

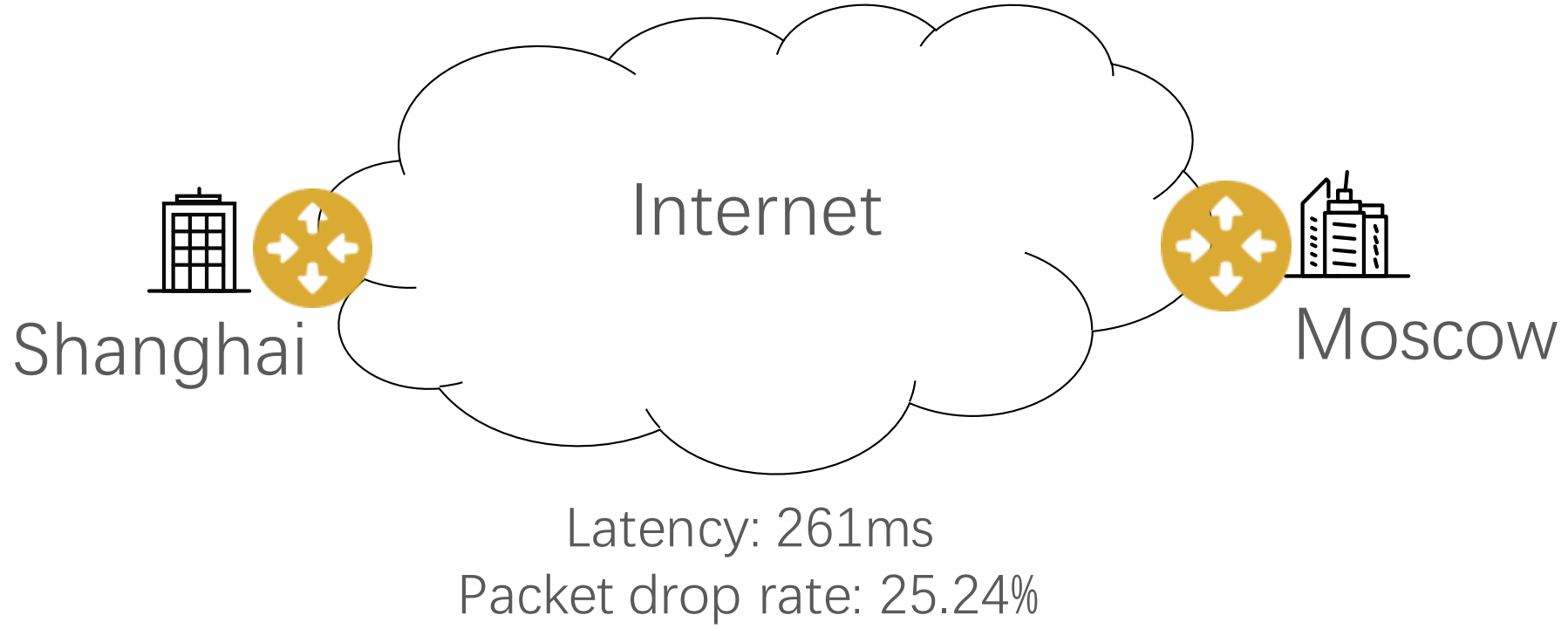
Provide a new general purpose transport layer with:

Secure[QUIC-TLS],

Reliable[QUIC-Transport],

Programmable[SR]

Simple Problem

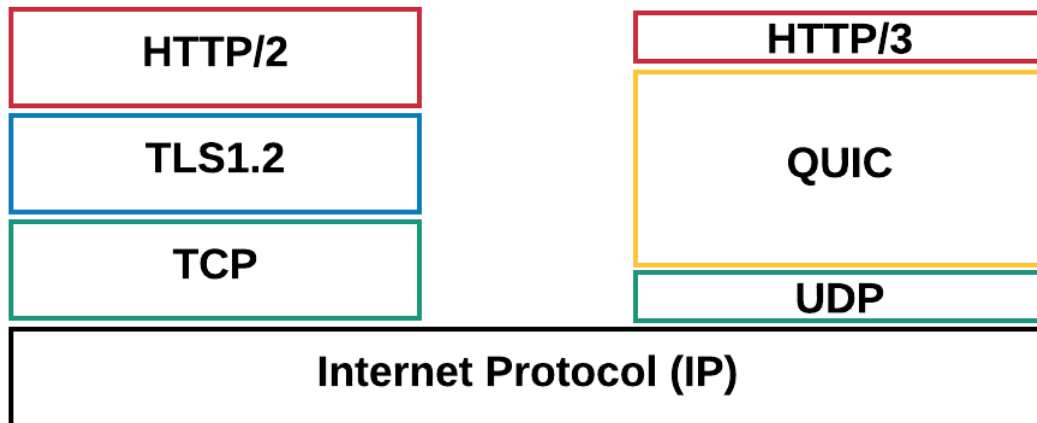


Resolve on Transportation Layer

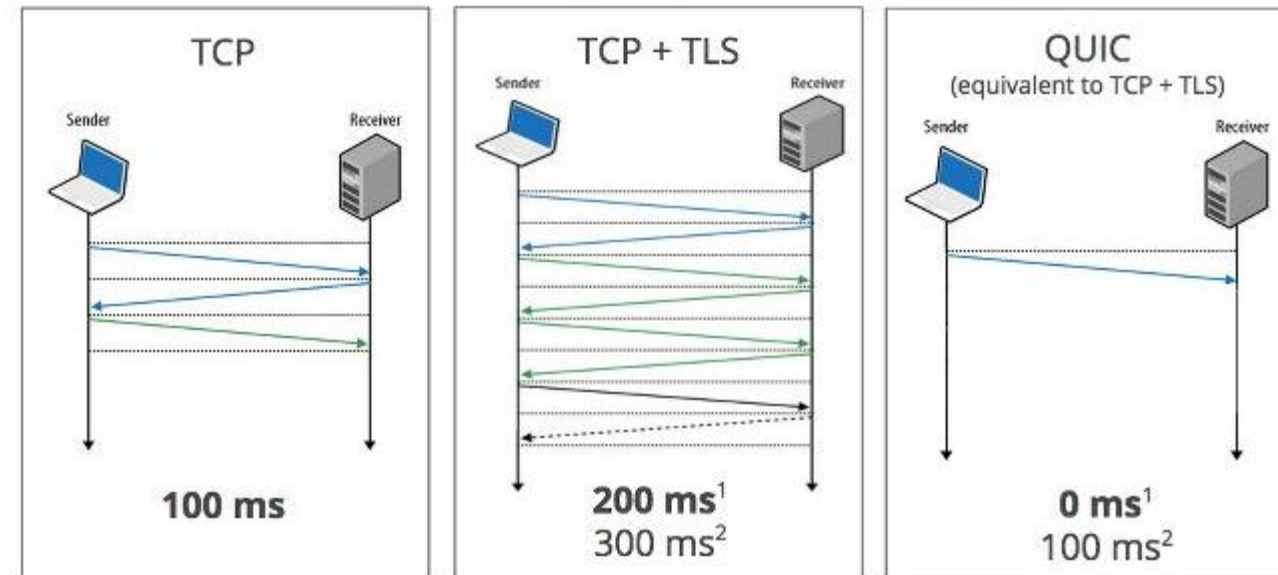


QUIC is a multiplexed and secure general-purpose transport protocol that provides:

- Stream multiplexing
- Stream and connection-level flow control
- Low-latency connection establishment



