

CH45

答疑(五)

**網路收發過程中, buffer
在哪裏?**

Agenda

- 問題一：網路收發過程中 buffer 的位置
- 問題二：protocol stack, 是通過一個 kernel thread 的方式運行的嗎？
- 問題三：最大的連線數是不是受限於 65535 個 ports？
- 問題四："如何改善 NAT 性能" 的效能思考
- Questions

問題一：網路收發過程中 buffer 的位置

What are these definition?

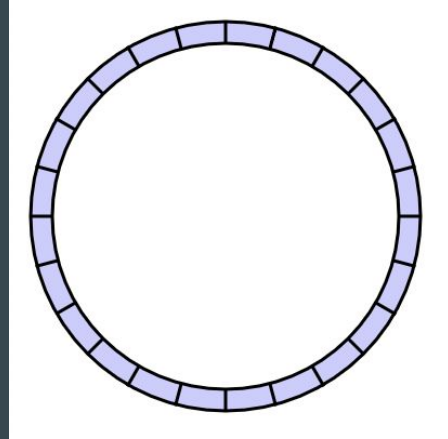
- **DMA (Direct Memory Access)**
- **Circular buffer**
- **sk_buff**
- **Socket() buffer**

DMA (Direct Memory Access)

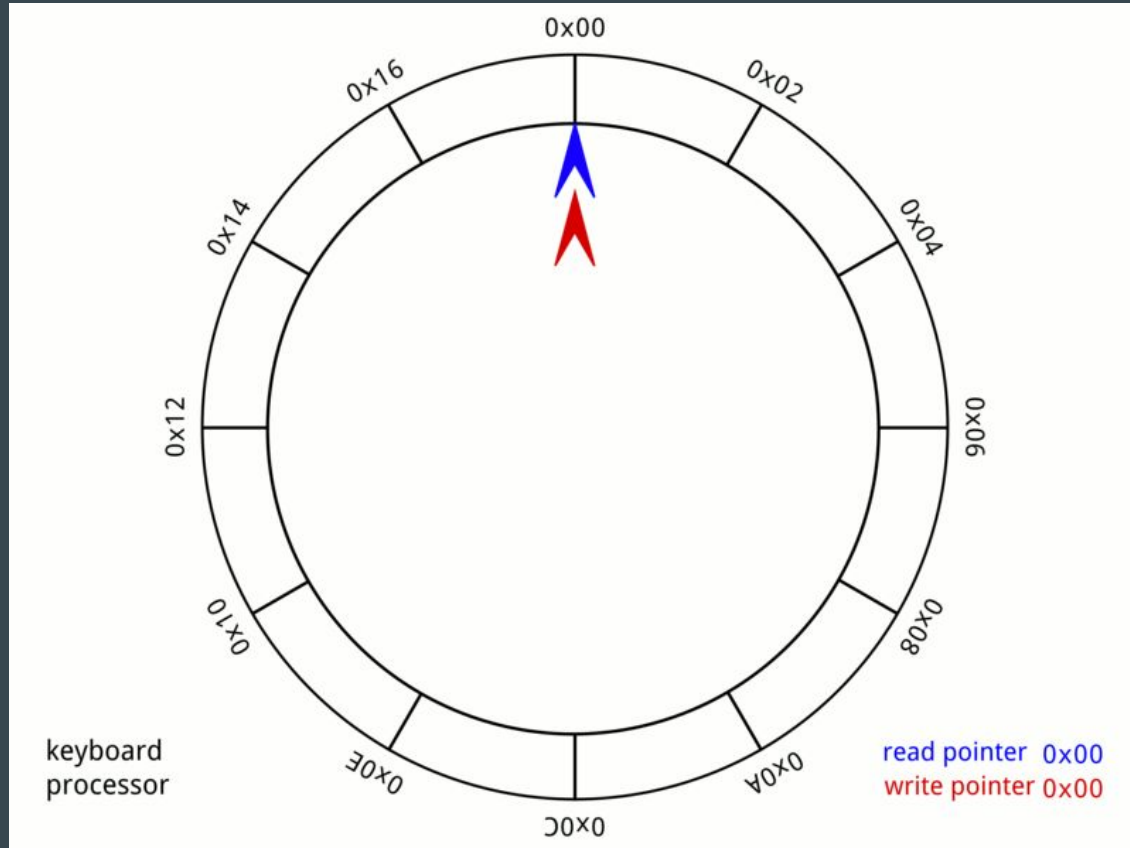
- **For high-speed propose, DMA channel allows device can access memory directly without involved CPU.**
- **DMA channel usually used by the sound card, hard disk drives, network interface.**

