1. zlm和ffmpeg建立连接

方案1：添加Starry的IPV6实现

ZLMediaKit 默认使用 IPv6：`socket(AF_INET6, SOCK_STREAM, IPPROTO_TCP) = 64`。在内核里参照目前的网络栈加个 ipv6 ，实在不行再去改 ZLMediaKit 配置看能否改 ipv4。

```
getsockname(23, {sa_family=AF_INET6, sin6_port=htons(9000), sin6_flowinfo=htonl(0), inet_pton(AF_INET6, "::", &sin6_addr), sin6_scope_id=0}, [128 => 28]) = 0
getpeername(23, 0x562abdb45000, [128])  = -1 ENOTCONN (传输端点尚未连接)
socket(AF_INET6, SOCK_DGRAM, IPPROTO_UDP) = 24
setsockopt(24, SOL_SOCKET, SO_REUSEADDR, [1], 4) = 0
setsockopt(24, SOL_SOCKET, SO_REUSEPORT, [1], 4) = 0
ioctl(24, FIONBIO, [1])                 = 0
setsockopt(24, SOL_SOCKET, SO_SNDBUF, [262144], 4) = 0
setsockopt(24, SOL_SOCKET, SO_RCVBUF, [262144], 4) = 0
setsockopt(24, SOL_SOCKET, SO_LINGER, {l_onoff=0, l_linger=0}, 8) = 0
fcntl(24, F_GETFD)                      = 0
fcntl(24, F_SETFD, FD_CLOEXEC)          = 0
setsockopt(24, SOL_IPV6, IPV6_V6ONLY, [0], 4) = 0
bind(24, {sa_family=AF_INET6, sin6_port=htons(9000), sin6_flowinfo=htonl(0), inet_pton(AF_INET6, "::", &sin6_addr), sin6_scope_id=0}, 28) = 0
prctl(PR_GET_NAME, "MediaServer")       = 0
prctl(PR_GET_NAME, "MediaServer")       = 0
write(14, "\0", 1)                      = 1
getsockname(24, {sa_family=AF_INET6, sin6_port=htons(9000), sin6_flowinfo=htonl(0), inet_pton(AF_INET6, "::", &sin6_addr), sin6_scope_id=0}, [128 => 28]) = 0
getpeername(24, 0x562abdb45eb0, [128])  = -1 ENOTCONN (传输端点尚未连接)
socket(AF_INET6, SOCK_DGRAM, IPPROTO_UDP) = 30
setsockopt(30, SOL_SOCKET, SO_REUSEADDR, [1], 4) = 0
setsockopt(30, SOL_SOCKET, SO_REUSEPORT, [1], 4) = 0
```

```
[ 22.747729 0:12 linux_syscall_api::syscall:53] [syscall] id = 157,return 0
[ 22.747835 0:12 linux_syscall_api::syscall:40] [syscall] id = GETTIMEOFDAY, args = [32093856, 0, 0, 0, 255, 128], entry
[ 22.747964 0:12 linux_syscall_api::syscall:40] [syscall] id = PRCTL, args = [16, 32094944, 32, 32094944, 32, 128], entry
[ 22.748099 0:12 linux_syscall_api::syscall:53] [syscall] id = 157,return 0
[ 22.748198 0:12 linux_syscall_api::syscall:40] [syscall] id = GETTIMEOFDAY, args = [32093856, 0, 0, 0, 255, 128], entry
[ 22.748353 0:12 linux_syscall_api::syscall:40] [syscall] id = PRCTL, args = [16, 32094944, 32, 32094944, 32, 128], entry
[ 22.748485 0:12 linux_syscall_api::syscall:53] [syscall] id = 157,return 0
[ 22.748723 0:12 linux_syscall_api::syscall:40] [syscall] id = GETTIMEOFDAY, args = [32093856, 0, 0, 70574902607872, 137438953471, 0], entry
[ 22.748875 0:12 linux_syscall_api::syscall:40] [syscall] id = PRCTL, args = [16, 32096944, 32, 32096944, 32, 32096944], entry
[ 22.749032 0:12 linux_syscall_api::syscall:53] [syscall] id = 157,return 0
[ 22.749160 0:12 linux_syscall_api::syscall:31] [syscall] id = WRITE, args = [6, 16699045, 1, 32096944, 0, 32097152], entry
[ 22.749292 0:12 linux_syscall_api::syscall_fs::imp::io:104] [write()] fd: 6, buf: 0xfecea5, len: 1
[ 22.749405 0:12 linux_syscall_api::syscall_fs::ctype::pipe:173] kernel: Pipe::write
[ 22.749499 0:12 linux_syscall_api::syscall:53] [syscall] id = 1,return 1
[ 22.749620 0:12 linux_syscall_api::syscall:12] [syscall] id = SOCKET, args = [10, 2, 17, 32096944, 0, 0], entry
[ 22.749787 0:12 linux_syscall_api::syscall:53] [syscall] id = 41,return 19
[ 22.749878 0:12 linux_syscall_api::syscall:12] [syscall] id = SETSOCKOPT, args = [19, 1, 2, 1073739336, 4, 0], entry
[ 22.750034 0:12 linux_syscall_api::syscall:53] [syscall] id = 54,return 0
[ 22.750151 0:12 linux_syscall_api::syscall:12] [syscall] id = SETSOCKOPT, args = [19, 1, 15, 1073739336, 4, 0], entry
[ 22.750282 0:12 linux_syscall_api::syscall_net::imp:480] [setsockopt()] option 15 not supported in Socket level
[ 22.750403 0:12 linux_syscall_api::syscall:53] [syscall] id = 54,return 0
[ 22.750515 0:12 linux_syscall_api::syscall:31] [syscall] id = IOCTL, args = [19, 21537, 1073739336, 1073739336, 4, 0], entry
[ 22.750648 0:12 linux_syscall_api::syscall_fs::imp::ctl:441] fd: 19, request: 21537, argp: 1073739336
[ 22.750761 0:12 linux_syscall_api::syscall:53] [syscall] id = 16,return 0
[ 22.750854 0:12 linux_syscall_api::syscall:12] [syscall] id = SETSOCKOPT, args = [19, 1, 7, 1073739320, 4, 0], entry
[ 22.750984 0:12 linux_syscall_api::syscall:53] [syscall] id = 54,return 0
[ 22.751072 0:12 linux_syscall_api::syscall:12] [syscall] id = SETSOCKOPT, args = [19, 1, 8, 1073739320, 4, 0], entry
[ 22.751213 0:12 linux_syscall_api::syscall:53] [syscall] id = 54,return 0
[ 22.751309 0:12 linux_syscall_api::syscall:12] [syscall] id = SETSOCKOPT, args = [19, 1, 13, 1073739336, 8, 0], entry
[ 22.751437 0:12 linux_syscall_api::syscall_net::imp:480] [setsockopt()] option 13 not supported in Socket level
[ 22.751557 0:12 linux_syscall_api::syscall:53] [syscall] id = 54,return 0
[ 22.751642 0:12 linux_syscall_api::syscall:31] [syscall] id = FCNTL64, args = [19, 1, 0, 1073739336, 8, 0], entry
[ 22.751770 0:12 linux_syscall_api::syscall_fs::imp::ctl:369] fd: 19, cmd: 1
[ 22.751858 0:12 linux_syscall_api::syscall:53] [syscall] id = 72,return 0
[ 22.751967 0:12 linux_syscall_api::syscall:31] [syscall] id = FCNTL64, args = [19, 2, 1, 1073739336, 8, 0], entry
[ 22.752093 0:12 linux_syscall_api::syscall_fs::imp::ctl:369] fd: 19, cmd: 2
[ 22.752183 0:12 linux_syscall_api::syscall:53] [syscall] id = 72,return 0
[ 22.752269 0:12 linux_syscall_api::syscall:12] [syscall] id = SETSOCKOPT, args = [19, 41, 26, 1073739144, 4, 0], entry
[ 22.752398 0:12 linux_syscall_api::syscall_net::imp:496] [setsockopt()] option 26 not supported in ipv6 level
[ 22.752520 0:12 linux_syscall_api::syscall:53] [syscall] id = 54,return 0
[ 22.752608 0:12 linux_syscall_api::syscall:12] [syscall] id = BIND, args = [19, 1073739280, 28, 0, 1073739136, 1073739794], entry
[ 22.752772 0:12 linux_syscall_api::syscall_net::imp:72] [bind()] binding Socket 19 to Endpoint { addr: Ipv6(Address([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])), port: 9000 }
[ 22.752961 0:12 linux_syscall_api::syscall:53] [syscall] id = 49,return 0
```