Zero RTT Connection Establishment

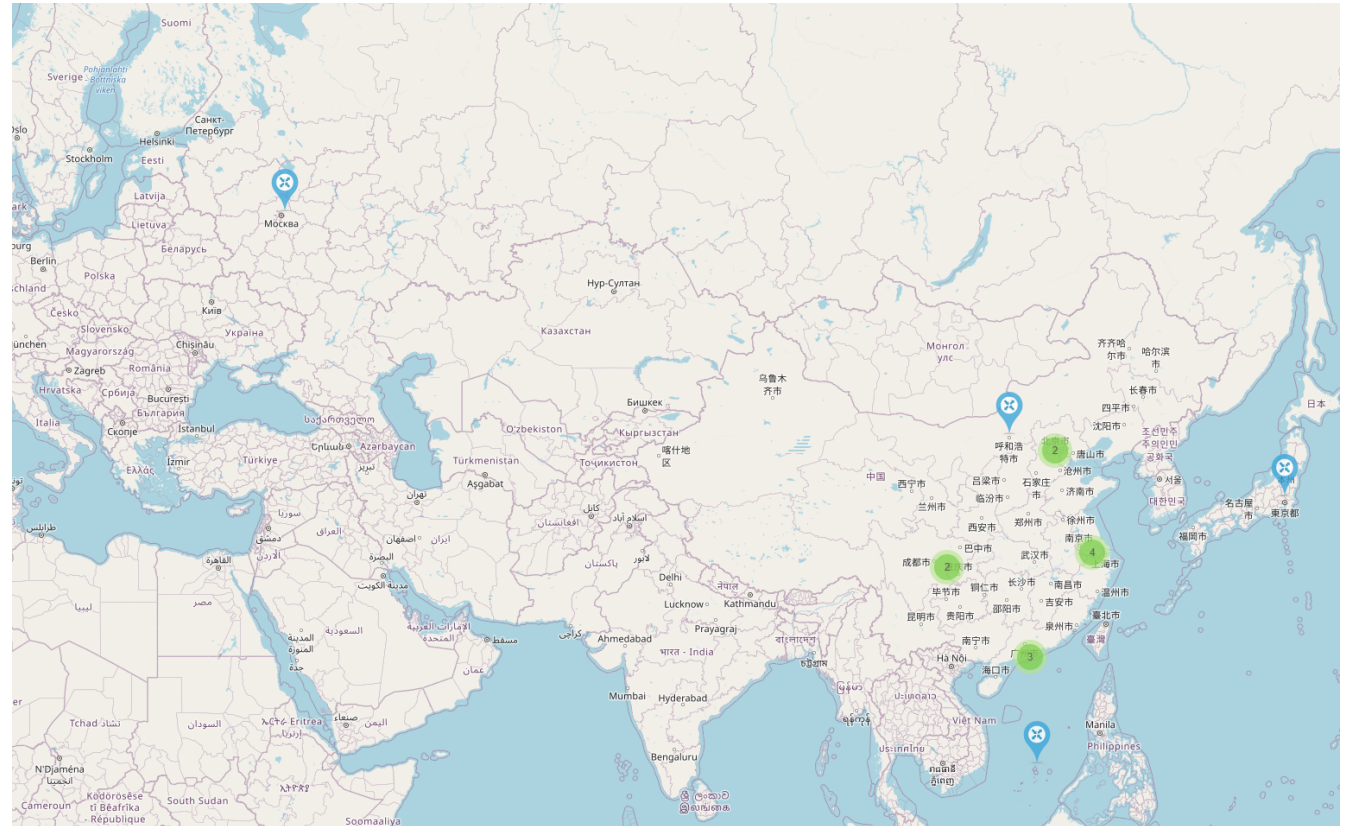


1. Repeat connection
2. Never talked to server before

Resolve on Network Layer

We built 15 virtual router(VR) in different regions.

5-VR in Ali-cloud
8-VR in Tencent-Cloud
1-VR in Private DC



Resolve on Network Layer

Latency Measurement

	AliCloud-BJ	AliCloud-HHHT	AliCloud-HK	AliCloud-HZ	AliCloud-SZ	CT-SH	HongkongDC	Tencent-BJ	Tencent-CD	Tencent-CQ	Tencent-GZ	Tencent-Moscow	Tencent-NJ	Tencent-SH	Tencent-Tokoyo
AliCloud-BJ	0.0ms	13.5ms	43.0ms	28.7ms	33.1ms	31.0ms	78.1ms	5.6ms	63.2ms	58.2ms	39.3ms	203.8ms	35.0ms	29.5ms	63.2ms
AliCloud-HHHT	13.6ms	0.0ms	55.8ms	35.3ms	63.0ms	38.0ms	83.0ms	13.4ms	69.0ms	76.2ms	50.8ms	223.5ms	46.4ms	37.1ms	71.1ms
AliCloud-HK	43.9ms	56.3ms	0.0ms	30.0ms	15.9ms	31.0ms	38.1ms	103.4ms	98.5ms	111.0ms	54.6ms	349.6ms	91.4ms	58.8ms	45.0ms
AliCloud-HZ	29.5ms	35.4ms	30.0ms	0.0ms	23.7ms	9.8ms	66.5ms	31.0ms	50.4ms	48.9ms	29.9ms	263.8ms	18.1ms	9.9ms	69.0ms
AliCloud-SZ	33.2ms	63.0ms	15.2ms	23.1ms	0.0ms	31.0ms	49.6ms	35.0ms	34.1ms	40.2ms	4.6ms	270.3ms	34.0ms	34.0ms	73.3ms
CT-SH	31.0ms	38.0ms	31.5ms	9.8ms	31.8ms	0.0ms	67.6ms	35.0ms	50.0ms	52.1ms	28.2ms	262.1ms	11.0ms	5.0ms	63.7ms
HongkongDC	78.3ms	83.2ms	37.3ms	66.4ms	49.8ms	67.0ms	0.0ms	79.8ms	83.2ms	79.1ms	40.8ms	299.6ms	69.3ms	64.2ms	91.7ms
Tencent-BJ	6.0ms	14.0ms	103.4ms	31.5ms	36.0ms	34.0ms	80.0ms	0.0ms	69.7ms	71.8ms	35.3ms	241.3ms	36.5ms	26.8ms	109.9ms
Tencent-CD	64.2ms	70.0ms	99.0ms	50.6ms	35.0ms	50.0ms	83.3ms	69.7ms	0.0ms	6.8ms	33.0ms	291.5ms	41.0ms	36.0ms	86.2ms
Tencent-CQ	59.2ms	77.1ms	111.5ms	49.8ms	41.1ms	51.1ms	79.1ms	71.8ms	6.8ms	0.0ms	30.0ms	263.1ms	34.3ms	30.0ms	103.8ms
Tencent-GZ	40.0ms	51.7ms	55.0ms	30.5ms	4.7ms	27.6ms	40.8ms	35.3ms	33.0ms	30.0ms	0.0ms	245.0ms	38.9ms	29.2ms	78.8ms
Tencent-Moscow	204.5ms	224.2ms	349.9ms	264.0ms	270.9ms	261.5ms	299.5ms	241.3ms	291.6ms	263.1ms	245.0ms	0.0ms	249.1ms	280.2ms	154.0ms
Tencent-NJ	36.0ms	47.3ms	91.7ms	18.8ms	35.0ms	10.0ms	69.3ms	36.4ms	41.0ms	34.3ms	38.9ms	249.1ms	0.0ms	8.0ms	111.7ms
Tencent-SH	30.2ms	38.0ms	58.9ms	10.3ms	35.0ms	4.0ms	64.3ms	26.8ms	36.0ms	30.0ms	29.1ms	280.2ms	8.0ms	0.0ms	90.3ms
Tencent-Tokoyo	63.2ms	71.1ms	44.0ms	69.0ms	73.3ms	63.1ms	91.3ms	109.4ms	85.8ms	103.3ms	78.3ms	153.0ms	111.6ms	90.1ms	0.0ms

