

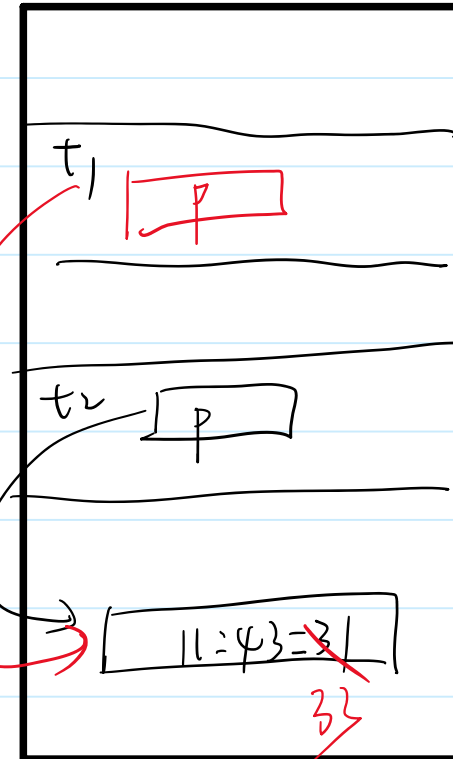
# 线程安全

2023年8月18日

9:27

```
void * threadFunc(void *arg){  
    time_t now = time(NULL);  
    char *p = ctime(&now); ← 线程安全  
    printf("before child, time = %s\n", p); → 42:00  
    sleep(5);  
    printf("after child, time = %s\n", p); → 42:00  
}
```

```
int main(int argc, char *argv[])  
{  
    pthread_t tid;  
    pthread_create(&tid, NULL, threadFunc, NULL);  
    sleep(2);  
    time_t now = time(NULL);  
    char *p = ctime(&now);  
    printf("main, time = %s\n", p); → 42:02  
    pthread_join(tid, NULL);  
    return 0;  
}
```



11:40.LinuxDay20p ./2\_thread\_safe

before child, time = Fri Aug 18 11:43:31 2023

main, time = Fri Aug 18 11:43:33 2023

after child, time = Fri Aug 18 11:43:33 2023

# 线程安全的版本

2023年8月18日 11:49

```
char *ctime(const time_t *timep);  
char *ctime_r(const time_t *timep, char *buf);
```

```
void * threadFunc(void *arg){  
    time_t now = time(NULL);  
    char buf[4096] = {0};  
    char *p = ctime_r(&now,buf);  
    printf("before child, time = %s\n", p);  
    sleep(5);  
    printf("after child, time = %s\n", p);  
}  
int main(int argc, char *argv[])  
{  
    pthread_t tid;  
    pthread_create(&tid,NULL,threadFunc,NULL);  
    sleep(2);  
    time_t now = time(NULL);  
    char buf[4096] = {0};  
    char *p = ctime_r(&now,buf);  
    printf("main, time = %s\n", p);  
    pthread_join(tid,NULL);  
    return 0;  
}
```