对比strace和starry的info信息，流程走向有所区别

```
getrusage(RUSAGE_SELF, {ru_utime={tv_sec=0, tv_usec=68948}, ru_stime={tv_sec=0, tv_usec=30164}, ...}) = 0
openat(AT_FDCWD, "/sys/devices/system/cpu/online", O_RDONLY|O_CLOEXEC) = 4
read(4, "0-3\n", 1024)                  = 4
close(4)                                = 0
rt_sigaction(SIGRT_1, {sa_handler=0x7f866fc00870, sa_mask=[], sa_flags=SA_RESTORER|SA_ONSTACK|SA_RESTART|SA_SIGINFO, sa_restorer=0x7f866fbb1520}, N
rt_sigprocmask(SIG_UNBLOCK, [RTMIN RT_1], NULL, 8) = 0
mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) = 0x7f8665659000
mprotect(0x7f866565a000, 8388608, PROT_READ|PROT_WRITE) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, child_t
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) = 0x7f8664e58000
mprotect(0x7f8664e59000, 8388608, PROT_READ|PROT_WRITE) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, child_t
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) = 0x7f8664657000
mprotect(0x7f8664658000, 8388608, PROT_READ|PROT_WRITE) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, child_t
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) = 0x7f8663e56000
mprotect(0x7f8663e57000, 8388608, PROT_READ|PROT_WRITE) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, child_t
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) = 0x7f8663655000
mprotect(0x7f8663656000, 8388608, PROT_READ|PROT_WRITE) = 0
rt_sigprocmask(SIG_BLOCK, ~[], [], 8)   = 0
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, child_t
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
write(2, "Stream mapping:\n", 16Stream mapping:
)       = 16
write(2, "  Stream #0:0 → #0:0", 21  Stream #0:0 → #0:0)   = 21
write(2, " (h264 (native) → h264 (libx264"..., 34 (h264 (native) → h264 (libx264))) = 34
write(2, "\n", 1
)                       = 1
write(2, "Press [q] to stop, [?] for help\n", 32Press [q] to stop, [?] for help
) = 32
pselect6(1, [0], NULL, NULL, {tv_sec=0, tv_nsec=0}, NULL) = 0 (Timeout)
futex(0x55b09a5938c0, FUTEX_WAKE_PRIVATE, 1) = 1
futex(0x55b09a593928, FUTEX_WAKE_PRIVATE, 1) = 1
futex(0x55b09a5938f0, FUTEX_WAIT_BITSET_PRIVATE|FUTEX_CLOCK_REALTIME, 0, NULL, FUTEX_BITSET_MATCH_ANY) = 0
futex(0x55b09a593950, FUTEX_WAKE_PRIVATE, 1) = 0
brk(0x55b09a68b000)                     = 0x55b09a68b000
futex(0x55b09a593a70, FUTEX_WAKE_PRIVATE, 1) = 1
futex(0x55b09a593aa0, FUTEX_WAIT_BITSET_PRIVATE|FUTEX_CLOCK_REALTIME, 0, NULL, FUTEX_BITSET_MATCH_ANY) = 0
futex(0x55b09a593b00, FUTEX_WAKE_PRIVATE, 1) = 0
brk(0x55b09a6ac000)                     = 0x55b09a6ac000
futex(0x55b09a593c20, FUTEX_WAKE_PRIVATE, 1) = 1
futex(0x55b09a593c88, FUTEX_WAKE_PRIVATE, 1) = 1
brk(0x55b09a6d2000)                     = 0x55b09a6d2000
```

```
[ 26.640722 0:8 linux_syscall_api::syscall:53] [syscall] id = 228,return 0
[ 26.640845 0:8 linux_syscall_api::syscall:40] [syscall] id = GETRUSAGE, args = [0, 1073740416, 640695, 218181056, 253574, 218035016], entry
[ 26.641386 0:8 linux_syscall_api::syscall:40] [syscall] id = CLOCK_GET_TIME, args = [1, 1073738800, 218171808, 218036552, 1, 0], entry
[ 26.641543 0:8 linux_syscall_api::syscall:53] [syscall] id = 228,return 0
[ 26.642145 0:8 linux_syscall_api::syscall:31] [syscall] id = OPENAT, args = [4294967196, 34799040, 524288, 0, 524288, 34799040], entry
[ 26.642336 0:8 linux_syscall_api::syscall_fs::imp::io:400] path: "/sys/devices/system/cpu/online"
[ 26.642468 0:8 axprocess::link:243] create_link: /sys/devices/system/cpu/online → /sys/devices/system/cpu/online
[ 26.642604 0:8 linux_syscall_api::syscall:53] [syscall] id = 257,return 5
[ 26.642696 0:8 linux_syscall_api::syscall:31] [syscall] id = READ, args = [5, 1073736848, 1024, 0, 524288, 34799040], entry
[ 26.642830 0:8 linux_syscall_api::syscall_fs::imp::io:32] [read()] fd: 5, buf: 0x3fffec90, len: 1024
[ 26.643031 0:8 linux_syscall_api::syscall:53] [syscall] id = 0,return 3
[ 26.643122 0:8 linux_syscall_api::syscall:31] [syscall] id = CLOSE, args = [5, 10, 1073736851, 34663104, 0, 0], entry
[ 26.643252 0:8 linux_syscall_api::syscall_fs::imp::io:461] into syscall_close. fd: 5
[ 26.643357 0:8 linux_syscall_api::syscall:53] [syscall] id = 3,return 0
[ 26.644250 0:8 linux_syscall_api::syscall:31] [syscall] id = WRITE, args = [2, 1073736516, 16, 0, 16, 1073732976], entry
[ 26.644388 0:8 linux_syscall_api::syscall_fs::imp::io:104] [write()] fd: 2, buf: 0x3fffeb44, len: 16
Stream mapping:
[ 26.644568 0:8 linux_syscall_api::syscall:53] [syscall] id = 1,return 16
[ 26.644747 0:8 linux_syscall_api::syscall:31] [syscall] id = WRITE, args = [2, 1073736500, 21, 0, 21, 1073732960], entry
[ 26.644885 0:8 linux_syscall_api::syscall_fs::imp::io:104] [write()] fd: 2, buf: 0x3fffeb34, len: 21
  Stream #0:0 → #0:0[ 26.645061 0:8 linux_syscall_api::syscall:53] [syscall] id = 1,return 21
[ 26.645217 0:8 linux_syscall_api::syscall:31] [syscall] id = WRITE, args = [2, 1073736500, 34, 0, 34, 1073732960], entry
[ 26.645352 0:8 linux_syscall_api::syscall_fs::imp::io:104] [write()] fd: 2, buf: 0x3fffeb34, len: 34
 (h264 (native) → h264 (libx264))[ 26.645537 0:8 linux_syscall_api::syscall:53] [syscall] id = 1,return 34
[ 26.645640 0:8 linux_syscall_api::syscall:31] [syscall] id = WRITE, args = [2, 1073736516, 1, 0, 1, 1073732976], entry
[ 26.645770 0:8 linux_syscall_api::syscall_fs::imp::io:104] [write()] fd: 2, buf: 0x3fffeb44, len: 1

[ 26.645921 0:8 linux_syscall_api::syscall:53] [syscall] id = 1,return 1
[ 26.646035 0:8 linux_syscall_api::syscall:31] [syscall] id = WRITE, args = [2, 1073736516, 32, 0, 32, 1073732976], entry
[ 26.646170 0:8 linux_syscall_api::syscall_fs::imp::io:104] [write()] fd: 2, buf: 0x3fffeb44, len: 32
Press [q] to stop, [?] for help
[ 26.646349 0:8 linux_syscall_api::syscall:53] [syscall] id = 1,return 32
[ 26.646447 0:8 linux_syscall_api::syscall:40] [syscall] id = CLOCK_GET_TIME, args = [1, 1073738800, 0, 0, 32, 1073732976], entry
[ 26.646608 0:8 linux_syscall_api::syscall:53] [syscall] id = 228,return 0
[ 26.646831 0:8 linux_syscall_api::syscall:40] [syscall] id = CLOCK_GET_TIME, args = [1, 1073738800, 218153440, 0, 32, 1073732976], entry
[ 26.646989 0:8 linux_syscall_api::syscall:53] [syscall] id = 228,return 0   直接跳到这里了
[ 26.647246 0:8 linux_syscall_api::syscall:31] [syscall] id = PSELECT6, args = [1, 1073733904, 0, 0, 1073733744, 0], entry
[ 26.647953 0:8 linux_syscall_api::syscall:53] [syscall] id = 270,return 1
[ 26.648093 0:8 linux_syscall_api::syscall:31] [syscall] id = READ, args = [0, 1073733884, 1, 0, 1073733744, 0], entry
[ 26.648227 0:8 linux_syscall_api::syscall_fs::imp::io:32] [read()] fd: 0, buf: 0x3fffe0fc, len: 1
QEMU: Terminated
```

```
socket(AF_INET, SOCK_STREAM|SOCK_CLOEXEC, IPPROTO_TCP) = 4
fcntl(4, F_GETFL)                       = 0x2 (flags O_RDWR)
fcntl(4, F_SETFL, O_RDWR|O_NONBLOCK)    = 0
connect(4, {sa_family=AF_INET, sin_port=htons(554), sin_addr=inet_addr("127.0.0.1")}, 16) = -1 EINPROGRESS (操作现在正在进行)
poll([{fd=4, events=POLLOUT}], 1, 100)  = 1 ([{fd=4, revents=POLLOUT}])
getsockopt(4, SOL_SOCKET, SO_ERROR, [0], [4]) = 0
getpeername(4, {sa_family=AF_INET, sin_port=htons(554), sin_addr=inet_addr("127.0.0.1")}, [128 => 16]) = 0
poll([{fd=4, events=POLLOUT}], 1, 100)  = 1 ([{fd=4, revents=POLLOUT}])
sendto(4, "OPTIONS rtsp://127.0.0.1:554/liv"..., 87, MSG_NOSIGNAL, NULL, 0) = 87
poll([{fd=4, events=POLLIN}], 1, 100)   = 1 ([{fd=4, revents=POLLIN}])
recvfrom(4, "R", 1, 0, NULL, NULL)      = 1
poll([{fd=4, events=POLLIN}], 1, 100)   = 1 ([{fd=4, revents=POLLIN}])
recvfrom(4, "T", 1, 0, NULL, NULL)      = 1
poll([{fd=4, events=POLLIN}], 1, 100)   = 1 ([{fd=4, revents=POLLIN}])
recvfrom(4, "S", 1, 0, NULL, NULL)      = 1
poll([{fd=4, events=POLLIN}], 1, 100)   = 1 ([{fd=4, revents=POLLIN}])
recvfrom(4, "P", 1, 0, NULL, NULL)      = 1
poll([{fd=4, events=POLLIN}], 1, 100)   = 1 ([{fd=4, revents=POLLIN}])
```

```
[libx264 @ 0xd012200] [304.385223 0:8 linux_syscall_api::syscall:53] [syscall] id = 1,return 22
[304.385319 0:8 linux_syscall_api::syscall:31] [syscall] id = WRITE, args = [2, 1073726132, 663, 0, 663, 63], entry
[304.385450 0:8 linux_syscall_api::syscall_fs::imp::io:104] [write()] fd: 2, buf: 0x3fffc2b4, len: 663
264 - core 155 r2917 0a84d98 - H.264/MPEG-4 AVC codec - Copyleft 2003-2018 - http://www.videolan.org/x264.html - options: cabac=1 ref=3 deblock=1:0:0 analyse=0x3:0x113
[304.386259 0:8 linux_syscall_api::syscall:53] [syscall] id = 1,return 663
[304.392488 0:8 linux_syscall_api::syscall:40] [syscall] id = CLOCK_GET_TIME, args = [1, 1073718064, 3, 34663104, 1073718692, 5912992], entry
[304.392666 0:8 linux_syscall_api::syscall:53] [syscall] id = 228,return 0
[304.394381 0:8 linux_syscall_api::syscall:12] [syscall] id = SOCKET, args = [2, 524289, 6, 0, 0, 1073716080], entry
[304.394769 0:8 linux_syscall_api::syscall:53] [syscall] id = 41,return 5
[304.394902 0:8 linux_syscall_api::syscall:40] [syscall] id = CLOCK_GET_TIME, args = [1, 1073718064, 6, 3, 0, 1073716080], entry
[304.395046 0:8 linux_syscall_api::syscall:53] [syscall] id = 228,return 0
[304.395173 0:8 linux_syscall_api::syscall:31] [syscall] id = FCNTL64, args = [5, 3, 395045, 3, 0, 1073716080], entry
[304.395320 0:8 linux_syscall_api::syscall_fs::imp::ctl:369] fd: 5, cmd: 3
[304.395444 0:8 linux_syscall_api::syscall:53] [syscall] id = 72,return 524288
[304.395565 0:8 linux_syscall_api::syscall:31] [syscall] id = FCNTL64, args = [5, 4, 526336, 3, 0, 1073716080], entry
[304.395698 0:8 linux_syscall_api::syscall_fs::imp::ctl:369] fd: 5, cmd: 4
[304.395819 0:8 linux_syscall_api::syscall:53] [syscall] id = 72,return 0
[304.395973 0:8 linux_syscall_api::syscall:12] [syscall] id = CONNECT, args = [5, 218150976, 16, 3, 0, 1073716080], entry
[304.396209 0:8 linux_syscall_api::syscall_net::imp:187] [connect()] socket 5 connecting to Endpoint { addr: Ipv4(Address([127, 0, 0, 1])), port: 554 }
[304.396949 0:8 axnet::smoltcp_impl::tcp:178] bound endpoint: ListenEndpoint { addr: None, port: 49152 }
[304.397170 0:8 axnet::smoltcp_impl::tcp:179] remote endpoint: Endpoint { addr: Ipv4(Address([127, 0, 0, 1])), port: 554 }
[304.397332 0:8 axnet::smoltcp_impl::tcp:180] Temporarily net bridge used
[304.397695 0:8 linux_syscall_api::syscall:53] [syscall] id = 42,return -115
[304.397837 0:8 linux_syscall_api::syscall:40] [syscall] id = CLOCK_GET_TIME, args = [1, 1073718064, 4294967181, 3, 0, 4], entry
[304.397986 0:8 linux_syscall_api::syscall:53] [syscall] id = 228,return 0
[304.398093 0:8 linux_syscall_api::syscall:40] [syscall] id = CLOCK_GET_TIME, args = [1, 1073718064, 397985, 3, 0, 4], entry
[304.398228 0:8 linux_syscall_api::syscall:53] [syscall] id = 228,return 0
[304.398752 0:8 linux_syscall_api::syscall:31] [syscall] id = POLL, args = [1073718240, 1, 100, 3, 0, 4], entry
[304.402361 0:8 linux_syscall_api::syscall:53] [syscall] id = 7,return 1
[304.402533 0:8 linux_syscall_api::syscall:12] [syscall] id = GETSOCKOPT, args = [5, 1, 4, 1073718232, 1073718236, 4], entry
[304.402941 0:8 linux_syscall_api::syscall:53] [syscall] id = 55,return 0
[304.403187 0:8 linux_syscall_api::syscall:12] [syscall] id = GETPEERNAME, args = [5, 1073722448, 1073722436, 1073722448, 219845248, 219845200], entry
[304.403636 0:8 linux_syscall_api::syscall:53] [syscall] id = 52,return -107
[304.404497 0:8 linux_syscall_api::syscall:31] [syscall] id = POLL, args = [1073712240, 1, 100, 0, 5, 1073712096], entry
[304.404860 0:8 linux_syscall_api::syscall:53] [syscall] id = 7,return 0
[304.404993 0:8 linux_syscall_api::syscall:31] [syscall] id = POLL, args = [1073712240, 1, 100, 0, 5, 1073712096], entry
[304.405305 0:8 linux_syscall_api::syscall:53] [syscall] id = 7,return 0
[304.405414 0:8 linux_syscall_api::syscall:31] [syscall] id = POLL, args = [1073712240, 1, 100, 0, 5, 1073712096], entry
[304.405704 0:8 linux_syscall_api::syscall:53] [syscall] id = 7,return 0
[304.405798 0:8 linux_syscall_api::syscall:31] [syscall] id = POLL, args = [1073712240, 1, 100, 0, 5, 1073712096], entry
[304.406100 0:8 linux_syscall_api::syscall:53] [syscall] id = 7,return 0
[304.406208 0:8 linux_syscall_api::syscall:31] [syscall] id = POLL, args = [1073712240, 1, 100, 0, 5, 1073712096], entry
[304.406507 0:8 linux_syscall_api::syscall:53] [syscall] id = 7,return 0
[304.406617 0:8 linux_syscall_api::syscall:31] [syscall] id = POLL, args = [1073712240, 1, 100, 0, 5, 1073712096], entry
[304.406914 0:8 linux_syscall_api::syscall:53] [syscall] id = 7,return 0
[304.407022 0:8 linux_syscall_api::syscall:31] [syscall] id = POLL, args = [1073712240, 1, 100, 0, 5, 1073712096], entry
[304.407325 0:8 linux_syscall_api::syscall:53] [syscall] id = 7,return 0
[304.407434 0:8 linux_syscall_api::syscall:31] [syscall] id = POLL, args = [1073712240, 1, 100, 0, 5, 1073712096], entry
[304.407737 0:8 linux_syscall_api::syscall:53] [syscall] id = 7,return 0
[304.407845 0:8 linux_syscall_api::syscall:31] [syscall] id = POLL, args = [1073712240, 1, 100, 0, 5, 1073712096], entry
[304.408146 0:8 linux_syscall_api::syscall:53] [syscall] id = 7,return 0
[304.408256 0:8 linux_syscall_api::syscall:31] [syscall] id = POLL, args = [1073712240, 1, 100, 0, 5, 1073712096], entry
```

