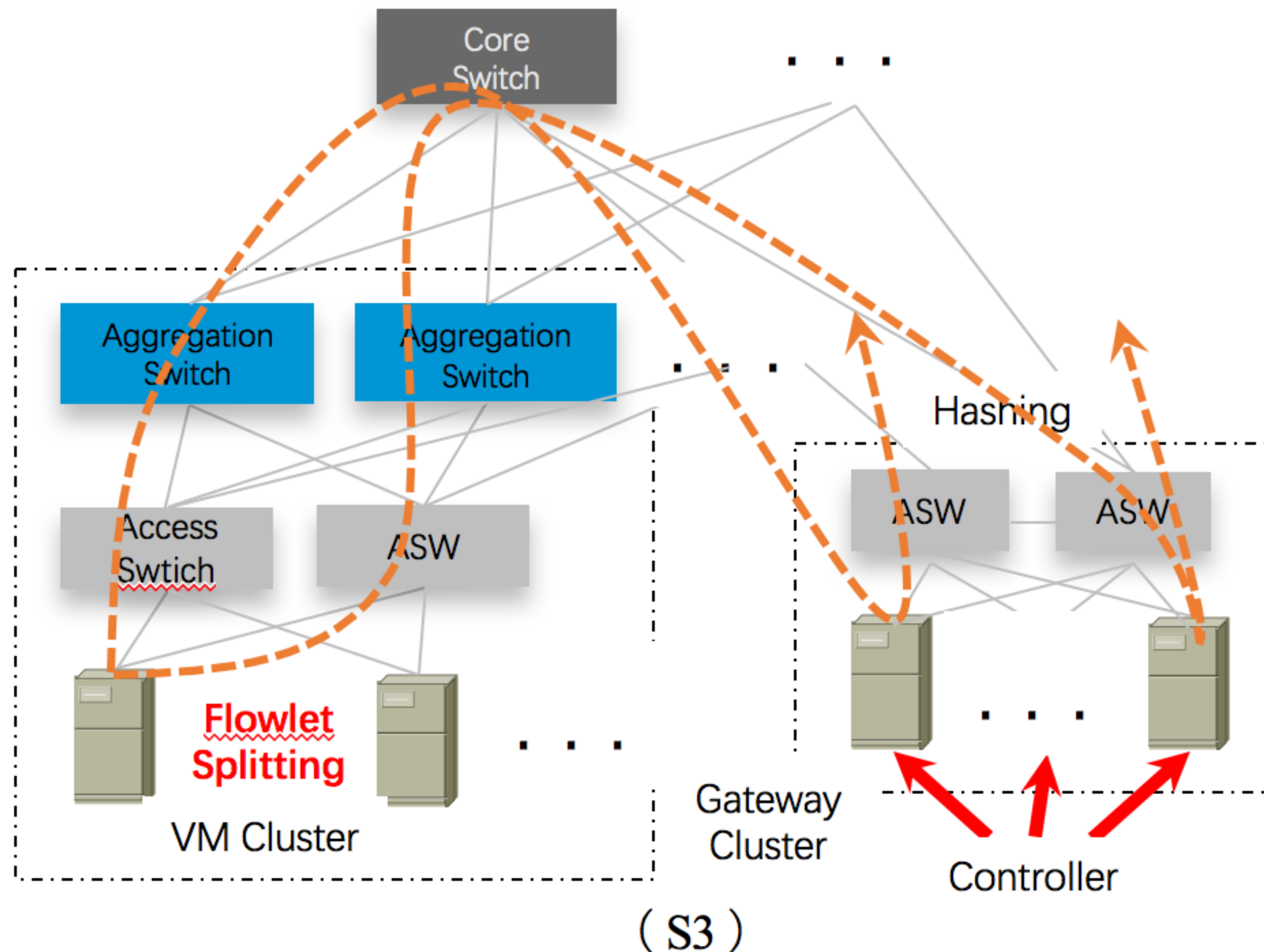
→ most flowlets are sub-windows

Two Cisco papers:

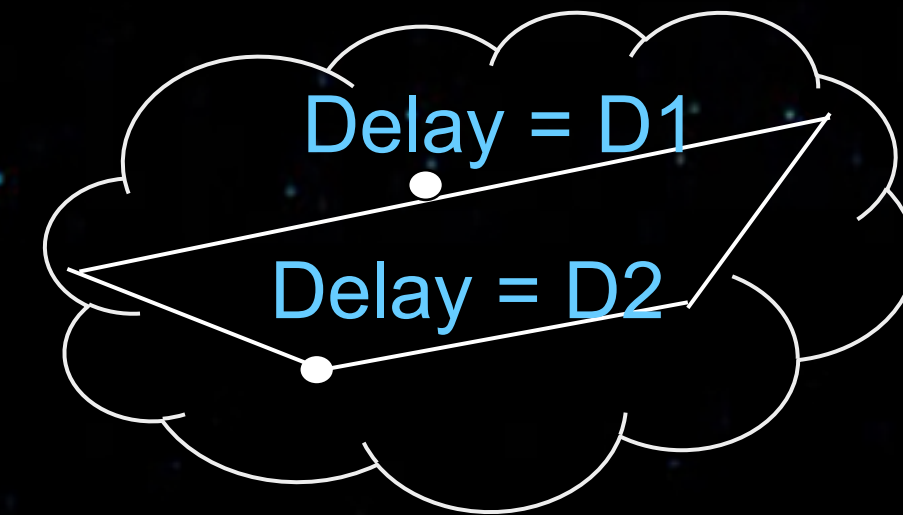
Let It Flow: Resilient Asymmetric Load Balancing with Flowlet Switching

CONGA: Distributed Congestion-Aware Load Balancing for Datacenters

FDRL(Flowlet based distributed rate-limiting)



Given $\delta > |D2-D1|$



- Select the static Time-Diff as 300us or 100 us
- In the VM side, virtual switch will do the FlowLet splitting
- Controller to dynamically change the rate-limiting ratio according to the cluster member changing