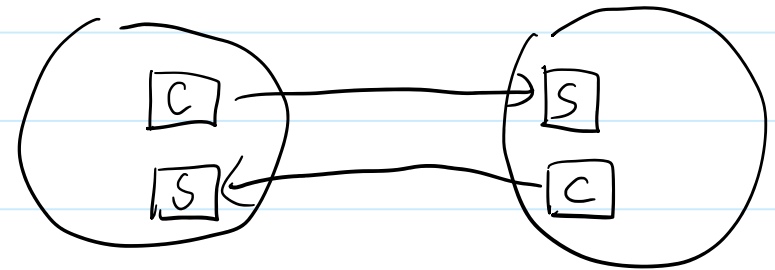
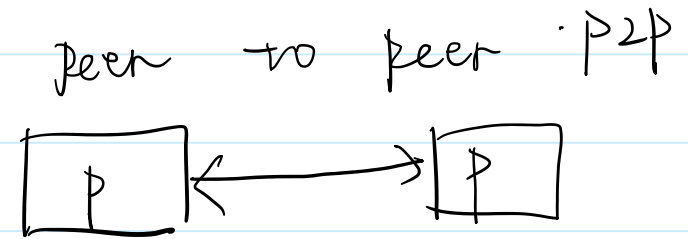
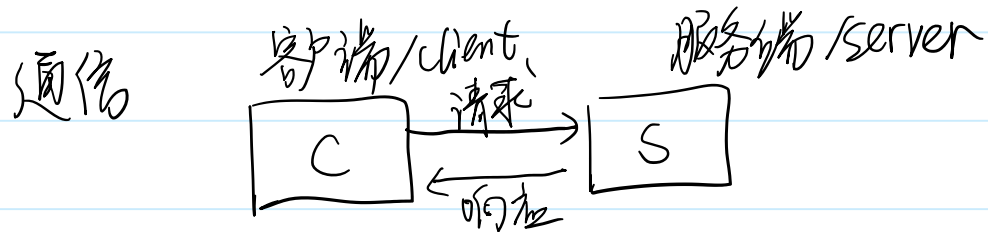
Interface	Attribute	Value
strtok()	Thread safety	MT-Unsafe race:strtok
strtok_r()	Thread safety	MT-Safe



