**Network**

**变更记录**

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 修改者 | 日期 | 描述 |
| v1.3 | 翁喆 | 2023年9月27日 | 不使用root权限启动bridge |
| v1.2 | 翁喆 | 2023年8月29日 | 第二章增加表格引导，内容统一为Goldfish |
| V1.1 | 翁喆 | 2023年8月15日 | 重新组织文档结构，精简内容，增加变更记录，补充原理 |
| V1.0 | 翁喆 | 2023年7月31日 | 完成三种联网方式的完整说明 |

**一、基础配置**

|  |
| --- |
| Makefile # Vela配置默认已经配好  CONFIG\_OPENAMP=y CONFIG\_DRIVERS\_VIRTIO=y CONFIG\_DRIVERS\_VIRTIO\_MMIO=y CONFIG\_DRIVERS\_VIRTIO\_NET=y |

**二、操作使用**

|  |  |  |
| --- | --- | --- |
| 实现方案 | 适用场景 | 注意事项 |
| 2.1 SLiRP user后端 | * **通用方案，建议没有特殊情况选择此方案** * Vela配置默认已经配好，可以直接从2.1.b开始 * 兼容macOS / Windows，命令简单 | * 两个Vela实例无法直接通过网络互相访问 * 可以做端口映射访问少量端口 * 不支持ping Vela、组播、广播 |
| 2.2 Bridge模式+宿主路由(NAT) | * 方便Host访问Vela，以及Vela实例之间互相通信 * 互联互通场景可能需要 | * Vela访问外网效果不如SLiRP，仅Linux * 没有Host提供的DNS * 需要确保各个Vela之间IP不冲突 |
| 2.3 Bridge模式+桥接到宿主网卡 | * **必须**要桥接到外部网络/**直接与外部网络通信**的场景 * 一般情况不建议使用此方案 | * 遇到过各种坑，参考[网络](https://xiaomi.f.mioffice.cn/wiki/wikk43TYrMcZm3qhW1kSh2ZRTpe#t9pql6) 2.3 已知问题 * 仅Linux Host支持 |

**2.1 通过SLiRP user后端进行网络通信（首选，默认）**

因为无需特殊配置，所以**不需要特殊应用场景的同学，建议用这种方式获得网络能力**。

1. Vela配置（**Vela配置默认已经配好**）

* Application Configuration -> Network Utilities -> Network initialization -> IP Address Configuration
* Target IPv4 address设置为0x0a00020f
* Router IPv4 address设置为0x0a000202
* 启用Application Configuration -> Network Utilities -> Network initialization -> Use DNS
* 并将DNS IP address设置为0x0a000203

|  |
| --- |
| Makefile CONFIG\_NETINIT\_DNS=y CONFIG\_NETINIT\_DNSIPADDR=0x0a000203 CONFIG\_NETINIT\_DRIPADDR=0x0a000202 CONFIG\_NETINIT\_IPADDR=0x0a00020f  # 一些优化项 ## Vela的第一个DNS有可能被ARP替代，因为没有延迟/丢包的可能，所以timeout可设为最小 CONFIG\_NETDB\_DNSCLIENT\_RECV\_TIMEOUT=1 |

1. 编译Vela

* 编译自己需要的版本即可./build.sh vendor/qemu/boards/vela/configs/**xxx** -j20

1. 启动Goldfish

* ./emulator.sh vela
* **此时Vela可访问外部网络，可以直接ping baidu.com**

1. 如果需要开启server port（对外提供服务），例如将Vela的23端口映射到host的10023端口：

* ./emulator.sh vela -network-user-mode-options hostfwd=tcp:127.0.0.1:10023-10.0.2.15:23,hostfwd=tcp:127.0.0.1:xxx-10.0.2.15:xxx

1. 如果需要多个Vela互联，可以参考[互连模拟器实例](https://developer.android.com/studio/run/emulator-networking#connecting)的思路，将所需端口映射到宿主机上即可

**2.2 通过Bridge模式+宿主路由(NAT)进行网络通信**

注：相比于SLiRP，此方法方便Host访问Guest，以及Guest之间互相通信，但Guest访问外网效果不如SLiRP，**仅有Guest向外访问需求的用户建议使用2.1 SLiRP的方法**。