Testing Result Analysis

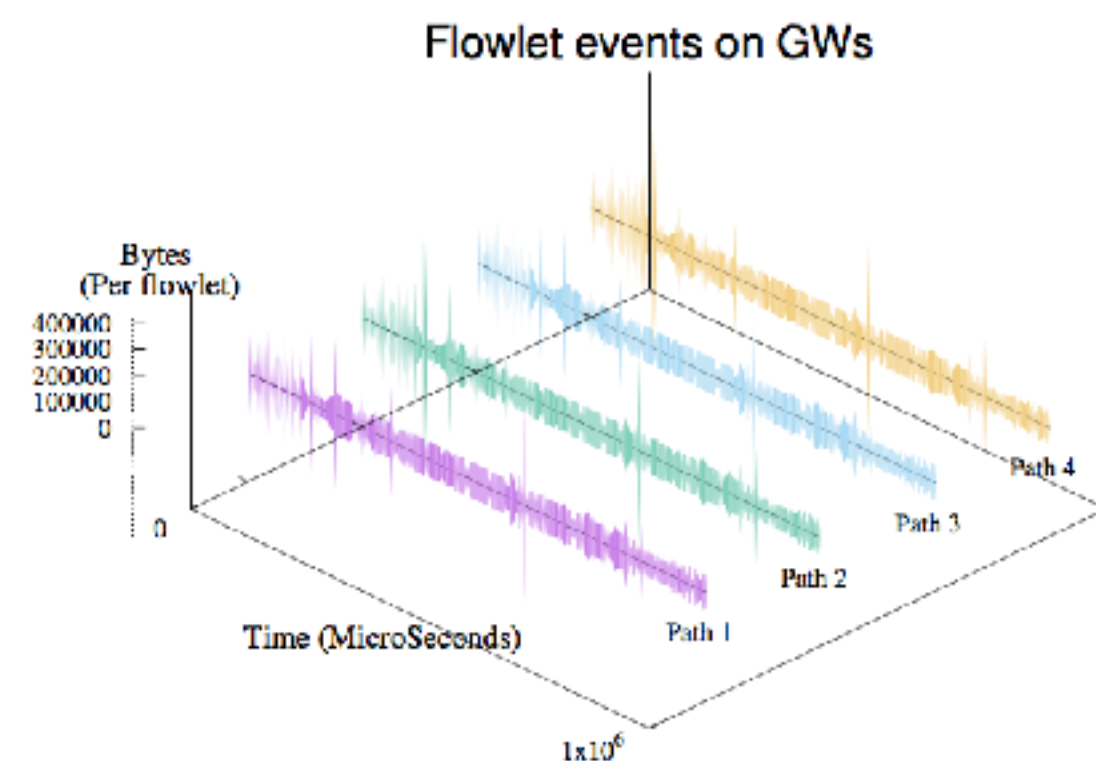


Figure 3: The process of flowlet splitting and dispatching to different paths over time