# 网络理论

2023年8月18日 14:30

多量多次



ARPA → Internet → Apple  
Web时代. Web2.0

# 分层

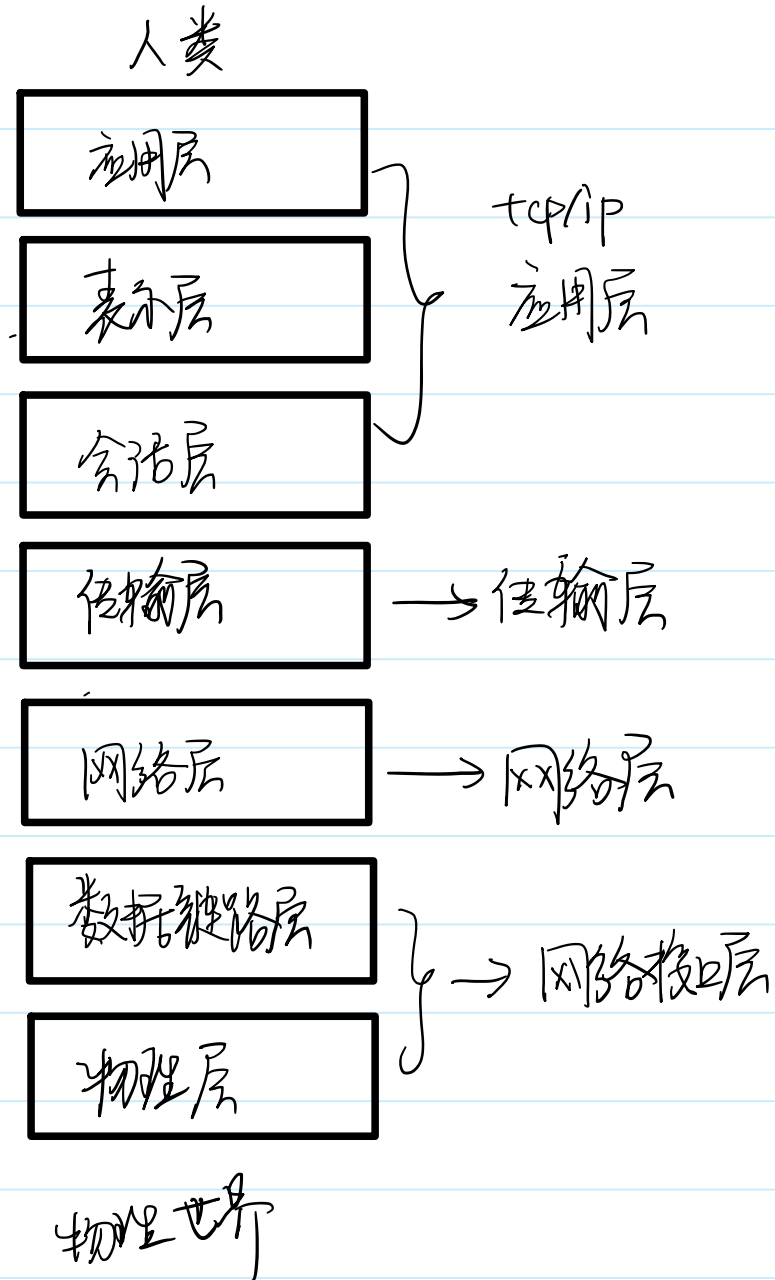
2023年8月18日

14:46

理论上 ISO/OSI 11分层 7层

实践上 TCP/IP 协议族 4层

{ 上层使用下层提供的接口  
下层实现接口.