Resolve on Network Layer

Drop Rate Measurement

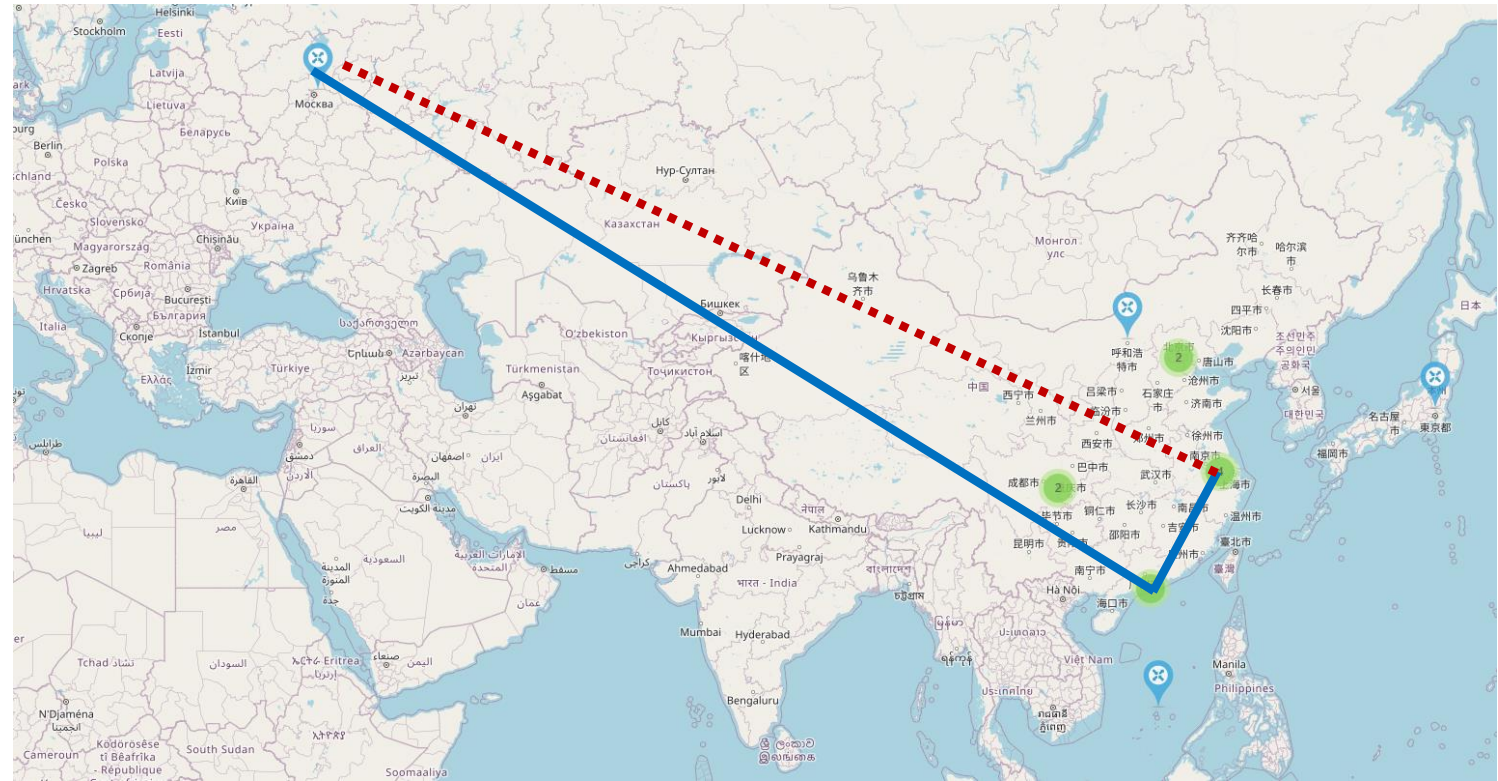
	AliCloud-BJ	AliCloud-HHHT	AliCloud-HK	AliCloud-HZ	AliCloud-SZ	CT-SH	HongkongDC	Tencent-BJ	Tencent-CD	Tencent-CQ	Tencent-GZ	Tencent-Moscow	Tencent-NJ	Tencent-SH	Tencent-Tokoyo
AliCloud-BJ	0.00%	0.00%	0.02%	0.00%	0.01%	0.01%	0.11%	0.03%	0.05%	0.06%	0.04%	8.33%	0.06%	0.05%	0.25%
AliCloud-HHHT	0.01%	0.00%	0.03%	0.01%	0.01%	0.01%	0.15%	0.05%	0.04%	0.05%	0.06%	8.76%	0.06%	0.05%	0.02%
AliCloud-HK	0.02%	0.01%	0.00%	23.45%	0.12%	0.43%	0.11%	5.66%	7.40%	0.44%	1.62%	2.42%	8.08%	12.02%	0.02%
AliCloud-HZ	0.01%	0.01%	23.55%	0.00%	0.01%	0.01%	0.85%	0.04%	0.04%	0.06%	0.04%	27.20%	0.05%	0.04%	0.22%
AliCloud-SZ	0.01%	0.01%	0.12%	0.01%	0.00%	0.13%	2.84%	0.02%	0.02%	0.05%	7.04%	14.29%	0.06%	0.04%	0.07%
CT-SH	0.01%	0.02%	0.43%	0.00%	0.15%	0.00%	12.85%	0.04%	0.04%	0.06%	0.04%	25.24%	0.04%	0.04%	4.14%
HongkongDC	0.11%	0.13%	0.15%	0.84%	2.85%	12.91%	0.00%	25.46%	8.45%	8.49%	1.81%	0.15%	3.71%	2.34%	0.12%
Tencent-BJ	0.04%	0.03%	5.66%	0.04%	0.04%	0.05%	25.35%	0.00%	0.06%	0.10%	0.08%	11.03%	0.09%	0.07%	12.38%
Tencent-CD	0.04%	0.04%	7.35%	0.04%	0.03%	0.03%	8.52%	0.06%	0.00%	0.06%	0.06%	30.72%	0.06%	0.07%	11.75%
Tencent-CQ	0.06%	0.08%	0.44%	0.05%	0.06%	0.06%	8.55%	0.08%	0.07%	0.00%	0.07%	10.06%	0.09%	0.08%	10.74%
Tencent-GZ	0.05%	0.05%	1.57%	0.05%	5.89%	0.04%	1.78%	0.08%	0.06%	0.08%	0.00%	0.68%	0.09%	0.07%	6.23%
Tencent-Moscow	8.41%	8.54%	2.49%	27.34%	14.45%	25.28%	0.15%	10.96%	30.91%	10.10%	0.75%	0.00%	25.57%	6.38%	0.03%
Tencent-NJ	0.05%	0.06%	8.31%	0.04%	0.06%	0.06%	3.54%	0.08%	0.06%	0.08%	0.09%	25.48%	0.00%	0.08%	7.47%
Tencent-SH	0.05%	0.04%	12.07%	0.04%	0.04%	0.04%	2.45%	0.06%	0.07%	0.07%	0.06%	6.53%	0.09%	0.00%	10.96%
Tencent-Tokoyo	0.29%	0.03%	0.02%	0.20%	0.07%	4.20%	0.11%	12.13%	11.73%	10.81%	6.13%	0.02%	7.59%	10.89%	0.00%

Resolve on Network Layer



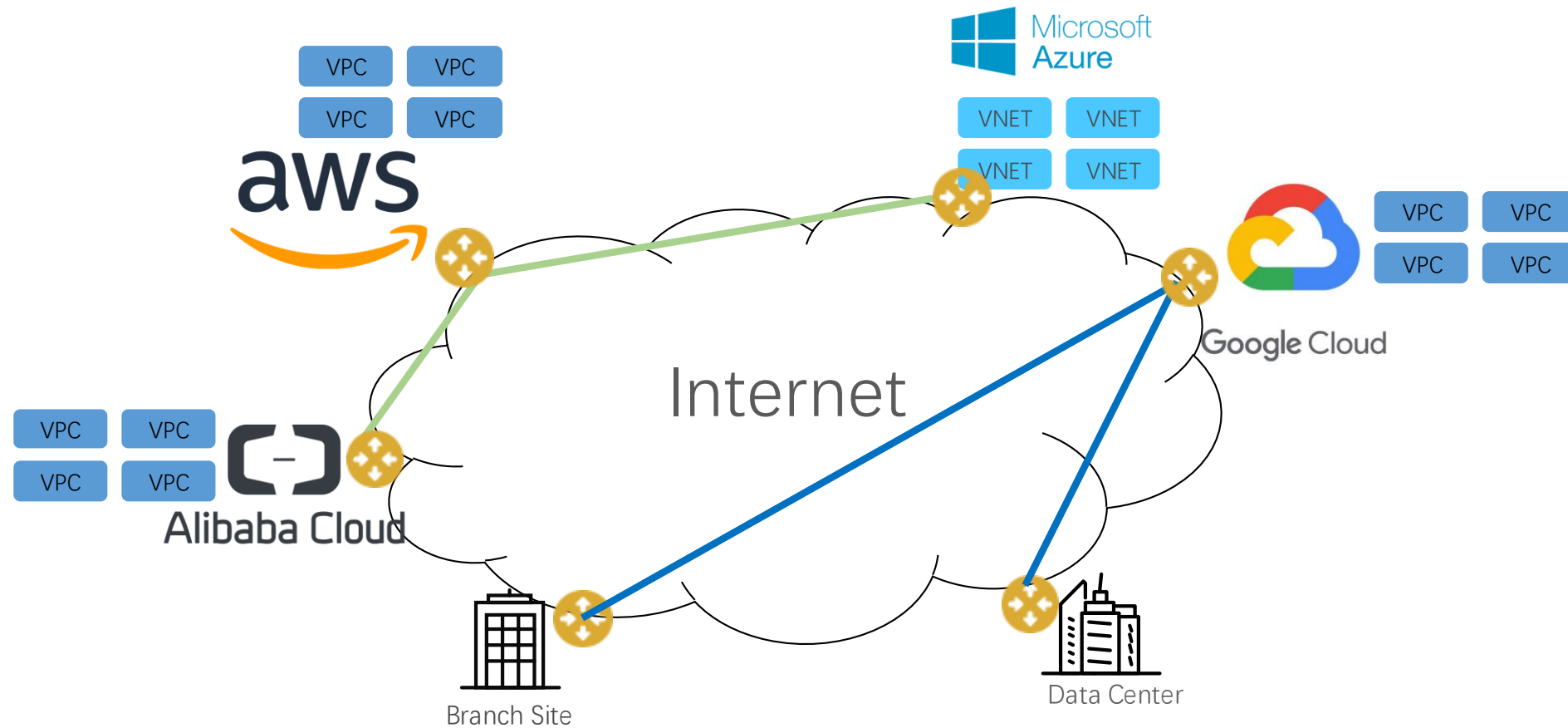
Segment routing to avoid congestion point

Shanghai -> Guangzhou/Hongkong -> Moscow



Problems

How to build overlay across multi-cloud?



Solution ? No!

How to build overlay across multi-cloud?

IPSec :

High latency on tunnel setup, scale problem on tunnel keepalive
Each interim site need to decrypt and encrypt which increase latency

VXLAN:

Lack of standard encryption mechanism and NAT-Traversal

Segment Routing:

Lack of IPv4 support, need use transport layer protocol for encryption

Solution ? **Yes!**

How to build overlay across multi-cloud?

