

Proxmox Network Best Practices

Complete Reference Guide for High-Performance Home Lab Networking

🏗️ Network Topology Planning

Recommended Segments

Segment	VLAN	Subnet	Purpose
Management	10	192.168.10.0/24	Proxmox GUI, SSH, Cluster
Workloads	20	192.168.20.0/24	VM/Container Traffic
Storage	30	10.0.1.0/24	iSCSI, NFS, Ceph
DMZ	40	10.0.2.0/24	Public Services

Best Practice: Use separate physical NICs for storage traffic when possible to avoid congestion.

🐧 Linux Bridge Configuration

Basic Management Bridge

```
auto vmbr0 iface vmbr0 inet static address
192.168.10.11/24 gateway 192.168.10.1 bridge-ports
eth0 bridge-stp off bridge-fd 0 bridge-maxwait 0
```

Key Parameters

- bridge-stp off - Disables Spanning Tree (faster startup)
- bridge-fd 0 - Zero forwarding delay
- bridge-maxwait 0 - No waiting for ports

Warning: Only disable STP if you're certain about your topology design to avoid loops.

🔌 VLAN Configuration

VLAN Sub-Interface

```
auto eth0.100 iface eth0.100 inet manual vlan-raw-
device eth0
```

VLAN Bridge

```
auto vmbr100 iface vmbr100 inet static address
10.100.0.1/24 bridge-ports eth0.100 bridge-stp off
bridge-fd 0
```

Switch Port Configuration

Cisco: switchport mode trunk
HP: vlan trunk allowed 10,20,30
Unifi: Set port profile to "All" VLANs

🔗 Link Aggregation (LACP)

Bond Configuration

```
auto bond0 iface bond0 inet manual bond-slaves eth0
eth1 bond-miimon 100 bond-mode 802.3ad bond-lacp-
rate 1 bond-xmit-hash-policy layer2+3
```

Bridge over Bond

```
auto vmbr0 iface vmbr0 inet static address
192.168.10.11/24 gateway 192.168.10.1 bridge-ports
bond0 bridge-stp off bridge-fd 0
```

Bond Modes

- 802.3ad - LACP (requires switch support)
- active-backup - Failover only
- balance-rr - Round-robin

Check Status: cat /proc/net/bonding/bond0

MTU and Jumbo Frames

Enable Jumbo Frames

```
auto bond0 iface bond0 inet manual mtu 9000 bond-  
slaves eth0 eth1 bond-mode 802.3ad bond-miimon 100
```

```
auto vmbr-storage iface vmbr-storage inet static  
address 10.0.1.11/24 bridge-ports bond0 mtu 9000  
bridge-stp off bridge-fd 0
```

Test Jumbo Frames

```
# Test 9000 byte MTU ping -M do -s 8972 10.0.1.1 #  
Test standard MTU ping -M do -s 1472 10.0.1.1
```

Important: All network devices in the path must support the same MTU size.

Security Configuration

SSH Hardening

```
# /etc/ssh/sshd_config PermitRootLogin no  
PasswordAuthentication no Port 2222 MaxAuthTries 3  
ClientAliveInterval 300 ClientAliveCountMax 2
```

Proxmox Firewall Rules

```
# Datacenter level INPUT rules IN ACCEPT -i lo IN  
ACCEPT -p tcp --dport 8006 -s 192.168.10.0/24 IN  
ACCEPT -p tcp --dport 22 -s 192.168.10.0/24 IN  
ACCEPT -p icmp -s 192.168.10.0/24 IN DROP
```

Management Network Isolation

- Separate VLAN for management traffic
- Firewall rules blocking inter-VLAN routing
- VPN access for remote management

Monitoring Commands

Interface Status

```
ip addr show - Interface states and IPs  
bridge link show - Bridge port status  
ethtool eth0 - NIC details and speed  
cat /proc/net/dev - Interface statistics
```

Traffic Analysis

```
iftop -i vmbr0 - Real-time bandwidth  
nload eth0 - Network load visualization  
tcpdump -i vmbr0 port 80 - Packet capture  
ss -tuln - Socket statistics
```

Performance Metrics

```
cat /proc/interrupts | grep eth - IRQ distribution  
ethtool -S eth0 - NIC statistics  
netstat -i - Interface packet counts
```

Troubleshooting Guide

Common Issues

- **VM can't reach network:** Check bridge ports and VLAN tags
- **Slow performance:** Verify MTU settings across path
- **Bond not working:** Check switch LACP configuration
- **Cluster issues:** Verify management network connectivity

Debug Commands

```
# Check bridge status brctl show # Verify VLAN  
configuration cat /proc/net/vlan/config # Test  
connectivity ping -c 4 -I vmbr0 gateway_ip # Check  
for packet drops netstat -i | grep -E "  
(RX|TX).*drop"
```

SR-IOV Configuration

Enable SR-IOV

```
# Enable in BIOS/UEFI first echo 4 >
/sys/class/net/eth0/device/sriov_numvfs #
Make persistent echo 'echo 4 >
/sys/class/net/eth0/device/sriov_numvfs' >
/etc/rc.local
```

VM Configuration

```
# Add to VM config net0:
bridge=vmbr0,firewall=1,model=virtio,queues=4
# Or use SR-IOV VF directly hostpci0: 01:10.0
```

Performance Tip: SR-IOV provides near-native network performance by bypassing the hypervisor.