# 协议

2023年8月18日

14:59

对等实体 关于通信内容的约定  
△△

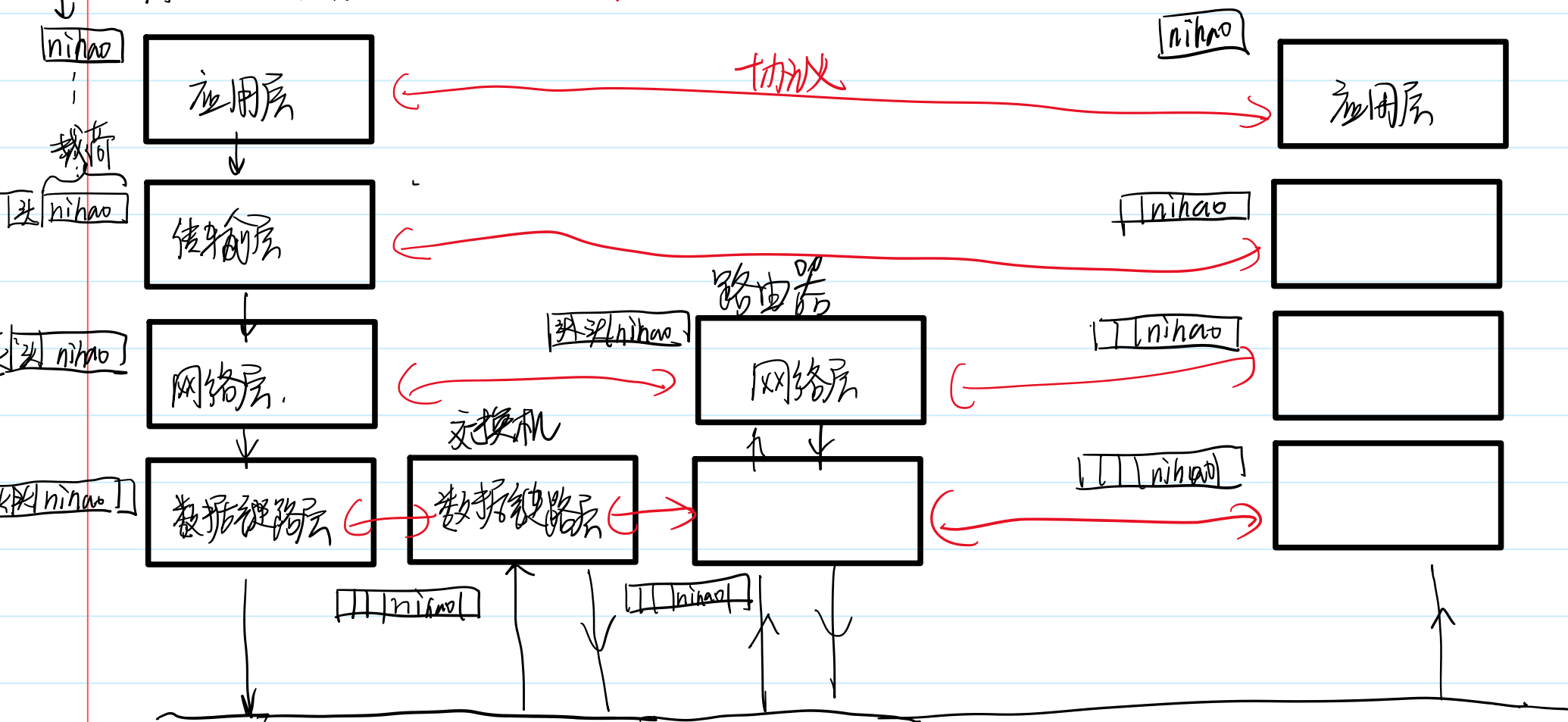
0x8869  
A

0x6869

short  
char [2]

2023年8月18日 15:04

协议数据单元 (pdus)  
↓ A "nihao"



# 应用层有哪些协议

2023年8月18日

15:20

tcp/ip协议族

{	http	超文本协议	明文	ftp
{	https	http + <u>TLS/SSL</u>	密文	pop3/smtp
				imap
				HLS
{	ssh	安全外壳	密文	:
		shell		:
{	telnet	明文		:

公开协议

<https://www.rfc-editor.org/>

<<tcp/ip详解 卷一>>

私有协议-

① 公司内部

② 游戏 安全 延迟 弱网

③ 金融



# 内核协议栈 ← OS内核的一部分

2023年8月18日

16:03

传输层 ~~tcp~~ "打电话" 有连接 可靠性.

udp "短信" 无连接 不可靠

sc-p 音视频...

网络层 ~~ip~~

ICMP/ICMP

数据链路层 ARP.

内核协议栈

# 地址

2023年8月18日 16:10

数据链路层 局域网 MAC地址/物理地址. 48bit.  $2^{48}$

ipconfig /all

```
以太网适配器 以太网:

    连接特定的 DNS 后缀 . . . . . : 
    描述. . . . . : Realtek PCIe GbE Family Controller
    物理地址. . . . . : E8-6A-64-76-C7-D0
    DHCP 已启用 . . . . . : 是
    自动配置已启用. . . . . : 是
    本地连接 IPv6 地址. . . . . : fe80::c6e2:2126:7cbc:a505%4(首选)
    IPv4 地址 . . . . . : 192.168.11.72(首选)
    子网掩码 . . . . . : 255.255.255.0
    获得租约的时间 . . . . . : 2023年8月16日 23:38:23