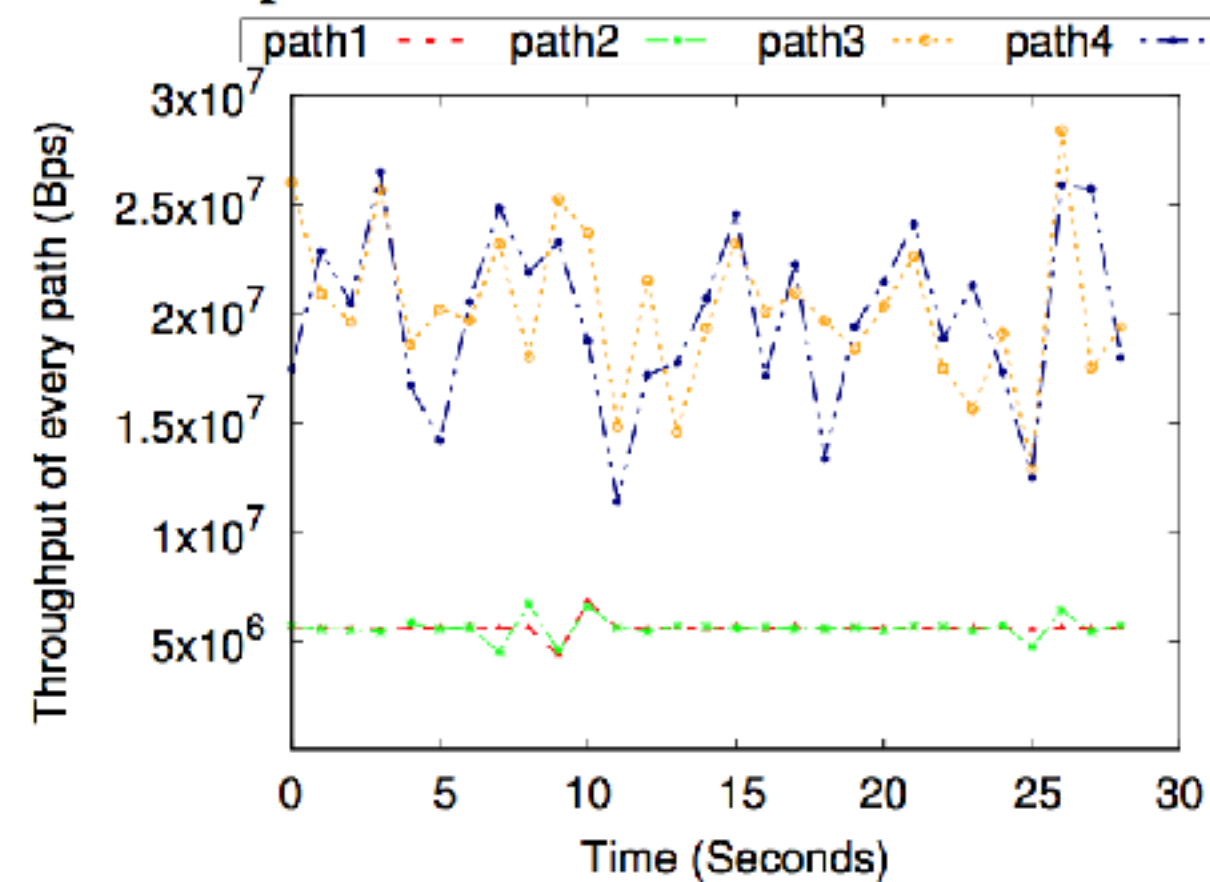


Figure 5: Throughput achieved by different paths under the asymmetric rate-limiting setting