连接过程中会出现不支持AF_NETLINK，这个在Starry中暂未实现

```
ffmpeg version 4.2.7-0ubuntu0.1 Copyright (c) 2000-2022 the FFmpeg developers
  built with gcc 9 (Ubuntu 9.4.0-1ubuntu1~20.04.1)
  configuration: --prefix=/usr --extra-version=0ubuntu0.1 --toolchain=hardened --libdir=/usr/lib/x86_64-linux-gnu --incdir=/usr/include/x86_64-linux-gnu
  libavutil      56. 31.100 / 56. 31.100
  libavcodec     58. 54.100 / 58. 54.100
  libavformat    58. 29.100 / 58. 29.100
  libavdevice    58.  8.100 / 58.  8.100
  libavfilter     7. 57.100 /  7. 57.100
  libavresample   4.  0.  0 /  4.  0.  0
  libswscale      5.  5.100 /  5.  5.100
  libswresample   3.  5.100 /  3.  5.100
  libpostproc    55.  5.100 / 55.  5.100
Input #0, mov,mp4,m4a,3gp,3g2,mj2, from '01.mp4':
  Metadata:
    major_brand     : mp42
    minor_version   : 0
    compatible_brands: mp42mp41
    creation_time   : 2018-04-09T16:53:03.000000Z
  Duration: 00:00:26.79, start: 0.000000, bitrate: 764 kb/s
    Stream #0:0(eng): Video: h264 (Main) (avc1 / 0x31637661), yuv420p(tv, bt709), 550x750, 755 kb/s, 29.97 fps, 29.97 tbr, 30k tbn, 59.94 tbc (default)
    Metadata:
      creation_time   : 2018-04-09T16:53:03.000000Z
      handler_name    : ?Mainconcept Video Media Handler
      encoder         : AVC Coding
Stream mapping:
  Stream #0:0 → #0:0 (h264 (native) → h264 (libx264))
Press [q] to stop, [?] for help
[libx264 @ 0xd012200] using cpu capabilities: MMX2 SSE2Slow
[libx264 @ 0xd012200] profile High, level 3.1
[libx264 @ 0xd012200] 264 - core 155 r2917 0a84d98 - H.264/MPEG-4 AVC codec - Copyleft 2003-2018 - http://www.videolan.org/x264.html - options: cabac=1
1970-01-01 00:05:12.170 W [MediaServer] [11-event poller 0] EventPoller.cpp:209 async_l | take time: 37ms, thread may be overloaded
[312.713012 0:18 linux_syscall_api::syscall_net::imp:27] [socket()] Address Family not supported: 16
[312.758490 0:18 axruntime::lang_items:5] panicked at crates/linux_syscall_api/src/syscall_net/socket.rs:851:28:
not implemented
root@josen:/code/main/Starry#
```

陈老师建议从现有的实现出发，通过ipv4实现连接

方案2：将ZLM改为IPV4的连接方式

修改zlm代码sockutil.cpp

```
502 int SockUtil::listen(const uint16_t port, const char *local_ip, int back_log) {
503     int fd = -1;
504     //int family = support_ipv6() ? (is_ipv4(local_ip) ? AF_INET : AF_INET6) : AF_INET;
505     int family = AF_INET;
506     if ((fd = (int)socket(family, SOCK_STREAM, IPPROTO_TCP)) == -1) {
507         WarnL << "Create socket failed: " << get_uv_errmsg(true);
508         return -1;
509     }
```

```
764  int SockUtil::bindUdpSock(const uint16_t port, const char *local_ip, bool enable_reuse) {
765      int fd = -1;
766      //int family = support_ipv6() ? (is_ipv4(local_ip) ? AF_INET : AF_INET6) : AF_INET;
767      int family = AF_INET;
768      if ((fd = (int)socket(family, SOCK_DGRAM, IPPROTO_UDP)) == -1) {
769          WarnL << "Create socket failed: " << get_uv_errmsg(true);
770          return -1;
771      }
```

编译后得到新的执行文件MediaServer，放入starry后出现如下报错

```
[ 62.825274 0:12 linux_syscall_api::syscall:30] [syscall] id = READLINK, args = [16534220, 1067450960, 8193, 16, 2, 0], entry
[ 62.825421 0:12 linux_syscall_api::syscall_fs::imp::io:634] read link at: /proc/self/exe
[ 62.825521 0:12 linux_syscall_api::syscall:57] [syscall] id = 89,return 0
[ 62.825612 0:12 linux_syscall_api::syscall:30] [syscall] id = READLINK, args = [16534220, 1067450656, 8193, 16, 2, 0], entry
[ 62.825756 0:12 linux_syscall_api::syscall_fs::imp::io:634] read link at: /proc/self/exe
[ 62.825859 0:12 linux_syscall_api::syscall:57] [syscall] id = 89,return 0
[ 62.826024 0:12 axmem:416] Page fault address VA:0x3f9fffd8 not found in memory set
[ 62.826493 0:12 axprocess::signal:178] cpu: 0, task: 12, handler signal: 11
[ 62.826728 0:12 axprocess::signal:257] use stack: 0x3f9ffdd0
[ 62.826844 0:12 axprocess::signal:264] restorer :0x1b88520, handler: 0x9cbeb5
[ 62.827016 0:12 axruntime::lang_items:5] panicked at crates/axtrap/src/arch/x86_64/mod.rs:61:17:
Kernel #PF @ 0xffffff800029d63b, fault_vaddr=0x3f9ffdc8, error_code=0x2:
TrapFrame {
    rax: 0x1b88520,
    rcx: 0xffffff80002d5d0c,
    rdx: 0x3f8,
    rbx: 0xffffff80147ea790,
    rbp: 0xffffff80147e9dc0,
    rsi: 0xffffff80002d5d0c,
    rdi: 0xb,
    r8: 0x97,
    r9: 0x1,
    r10: 0x8,
    r11: 0x3836,
    r12: 0x14000000,
    r13: 0xffffff80147e9a80,
    r14: 0xffffff80147ea910,
    r15: 0x3f9ffdd0,
    vector: 0xe,
    error_code: 0x2,
    rip: 0xffffff800029d63b,
    cs: 0x10,
    rflags: 0x46,
    rsp: 0xffffff80147e9770,
    ss: 0x0,
}
[ 62.828380 0:12 axtask::api:224] dump task: Task(12, "MediaServer"), stack range: 0xffffff80147a9a50: 0xffffff80147e9a50
[ 62.828577 0:12 axbacktrace:43] Call trace:
[ 62.828686 0:12 axbacktrace::x86:91] 0xFFFFFF800027D821
[ 62.828849 0:12 axhal::platform::x86_pc::misc:5] Shutting down...
root@josen:/code/Starry# ./zlm_start.sh
```

陷入重复的readlink，修改zlm的代码：

3rdpart/ZLToolKit/src/Util/util.cpp

```
145  #elif defined(__linux__)
146      n = readlink("/proc/self/exe", buffer, sizeof(buffer));
147  #endif
148
149      string filePath;
150      if (n <= 0) {
151          filePath = "/";
152      } else {
153          filePath = buffer;
154      }
```

修改之后在Starry中运行，发现starry需要支持 `AF_UNIX` 和 `AF_NETLINK`

```
Input #0, mov,mp4,m4a,3gp,3g2,mj2, from '01.mp4':
  Metadata:
    major_brand     : mp42
    minor_version   : 0
    compatible_brands: mp42mp41
    creation_time   : 2018-04-09T16:53:03.000000Z
  Duration: 00:00:26.79, start: 0.000000, bitrate: 764 kb/s
    Stream #0:0(eng): Video: h264 (Main) (avc1 / 0x31637661), yuv420p(tv, bt709), 550x750, 755 kb/s, 29.97 fps, 29.97 tbr,
    Metadata:
      creation_time   : 2018-04-09T16:53:03.000000Z
      handler_name    : ?Mainconcept Video Media Handler
      encoder         : AVC Coding
Stream mapping:
  Stream #0:0 → #0:0 (h264 (native) → h264 (libx264))
Press [q] to stop, [?] for help
[337.885746 0:17 linux_syscall_api::syscall_fs::imp::io:637] path.path() is /proc/self/exe
[337.885899 0:17 linux_syscall_api::syscall_fs::imp::io:642] file_real_path is
1970-01-01 00:05:38.063 W [MediaServer] [13-event poller 0] EventPoller.cpp:255 async_l | take time: 30ms, thread may be o
[338.609440 0:22 linux_syscall_api::syscall_net::imp:27] [socket()] Address Family not supported: 16
[338.652090 0:22 axruntime::lang_items:5] panicked at crates/linux_syscall_api/src/syscall_net/socket.rs:851:28:
not implemented
[338.652389 0:22 axbacktrace:43] Call trace:
[338.652517 0:22 axbacktrace::x86:91] 0xFFFFFF8000273D90
[338.652707 0:22 axbacktrace::x86:91] 0xFFFFFF80002A15DA
[338.652797 0:22 axbacktrace::x86:91] 0x0000000000000011
```

