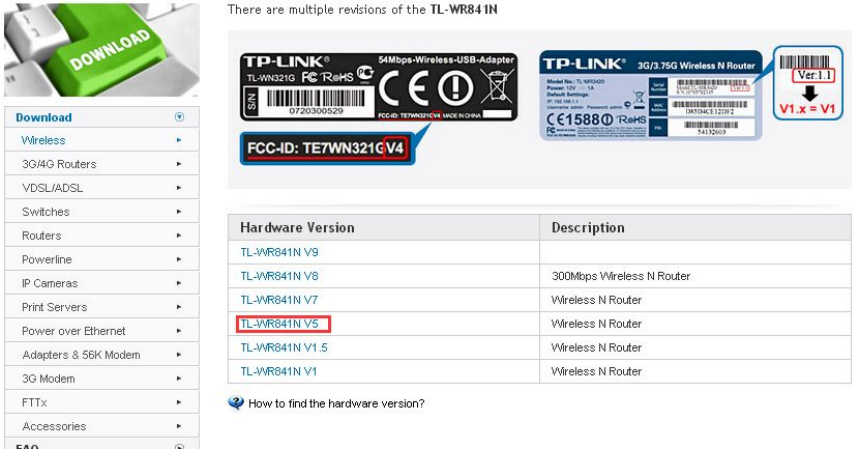
# Projeto KIntercept

Jonathan Santos Cunha
Carolina Santana Louzada

Disciplina: Interface Hardware/Software
Prof.: Bruno Otávio Piedade Prado

# Ideia

- Modificar o firmware de um roteador



# Obstáculos iniciais

- Encontrar código de firmware disponível, com documentação.
- Saber o que de fato seria útil extender ou modificar no firmware no espaço do kernel.

# O que foi decidido?

- Criar um módulo para o kernel de um firmware baseado em Linux chamado OpenWrt

# Quais ferramentas/plataformas foram utilizadas?

1. VSCode
2. OpenWRT SDK - > Linux 4.14.131
3. GNS3
    a. QEMU
    b. Docker para VMS
    c. Outras appliances

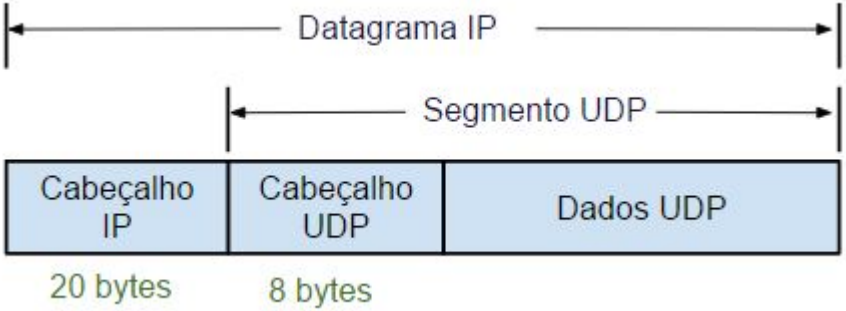
# Qual a funcionalidade desse módulo?

- Interceptar pacotes e modificar seu conteúdo (payload).