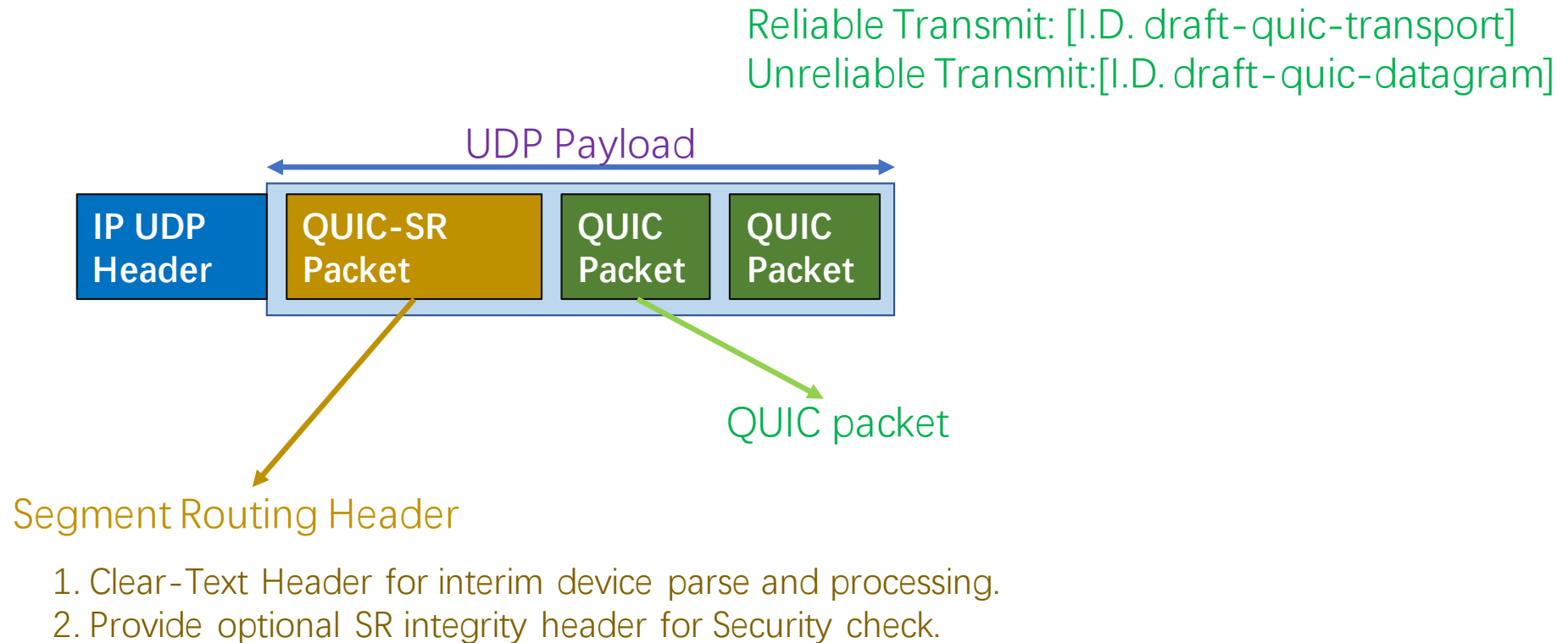


- Stream multiplexing
- Connection migration & Resilience
- Low-latency connection establishment
- Standard encryption mechanism



- Traffic engineering
- Network level programmability

draft-zartbot-quic-sr



Notice: QUIC “packet” is different with IP packet.
It is defined in QUIC RFC as a payload encapsulated in IP UDP Payload
In the following slides “QUIC-SR-Packet” means a QUIC “packet”, not an individual “ip packet”

Why not QUIC over SRv6?

- RFC4023(MPLS over GRE) and RFC7510(MPLS over UDP) does not support NAT-Traversals
- Multi-Cloud VPC inter-connection require secure/reliable/programmable transport layer over IPv4 internet.
- A standard SDWAN transport layer is required to interop with multiple vendors(include cloud and network equipment vendors)

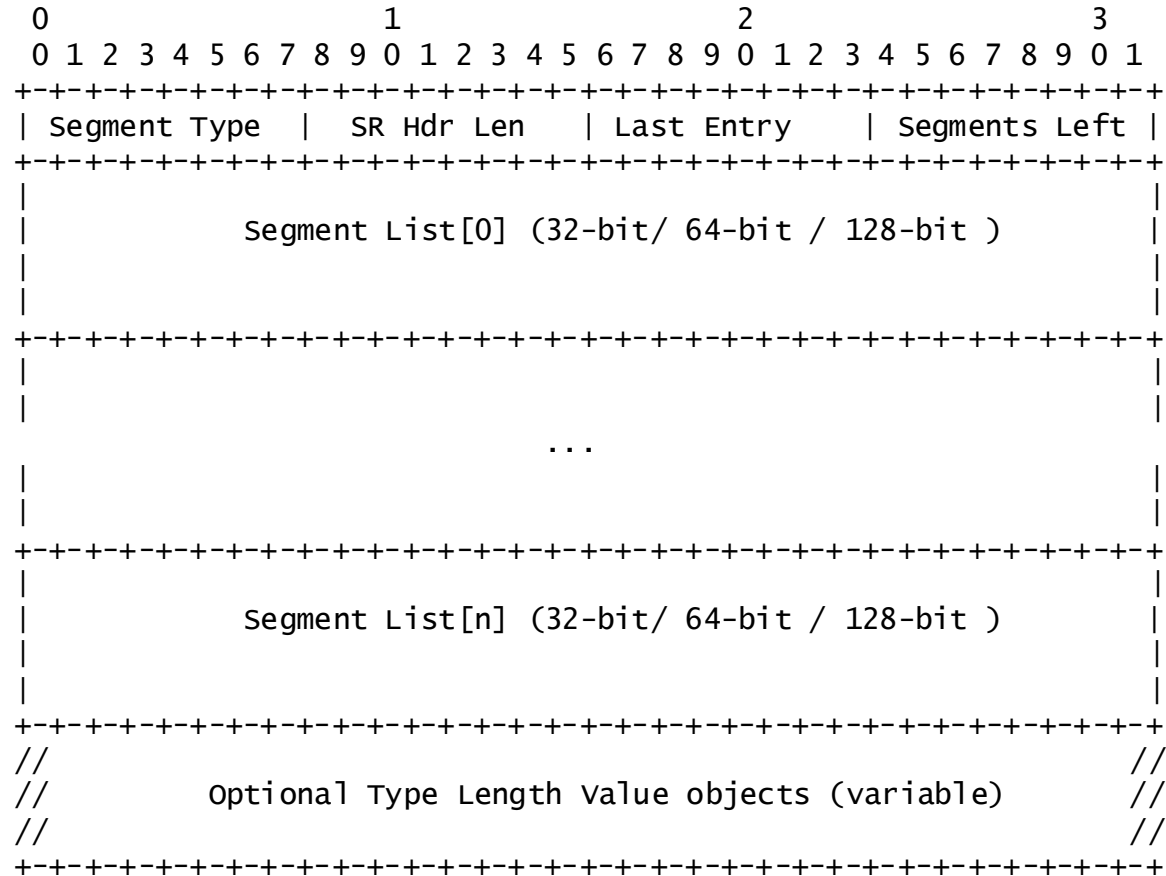
QUIC-SR Header

```
QUIC-SR Packet {  
  Header Form (1) = 1,  
  Fixed Bit (1) = 1,  
  Long Packet Type (2) = 0,  
  -----  
  QUIC-SR Flag(1) = 1,  
  Unused (3),  
  -----  
  Version (32),  
  DCID Length (8),  
  Destination Connection ID (0..160),  
  SCID Length (8),  
  Source Connection ID (0..160),  
  SR-QUIC Header (..),  
}
```

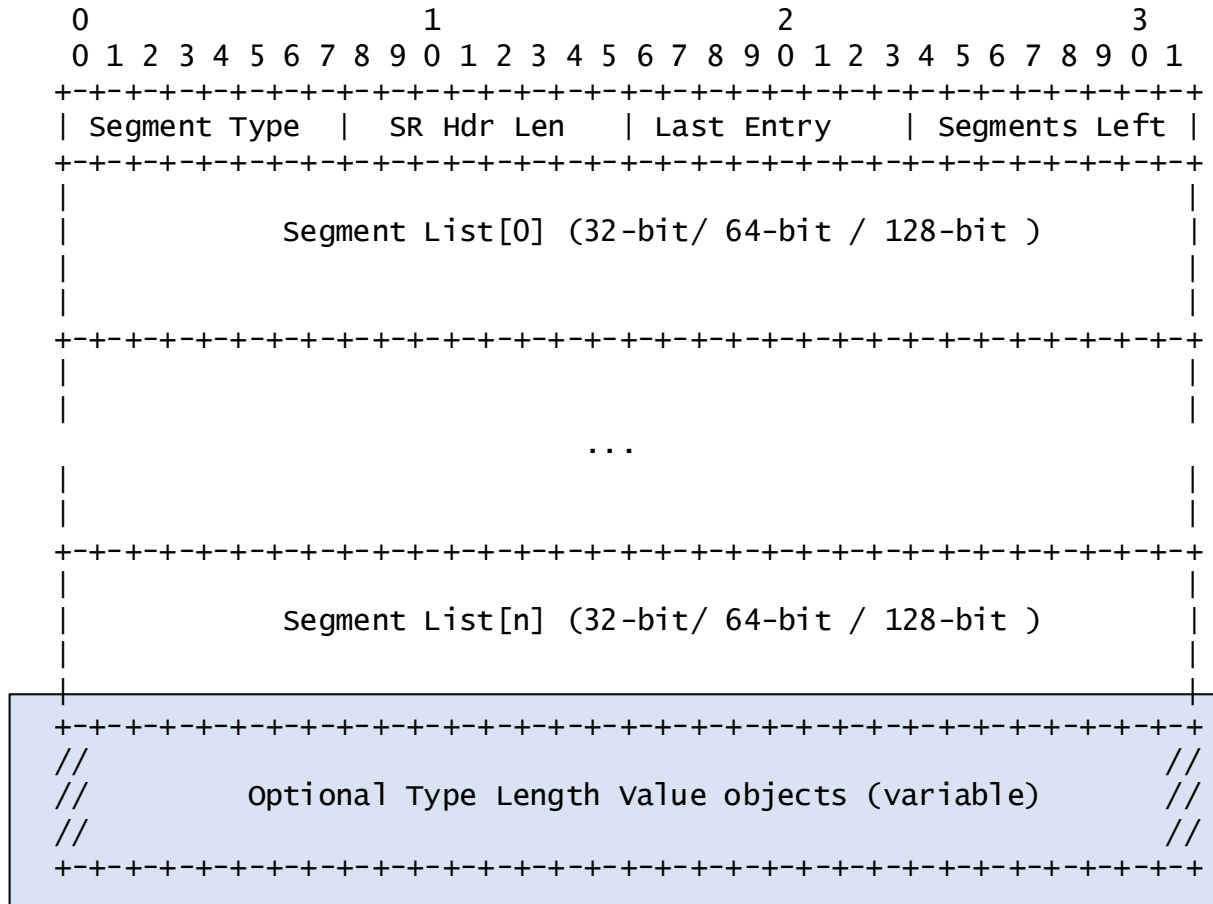
Minor Change on QUIC
Add one bit on Long Packet