1. Vela配置（与[网络](https://xiaomi.f.mioffice.cn/wiki/wikk43TYrMcZm3qhW1kSh2ZRTpe#xiGOkU) 2.2基本一致，与步骤e二选一即可）

* Application Configuration -> Network Utilities -> Network initialization -> IP Address Configuration
* Target IPv4 address设置为0x0a000102
* Router IPv4 address设置为0x0a000101
* 启用Application Configuration -> Network Utilities -> Network initialization -> Use DNS
* 并将DNS IP address设置为有效的DNS地址（比如0xdf050505）

|  |
| --- |
| Makefile CONFIG\_NETINIT\_IPADDR=0x0a000102 CONFIG\_NETINIT\_DRIPADDR=0x0a000101 CONFIG\_NETINIT\_DNS=y CONFIG\_NETINIT\_DNSIPADDR=0xdf050505  # 一些可选项 CONFIG\_NETINIT\_DHCPC=n CONFIG\_NET\_ICMPv6\_AUTOCONF=n |

1. 编译Vela

* 编译自己需要的版本即可./build.sh vendor/qemu/boards/vela/configs/**xxx** -j20

1. 执行脚本sudo tools/simhostroute.sh <interface> on

* <interface>为宿主机用于上网的网卡名称
* 同时运行多个模拟器也只需要on一次，然后参考2.2.e.i给每个Vela配置不同IP即可

1. 启动Goldfish

* 第一次使用时
* 需要在宿主机上允许bridge到nuttx0

|  |
| --- |
| Bash sudo mkdir -p /etc/qemu/ sudo sh -c "echo allow nuttx0 > /etc/qemu/bridge.conf" |

* 允许bridge脚本创建tap设备

|  |
| --- |
| Bash sudo chmod 0666 /dev/net/tun sudo setcap cap\_net\_admin,cap\_net\_raw+ep /usr/lib/qemu/qemu-bridge-helper |

* **运行**./emulator.sh vela -qemu -netdev bridge,**br=nuttx0,id=b1,helper=/usr/lib/qemu/qemu-bridge-helper** -device virtio-net-device,netdev=b1,bus=virtio-mmio-bus.3

1. 如果步骤a中已经配置好IP，此时Vela上默认配置的IP已经可以访问外网
2. 可以ping baidu.com进行测试
3. 若未配置IP或者想要改变地址（多个模拟器）则需要在Vela上执行（以.2为例，可用其他数字）
4. ifconfig eth0 10.0.1.2 dns 223.5.5.5
5. ifconfig eth0 inet6 fc00::2/112 dns 2001:da8::666
6. 若两个Vela均可以访问外网但无法互连，可以试试

* sudo iptables -P FORWARD ACCEPT

1. 完成调试后，关闭Vela，再执行sudo tools/simhostroute.sh <interface> off

**2.3 通过Bridge模式+桥接到宿主网卡进行网络通信（备选方案，容易踩坑）**

**注：仅建议必须要桥接到外部网络/直接与外部网络通信的场景使用，遇到过各种坑，参考**[**网络**](https://xiaomi.f.mioffice.cn/wiki/wikk43TYrMcZm3qhW1kSh2ZRTpe#t9pql6) **2.3 已知问题**

1. Vela配置（与[网络](https://xiaomi.f.mioffice.cn/wiki/wikk43TYrMcZm3qhW1kSh2ZRTpe#MS4sAi) 2.3基本一致）

* Application Configuration -> Network Utilities -> Network initialization -> IP Address Configuration -> Use DHCP to get IP address设置为开
* Application Configuration -> Network Utilities -> Network initialization -> Use DNS设置为关

|  |
| --- |
| Makefile CONFIG\_NETINIT\_DHCPC=y CONFIG\_NETUTILS\_DHCPC=y  CONFIG\_NETINIT\_DNS=n |

1. 编译Vela

* 编译自己需要的版本即可./build.sh vendor/qemu/boards/vela/configs/**xxx** -j20

1. 执行脚本sudo tools/simbridge.sh <interface> on

* interface为宿主机用于上网的网卡名称

1. 启动Goldfish

* 第一次使用时
* 需要在宿主机上允许bridge到nuttx0

|  |
| --- |
| Bash sudo mkdir -p /etc/qemu/ sudo sh -c "echo allow nuttx0 > /etc/qemu/bridge.conf" |

* 允许bridge脚本创建tap设备

|  |
| --- |
| Bash sudo chmod 0666 /dev/net/tun sudo setcap cap\_net\_admin,cap\_net\_raw+ep /usr/lib/qemu/qemu-bridge-helper |