但是ZLM与FFMPEG是通过 `AF_INET` 建立连接，用不到 `AF_UNIX` 和 `AF_NETLINK`

第二次运行又出现以下报错

```
Input #0, mov,mp4,m4a,3gp,3g2,mj2, from '01.mp4':
  Metadata:
    major_brand     : mp42
    minor_version   : 0
    compatible_brands: mp42mp41
    creation_time   : 2018-04-09T16:53:03.000000Z
  Duration: 00:00:26.79, start: 0.000000, bitrate: 764 kb/s
    Stream #0:0(eng): Video: h264 (Main) (avc1 / 0x31637661), yuv420p(tv, bt709), 550x750, 755 kb/s, 29.97 fps, 29.97 tbr, 30k tbr
    Metadata:
      creation_time   : 2018-04-09T16:53:03.000000Z
      handler_name    : ?Mainconcept Video Media Handler
      encoder         : AVC Coding
Stream mapping:
  Stream #0:0 → #0:0 (h264 (native) → h264 (libx264))
Press [q] to stop, [?] for help
[313.321444 0:17 linux_syscall_api::syscall_fs::imp::io:637] path.path() is /proc/self/exe
[313.321564 0:17 linux_syscall_api::syscall_fs::imp::io:642] file_real_path is
1970-01-01 00:05:13.501 W [MediaServer] [13-event poller 0] EventPoller.cpp:255 async_l | take time: 28ms, thread may be overloade
[314.065987 0:20 linux_syscall_api::syscall_net::socket:526] unimplemented SocketType: SOCK_RAW
[314.066146 0:20 axruntime::lang_items:5] panicked at crates/linux_syscall_api/src/syscall_net/socket.rs:527:17:
not implemented
[314.066417 0:20 axbacktrace:43] Call trace:
[314.066523 0:20 axbacktrace::x86:91] 0xFFFFFF8000273D90
[314.066682 0:20 axbacktrace::x86:91] 0xFFFFFF80002A3582
```

相关代码为

```rust
518     /// Create a new socket with the given domain and socket type.
519     pub fn new(domain: Domain, socket_type: SocketType) -> Self {
520         let inner: SocketInner = match socket_type {
521             SocketType::SOCK_STREAM | SocketType::SOCK_SEQPACKET => {
522                 SocketInner::Tcp(TcpSocket::new())
523             }
524             SocketType::SOCK_DGRAM => SocketInner::Udp(UdpSocket::new()),
525             _ => {
526                 error!("unimplemented SocketType: {:?}", socket_type);
527                 unimplemented!();
528             }
529         };
```

如果将 `UNIX` 全部换成 `INET` 后，出现了新的报错，缺少id为307的syscall

```
Input #0, mov,mp4,m4a,3gp,3g2,mj2, from '01.mp4':
  Metadata:
    major_brand     : mp42
    minor_version   : 0
    compatible_brands: mp42mp41
    creation_time   : 2018-04-09T16:53:03.000000Z
  Duration: 00:00:26.79, start: 0.000000, bitrate: 764 kb/s
    Stream #0:0(eng): Video: h264 (Main) (avc1 / 0x31637661), yuv420p(tv, bt709), 550x750, 755 kb/s
    Metadata:
      creation_time   : 2018-04-09T16:53:03.000000Z
      handler_name    : ?Mainconcept Video Media Handler
      encoder         : AVC Coding
Stream mapping:
  Stream #0:0 → #0:0 (h264 (native) → h264 (libx264))
Press [q] to stop, [?] for help
[315.947335 0:15 linux_syscall_api::syscall_fs::imp::io:637] path.path() is /proc/self/exe
[315.947529 0:15 linux_syscall_api::syscall_fs::imp::io:642] file_real_path is
1970-01-01 00:05:16.125 W [MediaServer] [11-event poller 0] EventPoller.cpp:255 async_l | take time
[317.310701 0:18 axruntime::lang_items:5] panicked at crates/linux_syscall_api/src/syscall.rs:53:9:
unknown syscall id: 307
[317.311101 0:18 axbacktrace:43] Call trace:
[317.311254 0:18 axbacktrace::x86:91] 0xFFFFFF80002ACCBC
```

id为307的系统调用为**syscall_sendto**，按理说不会直接用到这个系统调用，先使用**syscall_sendto**代替一下，ffmpeg的支持容易出现问题

2. 跨虚机连通

因此在starry中进行zlm监听，在同局域网的主机上使用ffmpeg推流

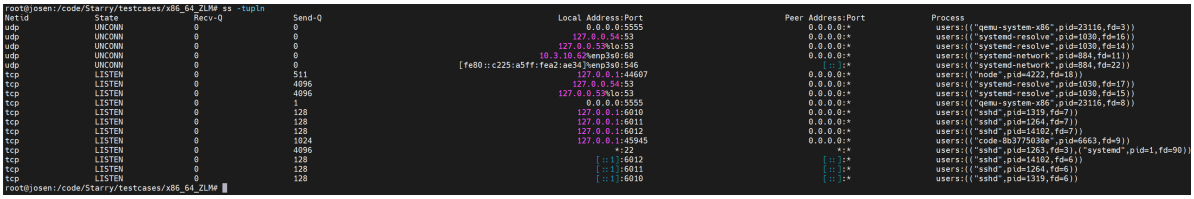
在linux中已验证，修改后的ZLM与ffmpeg通过ipv4建立连接

socket连接

```
tcp    LISTEN  0  1024  0.0.0.0:5554  0.0.0.0:*  users:(("MediaServer",pid=14525,fd=55))
tcp    LISTEN  0  1024  0.0.0.0:5566  0.0.0.0:*  users:(("MediaServer",pid=14525,fd=52))
tcp    LISTEN  0  1024  0.0.0.0:5525  0.0.0.0:*  users:(("MediaServer",pid=14525,fd=54))
```

```
root@josen:/code/Starry/testcases/x86_64_ZLM# ./ffmpeg -re -i "01.mp4" -vcodec h264 -acodec aac -f rtsp -rtsp_transport tcp rtsp://0.0.0.0:5566/live/
ffmpeg version 6.1.1-3ubuntu5 Copyright (c) 2000-2023 the FFmpeg developers
  built with gcc 13 (Ubuntu 13.2.0-23ubuntu3)
  configuration: --prefix=/usr --extra-version=3ubuntu5 --toolchain=hardened --libdir=/usr/lib/x86_64-linux-gnu --incdir=/usr/include/x86_64-linux-g
  libavutil      58. 29.100 / 58. 29.100
  libavcodec     60. 31.102 / 60. 31.102
  libavformat    60. 16.100 / 60. 16.100
  libavdevice    60.  3.100 / 60.  3.100
  libavfilter     9. 12.100 /  9. 12.100
  libswscale      7.  5.100 /  7.  5.100
  libswresample   4. 12.100 /  4. 12.100
  libpostproc    57.  3.100 / 57.  3.100
Input #0, mov,mp4,m4a,3gp,3g2,mj2, from '01.mp4':
  Metadata:
    major_brand     : mp42
    minor_version   : 0
    compatible_brands: mp42mp41
    creation_time   : 2018-04-09T16:53:03.000000Z
  Duration: 00:00:26.79, start: 0.000000, bitrate: 764 kb/s
  Stream #0:0[0x1](eng): Video: h264 (Main) (avc1 / 0x31637661), yuv420p(tv, bt709, progressive), 550x750, 755 kb/s, 29.97 fps, 29.97 tbr, 30k tbn (
    Metadata:
      creation_time   : 2018-04-09T16:53:03.000000Z
      handler_name    : ?Mainconcept Video Media Handler
      vendor_id       : [0][0][0][0]
      encoder         : AVC Coding
Stream mapping:
  Stream #0:0 → #0:0 (h264 (native) → h264 (libx264))
Press [q] to stop, [?] for help
[libx264 @ 0x61ecc0e2e580] using cpu capabilities: MMX2 SSE2Fast SSSE3 SSE4.2 AVX FMA3 BMI2 AVX2
[libx264 @ 0x61ecc0e2e580] profile High, level 3.1, 4:2:0, 8-bit
[libx264 @ 0x61ecc0e2e580] 264 - core 164 r3108 31e19f9 - H.264/MPEG-4 AVC codec - Copyleft 2003-2023 - http://www.videolan.org/x264.html - options:
Output #0, rtsp, to 'rtsp://0.0.0.0:5566/live/test':
  Metadata:
    major_brand     : mp42
    minor_version   : 0
    compatible_brands: mp42mp41
    encoder         : Lavf60.16.100
  Stream #0:0(eng): Video: h264, yuv420p(tv, bt709, progressive), 550x750, q=2-31, 29.97 fps, 90k tbn (default)
    Metadata:
      creation_time   : 2018-04-09T16:53:03.000000Z
      handler_name    : ?Mainconcept Video Media Handler
      vendor_id       : [0][0][0][0]
      encoder         : Lavc60.31.102 libx264
    Side data:
      cpb: bitrate max/min/avg: 0/0/0 buffer size: 0 vbv_delay: N/A
[out#0/rtsp @ 0x61ecc0da8680] video:1442kB audio:0kB subtitle:0kB other streams:0kB global headers:0kB muxing overhead: unknown
frame=  803 fps= 30 q=-1.0 Lsize=N/A time=00:00:26.69 bitrate=N/A speed=1.01x
[libx264 @ 0x61ecc0e2e580] frame I:22    Avg QP:16.53  size:  5428
[libx264 @ 0x61ecc0e2e580] frame P:215   Avg QP:18.95  size:  2851
[libx264 @ 0x61ecc0e2e580] frame B:566   Avg QP:19.44  size:  1314
[libx264 @ 0x61ecc0e2e580] consecutive B-frames:  4.0%  5.0%  3.4% 87.7%
[libx264 @ 0x61ecc0e2e580] mb I  I16..4: 36.3% 61.3%  2.3%
[libx264 @ 0x61ecc0e2e580] mb P  I16..4: 15.7% 22.6%  0.1%  P16..4: 30.2%  2.3%  0.7%  0.0%  0.0%    skip:28.5%
[libx264 @ 0x61ecc0e2e580] mb B  I16..4:  2.2%  2.3%  0.0%  B16..8: 28.2%  1.3%  0.0%  direct: 6.5%  skip:59.6%  L0:49.0% L1:50.3% BI: 0.8%
[libx264 @ 0x61ecc0e2e580] 8x8 transform intra:57.6% inter:99.7%
[libx264 @ 0x61ecc0e2e580] coded y,uvDC,uvAC intra: 9.8% 31.3% 3.1% inter: 2.8% 15.7% 0.0%
[libx264 @ 0x61ecc0e2e580] i16 v,h,dc,p: 33% 30% 17% 21%
[libx264 @ 0x61ecc0e2e580] i8 v,h,dc,ddl,ddr,vr,hd,vl,hu: 23% 15% 60%  1%  0%  0%  0%  0%  0%
[libx264 @ 0x61ecc0e2e580] i4 v,h,dc,ddl,ddr,vr,hd,vl,hu: 39% 24% 28%  3%  2%  1%  2%  1%  1%
[libx264 @ 0x61ecc0e2e580] i8c dc,h,v,p: 62% 20% 15%  3%
[libx264 @ 0x61ecc0e2e580] Weighted P-Frames: Y:5.1% UV:3.7%
[libx264 @ 0x61ecc0e2e580] ref P L0: 67.4%  2.1% 20.3% 10.1%  0.2%
[libx264 @ 0x61ecc0e2e580] ref B L0: 78.1% 18.6%  3.3%
[libx264 @ 0x61ecc0e2e580] ref B L1: 88.6% 11.4%
[libx264 @ 0x61ecc0e2e580] kb/s:440.75
```

qemu与主机也是通过ipv4通信，当前的情况为，通过qemu启动starry后，starry中启动MediaServer后在主机端检查不到它监听的地址和端口



Starry 已经支持和同局域网下的其他主机通信，经验证，Starry 和同局域网下的其他主机通信正常

```
Running testcase: busybox sh
[  0.346497 0:6 linux_syscall_api::syscall_fs::imp::ctl:416] file.F_DUPFD_CLOEXEC
[  0.347064 0:6 linux_syscall_api::syscall_fs::imp::ctl:416] file.F_DUPFD_CLOEXEC
[  0.347377 0:6 linux_syscall_api::syscall_fs::imp::ctl:391] file.set_close_on_exec
/ # ./cli[ 42.936960 0:6 linux_syscall_api::syscall_fs::imp::ctl:391] file.set_close_on_exec
/ # ./client
          7:     file=./client [0];  generating link map
          7:       dynamic: 0x0000000000004d60  base: 0x0000000000001000   size: 0x0000000000004018
          7:         entry: 0x0000000000002200  phdr: 0x0000000000001040  phnum:                 13
          7:
          7:
          7:     file=libc.so.6 [0];  needed by ./client [0]
          7:     file=libc.so.6 [0];  generating link map
          7:       dynamic: 0x0000000000021fbc0  base: 0x0000000000006000   size: 0x0000000000228e50
          7:         entry: 0x000000000002ff50  phdr: 0x0000000000006040  phnum:                 14
          7:
          7:
          7:     calling init: /lib64/ld-linux-x86-64.so.2
          7:
          7:
          7:     calling init: /lib/libc.so.6
          7:
          7:
          7:     initialize program: ./client
          7:
          7:
          7:     transferring control: ./client
          7:
          7:
          7:     calling fini: ./client [0]
          7:
Local IP address: 10.0.2.15
Message sent to server
Received from server: Hello from server
```

starry作为客户端可以与外面服务端通信，实现正常收发数据

但是作为客户端，外面访问不到Starry的网络

已能够作为客户端与同局域网下的其他主机通信，其他主机访问starry是本机所在的 IP 和 5555 号端口，访问 Starry 就是访问本机的 qemu 进程

syscall_sendmsg实现

还缺少 `syscall_sendmsg`，`id = 46`，接受三个参数`sockfd`， `msg`，`flags`

```
ssize_t sendmsg(int sockfd, const struct msghdr *msg, int flags);
```

