Locating Network Failures in Cloud

Long Wang
REASONS Lab, Tsinghua University
at IFIP WG 10.4 in Praia Do Forte, Brazil
Feb 9, 2025

Cloud Computing

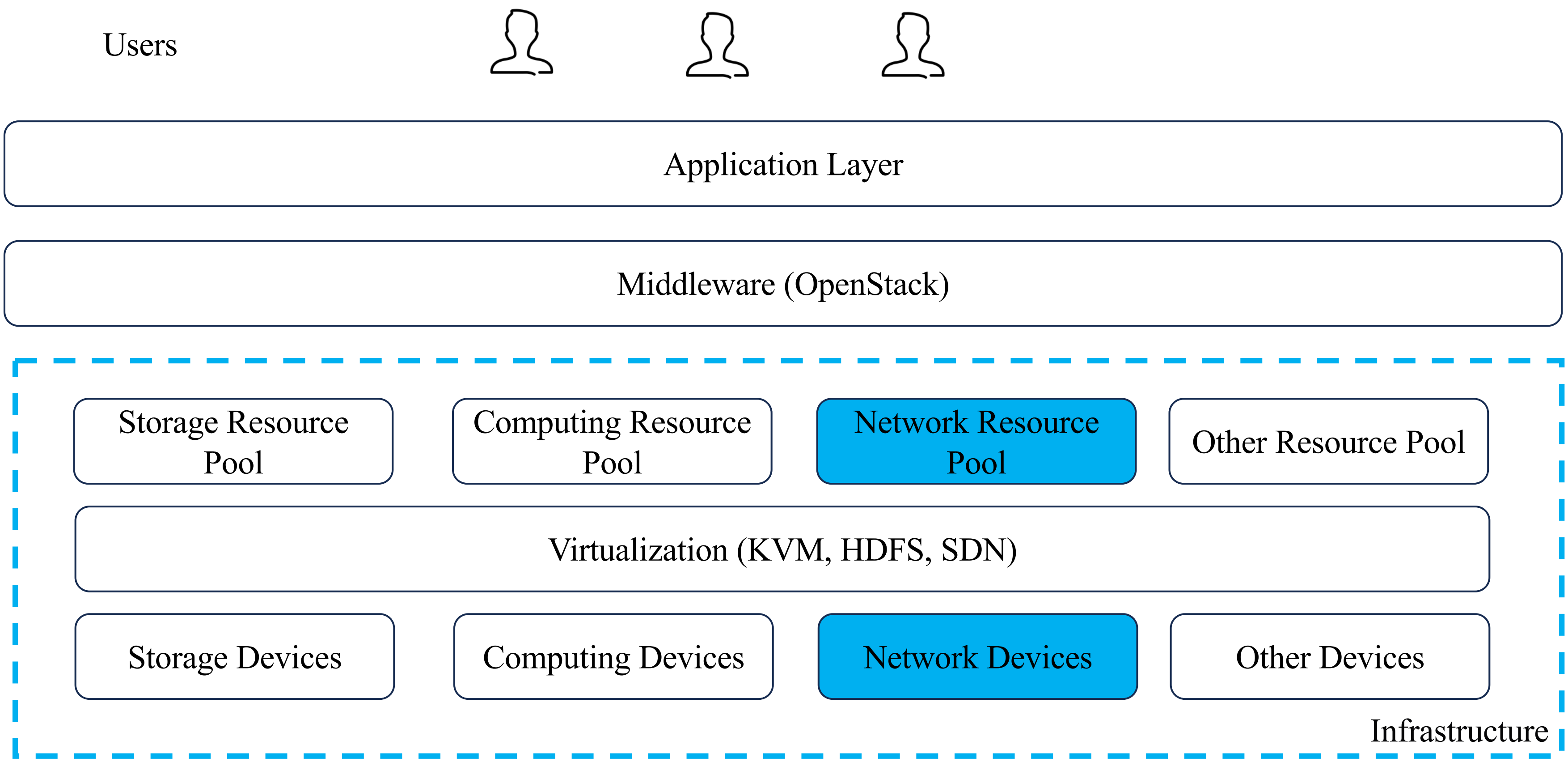
- Cloud computing is the on-demand delivery of IT resources.