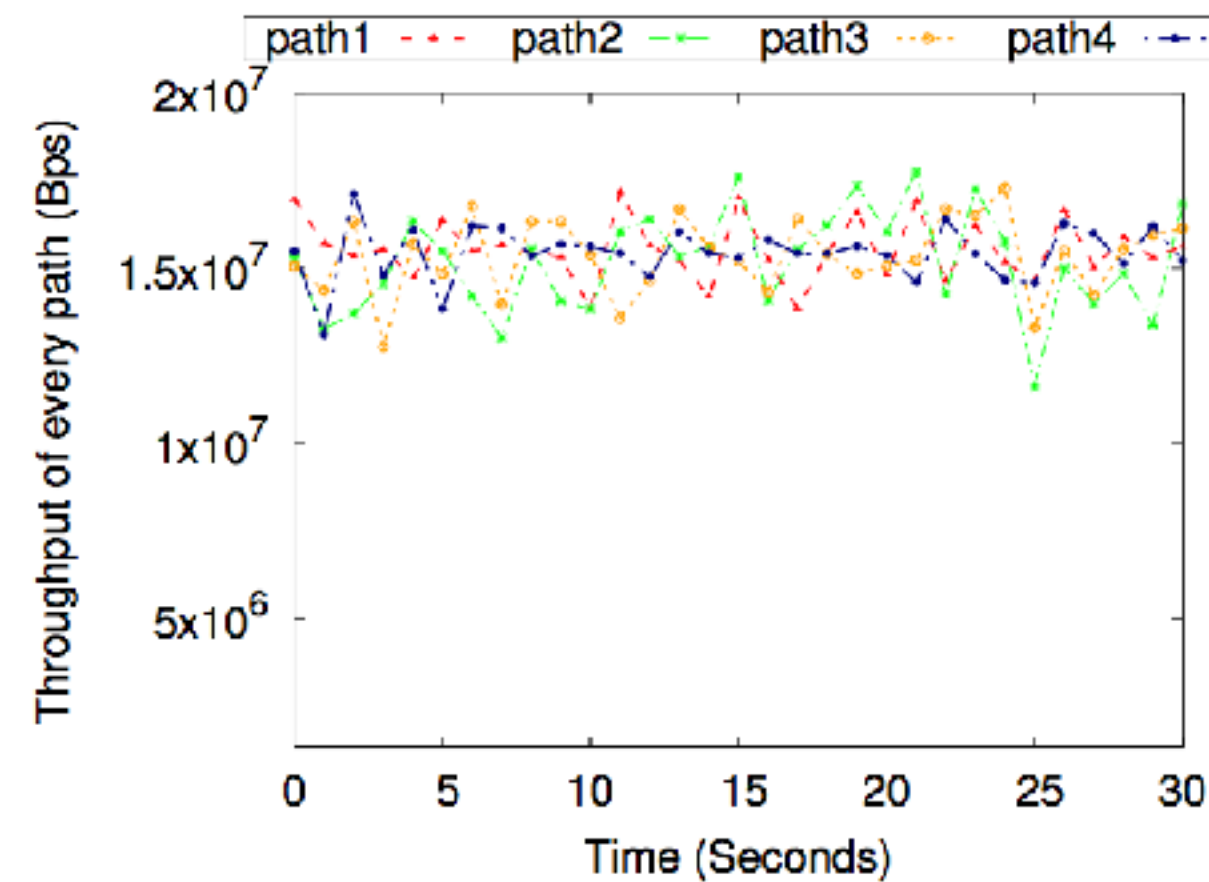


Figure 4: Throughput achieved by different paths under the symmetric rate-limiting setting