但是第四五六个参数依然有值，根据前三个参数的传入内容可以确定是`sendmsg`，而不是`sendto`，

flags的值为`MSG_DONTWAIT|MSG_NOSIGNAL`，即 `0x4040 = 16448`

```
[ 70.606061 0:17 linux_syscall_api::syscall_net::imp:449] sendmsg message_hdr name: 0x0, name_len: 0
[ 70.606176 0:17 linux_syscall_api::syscall_net::imp:450] endmsg message_hdr iovec: 0x100021e0, iovec_len: 1
[ 70.606299 0:17 linux_syscall_api::syscall_net::imp:451] endmsg message_hdr iovec.base: 0x10003660, iovec.len: 279
[ 70.606432 0:17 linux_syscall_api::syscall_net::imp:445] sendmsg fd: 13, message_hdr: 0x1f31a80, flags: 16448
[ 70.606557 0:17 linux_syscall_api::syscall_netQEMU: Terminated
```

添加 `sendmsg`，实现代码如下

```
pub fn syscall_sendmsg(args: [usize; 6]) -> SyscallResult {
    let fd = args[0];
    let msg = args[1] as *mut MessageHeader;
    let flags = args[2];
    let curr = current_process();
    let msg = unsafe { &*msg };
    error!("sendmsg fd: {fd}, msg: {msg:?}, flags: {flags}");
    error!("msg.msg_name: {:?}, msg_name_len: {:?}", msg.name, msg.name_len);

    let file = match curr.fd_manager.fd_table.lock().get(fd) {
        Some(Some(file)) => file.clone(),
        _ => return Err(SyscallError::EBADF),
    };

    let Some(socket) = file.as_any().downcast_ref::<Socket>() else {
        return Err(SyscallError::ENOTSOCK);
    };

    // let iov = unsafe { core::slice::from_raw_parts(msg.iovec, msg.iovec_len as
usize) };
    // let mut buf = Vec::new();

    // for iov in iov {
    //     let buf_part = unsafe { core::slice::from_raw_parts(iov.base as *const
u8, iov.len) };
    //     buf.extend_from_slice(buf_part);
    // }
    let msg_header = &*msg;
    let iove = unsafe{&*msg_header.iovec};
    let buf = iove.base;
    let len = iove.len;
    let Ok(buf) = curr
        .manual_alloc_range_for_lazy(
            (buf as usize).into(),
            unsafe { buf.add(len as usize) as usize }.into(),
        )
        .map(|_| unsafe { from_raw_parts(buf, len as usize) })
    else {
        error!("[sendto()] buf address {buf:?} invalid");
        return Err(SyscallError::EFAULT);
    };

    let addr =
axnet::SocketAddr::new(axnet::IpAddr::Ipv4(axnet::Ipv4Addr::new(10, 3, 10, 62)),
5555);

    match socket.sendto(buf, Some(addr)) {
        Ok(len) => {
            error!("[sendmsg()] socket {fd} sent {len} bytes to addr {:?}",
addr);
            Ok(len as isize)
        }
        Err(AxError::Interrupted) => Err(SyscallError::EINTR),
        Err(AxError::Again) | Err(AxError::WouldBlock) =>
Err(SyscallError::EAGAIN),
```

```
        Err(AxError::NotConnected) => Err(SyscallError::ENOTCONN),
        Err(AxError::ConnectionReset) => Err(SyscallError::EPIPE),
        Err(e) => {
            error!("[sendmsg()] socket {fd} send error: {e:?}");
            Err(SyscallError::EPERM)
        }
    }
}
```

sendmsg信息发送之后ffmpeg一直处于超时状态

```
recvfrom(4, "C", 1, 0, NULL, NULL)      = 1
poll([{fd=4, events=POLLIN}], 1, 100)   = 1 ([{fd=4, revents=POLLIN}])
recvfrom(4, "b", 1, 0, NULL, NULL)      = 1
poll([{fd=4, events=POLLIN}], 1, 100)   = 1 ([{fd=4, revents=POLLIN}])
recvfrom(4, "d", 1, 0, NULL, NULL)      = 1
poll([{fd=4, events=POLLIN}], 1, 100)   = 1 ([{fd=4, revents=POLLIN}])
recvfrom(4, "C", 1, 0, NULL, NULL)      = 1
poll([{fd=4, events=POLLIN}], 1, 100)   = 1 ([{fd=4, revents=POLLIN}])
recvfrom(4, "\r", 1, 0, NULL, NULL)     = 1
poll([{fd=4, events=POLLIN}], 1, 100)   = 1 ([{fd=4, revents=POLLIN}])
recvfrom(4, "\n", 1, 0, NULL, NULL)     = 1
poll([{fd=4, events=POLLIN}], 1, 100)   = 1 ([{fd=4, revents=POLLIN}])
recvfrom(4, "\r", 1, 0, NULL, NULL)     = 1
poll([{fd=4, events=POLLIN}], 1, 100)   = 1 ([{fd=4, revents=POLLIN}])
recvfrom(4, "\n", 1, 0, NULL, NULL)     = 1
write(2, "Output #0, rtsp, to 'rtsp://10.3"..., 56Output #0, rtsp, to 'rtsp://10.3.10.62:5555/live/test':
) = 56
write(2, "  Metadata:\n", 12  Metadata:
)           = 12
write(2, "    major_brand     : ", 22    major_brand     : )  = 22
write(2, "mp42", 4mp42)                     = 4
write(2, "\n", 1
```

```
recvfrom(4, "\r", 1, 0, NULL, NULL)     = 1
poll([{fd=4, events=POLLIN}], 1, 100)   = 1 ([{fd=4, revents=POLLIN}])
recvfrom(4, "\n", 1, 0, NULL, NULL)     = 1
poll([{fd=4, events=POLLOUT}], 1, 100)  = 1 ([{fd=4, revents=POLLOUT}])
sendto(4, "ANNOUNCE rtsp://10.3.10.62:5555/"..., 142, MSG_NOSIGNAL, NULL, 0) = 142
poll([{fd=4, events=POLLOUT}], 1, 100)  = 1 ([{fd=4, revents=POLLOUT}])
sendto(4, "v=0\r\no=- 0 0 IN IP4 127.0.0.1\r\ns"..., 289, MSG_NOSIGNAL, NULL, 0) = 289
poll([{fd=4, events=POLLIN}], 1, 100)   = 0 (Timeout)
poll([{fd=4, events=POLLIN}], 1, 100)   = 0 (Timeout)
poll([{fd=4, events=POLLIN}], 1, 100)   = 0 (Timeout)
poll([{fd=4, events=POLLIN}], 1, 100)   = 0 (Timeout)
poll([{fd=4, events=POLLIN}], 1, 100)   = 0 (Timeout)
poll([{fd=4, events=POLLIN}], 1, 100)   = 0 (Timeout)
poll([{fd=4, events=POLLIN}], 1, 100)   = 0 (Timeout)
poll([{fd=4, events=POLLIN}], 1, 100)   = 0 (Timeout)
poll([{fd=4, events=POLLIN}], 1, 100)   = 0 (Timeout)
poll([{fd=4, events=POLLIN}], 1, 100)   = 0 (Timeout)
poll([{fd=4, events=POLLIN}], 1, 100)   = 0 (Timeout)
poll([{fd=4, events=POLLIN}], 1, 100)   = 0 (Timeout)
poll([{fd=4, events=POLLIN}], 1, 100)   = 0 (Timeout)
poll([{fd=4, events=POLLIN}], 1, 100)   = 0 (Timeout)
poll([{fd=4, events=POLLIN}], 1, 100)   = 0 (Timeout)
poll([{fd=4, events=POLLIN}], 1, 100)   = 0 (Timeout)
poll([{fd=4, events=POLLIN}], 1, 100)   = 0 (Timeout)
poll([{fd=4, events=POLLIN}], 1, 100)   = 0 (Timeout)
poll([{fd=4, events=POLLIN}], 1, 100)   = 0 (Timeout)
poll([{fd=4, events=POLLIN}], 1, 100)   = 0 (Timeout)
poll([{fd=4, events=POLLIN}], 1, 100)   = 0 (Timeout)
poll([{fd=4, events=POLLIN}], 1, 100)   = 0 (Timeout)
poll([{fd=4, events=POLLIN}], 1, 100)   = 0 (Timeout)
poll([{fd=4, events=POLLIN}], 1, 100)   = 0 (Timeout)
poll([{fd=4, events=POLLIN}], 1, 100)   = 0 (Timeout)
poll([{fd=4, events=POLLIN}], 1, 100)   = 0 (Timeout)
poll([{fd=4, events=POLLIN}], 1, 100)   = 0 (Timeout)
poll([{fd=4, events=POLLIN}], 1, 100)   = 0 (Timeout)
poll([{fd=4, events=POLLIN}], 1, 100)   = 0 (Timeout)
poll([{fd=4, events=POLLIN}], 1, 100)   = 0 (Timeout)
poll([{fd=4, events=POLLIN}], 1, 100)   = 0 (Timeout)
poll([{fd=4, events=POLLIN}], 1, 100)   = 1 ([{fd=4, revents=POLLIN}])
recvfrom(4, "", 1, 0, NULL, NULL)       = 0
close(4)                                = 0
write(2, "\33[1;31mCould not write header fo"..., 126Could not write header for output file #0 (incorrect codec parameters ?): Invalid data found when processing input
) = 126
write(2, "\33[1;31mError initializing output"..., 52Error initializing output stream 0:0 --
) = 52
```

似乎还缺少个文件/dev/urandom

```
recvfrom(4, "\n", 1, 0, NULL, NULL)      = 1
openat(AT_FDCWD, "/dev/urandom", O_RDONLY) = 5
fcntl(5, F_SETFD, FD_CLOEXEC)           = 0
read(5, "K\265\325t", 4)                = 4
close(5)                                = 0
```