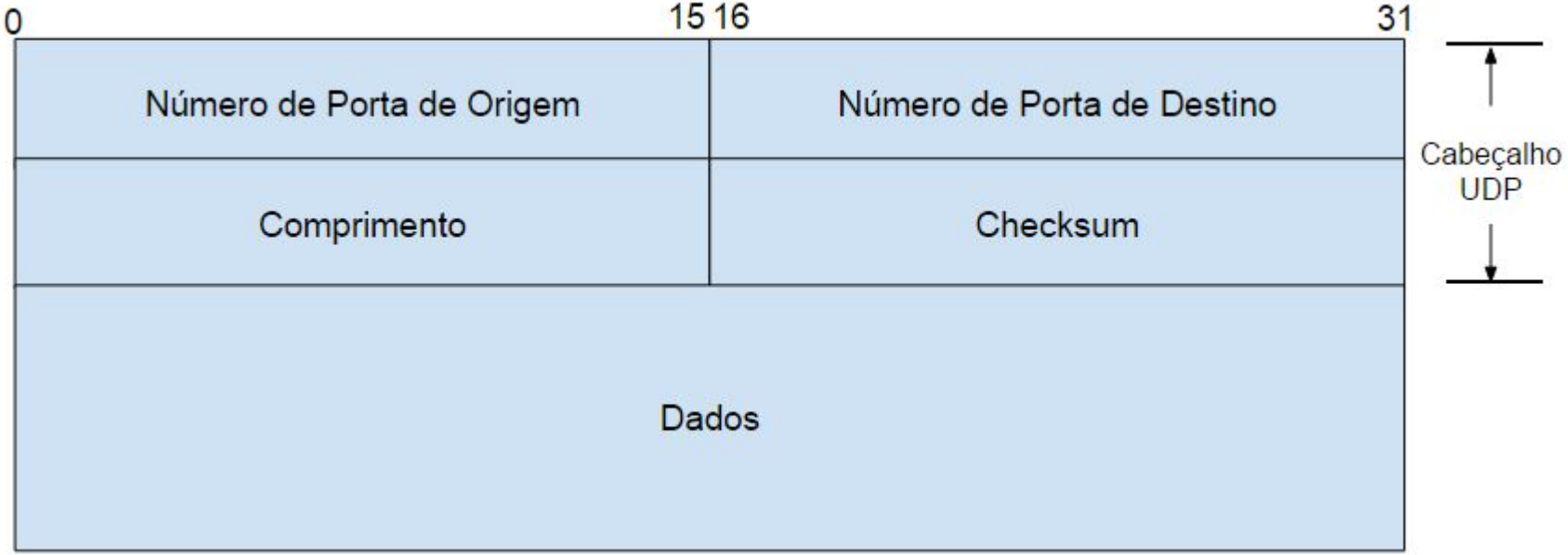
# Conceitos necessários para o desenvolvimento do módulo

- Interface entre roteador e firmware
- Noções sobre redes de computadores e configuração
- Conhecimento sobre compilação do Kernel do Linux -> Gerenciamento de dispositivos