aws Azure Google Cloud IBM Cloud

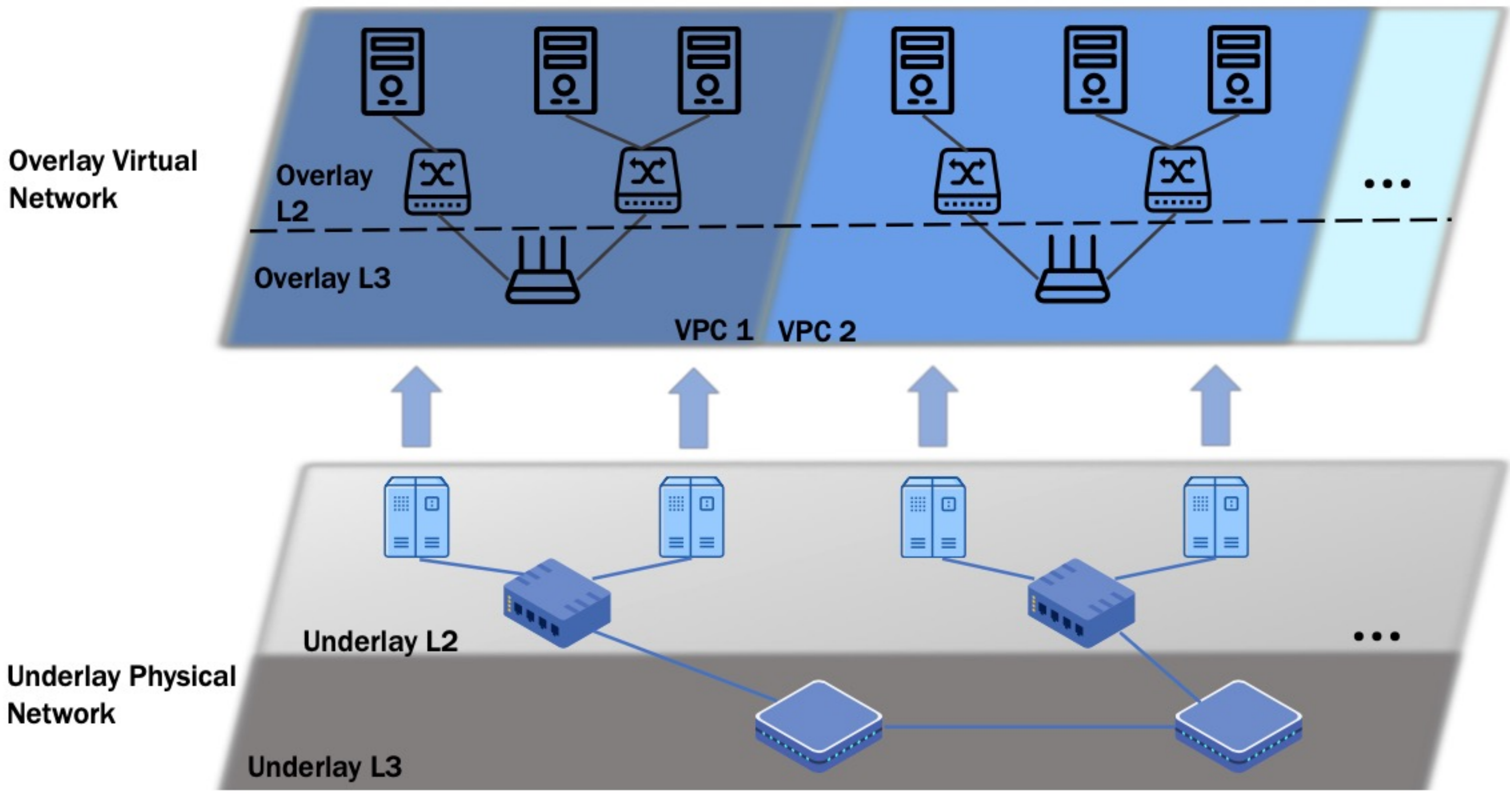
Tencent Cloud HUAWEI HUAWEI CLOUD Alibaba Cloud

- IDC: Global cloud spending hit $706B and is projected to exceed $1.3T by 2025.