SR-QUIC Header(almost same as SRv6)



Optional TLV for SecOps

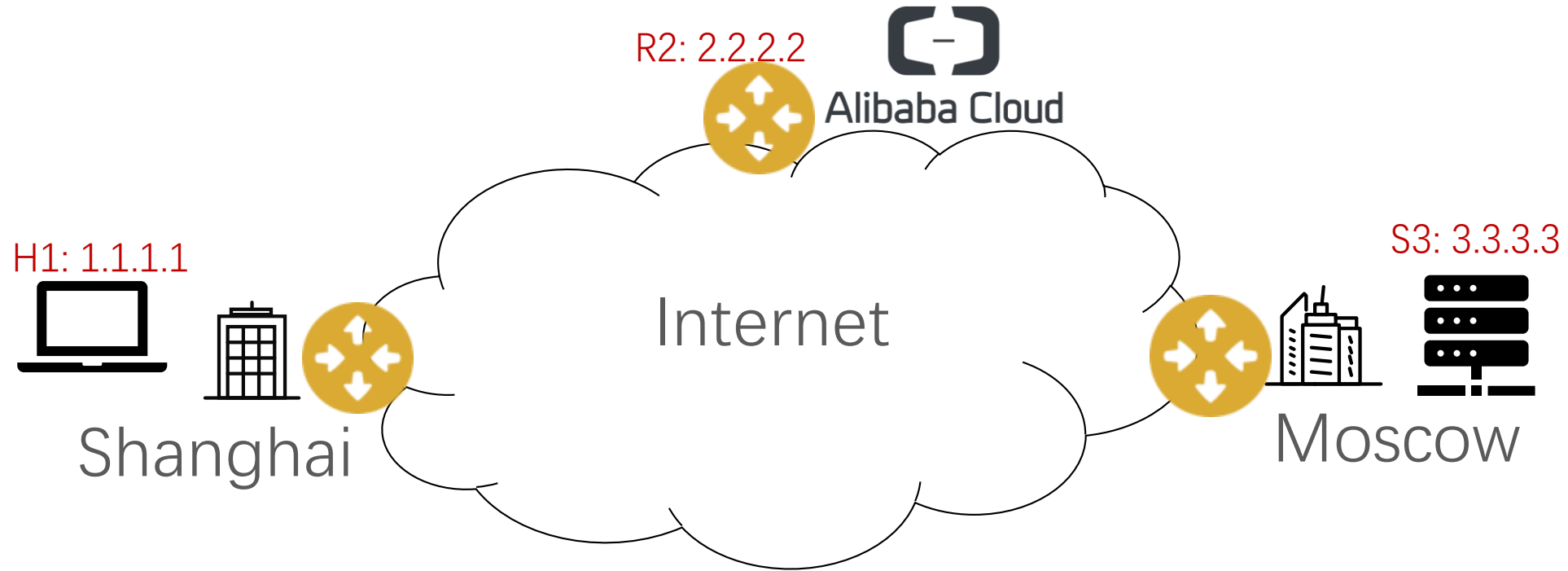


- Micro Segmentation(uSeg) Sub-TLV

```
{  
  0x0, Source Group ID  
  0x1, Destination Group ID  
  0x2, Application Group ID  
  0x3, Source Device ID  
  0x4, Destination Device ID  
  0x5, Application ID  
}
```

Micro-segmentation is a [network security](#) technique that enables security architects to logically divide the [data center](#) into distinct security segments down to the individual workload level,

Use case-1: Traffic engineering over internet



Step.1. Setup interim SR-QUIC enabled Router R2(2.2.2.2)

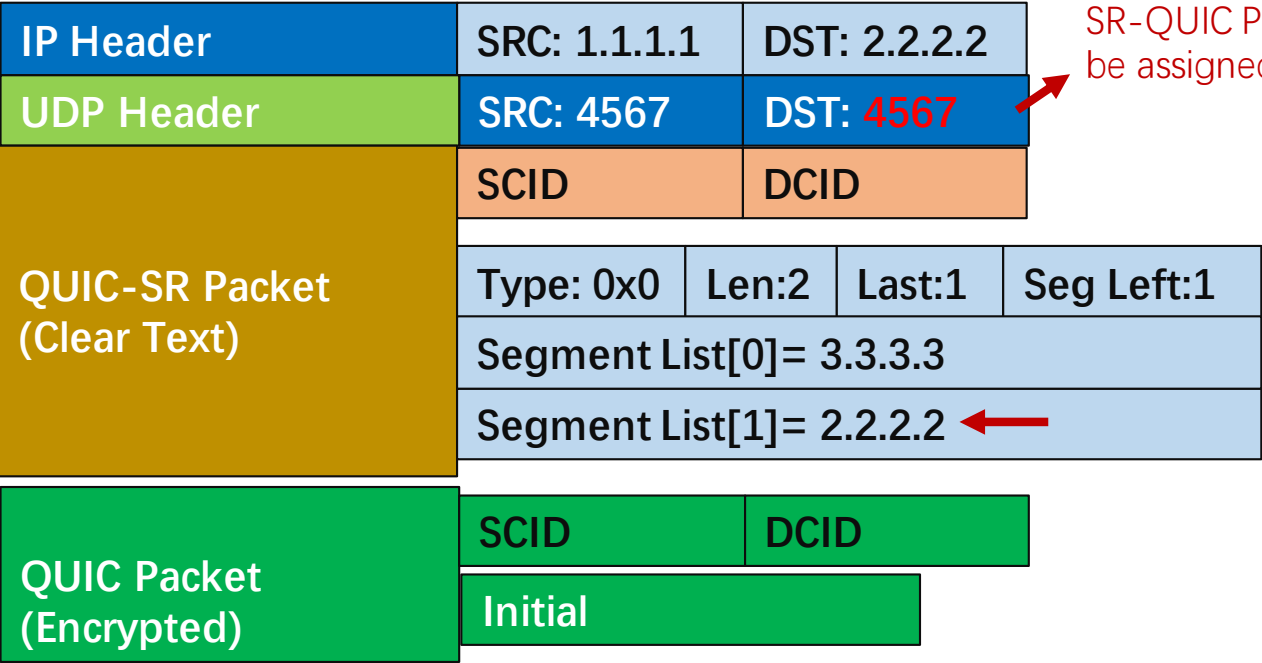
Use case-1: Traffic engineering over internet

