



# uCOS-II Report (3<sub>---porting lwIP</sub>)

---

Yihe.Chen

[yihect@gmail.com](mailto:yihect@gmail.com)



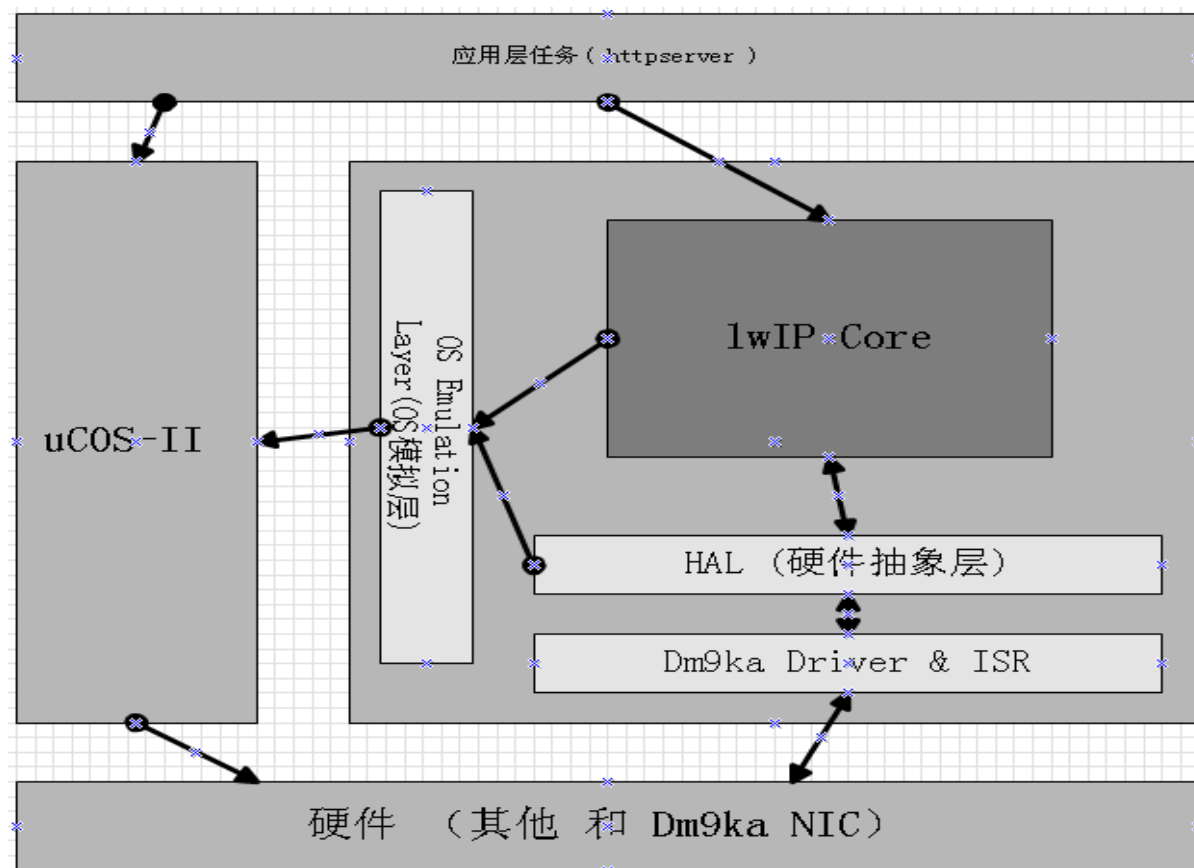
# 内容

---

- Overall structure
- The OS emulation layer
- Packet buffers
- Network Interface
- The HAL layer & Dm9k Nic Driver
- Interfacing the lwIP stack
- Q & A

# Overall structure

## ■ Overall structure





# The OS Emulation Layer

---

- Provide Timer for lwIP
  - 为TCP部分提供定时服务
  - 对每一种链路层，lwIP都会有一个lwIP任务与其对应，对这些任务都应该提供一个Timer。我们只要以太网类型。
- Provide Semaphores as copro\_sync\_mech
- Provide Mailbox as msg\_passing\_mech
- See sys\_arch.c/.h



# The Packet buffers

---

- Usage
  - Allocating dynamic memory to hold packet contents
  - Let packet data reside in static memory

# The Packet buffers

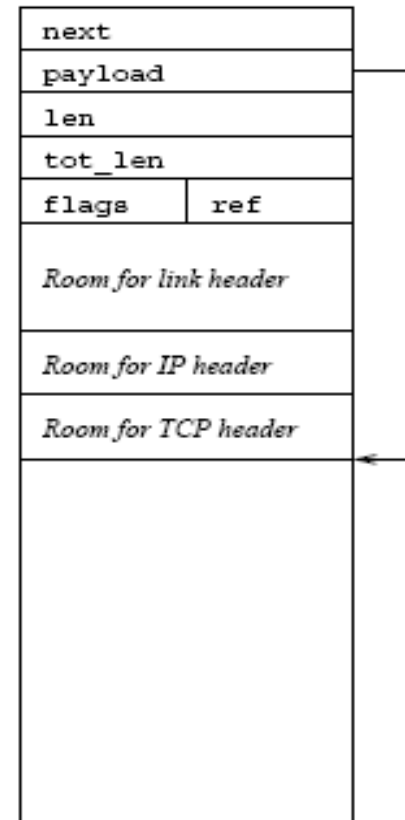
## ■ Three types

### ■ PBUF\_RAM

a, 为各种包头内容准备存储空间;