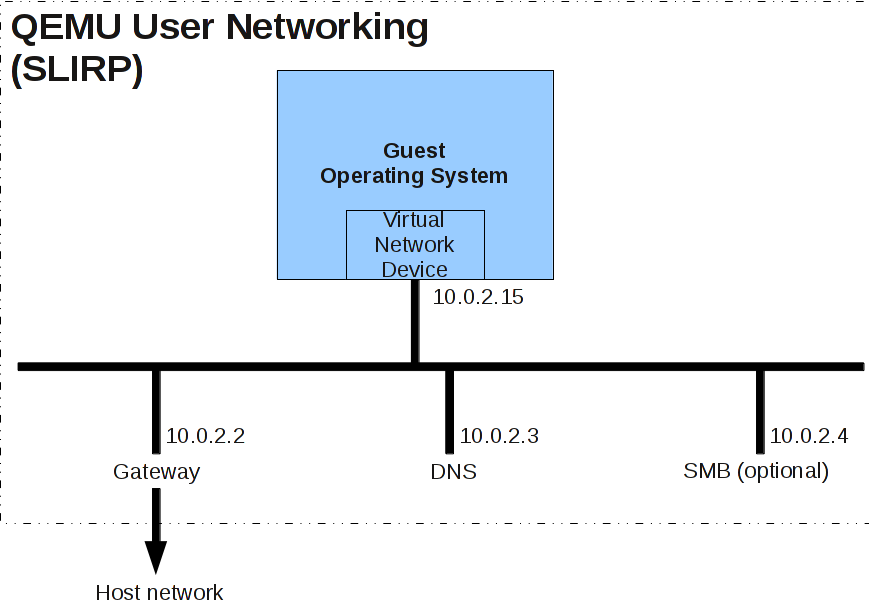
* **运行**./emulator.sh vela -qemu -netdev bridge,**br=nuttx0,id=b1,helper=/usr/lib/qemu/qemu-bridge-helper** -device virtio-net-device,netdev=b1,bus=virtio-mmio-bus.3

1. 此时Vela上会通过DHCP取得IP等参数并自动配置，此时已经可以访问外网
2. 可以ping baidu.com进行测试
3. 完成调试后，关闭Vela，再执行sudo tools/simbridge.sh <interface> off

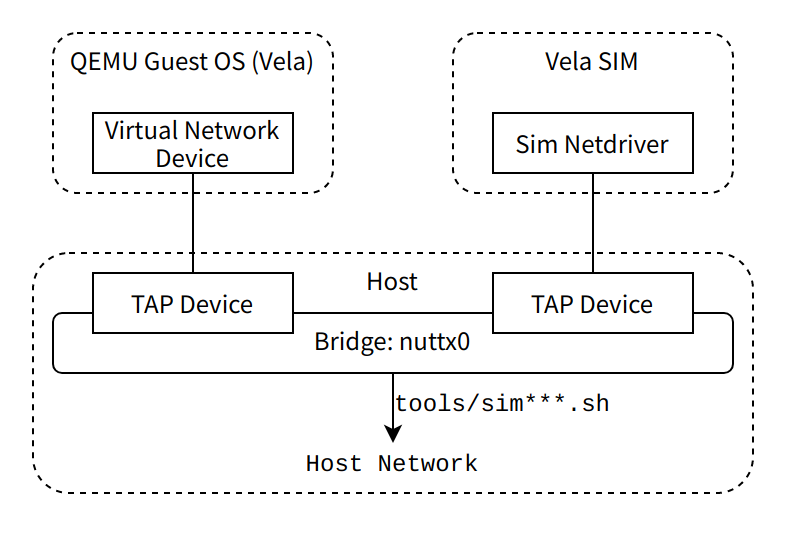
**三、实现原理**

**3.1 通过SLiRP user后端进行网络通信**

每个SLiRP user后端都默认生成以下私有网络，因此将对应的IP、网关、DNS都配置上即可。



**3.2 通过Bridge模式（TAP后端）进行网络通信**



**四、附录**

**4.1 如何抓包**

* 使用QEMU自带的dump功能
* https://wiki.qemu.org/Documentation/Networking#Network\_Monitoring
* -qemu -object filter-dump,id=f1,netdev=mynet,file=dump.dat
* netdev=mynet此id与命令中-netdev指定的id保持一致
* file=dump.dat可自己定义其他文件名
* 使用Vela内置的tcpdump
* [tcpdump](https://xiaomi.f.mioffice.cn/wiki/wikk4FiXpv8ujakn8ozuUc21Otr)
* Guest和Host的文件传输：[File](https://xiaomi.f.mioffice.cn/wiki/wikk49BdBjx5HGPyLkjD476A9dd)