这个文件应该不影响

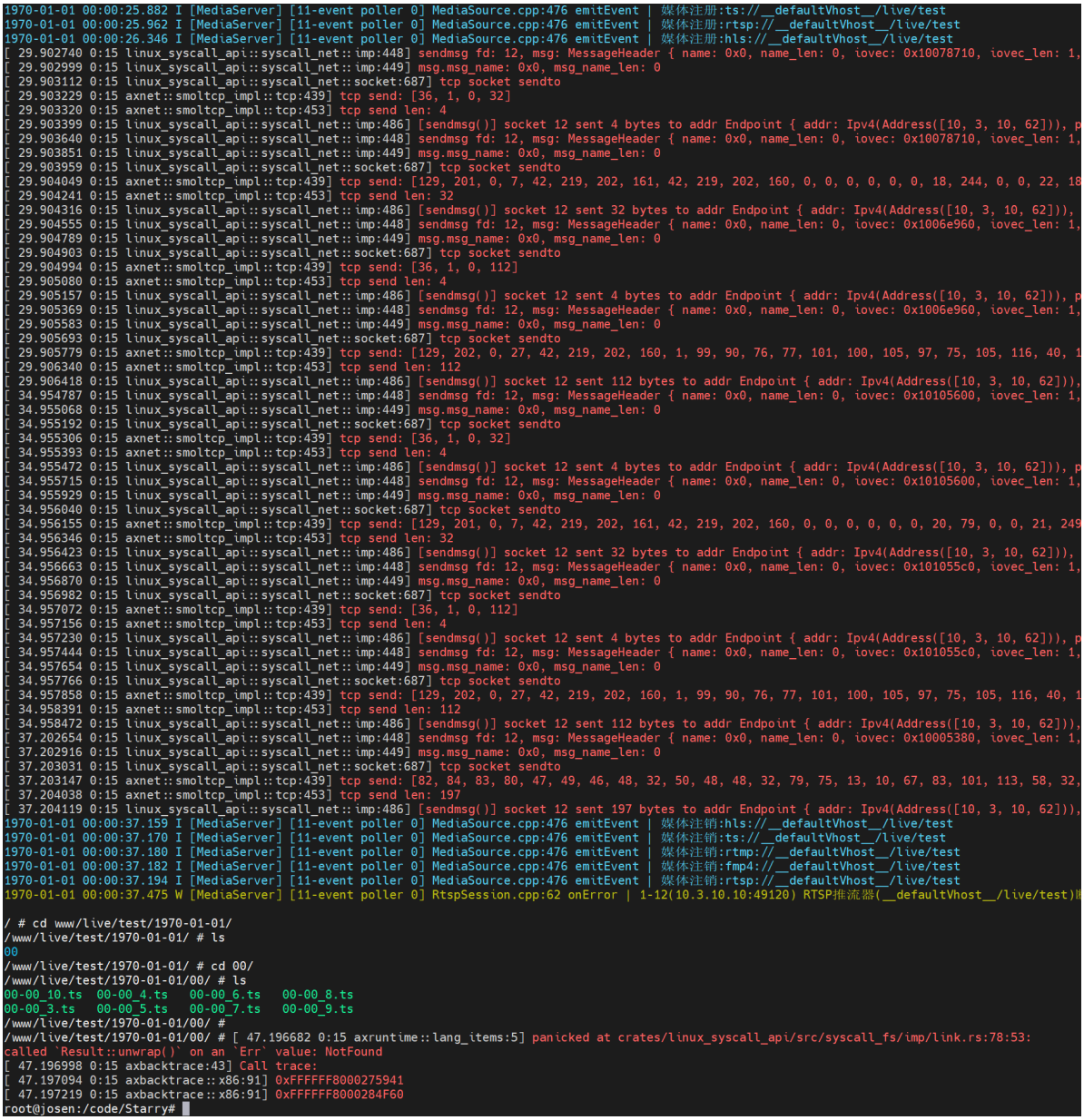
把日志关闭后starry的相应速度会更快，减小超时退出的可能性

3. 初步实现效果

```
root@josen:/home/josen/code/Starry-bqs/testcases/x86_64_ZLM# ./ffmpeg -re -i "./01.mp4" -vcodec h264 -acodec aac -f rtsp -rtsp_transport tcp rtsp://10.3.10.62:5555/live/test
ffmpeg version 4.2.7-0ubuntu0.1 Copyright (c) 2000-2022 the FFmpeg developers
  built with gcc 9 (Ubuntu 9.4.0-1ubuntu1~20.04.1)
  configuration: --prefix=/usr --extra-version=0ubuntu0.1 --toolchain=hardened --libdir=/usr/lib/x86_64-linux-gnu --incdir=/usr/include/x86_64-linux-gnu --arch=amd64 --enable-gpl --disable-stripping --enable-avresample --disable-filter=resample --enable-avisynth --enable-gnutls --enable-ladspa --enable-libaom --enable-libass --enable-libbluray --enable-libbs2b --enable-libcaca --enable-libcdio --enable-libcodec2 --enable-libflite --enable-libfontconfig --enable-libfreetype --enable-libfribidi --enable-libgme --enable-libgsm --enable-libjack --enable-libmp3lame --enable-libmysofa --enable-libopenjpeg --enable-libopenmpt --enable-libopus --enable-libpulse --enable-librsvg --enable-librubberband --enable-libshine --enable-libsnappy --enable-libsoxr --enable-libspeex --enable-libssh --enable-libtheora --enable-libtwolame --enable-libvidstab --enable-libvorbis --enable-libvpx --enable-libwavpack --enable-libwebp --enable-libx265 --enable-libxml2 --enable-libxvid --enable-libzmq --enable-libzvbi --enable-lv2 --enable-omx --enable-openal --enable-opencl --enable-opengl --enable-sdl2 --enable-libdc1394 --enable-libdrm --enable-libiec61883 --enable-nvenc --enable-chromaprint --enable-frei0r --enable-libx264 --enable-shared
  libavutil      56. 31.100 / 56. 31.100
  libavcodec     58. 54.100 / 58. 54.100
  libavformat    58. 29.100 / 58. 29.100
  libavdevice    58.  8.100 / 58.  8.100
  libavfilter     7. 57.100 /  7. 57.100
  libavresample   4.  0.  0 /  4.  0.  0
  libswscale      5.  5.100 /  5.  5.100
  libswresample   3.  5.100 /  3.  5.100
  libpostproc    55.  5.100 / 55.  5.100
Input #0, mov,mp4,m4a,3gp,3g2,mj2, from './01.mp4':
  Metadata:
    major_brand     : mp42
    minor_version   : 0
    compatible_brands: mp42mp41
    creation_time   : 2018-04-09T16:53:03.000000Z
  Duration: 00:00:26.79, start: 0.000000, bitrate: 764 kb/s
    Stream #0:0(eng): Video: h264 (Main) (avc1 / 0x31637661), yuv420p(tv, bt709), 550x750, 755 kb/s, 29.97 fps, 29.97 tbr, 30k tbn, 59.94 tbc (default)
    Metadata:
      creation_time   : 2018-04-09T16:53:03.000000Z
      handler_name    : ?Mainconcept Video Media Handler
      encoder         : AVC Coding
Stream mapping:
  Stream #0:0 -> #0:0 (h264 (native) -> h264 (libx264))
Press [q] to stop, [?] for help
[libx264 @ 0x564f40e40200] using cpu capabilities: MMX2 SSE2Fast SSSE3 SSE4.2 AVX FMA3 BMI2 AVX2
[libx264 @ 0x564f40e40200] profile High, level 3.1
[libx264 @ 0x564f40e40200] 264 - core 155 r2917 0a84d98 - H.264/MPEG-4 AVC codec - Copyleft 2003-2018 - http://www.videolan.org/x264.html - options: cabac=1 ref=3 deblock=1:0:0 analyse=0x3:0x113 me=hex subme=7 psy=1 psy_rd=1.00:0.00 mixed_ref=1 me_range=16 chroma_me=1 trellis=1 8x8dct=1 cqm=0 deadzone=21,11 fast_pskip=1 chroma_qp_offset=-2 threads=6 lookahead_threads=1 sliced_threads=0 nr=0 decimate=1 interlaced=0 bluray_compat=0 constrained_intra=0 bframes=3 b_pyramid=2 b_adapt=1 b_bias=0 direct=1 weightb=1 open_gop=0 weightp=2 keyint=250 keyint_min=25 scenecut=40 intra_refresh=0 rc_lookahead=40 rc=crf mbtree=1 crf=23.0 qcomp=0.60 qpmin=0 qpmax=69 qpstep=4 ip_ratio=1.40 aq=1:1.00
Output #0, rtsp, to 'rtsp://10.3.10.62:5555/live/test':
  Metadata:
    major_brand     : mp42
    minor_version   : 0
    compatible_brands: mp42mp41
    encoder         : Lavf58.29.100
    Stream #0:0(eng): Video: h264 (libx264), yuv420p, 550x750, q=-1--1, 29.97 fps, 90k tbn, 29.97 tbc (default)
    Metadata:
      creation_time   : 2018-04-09T16:53:03.000000Z
      handler_name    : ?Mainconcept Video Media Handler
      encoder         : Lavc58.54.100 libx264
    Side data:
      cpb: bitrate max/min/avg: 0/0/0 buffer size: 0 vbv_delay: -1
frame=  803 fps= 29 q=-1.0 size=N/A time=00:00:26.69 bitrate=N/A speed=0.973x
video:1444kB audio:0kB subtitle:0kB other streams:0kB global headers:0kB muxing overhead: unknown
[libx264 @ 0x564f40e40200] frame I:21    Avg QP:16.53  size:  5709
[libx264 @ 0x564f40e40200] frame P:216   Avg QP:18.91  size:  2832
[libx264 @ 0x564f40e40200] frame B:566   Avg QP:19.44  size:  1319
[libx264 @ 0x564f40e40200] consecutive B-frames:  3.7%  5.7%  3.4% 87.2%
[libx264 @ 0x564f40e40200] mb I  I16..4: 36.4% 60.5%  3.1%
[libx264 @ 0x564f40e40200] mb P  I16..4: 15.7% 22.4%  0.1%  P16..4: 30.1%  2.2%  0.7%  0.0%  0.0%    skip:28.8%
[libx264 @ 0x564f40e40200] mb B  I16..4:  2.3%  2.3%  0.0%  B16..8: 28.2%  1.3%  0.0%  direct: 6.5%  skip:59.5%  L0:48.8% L1:50.4% BI: 0.8%
[libx264 @ 0x564f40e40200] 8x8 transform intra:57.2% inter:99.6%
[libx264 @ 0x564f40e40200] coded y,uvDC,uvAC intra: 9.9% 31.5% 3.3% inter: 2.8% 15.7% 0.0%
[libx264 @ 0x564f40e40200] i16 v,h,dc,p: 33% 30% 17% 21%
[libx264 @ 0x564f40e40200] i8 v,h,dc,ddl,ddr,vr,hd,vl,hu: 22% 15% 60%  1%  0%  0%  0%  0%  0%
[libx264 @ 0x564f40e40200] i4 v,h,dc,ddl,ddr,vr,hd,vl,hu: 40% 24% 26%  4%  2%  1%  2%  1%  1%
[libx264 @ 0x564f40e40200] i8c dc,h,v,p: 62% 20% 15%  3%
[libx264 @ 0x564f40e40200] Weighted P-Frames: Y:4.6% UV:3.7%
[libx264 @ 0x564f40e40200] ref P L0: 67.2%  2.0% 28.5% 10.2%  0.1%
[libx264 @ 0x564f40e40200] ref B L0: 78.2% 18.5%  3.4%
[libx264 @ 0x564f40e40200] ref B L1: 88.7% 11.3%
[libx264 @ 0x564f40e40200] kb/s:441.36
root@josen:/home/josen/code/Starry-bqs/testcases/x86_64_ZLM#
```

```
1970-01-01 00:00:27.382 D [MediaServer] [11-event poller 0] MediaSink.cpp:162 emitAllTrackReady | All track ready use 6454ms
1970-01-01 00:00:27.871 I [MediaServer] [11-event poller 0] MediaSource.cpp:476 emitEvent | 媒体注册:fmp4://__defaultVhost__/live/test
1970-01-01 00:00:27.971 I [MediaServer] [11-event poller 0] MultiMediaSourceMuxer.cpp:562 onAllTrackReady | stream: rtsp://10.3.10.62:5555/live/test , codec info: mpeg4-generic[8000/1/16] H264[55
1970-01-01 00:00:28.061 I [MediaServer] [11-event poller 0] MediaSource.cpp:476 emitEvent | 媒体注册:rtmp://__defaultVhost__/live/test
