

# Container network architecture and performance analysis of Macvlan and IPvlan

Liangbin Zhang\*, Yuanming Wang, Shibin Liang, and Ran Jin

Zhejiang Wanli University, Ningbo, 315100, China

**Abstract:** Macvlan and IPvlan modes allow the host physical Network Interface Card (NIC) to create several virtual sub interfaces, which can meet the containers communication requirements across different network segment in complex application scenarios. The paper describes the architecture characteristics of these two modes in depth and builds container overlay network in single machine and multi machine environments. Around the key network indicators such as round-trip time (RTT), bandwidth and jitter, Ping and Iperf tools are used to test and analyse the end-to-end container communication performance. The experimental results show that IPvlan mode has better network performance and provides a comprehensive and rich application scenario for the container overlay network.

## 1 Introduction

Docker is the most popular lightweight virtualization container platform at present. Because Docker shares the host operating system, it greatly saves system resources such as CPU, memory and storage, and saves license and OS patch maintenance costs. Compared with traditional virtual machines, Docker has many advantages, such as less resources, fast startup and convenient migration. More and more applications are running in the form of container in the development, testing, and production environment [1-2].

However, Docker technology is not perfect, and container network has certain weaknesses. The background of modern application architecture is becoming more and more complex, and service provision often requires cooperation with multiple containers. How to build a secure and high-performance container network is the core problem to be solved in the industry. IPvlan and Macvlan are new modes of container overlay network [3], several virtual sub interfaces are allowed to be created from the host physical Network Interface Card (NIC) to interconnect the container, so that it is regarded as equal status of the container and the host in the physical network. Container can communicate with each other in Layer 2 and Layer 3 network between different hosts.

This paper constructs container overlay network environment for Macvlan and IPvlan within multi hosts, and uses Alpine and Iperf official container images to implement experimental testing and analysis of end-to-end container network transmission performance. IPvlan not only breaks through many limitations of Macvlan network in

---

\* Corresponding author: [zlb@zww.edu.cn](mailto:zlb@zww.edu.cn)

application, but also has excellent network transmission performance across different network segments, providing useful reference for building high-performance and production level container network architecture in the future. The remainder of this paper is organized as follows. Section 2 introduces the related work. Section 3 describes container network architecture of Macvlan and IPvlan. Section 4 shows the experimental performance test and result analysis of Macvlan and IPvlan. Section 5 summarizes our work.

## **2 Related work**

Docker containers are used in large-scale deployment, elastic scaling and other application scenarios. The degree of dispersion of container networks is significantly higher than that of traditional virtual machine network [4]. Management and control have become a difficult problem in large-scale container cluster deployment. Zhang weiqi [5] realized flexible adaptation and migration of container network based on Macvlan. Yang xin et al [6] pointed out that using the Macvlan network can solve the problem of docker network port conflict in the mesos cluster. Li wei et al [7] carried out experiments to verify that Macvlan container network does not cause significant loss of network transmission performance. Feng mingzhen [8] designed and implemented Docker container network system based on Macvlan, and the entire system can be seamlessly integrated with Docker daemon. However, Ipvlan is regarded as the upgrade iteration technology of Macvlan and there are few relevant references on IPvlan except the Docker official website. Therefore, research on the characteristics and performance of IPvlan and Macvlan can provide a better solution for container interconnection in complex networks.

## **3 Container network architecture of Macvlan and IPvlan**

### **3.1 Network architecture of Macvlan**

Typically, one host NIC can create only one Macvlan network to connect internal containers. Meanwhile, IEEE 802.1Q vlan trunk protocol allows one host NIC to create several virtual sub interfaces, and each virtual sub interface is bound to the designated Macvlan container network by labeling incoming and outgoing data frame of container with the designated Vlan Tag. With the route forwarding of network Layer device such as router or Layer 3 switch, Macvlan container can achieve cross Layer2 and Layer 3 network communication within multi hosts [9-11].