Overview of Networking in Cloud



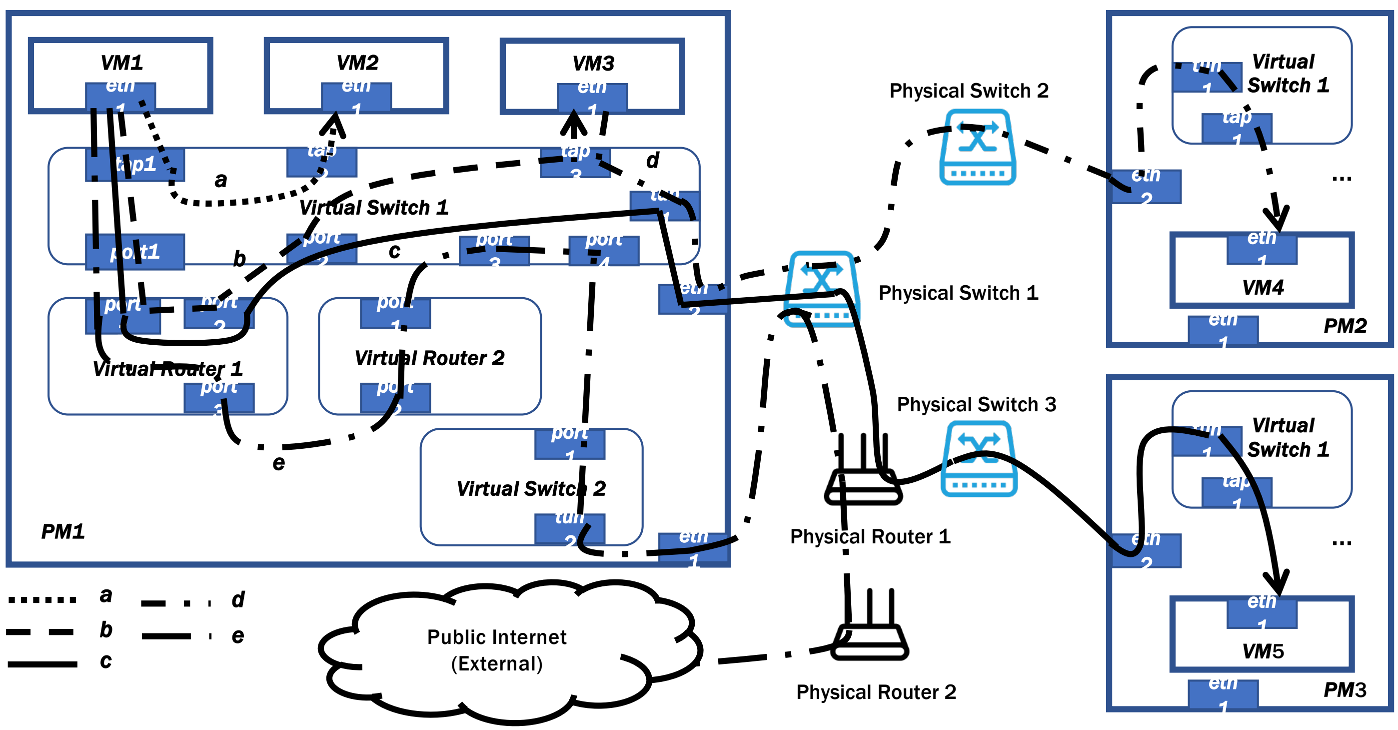
Overview of Networking in Cloud



What are missing in current research on failure diagnosis of cloud networking?

- Lack of systematic analysis of complicated cloud network scenarios
- Lack of systematic study of cloud network failures
- Lack of a unified fault diagnosis framework for the overall cloud network

Typical Networking Scenarios in Cloud



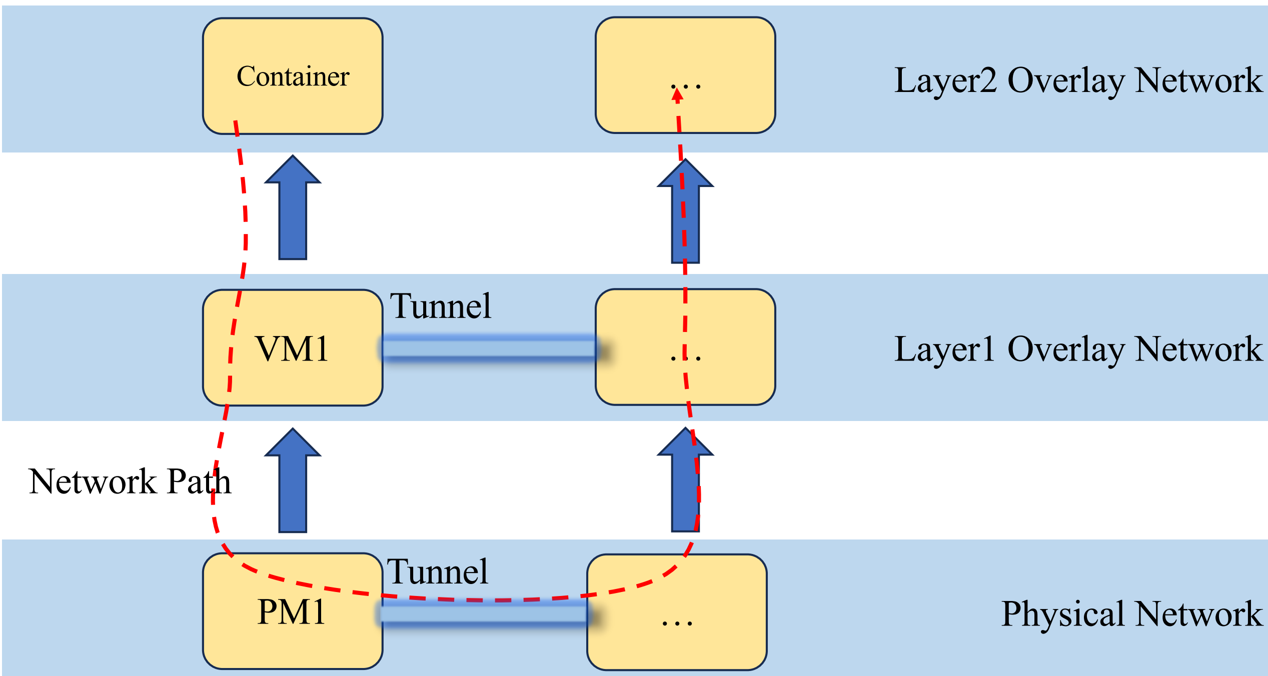
Virtual Link Mechanisms

1. veth-pair:
Bidirectional data channel
(e.g., VS1.port1 ↔ VR1.port1).
2. Tap Device:
OS kernel data structure for
VM network traffic
(e.g., VM1.eth1 ↔ VS1.tap1).
3. Tun Device:
Handles routing at Layer 3
(e.g., PM1.VS1.tun1 ↔
PM1.eth2).