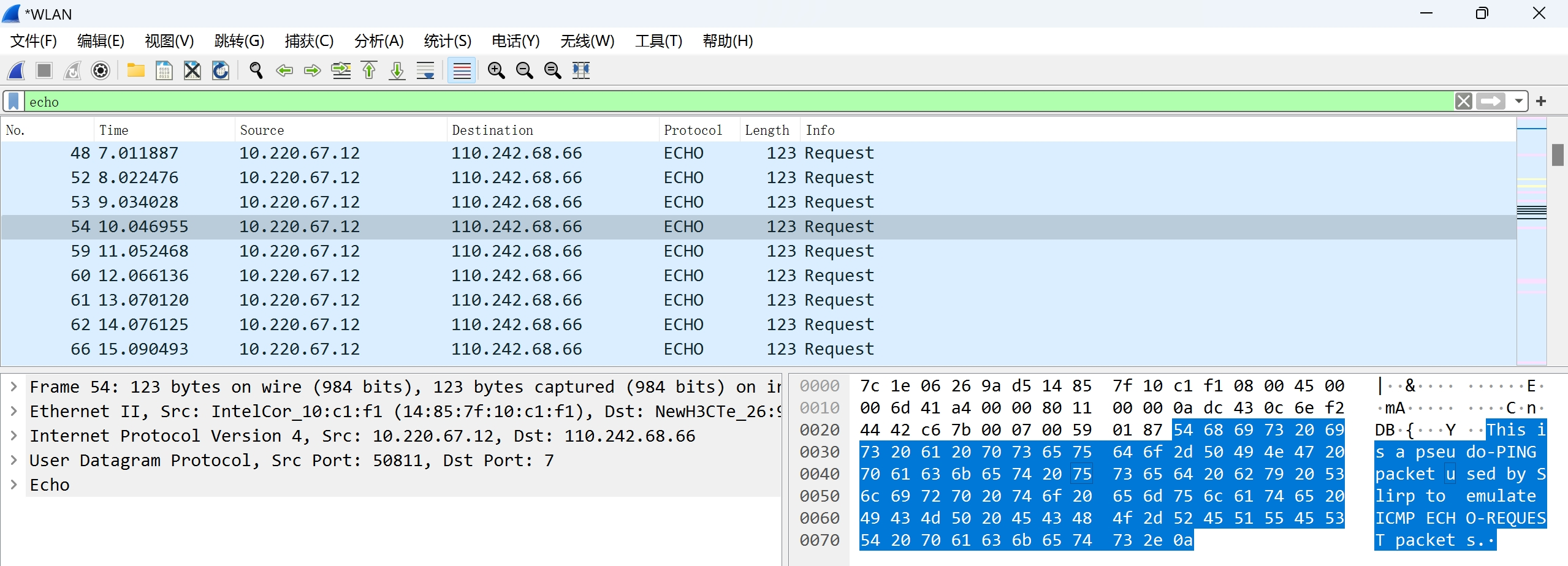
**4.2 已知问题**

**4.2.1 Goldfish Bridge**

* Bridge模式所需的工具qemu-bridge-helper功能异常（可能是编译时没有包含此工具）
* 当前规避办法是借用系统中QEMU的helper，即helper=/usr/lib/qemu/qemu-bridge-helper（文中均使用此方法）

**4.2.2 Windows上ping不通baidu.com**

* 普通服务器无法ping通，需要启动Echo Server
* Echo Server启动方式：sudo socat -v PIPE udp-recvfrom:7,fork
* 启动了Echo Server的服务器才能被Windows上的Goldfish ping通
* 发现所有发出去的ping都被goldfish换成了UDP的ECHO protocol，只有在UDP port 7启动了echo server的服务器才能响应ping
* "This is a pseudo-PING packet used by Slirp to emulate ICMP ECHO-REQUEST packets."



* https://gerrit.pt.mioffice.cn/plugins/gitiles/vela-qemu/platform/external/qemu/+/refs/heads/vela-emu-32-dev/slirp/ip\_icmp.c#39

**五、参考资料**

* [VirtIO Net](https://xiaomi.f.mioffice.cn/wiki/wikk46dCn1Cn01oukM8wfPkJkId)
* [NuttX Qemu virtio-net使用说明](https://xiaomi.f.mioffice.cn/wiki/wikk4cQ7Mck6gKkBRuHU30gkfCg)
* https://wiki.qemu.org/Documentation/Networking
* https://developer.android.com/studio/run/emulator-networking
* https://man.archlinux.org/man/slirp4netns.1.en
* https://wiki.qemu.org/Features/HelperNetworking