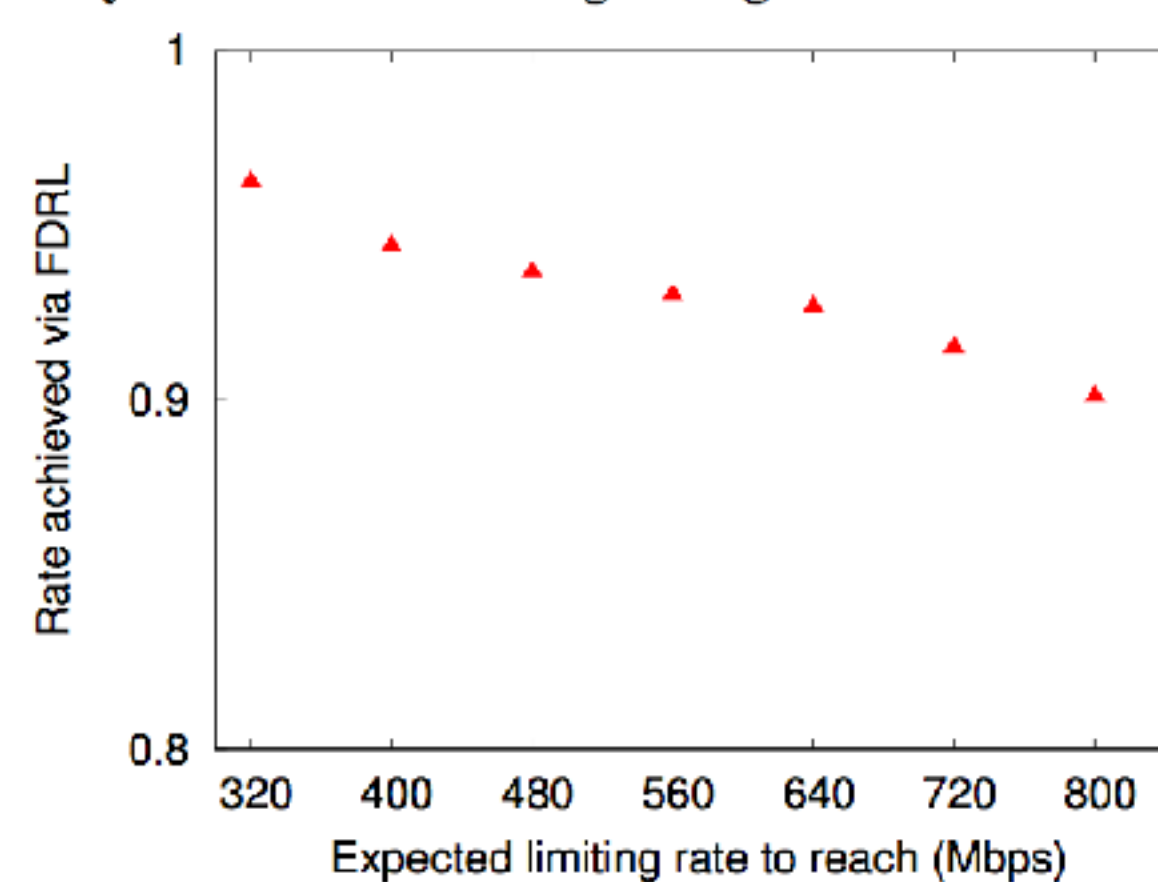


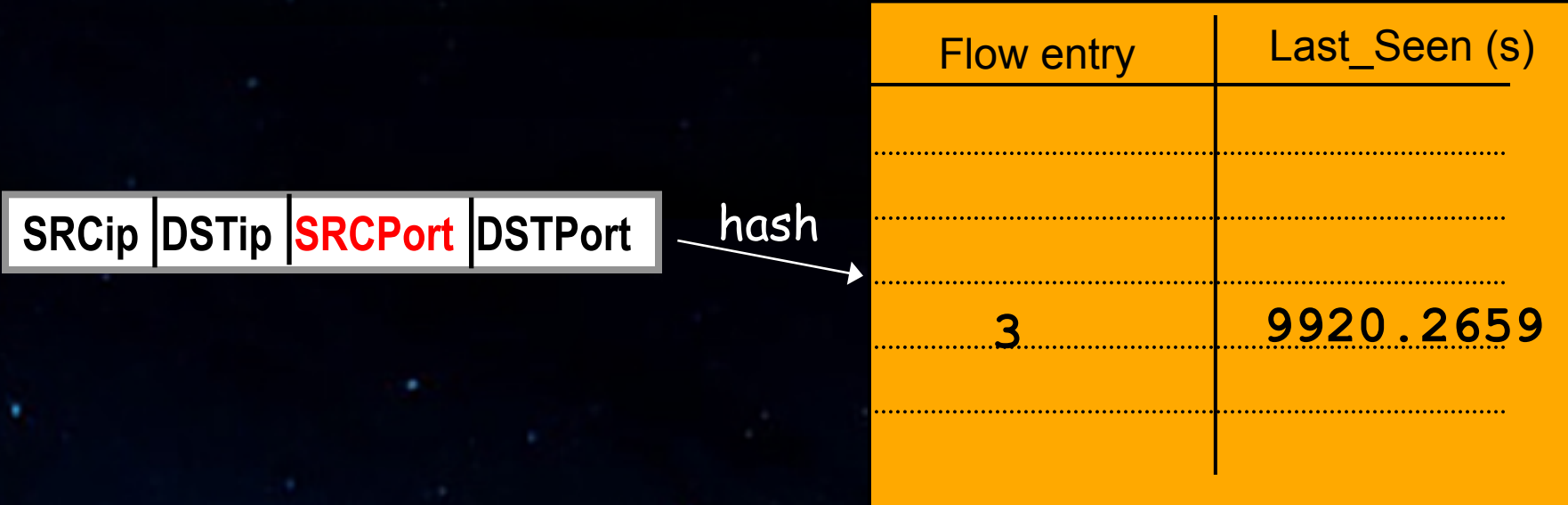
Figure 6: Total rate achieved via FDRL normalized to the limiting rate expected to reach

FDRL leverage the simple FlowLet mechanism using the overlay UDP src port as the entropy.

- Simply implement
- Self-adaption
- Great performance improvement
- Great BW coefficient of utilization

How to implement the FDRL in VPP

Implementing Flowlet in VPP is Simple



- Record the flow and timestamp for the time of last receive pkt
- If $(\text{Now} - \text{Last_Seen}) > \delta$, flow can change path
- Change the overlay encap filed of **src_port**
- Reassign path proportionally to the desired split ratios

Plugin	Version	Description
1. ioam_plugin.so	18.07-7~g004aa8f-dirty	Inbound OAM
2. memif_plugin.so	18.07-7~g004aa8f-dirty	Packet Memory Inter
face (experimental)		
3. avf_plugin.so	18.07-7~g004aa8f-dirty	Intel Adaptive Virt
ual Function (AVF) Device Plugin		
4. pppoe_plugin.so	18.07-7~g004aa8f-dirty	PPPoE
5. flowtable_plugin.so	18.07-7~g004aa8f-dirty	Flowtable
6. abf_plugin.so	18.07-7~g004aa8f-dirty	ACL based Forwardin