As shown in Figure 1, host 1 and host 2 create two new Macvlan networks with two sub virtual interfaces, host 3 creates two new sub interfaces as gateway of corresponding Macvlan network respectively, and host 3 enables routing and forwarding, which can realize container communication across different network segments within multi hosts. Docker network isolation method based on Macvlan can effectively use the bandwidth of the host NIC to achieve nearly lossless network performance, which is particularly suitable for applications such as network traffic monitoring. However, Macvlan network requires the host NIC to enable promiscuous mode, and allows uplink physical switch port associated with multi Mac addresses, which will bring certain hidden danger to the container's network communication security.

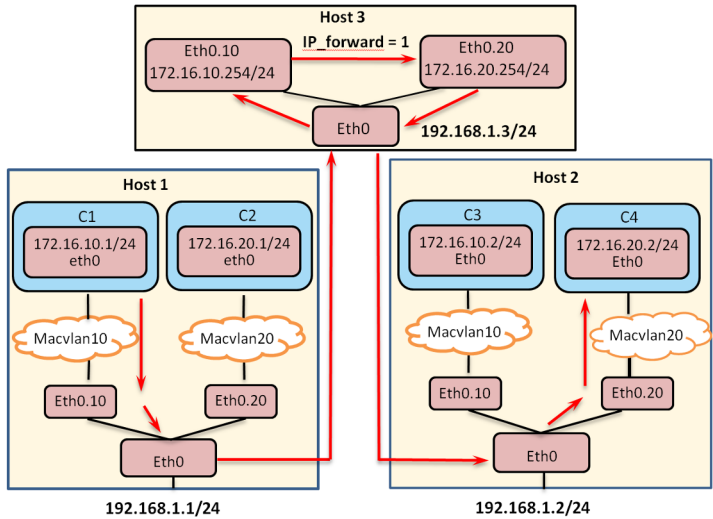


Fig. 1. Network architecture of Macvlan within multi hosts.

### 3.2 Network architecture of IPvlan

Because the application scenarios of Macvlan container network are subject to many restrictions, the new mode IPvlan network is generated iteratively to improve the performance and security of container network communication.

IPvlan requires at least 4.2 Linux kernel, and allows the host NIC to create several virtual sub interfaces to connect internal container of IPvlan network. All containers have the same Mac address. IPvlan network does not receive multicast or broadcast messages, and all ARP processes or multicast messages on the network are completed at the host NIC [12].

The working principle of the Layer 2 mode of the IPvlan is similar to that of the Bridge mode of the Macvlan. The Layer 3 mode of the IPvlan supports the message forwarding across network segments within current host. If the routing rule to reach the IPvlan container is configured on the external router, the IPvlan network can realize communication between the container and the external network. Figure.2 shows that different IPvlan network container can communicate with each other within the current host.

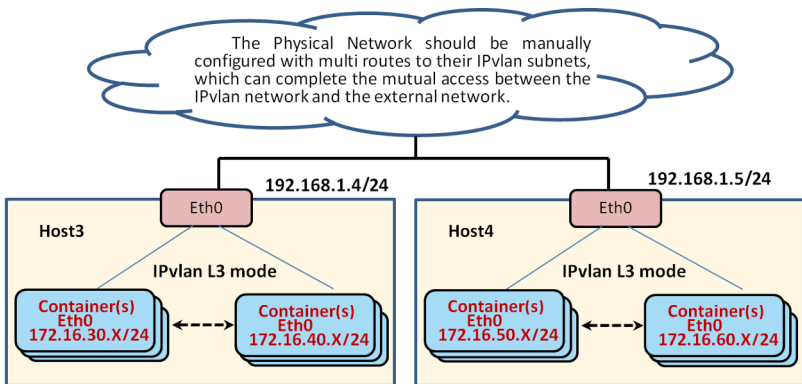


Fig. 2. Network architecture of IPvlan within multi hosts.