Network Namespaces

Create Isolated Network

```
# Create namespace ip netns add isolated # Create veth pair
ip link add veth0 type veth peer name veth1 # Move to
namespace ip link set veth1 netns isolated # Configure in
namespace ip netns exec isolated ip addr add 10.0.0.1/24
dev veth1 ip netns exec isolated ip link set veth1 up
```

Bridge Integration

```
# Add to bridge ip link set veth0 master vmbr0 ip link set
veth0 up
```

QoS and Traffic Shaping

TC Traffic Control

```
# Create root qdisc tc qdisc add dev eth0
root handle 1: htb default 30 # Create
classes tc class add dev eth0 parent 1:
classid 1:1 htb rate 1gbit tc class add dev
eth0 parent 1:1 classid 1:10 htb rate 500mbit
ceil 1gbit tc class add dev eth0 parent 1:1
classid 1:20 htb rate 300mbit ceil 600mbit tc
class add dev eth0 parent 1:1 classid 1:30
htb rate 200mbit ceil 400mbit
```

Bandwidth Limiting

```
# Limit VM bandwidth tc qdisc add dev
tap100i0 root handle 1: tbf rate 100mbit
burst 10mb latency 50ms
```

Network Redundancy

Multi-Path Configuration

```
# Enable IP forwarding echo 1 >
/proc/sys/net/ipv4/ip_forward # Configure routing tables
echo "200 storage" >> /etc/iproute2/routing echo "201
backup" >> /etc/iproute2/routing # Add routes ip route
add 10.0.1.0/24 dev bond0-storage table storage ip route
add 10.0.2.0/24 dev bond1-backup table backup
```

Failover Rules

```
# Priority-based routing ip rule add from 10.0.1.0/24 table
storage priority 100 ip rule add from 10.0.2.0/24 table
backup priority 200
```



DPDK Integration

DPDK Setup

```
# Install DPDK apt install dpdk dpdk-dev #
Bind interface to DPDK dpdk-devbind --
bind=vfio-pci 0000:01:00.0 # Configure
hugepages echo 1024 >
/sys/kernel/mm/hugepages/hugepages-
2048kB/nr_hugepages
```

VM Configuration

```
# VM args for DPDK args: -object memory-
backend-file,id=mem,size=4G,mem-
path=/dev/hugepages,share=on -numa
node,memdev=mem -mem-prealloc
```



Advanced Monitoring

SNMP Configuration

```
# Install SNMP apt install snmpd snmp-mibs-downloader #
Configure /etc/snmp/snmpd.conf rocommunity public localhost
syslocation "Home Lab" syscontact "admin@lab.local"
```

Prometheus Integration

```
# Node Exporter wget
https://github.com/prometheus/node_exporter/releases/latest
./node_exporter --web.listen-address=":9100"
```

```
# Custom metrics echo
"network_bridge_ports{bridge=\"vmbr0\"} 1" >
/var/lib/node_exporter/bridge.prom
```



Performance Tuning

Network Buffer Tuning

```
# /etc/sysctl.conf net.core.rmem_default = 262144
net.core.rmem_max = 16777216 net.core.wmem_default = 262144
net.core.wmem_max = 16777216 net.core.netdev_max_backlog =
5000 net.ipv4.tcp_rmem = 4096 65536 16777216 net.ipv4.tcp_wmem
= 4096 65536 16777216
```

VM Network Optimization

```
# VM config optimizations net0:
virtio=AA:BB:CC:DD:EE:FF,bridge=vmbr0,firewall=1,queues=4 #
Enable virtio-net multiqueue virtio-net-
pci,netdev=net0,id=net0,mac=AA:BB:CC:DD:EE:FF,mq=on,vectors=10
```

Interrupt Optimization

```
# CPU affinity for NICs echo 2 >
/proc/irq/24/smp_affinity echo 4 >
/proc/irq/25/smp_affinity # Enable receive
packet steering echo 1 >
/sys/class/net/eth0/queues/rx-0/rps_cpus
```

Storage Network Tuning

```
# iSCSI optimizations echo deadline >
/sys/block/sda/queue/scheduler echo 32 >
/sys/block/sda/queue/nr_requests echo 2 >
/sys/block/sda/queue/rq_affinity
```

Quick Reference Commands

Network Status

```
pvesh get /nodes/node1/network - Proxmox network info
pvesh get /cluster/resources - Cluster resources
ip route show table all - All routing tables
bridge vlan show - VLAN information
```

Performance Testing

```
iperf3 -s - Network throughput server
iperf3 -c target_ip - Throughput test
netperf -H target_ip - Network performance
hping3 -S -p 80 target_ip - Latency test
```

Cluster Operations

```
pvecm nodes - List cluster nodes
pvecm status - Cluster status
corosync-quorumtool -s - Quorum status
journalctl -u corosync - Corosync logs
```

Emergency Recovery

```
systemctl restart networking - Restart network
ifdown vmbro && ifup vmbro - Restart bridge
echo 0 > /proc/sys/net/ipv4/ip_forward - Disable forwarding
iptables -F - Flush firewall rules
```

 **Pro Tips:** Always test network changes in a staging environment first • Keep network diagrams updated • Use version control for network configurations • Monitor bandwidth usage trends • Document firewall rules and their purposes • Regular backup of network configurations