Typical Networking Scenarios in Cloud

Scenario	Type	Technologies Used	Description
a	Intra-Node, Same Subnet	Tap, virtual switching	Traffic between VMs in the same subnet via virtual switch (no routing).
b	Intra-Node, Cross-Subnet	veth-pair, SNAT, virtual routing	Traffic crosses subnets via virtual router (VR1), modifies MAC/IP via SNAT.
c	Inter-Node, Cross-Subnet	Tunneling (tun), physical routing, virtual routing	Overlay traffic tunnels through underlay network, uses encapsulation.
d	Inter-Node, Same Subnet	Virtual switching, physical switching	Overlay traffic tunnels through underlay network, uses encapsulation.
e	Traffic to External Networks	Physical routing (PR2), NAT, veth-pair	Exits cloud via edge router (VR2 → PR2.

Multilayer Overlay Networking in Cloud



- Both layers rely on tunneling mechanisms (e.g., GRE, Geneve) to abstract virtual networks from the underlay.
- Example: Traffic from VM1 traverses Layer 2 tunnels, then Layer 1 tunnels, before reaching the physical network.

Failure Analysis for Cloud Networking

- Data Sources
  - Public incident reports from Google Cloud
  - Public incident reports from AWS
  - Internal data from China Telecom Cloud
  - Public incident reports in research papers
- Analysis Methodologies
  - Classification of network incidents into categories
  - Injection of network faults/failures into cloud and result analysis

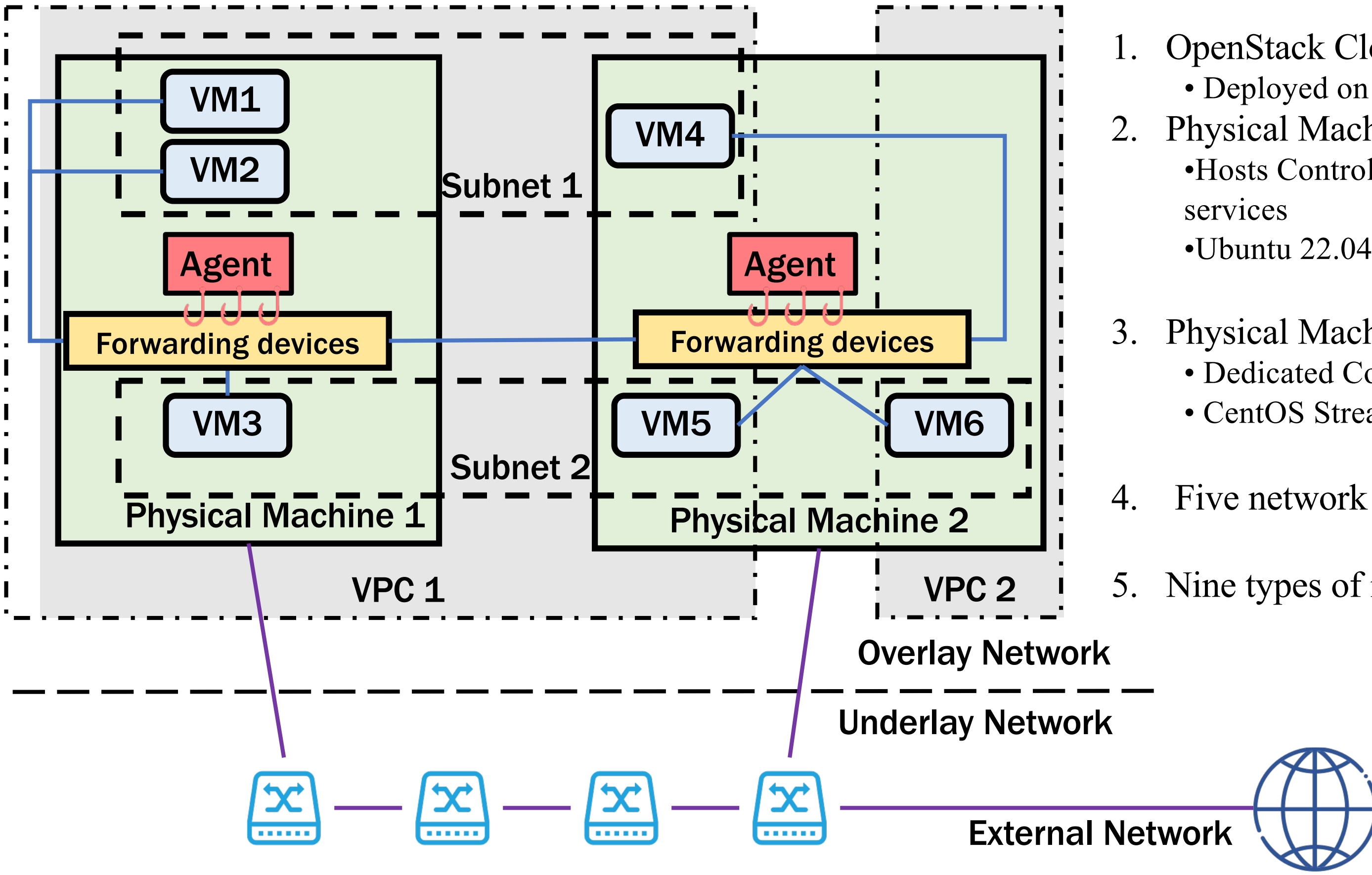
Categories of Cloud Network Failures

- Software Errors:
  - Memory leaks, bugs.
  - Symptoms: Cloud components down or restarting repeatedly.
- Configuration Errors:
  - User-side:
    - VM routing errors, security group misconfigurations.
  - Provider-side:
    - Misconfigured cloud components (e.g., wrong listening ports, outdated database configs).
    - Network topology errors (e.g., incorrect VPC peering IP, faulty traffic routing).
- Hardware Errors:
  - Power outages, fiber cuts, hardware damage.