1970-01-01 00:00:28.149 W [MediaServer] [11-event poller 0] EventPoller.cpp:255 async_l | take time: 51ms, thread may be overloaded
1970-01-01 00:00:28.150 I [MediaServer] [11-event poller 0] MediaSource.cpp:476 emitEvent | 媒体注册:ts://__defaultVhost__/live/test
1970-01-01 00:00:28.230 I [MediaServer] [11-event poller 0] MediaSource.cpp:476 emitEvent | 媒体注册:rtsp://__defaultVhost__/live/test
1970-01-01 00:00:28.618 I [MediaServer] [11-event poller 0] MediaSource.cpp:476 emitEvent | 媒体注册:hls://__defaultVhost__/live/test
1970-01-01 00:00:30.627 W [MediaServer] [11-event poller 0] Stamp.cpp:334 getNtpStampUS | rtp stamp abnormal reduced:39403826 → 38927216
[ 32.119267 0:15 linux_syscall_api::syscall_net::imp:448] sendmsg fd: 15, msg: MessageHeader { name: 0x0, name_len: 0, iovec: 0x1009c150, iovec_len: 1, control: 0x0, control_len: 0, flags: 0 }, f
[ 32.119511 0:15 linux_syscall_api::syscall_net::imp:449] msg.msg_name: 0x0, msg_name_len: 0
[ 32.119630 0:15 linux_syscall_api::syscall_net::socket:687] tcp socket sendto
[ 32.119750 0:15 axnet::smoltcp_impl::tcp:439] tcp send: [36, 1, 0, 32]
[ 32.119843 0:15 axnet::smoltcp_impl::tcp:453] tcp send len: 4
[ 32.119925 0:15 linux_syscall_api::syscall_net::imp:486] [sendmsg()] socket 15 sent 4 bytes to addr Endpoint { addr: Ipv4(Address([10, 3, 10, 62])), port: 5555 }
[ 32.120168 0:15 linux_syscall_api::syscall_net::imp:448] sendmsg fd: 15, msg: MessageHeader { name: 0x0, name_len: 0, iovec: 0x1009c150, iovec_len: 1, control: 0x0, control_len: 0, flags: 0 }, f
[ 32.120389 0:15 linux_syscall_api::syscall_net::imp:449] msg.msg_name: 0x0, msg_name_len: 0
[ 32.120506 0:15 linux_syscall_api::syscall_net::socket:687] tcp socket sendto
[ 32.120601 0:15 axnet::smoltcp_impl::tcp:439] tcp send: [129, 201, 0, 7, 52, 16, 246, 8, 52, 16, 246, 7, 0, 0, 0, 0, 0, 0, 8, 218, 0, 0, 20, 78, 91, 174, 105, 251, 0, 1, 133, 30]
[ 32.120798 0:15 axnet::smoltcp_impl::tcp:453] tcp send len: 32
[ 32.120882 0:15 linux_syscall_api::syscall_net::imp:486] [sendmsg()] socket 15 sent 32 bytes to addr Endpoint { addr: Ipv4(Address([10, 3, 10, 62])), port: 5555 }
[ 32.121127 0:15 linux_syscall_api::syscall_net::imp:448] sendmsg fd: 15, msg: MessageHeader { name: 0x0, name_len: 0, iovec: 0x10002900, iovec_len: 1, control: 0x0, control_len: 0, flags: 0 }, f
[ 32.121350 0:15 linux_syscall_api::syscall_net::imp:449] msg.msg_name: 0x0, msg_name_len: 0
[ 32.121462 0:15 linux_syscall_api::syscall_net::socket:687] tcp socket sendto
[ 32.121557 0:15 axnet::smoltcp_impl::tcp:439] tcp send: [36, 1, 0, 112]
[ 32.121652 0:15 axnet::smoltcp_impl::tcp:453] tcp send len: 4
[ 32.121734 0:15 linux_syscall_api::syscall_net::imp:486] [sendmsg()] socket 15 sent 4 bytes to addr Endpoint { addr: Ipv4(Address([10, 3, 10, 62])), port: 5555 }
[ 32.121947 0:15 linux_syscall_api::syscall_net::imp:448] sendmsg fd: 15, msg: MessageHeader { name: 0x0, name_len: 0, iovec: 0x10002900, iovec_len: 1, control: 0x0, control_len: 0, flags: 0 }, f
[ 32.122167 0:15 linux_syscall_api::syscall_net::imp:449] msg.msg_name: 0x0, msg_name_len: 0
[ 32.122282 0:15 linux_syscall_api::syscall_net::socket:687] tcp socket sendto
[ 32.122381 0:15 axnet::smoltcp_impl::tcp:439] tcp send: [129, 202, 0, 27, 52, 16, 246, 7, 1, 99, 90, 76, 77, 101, 100, 105, 97, 75, 105, 116, 40, 103, 105, 116, 32, 104, 97, 115, 104, 58, 54, 50
[ 32.122923 0:15 axnet::smoltcp_impl::tcp:453] tcp send len: 112
[ 32.123073 0:15 linux_syscall_api::syscall_net::imp:486] [sendmsg()] socket 15 sent 112 bytes to addr Endpoint { addr: Ipv4(Address([10, 3, 10, 62])), port: 5555 }
[ 36.733774 0:15 linux_syscall_api::syscall_net::imp:448] sendmsg fd: 15, msg: MessageHeader { name: 0x0, name_len: 0, iovec: 0x10005350, iovec_len: 1, control: 0x0, control_len: 0, flags: 0 }, f
[ 36.734061 0:15 linux_syscall_api::syscall_net::imp:449] msg.msg_name: 0x0, msg_name_len: 0
[ 36.734229 0:15 linux_syscall_api::syscall_net::socket:687] tcp socket sendto
[ 36.734397 0:15 axnet::smoltcp_impl::tcp:439] tcp send: [82, 84, 83, 80, 47, 49, 46, 48, 32, 50, 48, 48, 32, 79, 75, 13, 10, 67, 83, 101, 113, 58, 32, 53, 13, 10, 68, 97, 116, 101, 58, 32, 84,
[ 36.735394 0:15 axnet::smoltcp_impl::tcp:453] tcp send len: 197
[ 36.735477 0:15 linux_syscall_api::syscall_net::imp:486] [sendmsg()] socket 15 sent 197 bytes to addr Endpoint { addr: Ipv4(Address([10, 3, 10, 62])), port: 5555 }
1970-01-01 00:00:36.690 I [MediaServer] [11-event poller 0] MediaSource.cpp:476 emitEvent | 媒体注销:hls://__defaultVhost__/live/test
1970-01-01 00:00:36.701 I [MediaServer] [11-event poller 0] MediaSource.cpp:476 emitEvent | 媒体注销:ts://__defaultVhost__/live/test
1970-01-01 00:00:36.711 I [MediaServer] [11-event poller 0] MediaSource.cpp:476 emitEvent | 媒体注销:rtmp://__defaultVhost__/live/test
1970-01-01 00:00:36.713 I [MediaServer] [11-event poller 0] MediaSource.cpp:476 emitEvent | 媒体注销:fmp4://__defaultVhost__/live/test
1970-01-01 00:00:36.725 I [MediaServer] [11-event poller 0] MediaSource.cpp:476 emitEvent | 媒体注销:rtsp://__defaultVhost__/live/test
1970-01-01 00:00:37.007 W [MediaServer] [11-event poller 0] RtspSession.cpp:62 onError | 1-15(10.3.10.10:56234) RTSP推流器(__defaultVhost__/live/test)断开:recv teardown request,耗时(s):20

/ # ls
01.mp4              fionbio.c           log
MediaServer         info.txt            oom_openat
busybox             ioctl               oom_openat.c
client              ioctl.c             proc
client.c            lat_sig             sched_getscheduler.c
config.ini          lib                 server
dev                 lib64               server.c
downloads.sh        libc.so.6           sys
fcntl               libcrypto.so.3      testfile.txt
fcntl.c             libgcc_s.so.1       tmp
ffmpeg              libm.so.6           var
file.txt            libssl.so.3         www
fionbio             libstdc++.so.6
/ # cd www/live/test/
/www/live/test/ # ls
1970-01-01  hls.m3u8
/www/live/test/ # cd 1970-01-01/00/[ 46.731768 0:15 axruntime::lang_items:5] panicked at crates/linux_syscall_api/src/syscall_fs/imp/link.rs:78:53:
called `Result::unwrap()` on an `Err` value: NotFound
[ 46.732215 0:15 axbacktrace:43] Call trace:
[ 46.732366 0:15 axbacktrace::x86:91] 0xFFFFFF8000275941
[ 46.732542 0:15 axbacktrace::x86:91] 0xFFFFFF8000284F60
```