    租约过期的时间 . . . . . : 2023年8月21日 11:38:23
    默认网关. . . . . : 192.168.11.1
    DHCP 服务器 . . . . . : 192.168.11.1
    DHCPv6 IAID . . . . . : 216558180
    DHCPv6 客户端 DUID . . . . . : 00-01-00-01-23-A4-35-FD-E8-6A-64-76-C7-D0
    DNS 服务器 . . . . . : 202.103.24.68
                           202.103.44.150
    TCP/IP 上的 NetBIOS . . . . . : 已启用
```

ip地址 视在互联网中的位置.

IPv4 32bit  $2^{32}$  NAT  
IPv6 128bit  $2^{128}$  -

# 端口号

2023年8月18日

16:19

传输层.

区别: 一个主机内不同的进程 16bit  $0 \sim 65535$ .

$0 \sim 1023$  知名端口号

80. Http

443 https

22 ssh

# Linux下查看网卡信息

2023年8月18日 16:13

ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
inet 192.168.118.128 netmask 255.255.255.0 broadcast 192.168.118.255  
inet6 fe80::1e76:f607:177f:b610 prefixlen 64 scopeid 0x20<link>  
ether 00:0c:29:bf:f5:c8 txqueuelen 1000 (Ethernet)  
RX packets 163531 bytes 151600759 (151.6 MB)  
RX errors 0 dropped 0 overruns 0 frame 0  
TX packets 97985 bytes 46241281 (46.2 MB)  
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