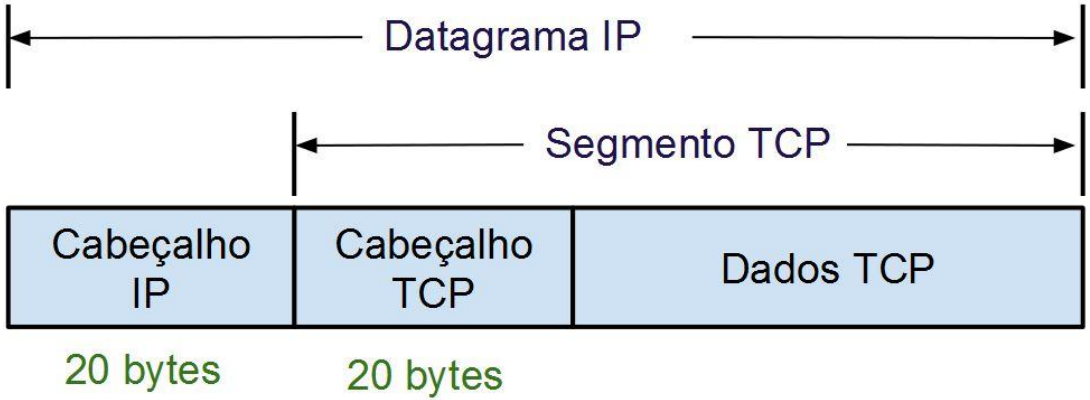
# Protocolo UDP



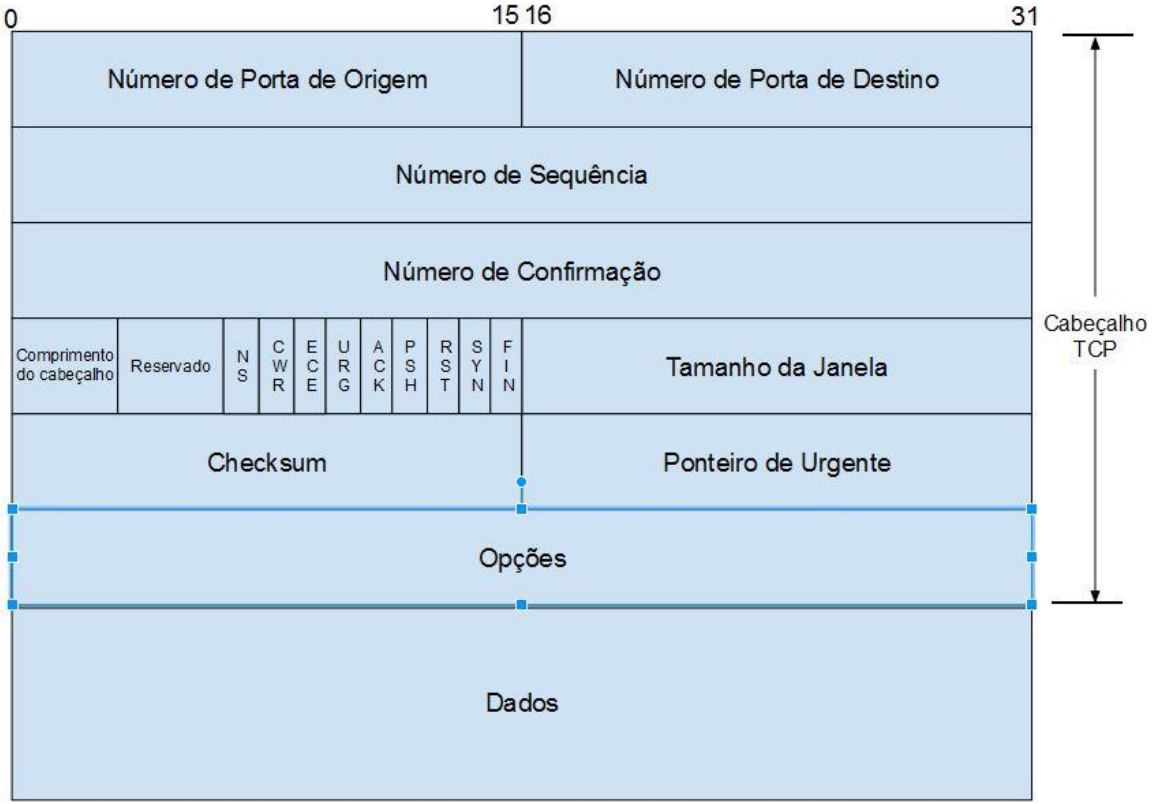
# Cabeçalho UDP + payload



# Protocolo TCP



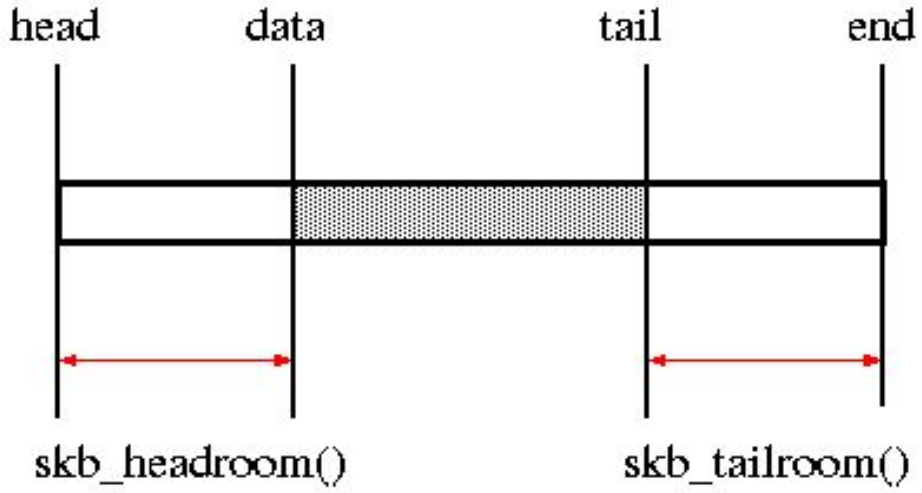
# Cabeçalho TCP + payload



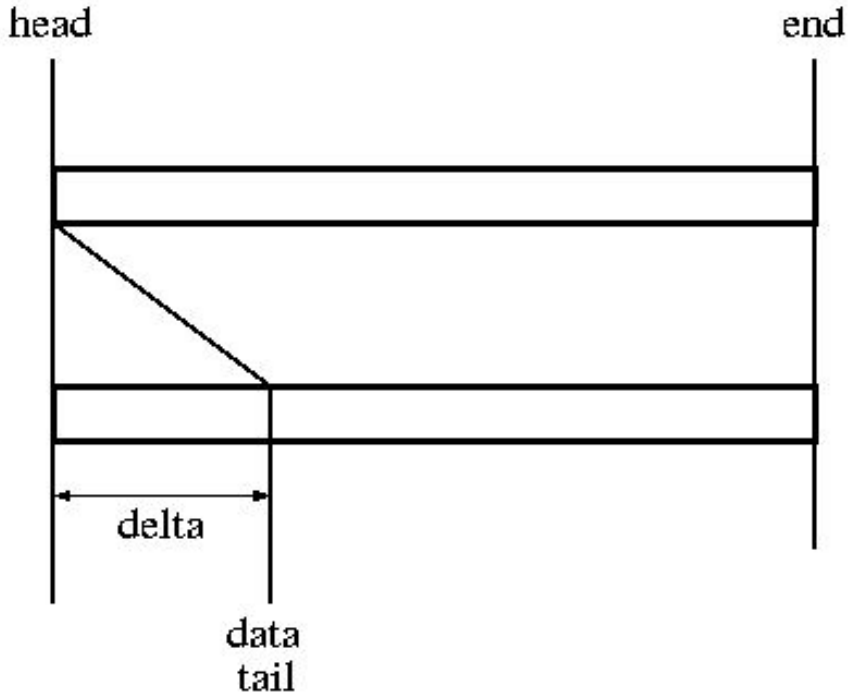
# Utilizando NetFilter

```
struct nf_hook_ops {
        /* User fills in from here down. */