Fault Injection

Scenario / Fault Type	**Injection Environment**	**Injection Method**
SDN Controller Software Bug	Management network	Modify flow tables via SDN Python script (e.g., ovs-vsctl add-flow ... action=drop).
User VM Routing Misconfiguration	User network	Delete default route in VM.
User Security Group Misconfiguration	User network	Remove security group.
Component Port Misconfiguration	Management network	Modify port and restart component service.
Monitoring Database Outdated Config	Management network	Alter VM IP mappings in monitoring scripts.
VPC Peering IP Misconfiguration	Management network	Modify Geneve tunnel local_ip and restart OVN controller.
Faulty Forwarding Route Configuration	Management network	Add incorrect gateway.
Packet Loss (Drop)	User network	Use tc to drop packets on target device.
Packet Latency	User network	Use tc to add delay to packets on target device.

Fault Injection Experimental Setup



1. OpenStack Cloud Environment:
   - Deployed on 2 Physical Machines (PMs)
2. Physical Machine 1 (PM1):
   - Hosts Control, Compute, Storage & Network services
   - Ubuntu 22.04 LTS
3. Physical Machine 2 (PM2):
   - Dedicated Compute Node
   - CentOS Stream release 9
4. Five network scenarios.
5. Nine types of network faults.

Fault Injection Results – Observed Phenomena

1. VM Routing Error (Drop Rule):
   • Faulty VM’ s packets appear only on its NIC; connectivity fails bidirectionally.
2. Default Route Deletion in VM:
   • Outgoing packets display broadcast MAC (ff:ff:ff:ff:ff:ff); incoming packet can reach fault VM.
3. Security Group Deletion:
   • Outgoing path shows only the faulty VM’ s NIC; incoming path from others lacks the faulty NIC; connectivity fails bidirectionally.
4. Nova Port Listening Error:
   • No impact on existing connectivity; affects new VM creation and VNC access.
5. Monitoring DB Error:
   • Monitoring reports VMs as unreachable by name despite normal IP connectivity.
6. Geneve Tunnel Misconfiguration:
   • VMs on the affected PM lose connectivity with other PM’ s VMs; path breaks at the bridge or Geneve interface.
7. User Traffic Forwarding Route Misconfiguration:
   • Faulty subnet VMs cannot connect to VMs in normal subnets.
8. Packet Loss.
9. Packet Delay.

Cloud Network Failure Diagnosis

- We design an end-to-end packet tracing mechanism to diagnose cloud network failures.
- Our approach targets three failure types:
  - Network connectivity broken: Packet loss of 100% (no probe response).
  - Delay: Packets arriving later than expected (excessive latency).
  - Packet Loss: Probabilistic drops along the path (partial loss).
- By collecting events from all forwarding devices, we can pinpoint the failure point.

Basic Idea

- Objective: Locate network failure points when traffic fails to reach the destination.
- Steps:
  - Probe Packet: Send a probe packet from source VM to destination VM from the host PM (without entering tenant VM).
  - End-to-End Path Collection: Collect the complete path at the port-level across both overlay and underlay networks.
  - Failure Identification: Detect where the path is broken or deviates from the intended route.

Packet Tracing

- Deployment:
  - Load eBPF functions into forwarding devices (virtual switches/routers, PMs) and use P4 programs on programmable PS/PR.
- Probe Packet Identification:
  - Use the IP Options field with a hard-coded magic string and unique identifier.
- Event Generation:
  - When a probe packet is matched, the device sends an event (containing fields such as probe_ID, in_port, out_port, phy_device_ID, sequence_num, boundary_type, packet_info) to a central Path Generator.
- Ordering:
  - Use Ringbuffer mechanism to assign sequence numbers, ensuring the correct event sequence across multiple CPUs.

Event Linking

- Event Linking
  - use the common probe_ID and ordered by sequence_num; matching fields (e.g. dst_IP in packet_info) connect events from one device to the next, forming the complete end-to-end path.
- Failure Point Locating
  - The final end-to-end path reveals the break point (e.g. out_port set to "dropped") or a deviation from the intended path.

Steps

1.Collect Events:
  •Gather all events with the same probe_ID.

2.Partition by Device:
  •Group events by phy_device_ID (e.g. PM, PR, PS).

3.Local Linking:
  •Within each group, sort events by sequence_num and link them in time order.

4.Global Linking (Overlay):
  •Link the last event from the source PM’s bucket to the first event of the next PM (using destination IP in packet_info).