## 4 Experimental performance test and analysis

### 4.1 Experimental environment and test scheme

The experimental environment involves three host nodes. Node1 and Node2 install the Docker engine and create new container overlay network of Macvlan and IPvlan. Node3 creates two sub interfaces of the host NIC as the corresponding gateway of the Macvlan network respectively, and simultaneously enables routing and forwarding function. The node configuration information is shown in Table 1.

**Table 1.** Node configuration information.

Node properties	Configuration
CPU	Intel I7-11700
Memory	16G
Hard disk	2T
NIC	1Gbps
OS	Ubuntu 18.04.6
Docker	Docker-CE-20.10.7
Go	Go1.13.8
Test container image	alpine, clearlinux/iperf

The measurement of network performance includes availability, round-trip time (RTT), utilization, throughput and bandwidth. Ping command is an Internet packet explorer used to test network connectivity. Iperf is an open source tool developed by the University of Illinois to test network performance. Client and server modes are enabled at both ends of the test, TCP connection is tested bandwidth and transmitted data, UDP connection is tested jitter and packet loss rate [13-15]. For the two types of Macvlan and IPvlan container networks, our experimental test scheme is as follows: Ping command is used to test RTT, and Iperf tool is used to test bandwidth, throughput, jitter, packet loss rate and other network performance between end-to-end containers.

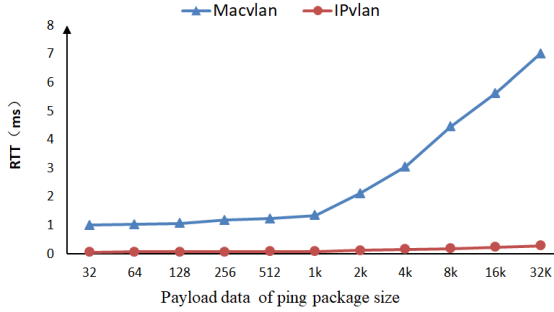
In our experimental test, the source container is set to send 100 ping request packets to the target container each time to get the RTT average, and the payload data size increases from 32 bytes to 32K bytes. The test parameters of Iperf are set as follows: the read/write buffer length option is the maximum allowed value, namely 63KB (UDP) and 1MB (TCP), Maximum Segment Size (MSS) of the transport Layer is set to 1460 bytes, and the numbers of concurrent threads is 1, 2, 4, 8, 16, and 32 in turn. One container runs the server side and the other container runs the client side. It outputs data communication statistics of the sending container and the receiving container within 10 seconds

### 4.2 Test results and analysis

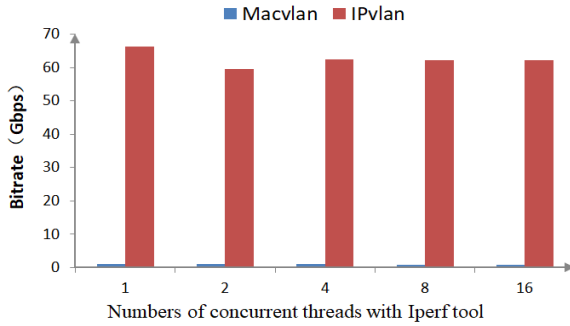
For RTT, bandwidth, UDP jitter and other test dimensions, our experiment gives a comparison of the container network performance of Macvlan and IPvlan across different network segment as shown in Figure3-5.