        nf_hookfn               *hook;
        struct net_device       *dev;
        void                    *priv;
        u_int8_t        pf;
        unsigned int            hooknum;
        /* Hooks are ordered in ascending priority. */
        int                     priority;
};
```

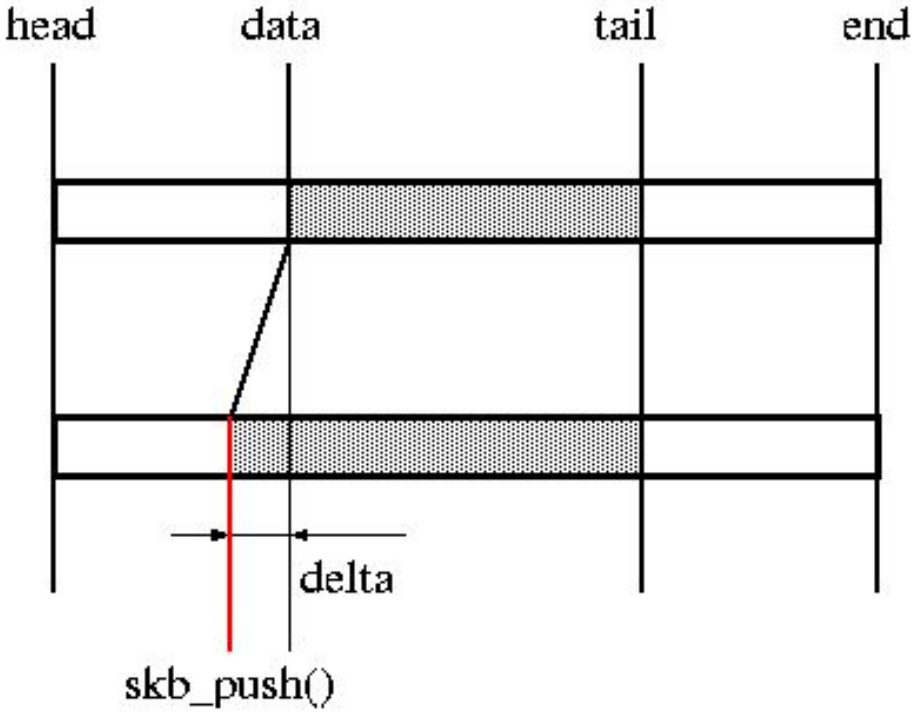
# Utilizando sk_buff para lidar com os pacotes