↑  
↓  
外网网卡

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
inet 127.0.0.1 netmask 255.0.0.0  
inet6 ::1 prefixlen 128 scopeid 0x10<host>  
loop txqueuelen 1000 (Local Loopback)  
RX packets 51175 bytes 50124480 (50.1 MB)  
RX errors 0 dropped 0 overruns 0 frame 0  
TX packets 51175 bytes 50124480 (50.1 MB)  
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

↑  
loopback

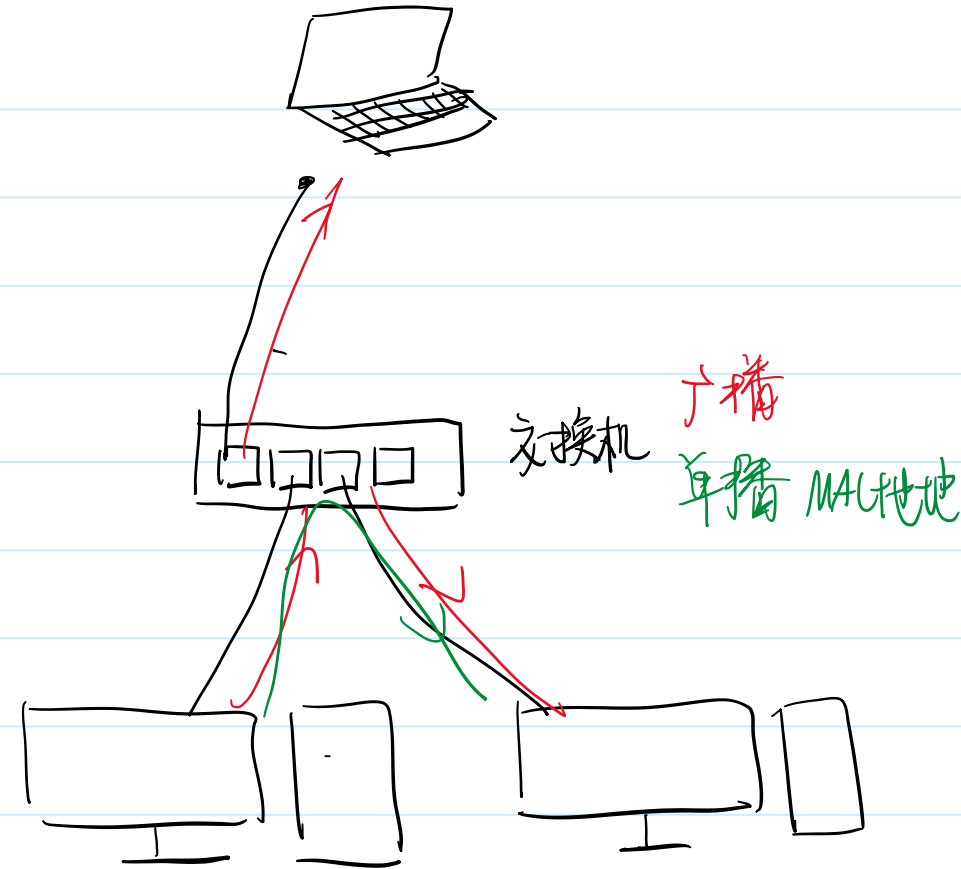
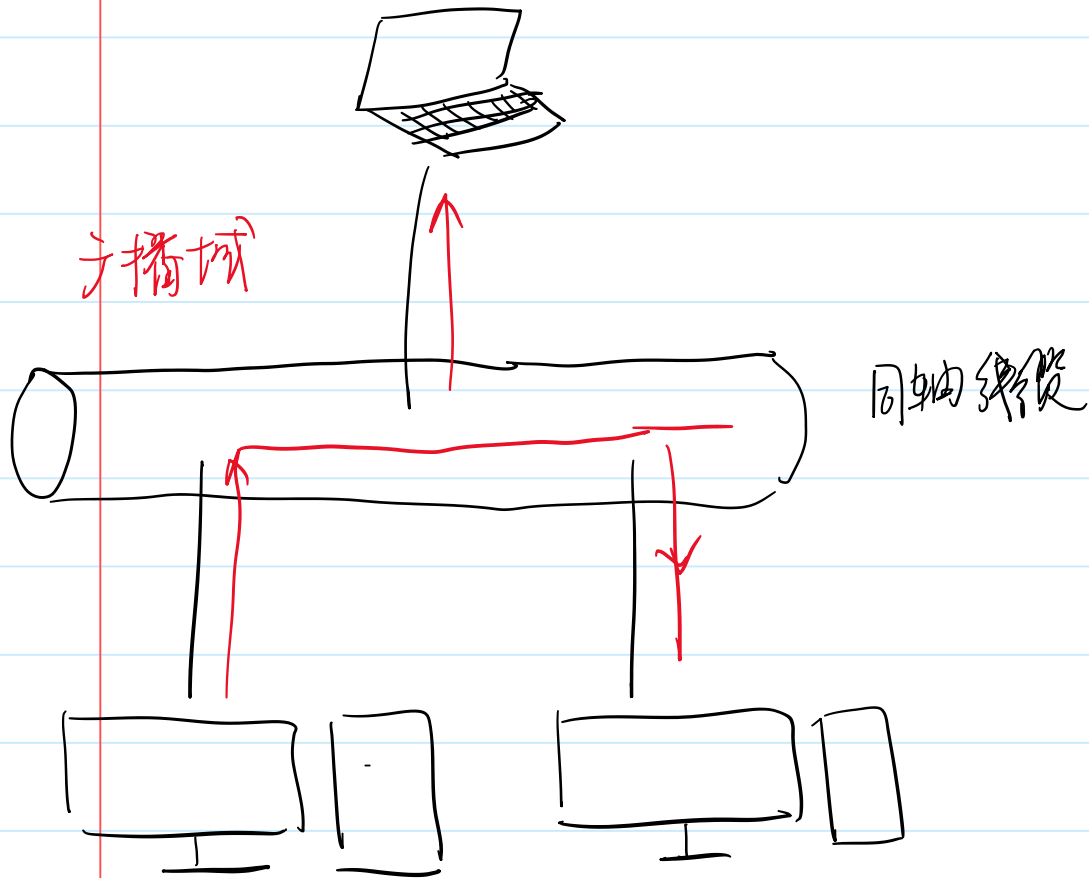
环回设备