The experimental results show that the transmission performance of IPvlan container network is significantly better than that of Macvlan container network. Macvlan relies on two sub interfaces of the third host to complete the route forwarding between containers across different network segments, which will inevitably increase the communication delay between end-to-end containers. In the Layer 3 mode of IPvlan, a router is added to current host to forward the routing message of each endpoint. As long as the parent interface is the

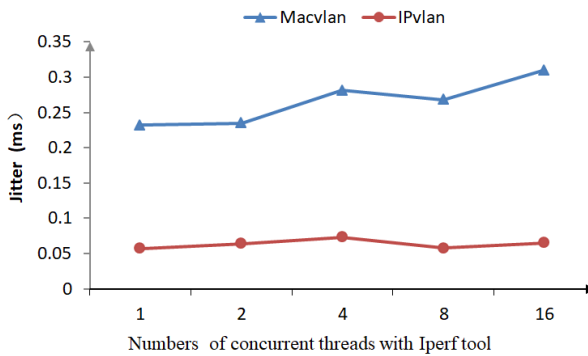
same, containers in different network segments can also complete the message forwarding within the local host.



**Fig. 3.** RTT test of container across different network segment.



**Fig. 4.** Bandwidth test of container across different network segment.



**Fig. 5.** UDP Jitter test of Container across different network segment.

IPvlan is a new twist on the tried and true network virtualization technique. The Linux implementations are extremely lightweight because rather than using the traditional Linux bridge for isolation, they are associated to a Linux Ethernet interface or sub-interface to enforce separation between networks and connectivity to the physical network.

For the following container application scenarios in the production environment, IPvlan should be used instead of Macvlan: (1) The uplink switch of the host NIC has set the binding between the current port and the designed Mac address; (2) The uplink switch of the host NIC has limited the numbers of Mac addresses associated with current port; (3)

Physical host NIC works on IEEE 802.11 wireless network protocol; (4) Constructing complex network topology, etc.

## 5 Conclusion

Macvlan is the main mode for multi host container overlay network. Without the help of traditional Linux virtual bridge devices, Macvlan can achieve nearly the lossless transmission performance of the local host. However, Macvlan is limited by port binding, the numbers of Mac addresses, and the lack of support for wireless networks. New mode IPvlan can better solve such problems instead of Macvlan, enabling administrator to fully control Layer 2 network, and even Layer3 network and has excellent network performance test across different network segment. Deeply exploring the special function of IPvlan and carrying out innovative application research on multi container network to better realize the system integration of user network, which will be our future research direction.

This work is supported by the Public Technology Research Project of Zhejiang under Grant No.LGF20F020005. and in part by Virtual Simulation Experiment Teaching Project of Zhejiang under Grant No.2019-250.

## Reference

1. Wu Song, Wang Kun, Jin Hai. Journal of Computer Research and Development 56,58-68 (2019)
2. Wang Baosheng, Zhang Weiqi. Journal of National University of Defense Technology 41,142-152 (2019)
3. SEL Laboratory of Zhejiang University. Docker container and container cloud (version 2) ,108-127 (2020)
4. Zhang Qing, Liu Jian, Zhu Xiaomin. Telecom Engineering Technics and Standardization 28, 74-77 (2015)
5. Zhang Weiqi. National University of Defense Technology Master Dissertation,31-38 (2018)
6. Yang Xin, Wu Zhinan, Qian Songrong. Microcomputer Applications 32, 58-60 (2016)
7. Li Wei, Zhao Yongbin, Wang Ou, Gang Yining. Machinery Design & Manufacture 5,270-272 (2017)
8. Feng Mingzhen. Zhejiang University Master Dissertation,48-55 (2016)
9. Zang Di, Yang Zhigang, Wang Jing, Yao Zhicheng, Zhang Weigong. Computer Engineering 48,214-219 (2022)
10. Nigel Poulton. Docker Deep Dive,157-168 (2019)
11. Use Macvlan networks[EB/OL].<https://docs.docker.com/network/macvlan> (2022)
12. Use Ipvlan networks[EB/OL].<https://docs.docker.com/network/ipvlan> (2022)
13. Jian Caili. Huazhong University of Science & Technology Master Dissertation,44-52 (2020)
14. Wang Peng. Research on Digital Media 39, 67-71 (2022)
15. Iperf ultimate speed test tool for TCP, UDP and SCTP [EB/OL].<https://iperf.fr> (2022)