H1: 1.1.1.1  Shanghai

R2: 2.2.2.2  Alibaba Cloud

S3: 3.3.3.3  MOSCOW

Step.2. H1 → R2



SR-QUIC Port: need to be assigned by IANA

Use case-1: Traffic engineering over internet

H1: 1.1.1.1  Shanghai

R2: 2.2.2.2  Google Cloud

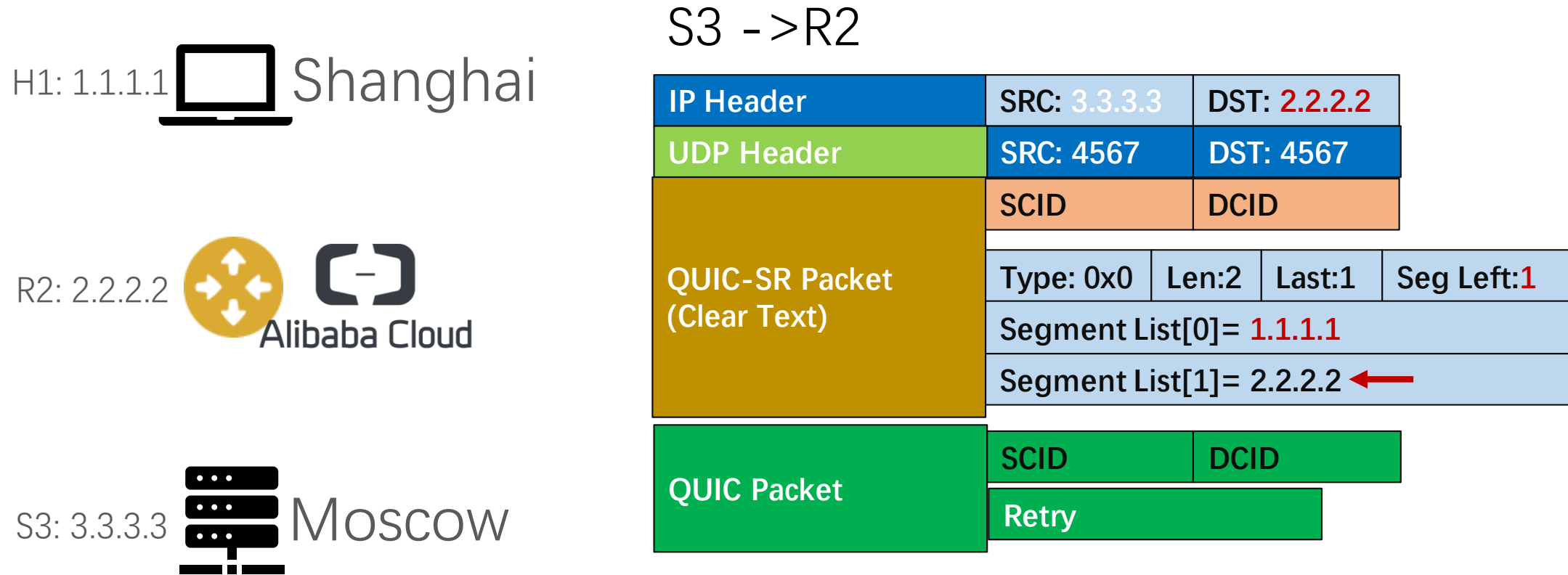
S3: 3.3.3.3  MOSCOW

R2 → S3

IP Header	SRC: 1.1.1.1	DST: 3.3.3.3		
UDP Header	SRC: 4567	DST: 4567		
QUIC-SR Packet (Clear Text)	SCID	DCID		
	Type: 0x0	Len:2	Last:1	Seg Left:0
	Segment List[0]= 3.3.3.3 ←			
	Segment List[1]= 2.2.2.2			
QUIC Packet (Encrypted)	SCID	DCID		
	Initial			

R2 based on Segment list modify destination address
then reduce seg-left field to indicate the offset in Segment List for nexthop device

Use case-1: Traffic engineering over internet



S3 Receive the initial packet from R2, It MUST store the CONNECT-ID with Received IP/Port and Segment List in the session table.

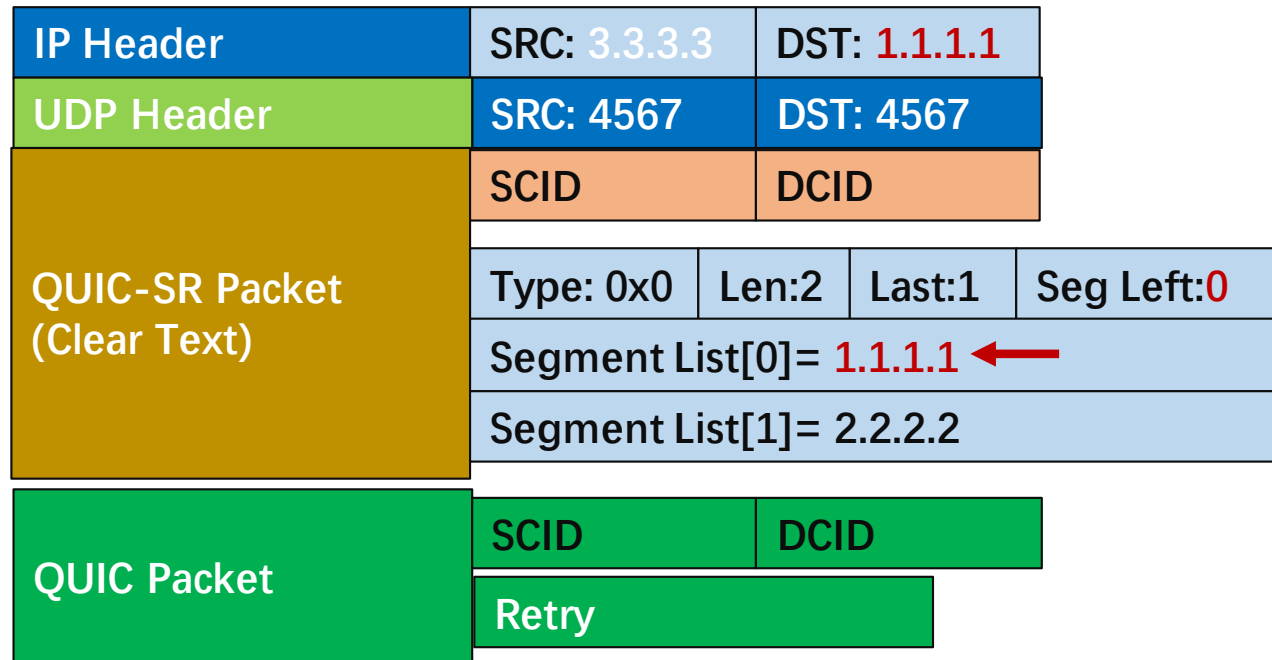
Use case-1: Traffic engineering over internet

H1: 1.1.1.1  Shanghai

R2: 2.2.2.2  Alibaba Cloud

S3: 3.3.3.3  MOSCOW

R2 -> H1




S3 Receive the initial packet from R2, It MUST store the CONNECT-ID with Received IP/Port and Segment List in the session table.

Use case-1: Traffic engineering over internet

NAT Traversals : External STUN server may used to be sync the
H1 Private Address(192.168.1.2:4567) and Public Address mapping(1.1.1.1:45678)
A uSID key-value mapping table cloud be used for NAT-T

H1: 192.168.1.2  Shanghai

1.1.1.1  -NAT

R2: 2.2.2.2  Alibaba Cloud


S3: 3.3.3.3  Moscow


Step.2. H1 → R2

IP Header	SRC: 1.1.1.1	DST: 2.2.2.2		
UDP Header	SRC: 45678	DST: 4567		
QUIC-SR Packet (Clear Text)	SCID	DCID		
	Type: 0x0	Len:2	Last:1	Seg Left:1
	Segment List[0]= 3.3.3.3			
	Segment List[1]= 2.2.2.2 ◀			
QUIC Packet (Encrypted)	SCID	DCID		
	Initial			

uSID mapping table

transport over any protocol(v4/v6/MPLS)

 **etcd**

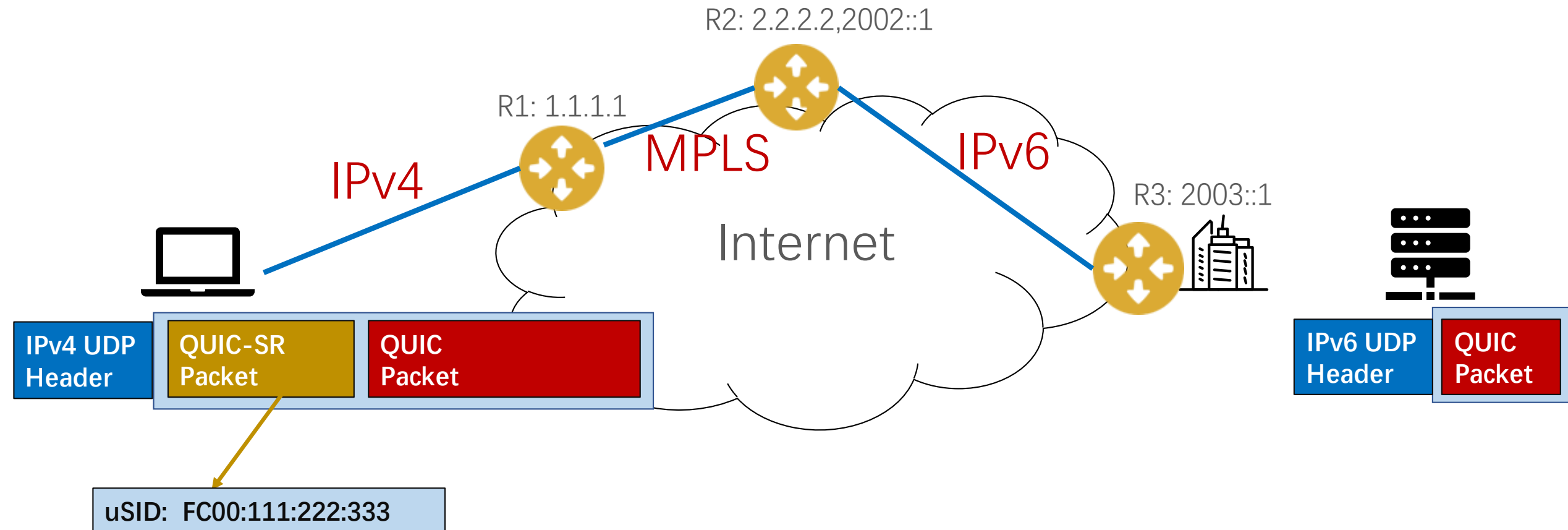


SID	Private IP Port	Public IP Port or Label
111	192.168.1.2:4567	1.1.1.1:24567
222	2.2.2.2/2001::1 port 4567	2.2.2.2/2001::1 port 4567
333	3.3.3.3	MPLS-SR Label: 16333
444	192.168.4.5:4567	4.4.4.4