Circular buffer

- Is a data structure that uses a single, fixed-size buffer as if it were connected end-to-end.



HELLO WIKIPEDIA with keyboard



sk_buff

- struct “sk_buff” is one of skbuff.h function
- “sk_buff” is a large struct containing all the control information required for the packet

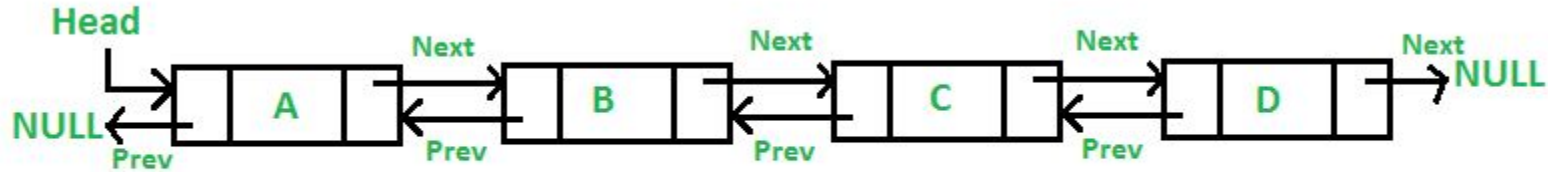
sk_buff & sk_buff_head

The sk_buff elements are organized as a doubly linked list, in such a way that it is very efficient to move an sk_buff element from the beginning/end of a list to the beginning/end of another list.

A queue is defined by struct sk_buff_head, which includes a head and a tail pointer to sk_buff elements.

Doubly linked list

Doubly linked list is a linked data structure that consists of a set of sequentially linked records called nodes