5.Underlay Integration:
  •Identify boundary events (boundary_type 1 and 2) and consult underlay topology to link neighbor PS/PR events.

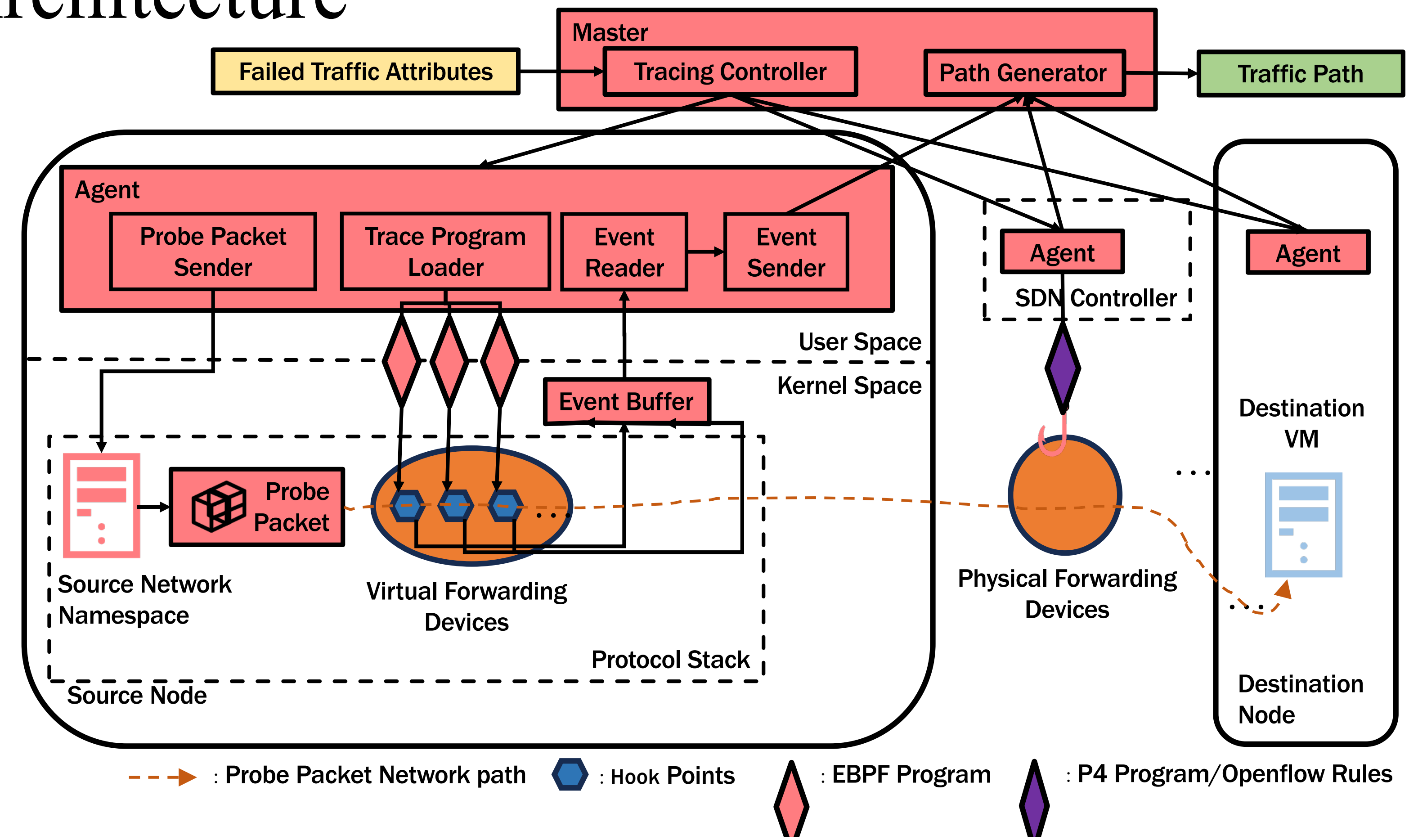
Event Structure

Each event is represented as a tuple:

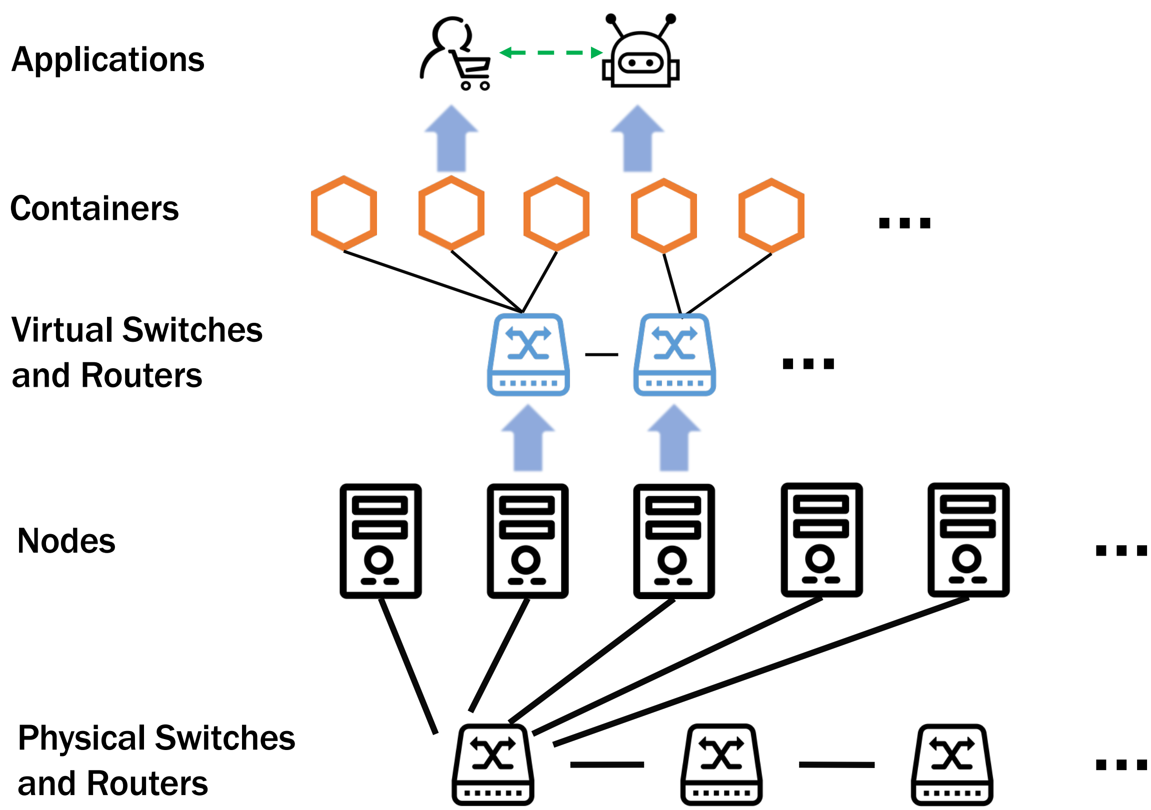
<probe_ID, in_port_name, out_port_name, phy_device_ID, sequence_num, boundary_type, packet_info>

Field	Description
probe_ID	the ID that differentiates a probe packet from other ones. It is the ICMP packet's *identifier*.
in_port_name	the unique name, or ID, of the port from which the packet comes into the forwarding device
out_port_name	the unique name, or ID, of the port to which the packet will be transmitted by the forwarding device. If this packet is dropped at this device, this field is set to the "dropped" string
phy_device_ID	the ID of the physical device where the packet is processed. For a physical switch/router it is its unique ID or name; for other devices it is the IP of the PM that hosts the device
sequence_num	when phy_device_ID is a PM's IP, this is the sequence number of the event, used to indicate the order in which an event occurs on a PM. The larger the sequence number, the later the event occurs; otherwise, this field is not set
boundary_type	this field indicates whether the current path segment is at the overlay-underlay boundary. 0: not at boundary; 1: from overlay to underlay; 2: from underlay to overlay
packet_info	the current packet's header at this hook point (usually at the exit port of the forwarding device). It includes at least *ethernet: (src_mac, dst_mac), IP: (src_IP, dst_IP)*. Certain payload data, e.g. the overlay network packet information encapsulated in an underlay network tunnel, may also be included. This field is used for both event linking and conveying diagnosis information for help engineers figure out potential root cause of the failure.

Architecture



Experimental Setup



Experimental Setup (cont.)

- Cloud Environment:
  - 30 nodes (VMs) with 300 containers in 2 subnets
  - 6 physical switches (each connects to 5 nodes)
  - Built with Mininet (physical network simulation), OVS, Docker, and BMV2 for P4 switches
- Node Hardware (per node):
  - 4 CPUs, 4GB Memory, 128GB Disk
  - OS: Ubuntu 22.04 LTS
- Workload:
  - Stan's Robot Shop: 20 services, 60 clients generating randomized web requests
- CloudNetPath Deployment:
  - Agents deployed on all nodes; Master & Underlay SDN Controller run on a dedicated PM (Xeon Silver 4214R, 24 logical cores, 128GB memory, Ubuntu 22.04.1 LTS)

Failure Scenarios in Experimental Evaluation

R1: Misconfiguration of forwarding rules (e.g., drop/wrong port)

R2: Controller–device disconnection (forwarding device failure)

R3: User misconfiguration in VM networking (e.g., route deletion)

R4: Physical link cut (optical fiber break)

R5: PM power failure (node shutdown)

Additional Simulations:

- S1: Network card failure (using "ip link set … down")
- S2: Link failure (delete port in OVS / set 100% loss via Mininet)
- S3: Forwarding device failure (clear flow/routing table)

User Traffic Forwarding Route Misconfiguration: Erroneous route added on router (faulty subnet gateway)