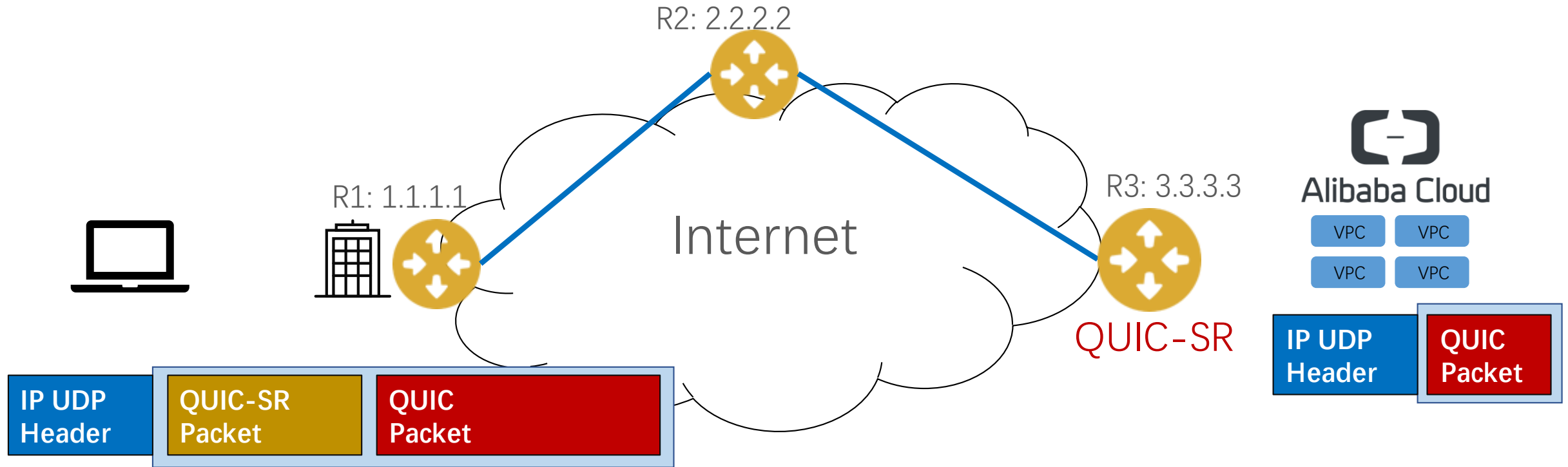
Distributed K-V store

IP Header	SRC: 1.1.1.1	DST: 2.2.2.2
UDP Header	SRC: 24567	DST: 4567
QUIC-SR Packet (Clear Text)	SCID	DCID
	uSID: FC00:222:333:444	
QUIC Packet (Encrypted)	SCID	DCID
	Initial	

uSID mapping table: IPv4-IPv6 interworking with pre-allocated SID and translation K-V store

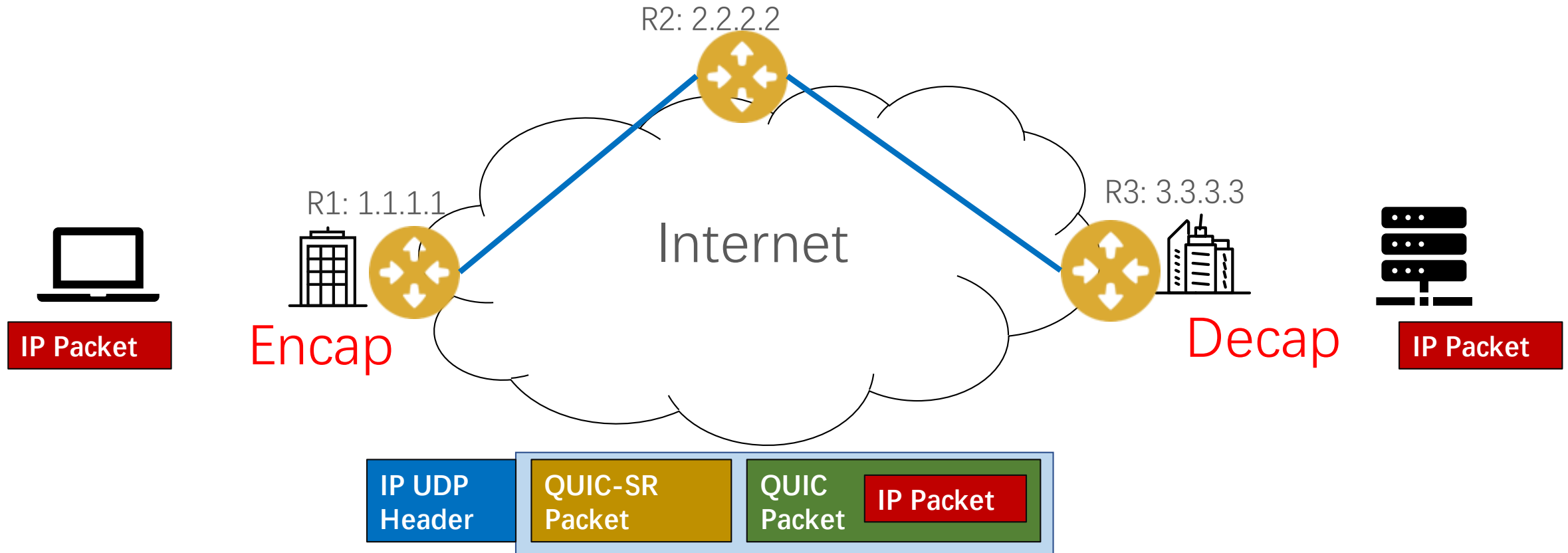


Use case-2: Client-less VPN to VPC Hosts



With SR programmability, Endpoint could encode VPC information in QUIC-SR Packet. R3 based on SRH to setup pinhole or proxy to directly access server inside VPC

Use case-3: SDWAN Tunnel via QUIC-SR



R1 encap packet from client, then send over QUIC-SR socket
R3 decap packet and send the original IP packet to server

Overlay or Tunnel-Less ?

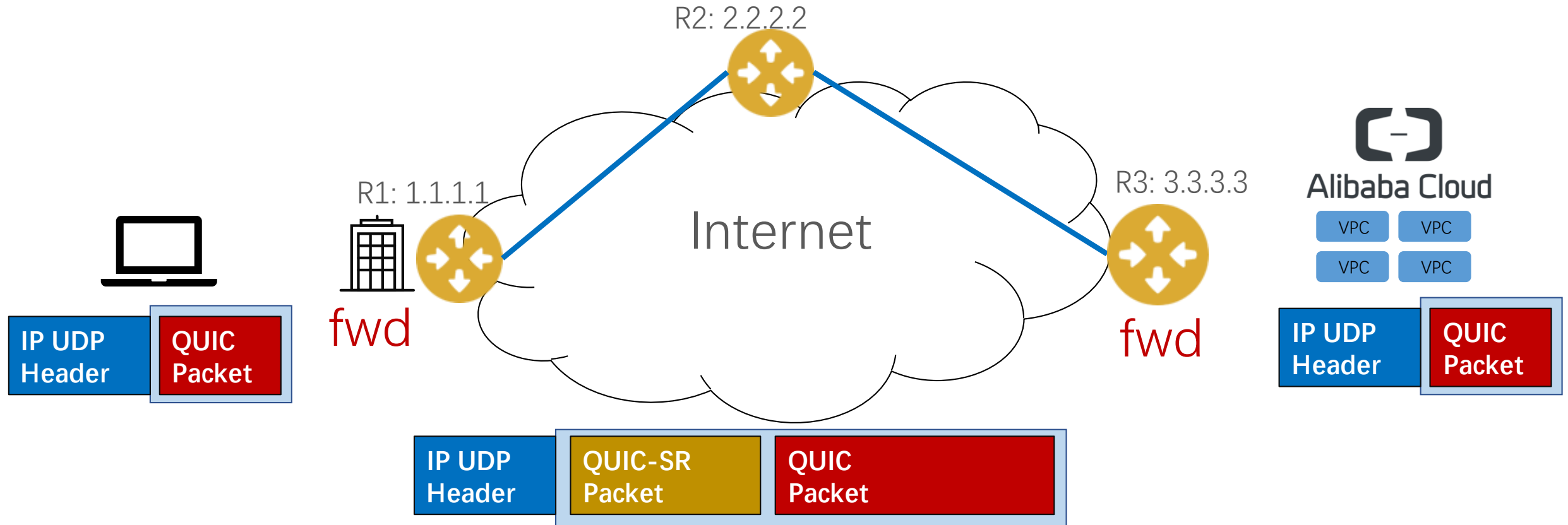
- We've been heavily use overlay technology in the SDN era for a decade.
- Now, **overlay is EVERYWHERE** in:
 - **Access Network**: wire and wireless converged
 - **WAN**: IPsec Tunnel with many private encapsulation
 - **Datacenter**: BGP-EVPN
 - **SmartNIC**: Host Overlay
 - **Container Network**: VXLAN/GRE

Proposal?

QUIC-SR could be used reduce overlay encapsulation by QUIC CONNECTION-ID and SR programmability.