b, 为动态产生的待发送数据准备存储空间

c, 一般用于发送



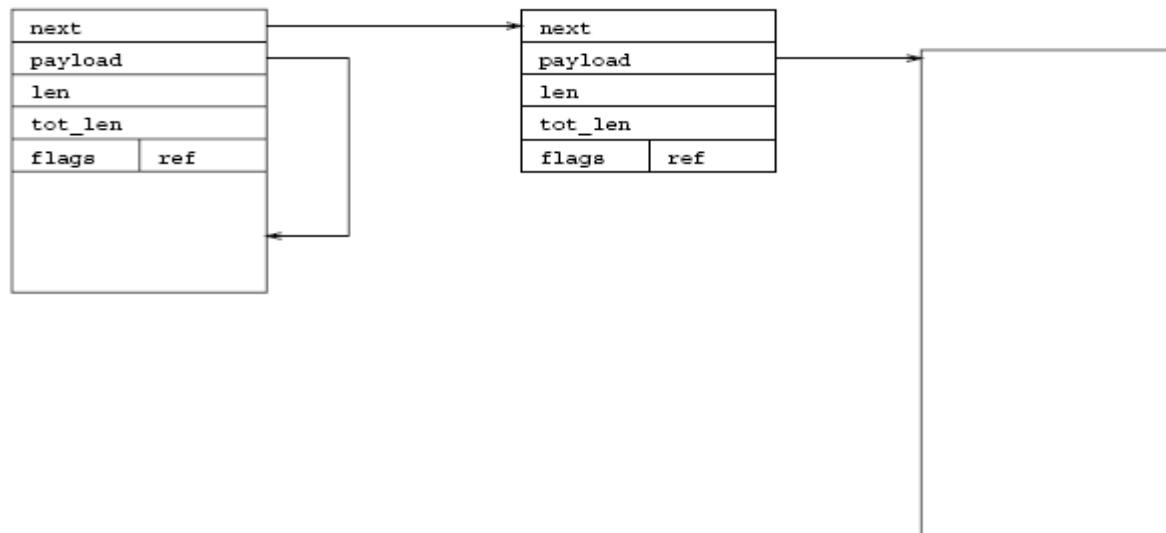
# The Packet buffers

- Three types

- PBUF\_ROM

a, 为只读的待发送数据准备存储空间；

b, 一般用于发送



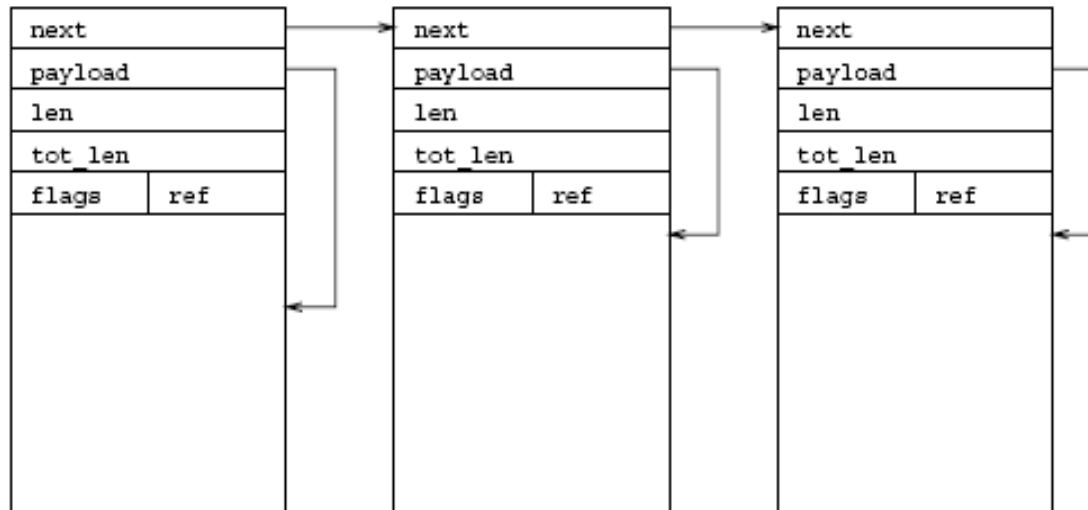
# The Packet buffers

- Three types

- PBUF\_POOL

- a, 在Nic Driver中使用

- b, 一般用于接收





# Network Interface

## ■ netif structure

```
struct netif {
    struct netif *next;

    struct ip_addr ip_addr;
    struct ip_addr netmask;
    struct ip_addr gw;

    err_t (* input) (struct pbuf *p, struct netif *inp);
    err_t (* output) (struct netif *netif, struct pbuf *p,
        struct ip_addr *ipaddr);
    err_t (* linkoutput) (struct netif *netif, struct pbuf *p);

    void *state;

    u8_t hwaddr_len;
    u8_t hwaddr[NETIF_MAX_HWADDR_LEN];
    u16_t mtu;
    u8_t flags;
    char name[2];
    u8_t num;
};
```

tcpip\_input()  
in Tcpip.c

etharp\_output()  
in etharp.c

low\_level\_output()  
in ethernetif.c(HAL)