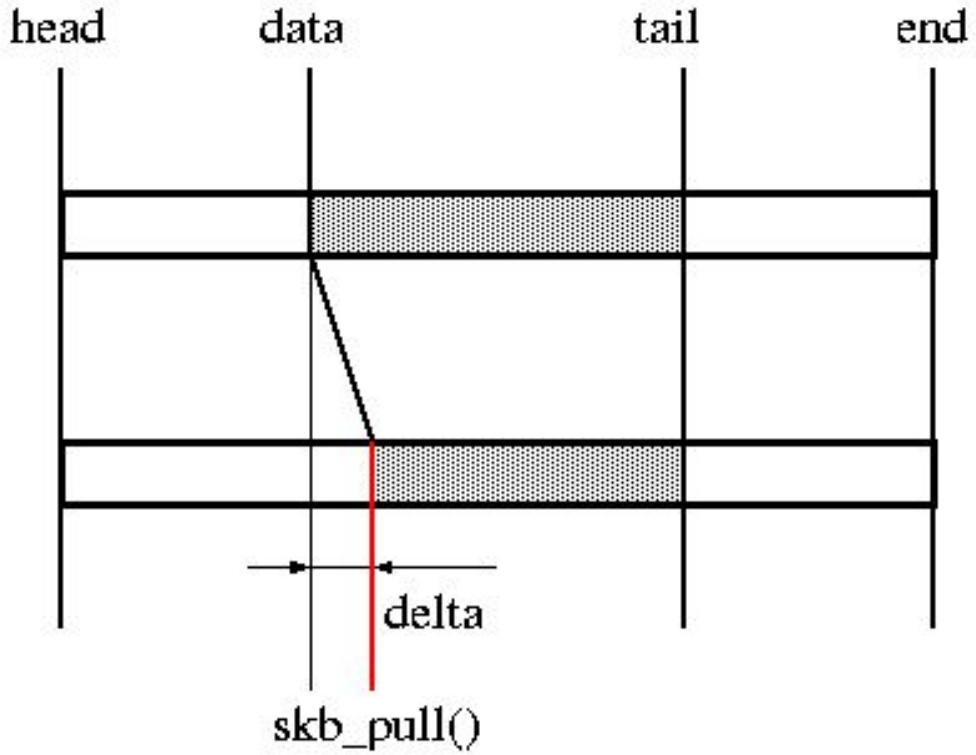
# skb_reserve( )