已初步实现推流，starry中存在推流后的文件

4. 目前仍需解决的问题

1. starry运行不稳定，并不是每次都可以成功推流

   原因：

   starry中zlm运行速度较慢，等它进入epoll之后再使用ffmpeg进行推流；

   还有一种是starry的抢占问题不够完善；

   其他不稳定的因素

2. 推流后直接退出了

   原因：starry仍然会去读取hls.m3u8，但是zlm的配置中设置为10s后删除www目录，因此link失败，将删除时间设置为-1，zlm不会去删除推流文件，不过后期还是要解决删除崩溃的问题

3. /www/live/test/ , hls.m3u8和hls_delay.m3u8是手动添加而不是自动生成的

5. 当前推流状态

推流比较稳定，接收视频文件后不会自动删除，之后需要将视频文件导出到本机，播放验证推流的正确性

```
1970-01-01 00:00:20.250 D [MediaServer] [9-MediaServer] System.cpp:148 startDaemon | 启动子进程:13
1970-01-01 00:00:20.267 I [MediaServer] [13-MediaServer] System.cpp:192 systemSetup | core文件大小设置为:18446744073709551615
1970-01-01 00:00:20.306 I [MediaServer] [13-MediaServer] System.cpp:201 systemSetup | 文件最大描述符个数设置为:18446744073709551615
1970-01-01 00:00:20.398 I [MediaServer] [13-MediaServer] main.cpp:256 start_main | ZLMediaKit(git hash:6889afb/2024-08-14T20:11:24+08:00,bran
1970-01-01 00:00:20.667 W [MediaServer] [13-MediaServer] SSLUtil.cpp:100 loadPublicKey | BIO_new_file failed: error:80000002:system library::
1970-01-01 00:00:20.670 W [MediaServer] [13-MediaServer] SSLUtil.cpp:128 loadPrivateKey | BIO_new_file failed: error:10000080:BIO routines::n
1970-01-01 00:00:20.672 W [MediaServer] [13-MediaServer] SSLUtil.cpp:202 makeSSLContext | SSL_CTX_check_private_key failed: error:80000002:sy
1970-01-01 00:00:20.836 D [MediaServer] [13-stamp thread] util.cpp:366 operator() | Stamp thread started
1970-01-01 00:00:21.043 I [MediaServer] [13-MediaServer] EventPoller.cpp:594 EventPollerPool | EventPoller created size: 1
1970-01-01 00:00:21.592 I [MediaServer] [13-MediaServer] main.cpp:360 start_main | 已启动http api 接口
1970-01-01 00:00:21.963 I [MediaServer] [13-MediaServer] main.cpp:362 start_main | 已启动http hook 接口
1970-01-01 00:00:22.142 W [MediaServer] [13-MediaServer] sockutil.cpp:442 bind_sock4 | inet_pton to ipv4 address failed: 0:0:0:0
1970-01-01 00:00:22.193 W [MediaServer] [13-MediaServer] EventPoller.cpp:255 async_l | take time: 82ms, thread may be overloaded
1970-01-01 00:00:22.192 W [MediaServer] [13-MediaServer] EventPoller.cpp:112 addEvent | take time: 82ms, thread may be overloaded
1970-01-01 00:00:22.243 I [MediaServer] [13-MediaServer] TcpServer.cpp:221 start_l | TCP server listening on [0:0:0:0]: 5555
1970-01-01 00:00:22.293 W [MediaServer] [13-MediaServer] EventPoller.cpp:255 async_l | take time: 90ms, thread may be overloaded
1970-01-01 00:00:22.342 W [MediaServer] [13-MediaServer] sockutil.cpp:442 bind_sock4 | inet_pton to ipv4 address failed: 0:0:0:0
1970-01-01 00:00:22.342 I [MediaServer] [13-MediaServer] TcpServer.cpp:221 start_l | TCP server listening on [0:0:0:0]: 1935
1970-01-01 00:00:22.384 W [MediaServer] [13-MediaServer] EventPoller.cpp:255 async_l | take time: 57ms, thread may be overloaded
1970-01-01 00:00:22.433 W [MediaServer] [13-MediaServer] sockutil.cpp:442 bind_sock4 | inet_pton to ipv4 address failed: 0:0:0:0
1970-01-01 00:00:22.433 I [MediaServer] [13-MediaServer] TcpServer.cpp:221 start_l | TCP server listening on [0:0:0:0]: 5525
1970-01-01 00:00:22.442 W [MediaServer] [13-MediaServer] sockutil.cpp:442 bind_sock4 | inet_pton to ipv4 address failed: 0:0:0:0
1970-01-01 00:00:22.442 I [MediaServer] [13-MediaServer] TcpServer.cpp:221 start_l | TCP server listening on [0:0:0:0]: 5554
1970-01-01 00:00:22.453 W [MediaServer] [13-MediaServer] sockutil.cpp:442 bind_sock4 | inet_pton to ipv4 address failed: 0:0:0:0
1970-01-01 00:00:22.497 W [MediaServer] [13-MediaServer] EventPoller.cpp:255 async_l | take time: 97ms, thread may be overloaded
1970-01-01 00:00:22.497 W [MediaServer] [13-MediaServer] EventPoller.cpp:112 addEvent | take time: 97ms, thread may be overloaded
1970-01-01 00:00:22.549 W [MediaServer] [13-MediaServer] sockutil.cpp:442 bind_sock4 | inet_pton to ipv4 address failed: 0:0:0:0
1970-01-01 00:00:22.549 W [MediaServer] [13-MediaServer] sockutil.cpp:442 bind_sock4 | inet_pton to ipv4 address failed: 0:0:0:0
1970-01-01 00:00:22.815 I [MediaServer] [13-MediaServer] UdpServer.cpp:123 start_l | UDP server bind to [0:0:0:0]: 10000
1970-01-01 00:00:22.858 W [MediaServer] [13-MediaServer] sockutil.cpp:442 bind_sock4 | inet_pton to ipv4 address failed: 0:0:0:0
1970-01-01 00:00:22.858 I [MediaServer] [13-MediaServer] TcpServer.cpp:221 start_l | TCP server listening on [0:0:0:0]: 10000
1970-01-01 00:00:22.859 W [MediaServer] [13-MediaServer] sockutil.cpp:442 bind_sock4 | inet_pton to ipv4 address failed: 0:0:0:0
1970-01-01 00:00:22.859 I [MediaServer] [13-MediaServer] UdpServer.cpp:123 start_l | UDP server bind to [0:0:0:0]: 9000
1970-01-01 00:00:31.990 W [MediaServer] [13-event poller 0] EventPoller.cpp:255 async_l | take time: 324ms, thread may be overloaded
1970-01-01 00:00:34.455 W [MediaServer] [13-event poller 0] EventPoller.cpp:255 async_l | take time: 1947ms, thread may be overloaded
1970-01-01 00:00:44.106 W [MediaServer] [13-event poller 0] Stamp.cpp:334 getNtpStampUS | rtp stamp abnormal reduced:1205587715 → 1204394111
1970-01-01 00:00:44.402 D [MediaServer] [13-event poller 0] MediaSink.cpp:162 emitAllTrackReady | All track ready use 8115ms
1970-01-01 00:00:44.922 I [MediaServer] [13-event poller 0] MediaSource.cpp:476 emitEvent | 媒体注册:fmp4://__defaultVhost__/live/test
1970-01-01 00:00:45.028 I [MediaServer] [13-event poller 0] MultiMediaSourceMuxer.cpp:562 onAllTrackReady | stream: rtsp://10.3.10.62:5555/li
1970-01-01 00:00:45.123 I [MediaServer] [13-event poller 0] MediaSource.cpp:476 emitEvent | 媒体注册:rtmp://__defaultVhost__/live/test
1970-01-01 00:00:45.217 I [MediaServer] [13-event poller 0] MediaSource.cpp:476 emitEvent | 媒体注册:ts://__defaultVhost__/live/test
1970-01-01 00:00:45.301 I [MediaServer] [13-event poller 0] MediaSource.cpp:476 emitEvent | 媒体注册:rtsp://__defaultVhost__/live/test
1970-01-01 00:00:45.706 I [MediaServer] [13-event poller 0] MediaSource.cpp:476 emitEvent | 媒体注册:hls://__defaultVhost__/live/test
1970-01-01 00:00:47.627 W [MediaServer] [13-event poller 0] Stamp.cpp:334 getNtpStampUS | rtp stamp abnormal reduced:1206039425 → 1205346062
1970-01-01 00:00:52.885 I [MediaServer] [13-event poller 0] MediaSource.cpp:476 emitEvent | 媒体注销:hls://__defaultVhost__/live/test
1970-01-01 00:00:52.896 I [MediaServer] [13-event poller 0] MediaSource.cpp:476 emitEvent | 媒体注销:ts://__defaultVhost__/live/test
1970-01-01 00:00:52.907 I [MediaServer] [13-event poller 0] MediaSource.cpp:476 emitEvent | 媒体注销:rtmp://__defaultVhost__/live/test
1970-01-01 00:00:52.909 I [MediaServer] [13-event poller 0] MediaSource.cpp:476 emitEvent | 媒体注销:fmp4://__defaultVhost__/live/test
1970-01-01 00:00:52.922 I [MediaServer] [13-event poller 0] MediaSource.cpp:476 emitEvent | 媒体注销:rtsp://__defaultVhost__/live/test
1970-01-01 00:00:53.212 W [MediaServer] [13-event poller 0] RtspSession.cpp:62 onError | 1-12(10.3.10.10:42358) RTSP推流器(__defaultVhost__/l
1970-01-01 00:00:53.212 W [MediaServer] [13-event poller 0] EventPoller.cpp:255 async_l | take time: 54ms, thread may be overloaded

/ # cd www/live/test/1970-01-01/00/
/www/live/test/1970-01-01/00/ # ls
00-00_10.ts  00-00_4.ts   00-00_6.ts   00-00_8.ts
00-00_11.ts  00-00_5.ts   00-00_7.ts   00-00_9.ts
/www/live/test/1970-01-01/00/ #
```

有几种思路：

- 比较优雅的方式：在 Starry 内部支持某种类似 scp 的指令，在运行时把文件拖出来（需要支持openssh，实现较为复杂，需要编译安装openssh，以及对openssh的系统调用支持）
- 更简单直接的方式：在播放完成后关机，此时文件会保存在文件系统镜像上。写一个脚本把它挂载到本机。就可以把文件复制出来了（注意将ramdisk改成virtio，把镜像 disk.img 挂载到本机直接导出即可）

  （最容易实现的方式，可行，但是在virtio模式下执行zlm运行速度太慢了）
- 创建共享文件夹，通过qemu创建虚拟机时加上以下参数

  `-virtfs local,path=/mnt/bqs,mount_tag=host0,security_model=passthrough,id=host0`

  此外还缺少系统调用`mknod`，用于创建特殊文件（如设备文件）。它创建字符设备文件或块设备文件，这些文件在 /dev 目录下用于与硬件设备进行交互

```
/ # busybox mount -t 9p -o trans=virtio host0 /mnt/shared
[232.191587 0:30 linux_syscall_api::syscall_fs::imp::mount:27] syscall_mount dev: [1073741345, 1073741333, 1073741367, 32768, 4688, 9259542123273814144]
[232.192014 0:30 linux_syscall_api::syscall_fs::imp::mount:30] syscall_mount mount: [1073741345, 1073741333, 1073741367, 32768, 4688, 9259542123273814144]
mount: permission denied (are you root?)
/ #
/ # busybox su root
su: unknown user root
/ #
```

- 使用socket把视频文件传送到外面机器上

  server.c

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#include <fcntl.h>
```

```
#define PORT 5555
#define BUFFER_SIZE 1024

int main() {
    int serverfd, clientfd, filefd;
    struct sockaddr_in servaddr, cliaddr;
    socklen_t len;
    char buffer[BUFFER_SIZE];
    ssize_t bytesRead;

    // 创建socket
    serverfd = socket(AF_INET, SOCK_STREAM, 0);
    if (serverfd < 0) {
        perror("socket creation failed");
        exit(EXIT_FAILURE);
    }

    // 服务器地址设置
    memset(&servaddr, 0, sizeof(servaddr));
    servaddr.sin_family = AF_INET;
    servaddr.sin_addr.s_addr = INADDR_ANY;
    servaddr.sin_port = htons(PORT);

    // 绑定socket
    if (bind(serverfd, (struct sockaddr *)&servaddr, sizeof(servaddr)) < 0)
{
        perror("bind failed");
        close(serverfd);
        exit(EXIT_FAILURE);
    }

    // 监听端口
    if (listen(serverfd, 5) < 0) {
        perror("listen failed");
        close(serverfd);
        exit(EXIT_FAILURE);
    }

    // 接受连接
    len = sizeof(cliaddr);
    clientfd = accept(serverfd, (struct sockaddr *)&cliaddr, &len);
    if (clientfd < 0) {
        perror("server accept failed");
        close(serverfd);
        exit(EXIT_FAILURE);
    }

    // 打开文件以写入接收到的数据
    filefd = open("/tmp/11-20_7.ts", O_WRONLY | O_CREAT | O_TRUNC, 0644);
    if (filefd < 0) {
        perror("file open failed");
        close(clientfd);
        close(serverfd);
        exit(EXIT_FAILURE);
```

```
    }

    // 从socket读取数据并写入文件
    while ((bytesRead = recv(clientfd, buffer, BUFFER_SIZE, 0)) > 0) {
        if (write(filefd, buffer, bytesRead) < 0) {
            perror("file write failed");
            close(filefd);
            close(clientfd);
            close(serverfd);
            exit(EXIT_FAILURE);
        }
    }

    if (bytesRead < 0) {
        perror("recv failed");
    }

    // 关闭文件和socket
    close(filefd);
    close(clientfd);
    close(serverfd);
    return 0;
}
```

client.c

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#include <sys/stat.h>
#include <fcntl.h>

#define PORT 5555
#define BUFFER_SIZE 1024

int main() {
    int sockfd, filefd;
    struct sockaddr_in servaddr;
    char buffer[BUFFER_SIZE];
    ssize_t bytesRead;

    // 创建socket
    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if (sockfd < 0) {
        perror("socket creation failed");
        exit(EXIT_FAILURE);
    }

    // 服务器地址设置
    memset(&servaddr, 0, sizeof(servaddr));
    servaddr.sin_family = AF_INET;
    servaddr.sin_port = htons(PORT);
```

```
      if (inet_pton(AF_INET, "10.3.10.10", &servaddr.sin_addr) <= 0) {
          perror("invalid address or address not supported");
          close(sockfd);
          exit(EXIT_FAILURE);
      }

      // 连接到服务器
      if (connect(sockfd, (struct sockaddr *)&servaddr, sizeof(servaddr)) <
0) {
          perror("connection failed");
          close(sockfd);
          exit(EXIT_FAILURE);
      }

      // 打开文件
      filefd = open("/www/live/test/1970-01-01/00/00-00_10.ts", O_RDONLY);
      if (filefd < 0) {
          perror("file open failed");
          close(sockfd);
          exit(EXIT_FAILURE);
      }

      // 读取文件并通过socket发送
      while ((bytesRead = read(filefd, buffer, BUFFER_SIZE)) > 0) {
          if (send(sockfd, buffer, bytesRead, 0) < 0) {
              perror("send failed");
              close(filefd);
              close(sockfd);
              exit(EXIT_FAILURE);
          }
      }

      if (bytesRead < 0) {
          perror("file read failed");
      }

      // 关闭文件和socket
      close(filefd);
      close(sockfd);
      return 0;
  }
```