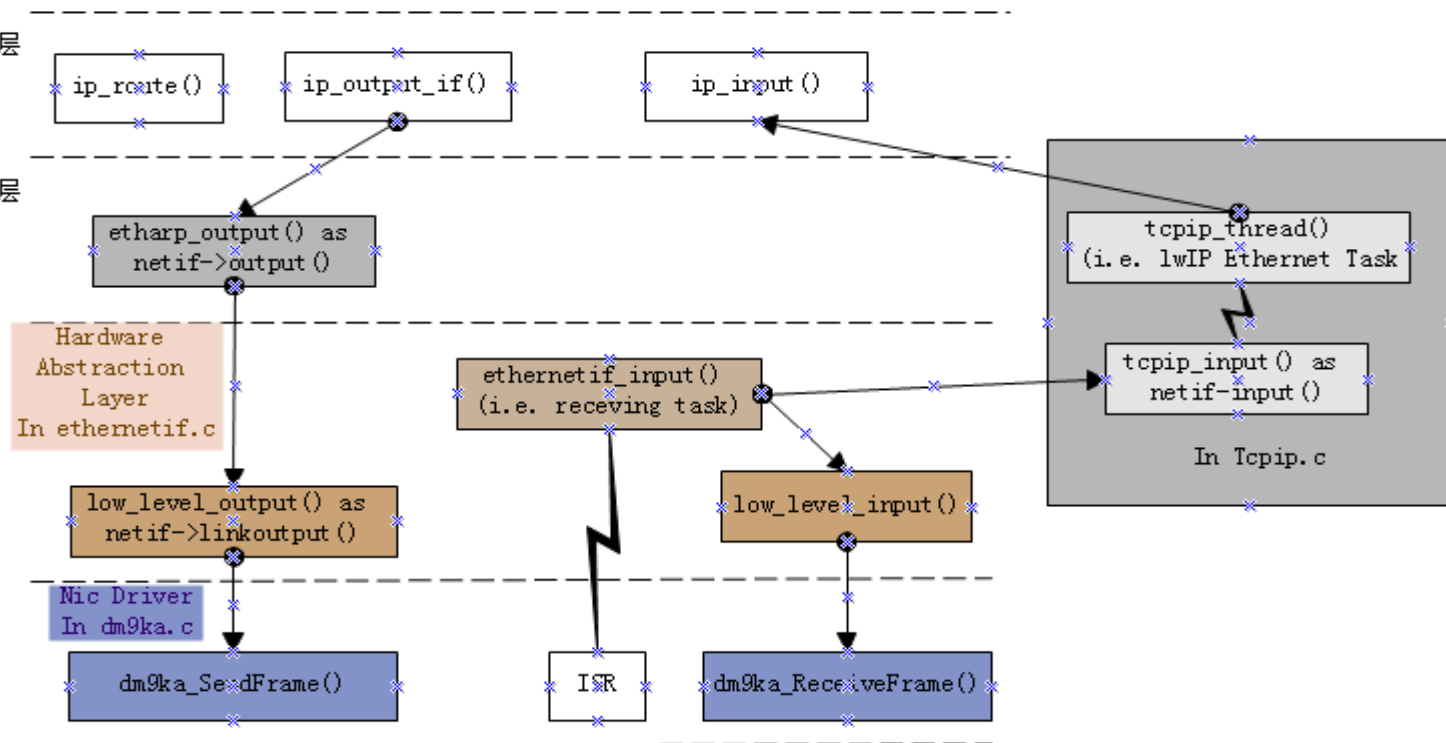
# The HAL layer & Dm9k Driver

## ■ Structure

传输层

网络层

接口层



硬件层



# Interfacing the lwIP stack

---

- **Raw/Native API**
  - 依赖于Callbacks；
  - APP 必须 和 lwIP 在同一个任务内，能实现 Zero-copy；
  - 难于理解和编码；
  - 在不用OS时，这是唯一适用的API，但在有OS支持的情况下，不推荐使用



# Interfacing the lwIP stack

---

## ■ Sequential API

- 类似于BSD Socket API ,易于使用；
- APP 和 lwIP stack 在两个任务内，他们用由 OSEL提供的消息机制进行通讯。也能实现 Zero-copy;
- APP 通过 netconn/netbuf 使用lwIP 提供的服务。
- 在OS下推荐使用



# Interfacing the lwIP stack

---

- **Socket API**

- 通过lwIP API 实现；
- 已有网络程序基本上是Socket API写的，lwIP提供同样的接口，便于移植



**Q & A**

---

**THE END**

**Thanks a lot, any suggestion?**