Queue of sk_buff elements

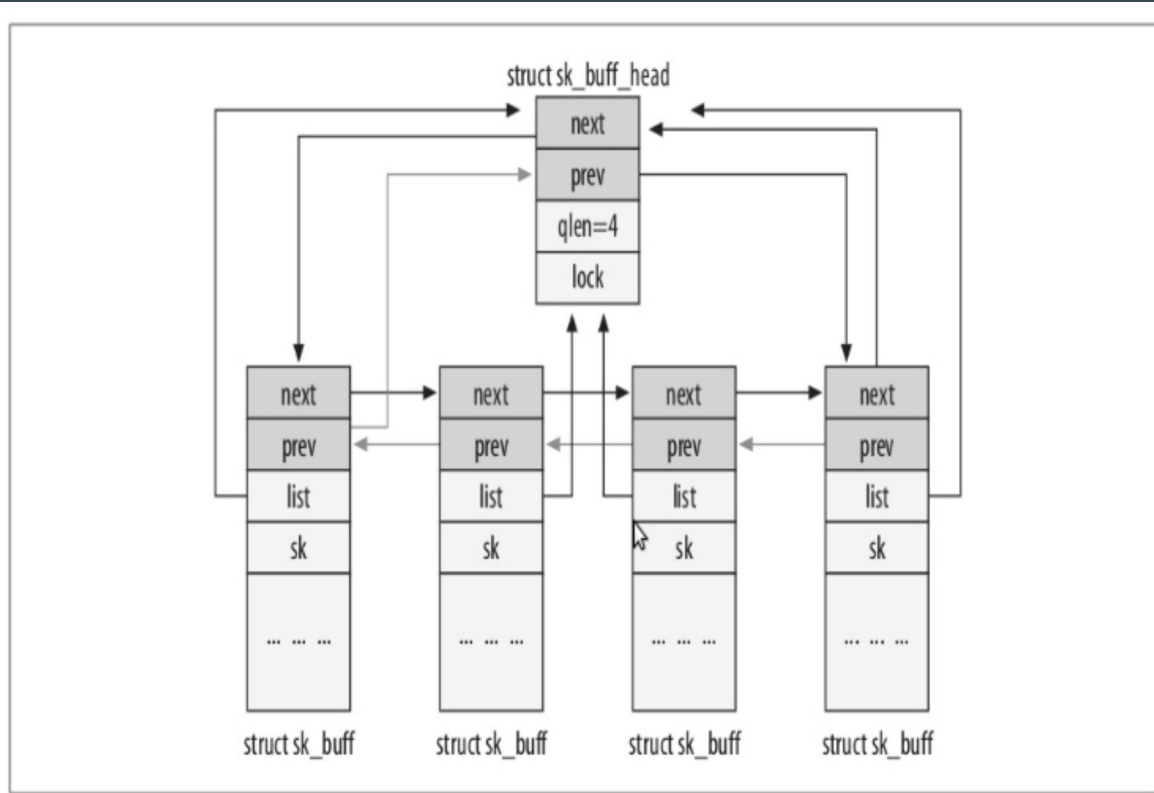
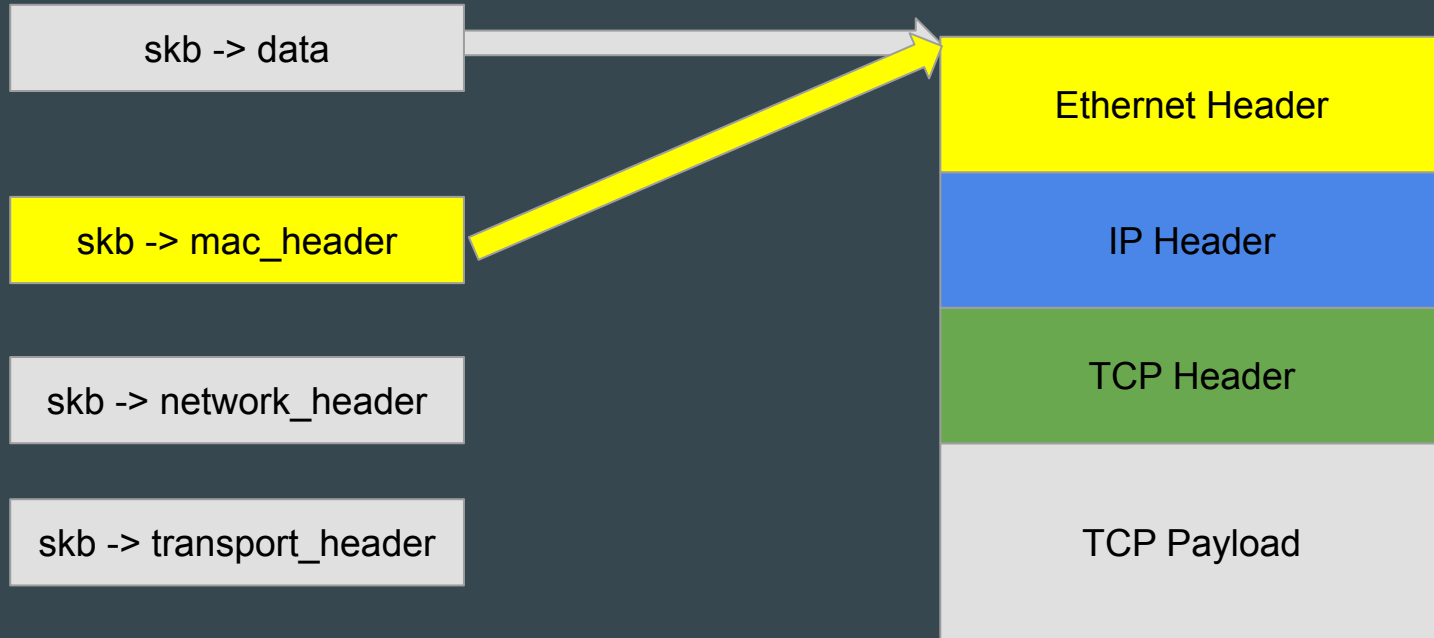