C和S都位于同一台虚拟机。

# 数据链路层

2023年8月18日 16:27

局域网 以太网



# 以太网的协议数据单元

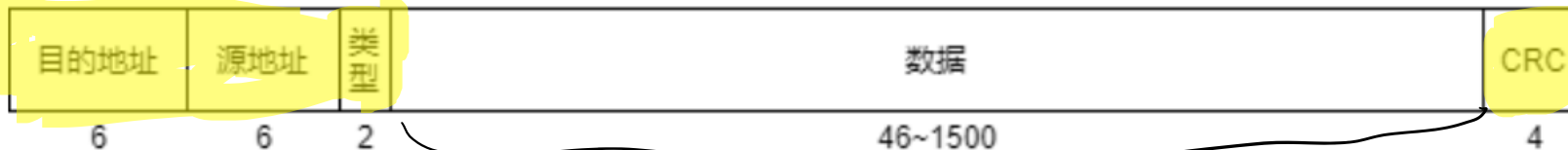
2023年8月18日

16:36

帧

MTU. 最大传输单元.

以太网帧的一般格式: RFC 894



帧头

载荷

一般放网络层 pdu.

哈希算法.

CRC

MD

SHA



效率低

碰撞概率低

# ARP 地址解析协议

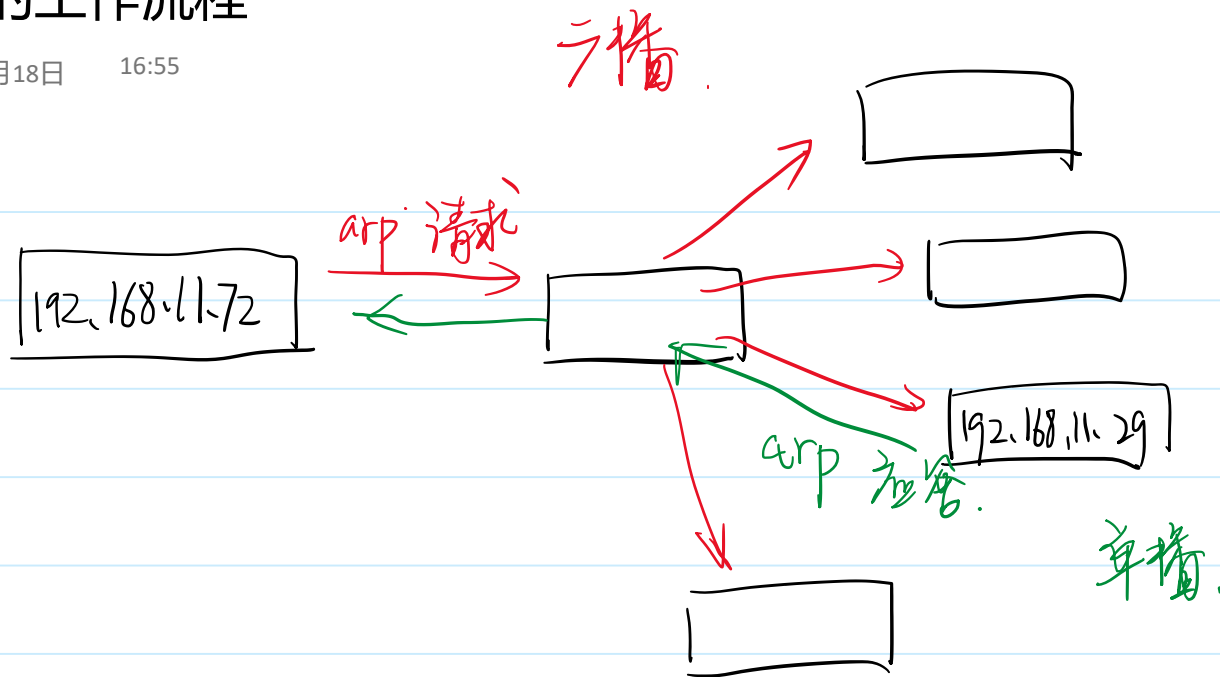
2023年8月18日 16:44

MAC地址 计算机自带的  
IP地址 人定的  
根据IP地址 → MAC地址

- ✓ Ethernet II, Src: e8:6a:64:76:c7:d0 (e8:6a:64:76:c7:d0), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  - > Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  - > Source: e8:6a:64:76:c7:d0 (e8:6a:64:76:c7:d0)
  - Type: ARP (0x0806)
- ✓ Address Resolution Protocol (request)
  - Hardware type: Ethernet (1)
  - Protocol type: IPv4 (0x0800)
  - Hardware size: 6
  - Protocol size: 4
  - Opcode: request (1)
  - Sender MAC address: e8:6a:64:76:c7:d0 (e8:6a:64:76:c7:d0)
  - Sender IP address: 192.168.11.72
  - Target MAC address: 00:00:00\_00:00:00 (00:00:00:00:00:00)
  - Target IP address: 192.168.11.29