Use case-4: Tunnel-Less SDWAN

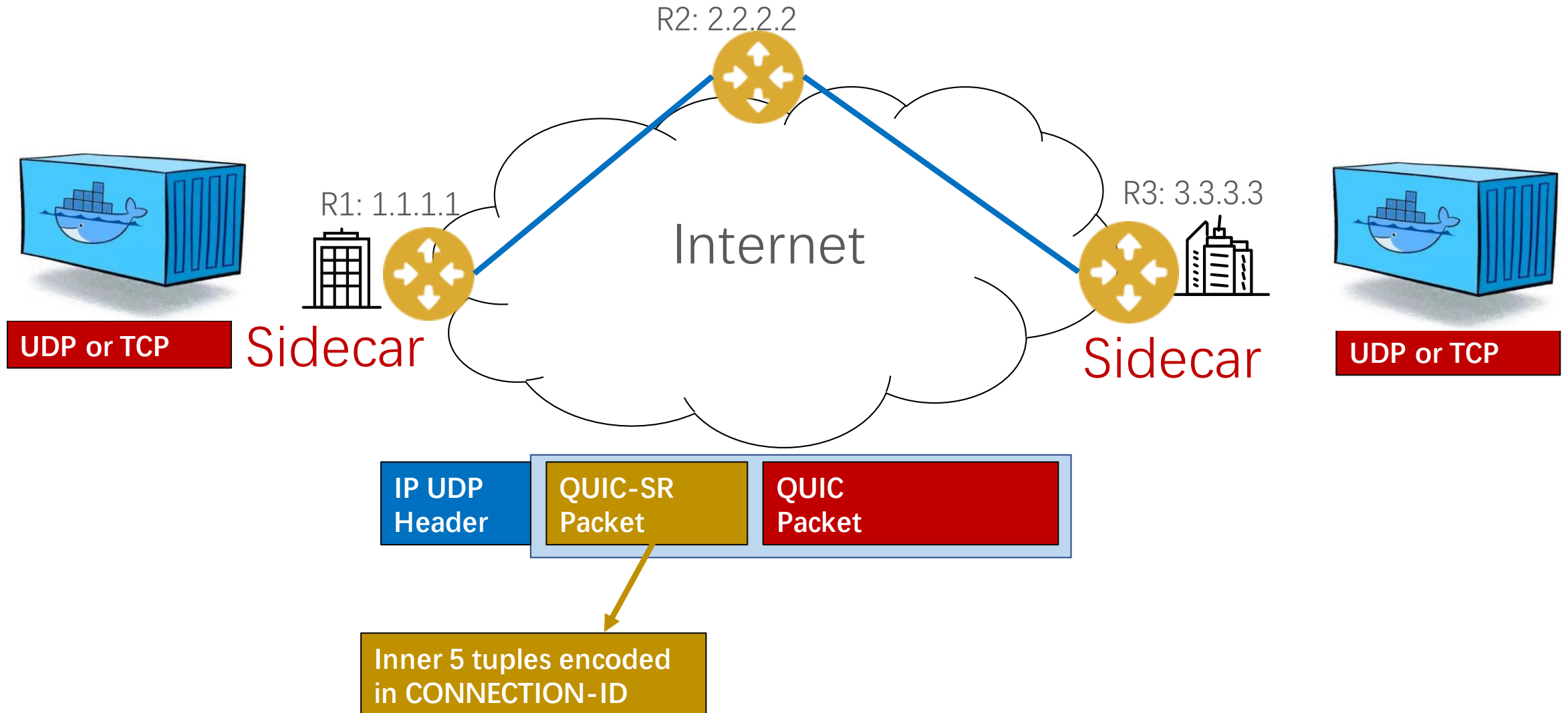
Reduce overlay overhead



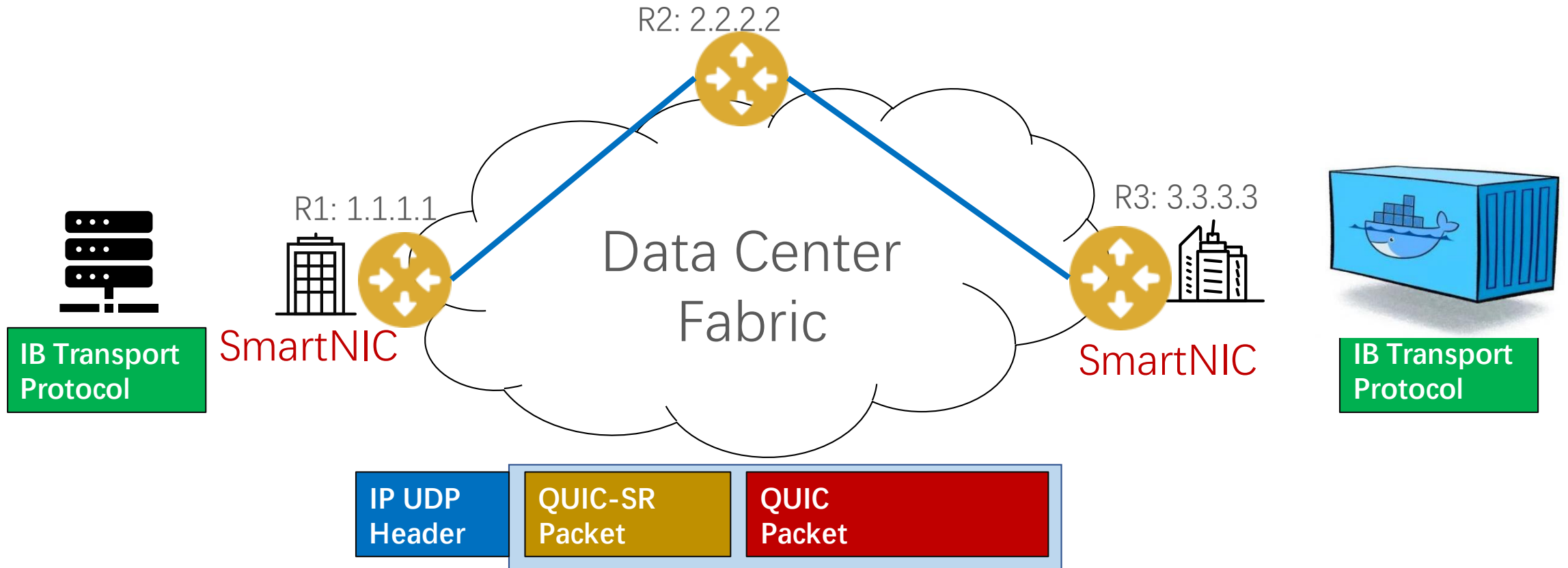
The Interim Router could aware QUIC packet and add QUIC-SR Packet for SR

Use case-5: Proxy Mode

Service-mesh Sidecar and Container Network Interface



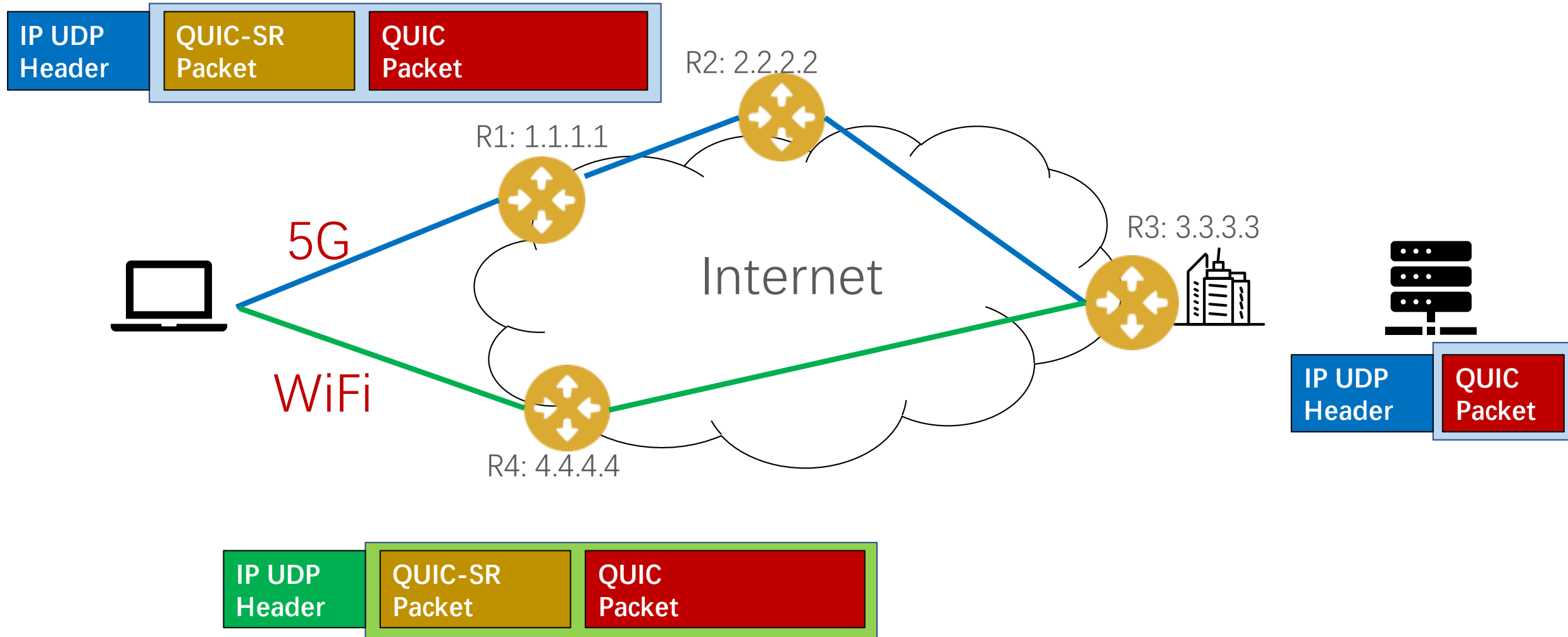
Use case-6: SmartNIC & RDMA over QUIC-SR



SmartNIC could offload crypto function and quic packet encapsulation and flow-control
SR provide path selection to avoid buffer overflow and congestion point.

Use case-7: Converged Access

Reduce overlay overhead & Multipath with same CONNECTION-ID



Use case-8: App Performance Monitor

Same CONNECTION-ID could be used for telemetry data correlation
In-band telemetry could be added in SRH optional header