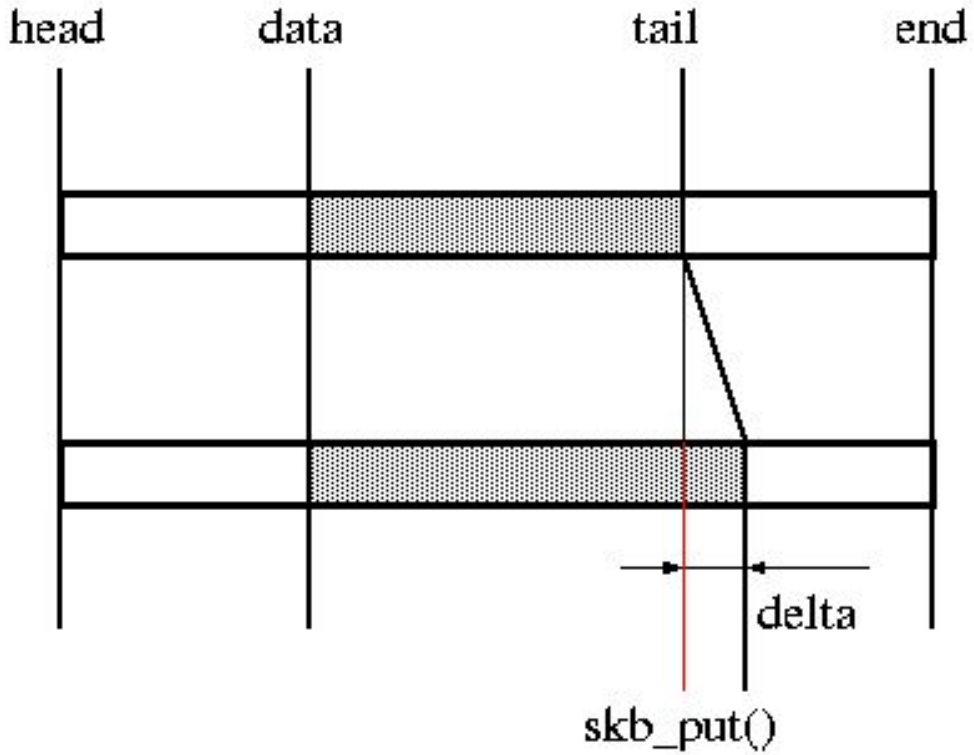
# skb_push()



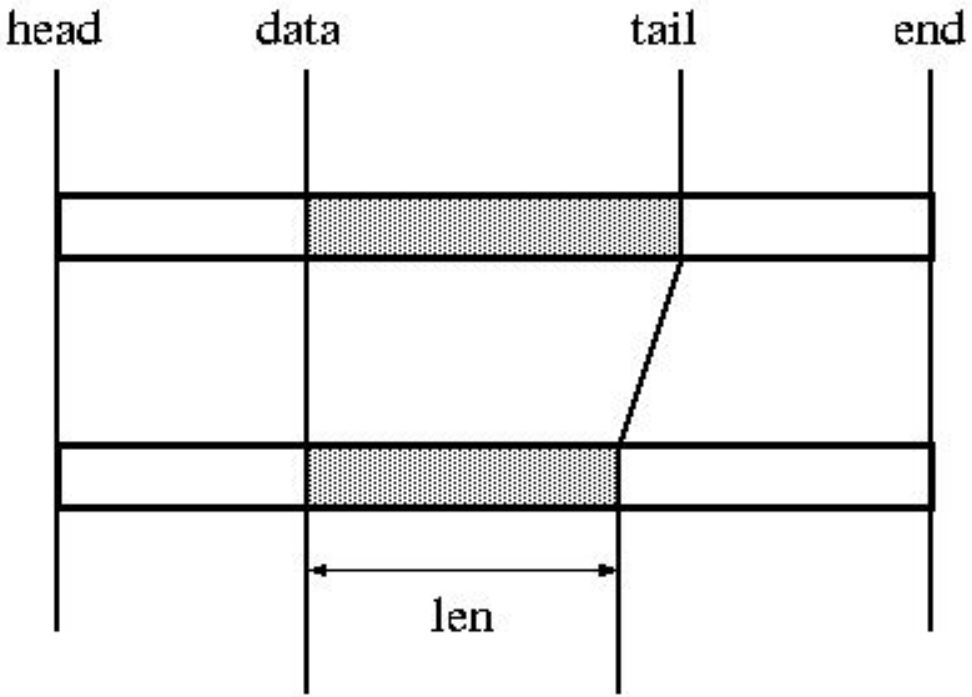
# skb_pull()



# skb_put()



# skb_trim()



# Como foi montado o experimento de teste?