# arp的工作流程

2023年8月18日 16:55



- ▼ Ethernet II, Src: 98:fa:9b:7f:68:60 (98:fa:9b:7f:68:60), Dst: e8:6a:64:76:c7:d0 (e8:6a:64:76:c7:d0)
  - > Destination: e8:6a:64:76:c7:d0 (e8:6a:64:76:c7:d0)
  - > Source: 98:fa:9b:7f:68:60 (98:fa:9b:7f:68:60)
  - Type: ARP (0x0806)
  - Padding: 00000000000000000000000000000000
- ▼ Address Resolution Protocol (reply)
  - Hardware type: Ethernet (1)
  - Protocol type: IPv4 (0x0800)
  - Hardware size: 6
  - Protocol size: 4
  - Opcode: reply (2)
  - Sender MAC address: 98:fa:9b:7f:68:60 (98:fa:9b:7f:68:60)
  - Sender IP address: 192.168.11.29
  - Target MAC address: e8:6a:64:76:c7:d0 (e8:6a:64:76:c7:d0)
  - Target IP address: 192.168.11.72



# wireshark

2023年8月18日 17:17

抓包

分析

Ncap 抓包 (Windows) → Wireshark

tcpdump (Linux) → Wireshark

① 选择网卡

以太网	有线
WLAN	无线
VMnet8	虚拟机

② 开始/停止抓包

③ 过滤

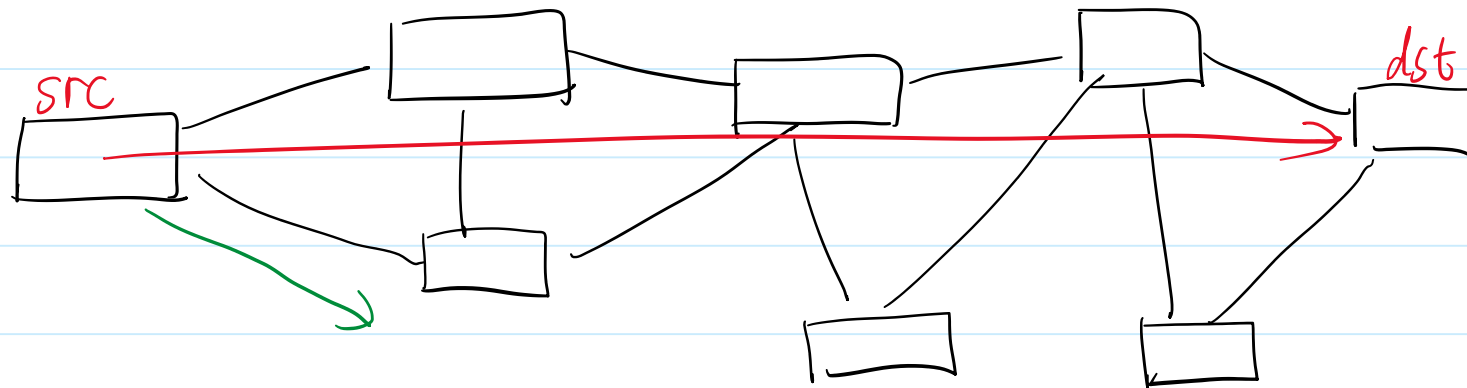
arp

ip.addr == 192.168.11.72

# 网络层 IP层

2023年8月18日

17:27



路由.

网络底层只能做逐跳传输  
△△

① 没有全局, 尽量保证下一跳更接近终端

② 丢包了不管, 不可靠

③ 可能会后发失至, 乱序.

# ip地址

2023年8月18日 17:37

```
Source: 192.168.11.37
Destination: 192.168.11.72
Transmission Control Protocol, Src Port: 57409,
0 e8 6a 64 76 c7 d0 00 e0 4c 36 05 f2 08 00
0 00 28 14 70 40 00 80 06 4e a2 c0 a8 0b 25
```

32bit ip地址

C0 a8 0b 25

"192.168.11.37" 人类阅读 二进制

255台机器  
子网.  
24bit 8bit  
子网号 主机号

子网掩码 →

192.168.11.72/24

ip地址  
1 - - - - - 0 - - - - 0  
255-255-255 10  
子网号 0 - - - 0

# NAT技术

2023年8月18日

17:50

缓解ip地址不足, 内网不能做服务端