Base on the flowtable plugin

- Support Dynamical session for TCP flow, with session aging/timeout
- TCP stateful session , TCP state update.
- Record the timestamp for last pkt receiving based on flow
- Make the judgement for the action: when we need to change the path
- How to change the path: just change the overlay udp src port.

```
DBGvpp# show flowtable
Number of flows cache allocated:256
active_flow: 0
run_show_cmd_time(s):68229237
DBGvpp#
DBGvpp# flowtable ?
flowtable [max-flows <n>] [intf <name>] [next-node <nam
e>] [disable]
DBGvpp# flowtable intf VirtualFunctionEthernet7/10/2
DBGvpp# show flowtabl
Number of flows cache allocated:256
active_flow: 0
run_show_cmd_time(s):68229279
DBGvpp# show flowtabl
Number of flows cache allocated:256
active_flow: 1
run_show_cmd_time(s):68229297
sig_src:80.0.0.159, sig_dst:80.0.0.160, sig_proto:1, sig_port_src:0, sig_port_dst:0
tcp_state: 0, expire(s):68229353, lifetime:60
flow_id:1, cpu_index:0, offloaded:0
stats[0].pkts:0, stats[0].bytes:0
stats[1].pkts:4, stats[1].bytes:256
last_pkt_dispatch_clock(tickes):156927383116153060, last_pkt_dispatch_time(us):68229297007023
DBGvpp#
```

How to implement the FDRL in VPP

```
[root@rs7h11514.et2sqa:/home/xiyun.xxy/vpp]
#git diff src/vlib/buffer.h
diff --git a/src/vlib/buffer.h b/src/vlib/buffer.h
index 9555cd7..8269180 100644
--- a/src/vlib/buffer.h
+++ b/src/vlib/buffer.h
@@ -156,7 +156,10 @@ typedef struct
     vlib_buffer_free_list_index_t free_list_index; /** < only used if
                                                    VLIB_BUFFER_NON_DEFAULT_FREELIST
                                                    flag is set */

-   u8 align_pad[3]; /**< available */
+
+#define FLOW_FLAGS_SWITCH_PATH 0x01
+   u8 flow_flags;
+   u8 align_pad[2]; /**< available */
+   u32 opaque2[12]; /**< More opaque data, see ../vnet/vnet/buffer.h */

/***** end of second cache line */
```

Base on the flowtable plugin

- Support Dynamical session for TCP flow, with session aging/timeout
- TCP stateful session , TCP state update.
- Record the timestamp for last pkt receiving based on flow
- Make the judgement for the action: when we need to change the path
- How to change the path: just change the overlay udp src port.

```
677 inline void set_pkt_buffer_flow_flags(flow_entry_t * flow,
678                                         flowtable_main_t *fm, vlib_buffer_t *b)
679 {
680     if ((fm->vlib_main->cpu_time_last_node_dispatch <
681         flow->last_pkt_dispatch_clock) ||
682         ((fm->vlib_main->cpu_time_last_node_dispatch -
683         flow->last_pkt_dispatch_clock) > fm->diff_clock)) {
684         b->flow_flags |= FLOW_FLAGS_SWITCH_PATH;
685     }
686     flow->last_pkt_dispatch_clock =
687         fm->vlib_main->cpu_time_last_node_dispatch;
688     return;
689 }
690
```

What is the next Step?

Call for community to join this direction, and make more improvement:

- Dynamically change the time diff for different Elephants/Mice flow
- Asymmetry rate-limiting scenario
- Flowlet is still working or not after BBQ?
- Use the same logic to quickly bypass the failure path through changing the overlay src-port
- VPP implementation, and integration to support more FDRL features

Our Vision

Simply The Network



Blog

Scan to Learn More



DingDing User Group

Scan to Learn More

Thanks