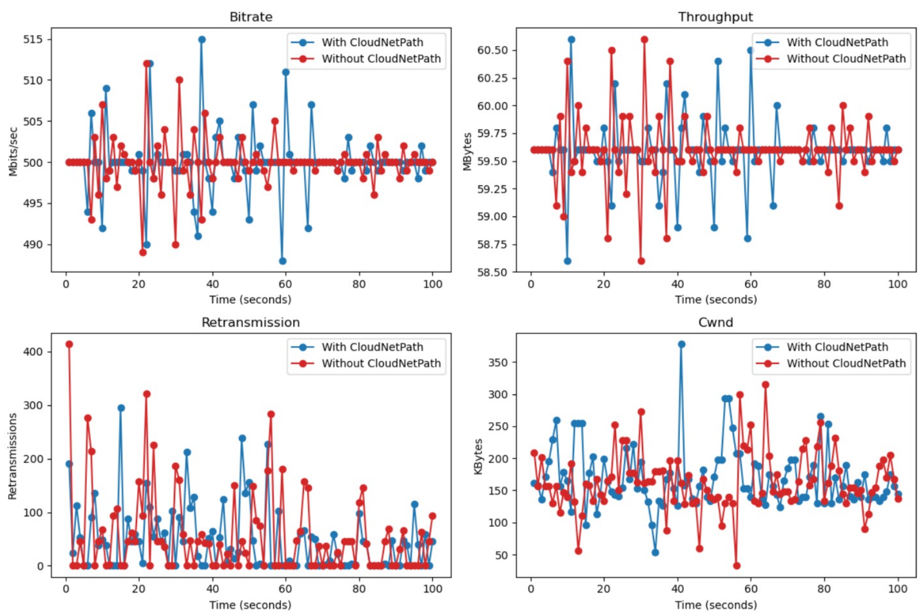
Preliminary Experimental Results

- For all of the five categories of cloud network scenarios
  - Intra-Node, Same Subnet
  - Intra-Node, Cross-Subnet
  - Inter-Node, Cross-Subnet
  - Inter-Node, Same Subnet
  - Traffic to External Networks
- Event Linking Effectiveness: Reconstruct the complete end-to-end/broken path by linking events.
- Network Fault diagnosis in all scenarios accurately pinpointed failure locations, e.g., truncated paths or missing NIC events.

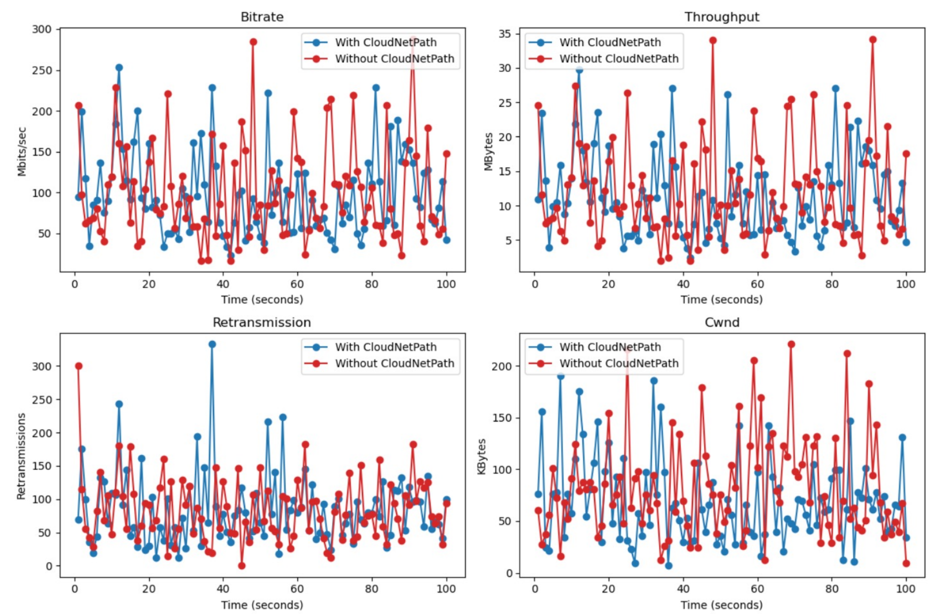
Preliminary Experimental Results (cont.)

- Overall Experiments:
  - Total experiments: 1110 (automated environment creation, fault injection & simulation)
- Key Questions & Answers:
  - Q1: CloudNetPath generates complete port-level paths (89,700 paths collected)
  - Q2: Accurately locates network failures in both overlay and underlay (1110/1110 experiments correct)
  - Q3: Distinguishes forwarding failure vs. packet drop failure:
    - Forwarding failure: Path shows erroneous forwarding (wrong port)
    - Packet drop: Path directly shows a break at the drop point
  - Q4: Performance Overhead:
    - Same-node throughput decreases by 1.63%
    - Different-node throughput decreases by 5.94%

End-to-end Performance Overhead of the Proposed Diagnosis



(a) The server and the client are on the same node



(b) The server and client are on the different node

A minimal overhead (3–5%) is incurred

Summary of the Cloud Network Diagnosis Approach

- Mechanism Overview:
  - Traces the end-to-end path of a probe packet at port granularity.
- Key Components:
  - Packet Tracing: Uses eBPF/P4 programs in forwarding devices to generate detailed events.
  - Event Linking: Aggregates and orders events to reconstruct complete paths.
  - Fault Localization: Accurate identification of failure points in both overlay and underlay networks.
- What we provide:
  - Fine-grained, port-level diagnostic information to assist engineers in quickly locating network failures.

Thanks!