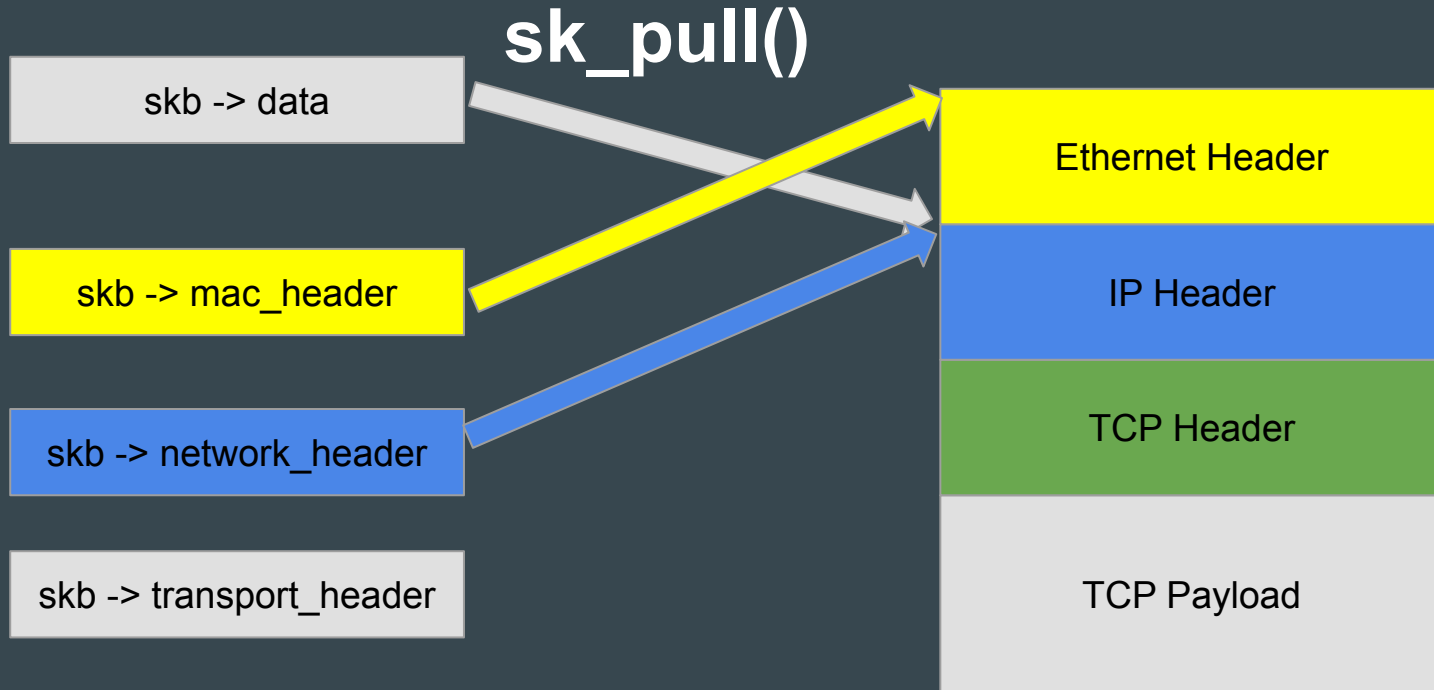


Figure 2-1. List of `sk_buff` elements

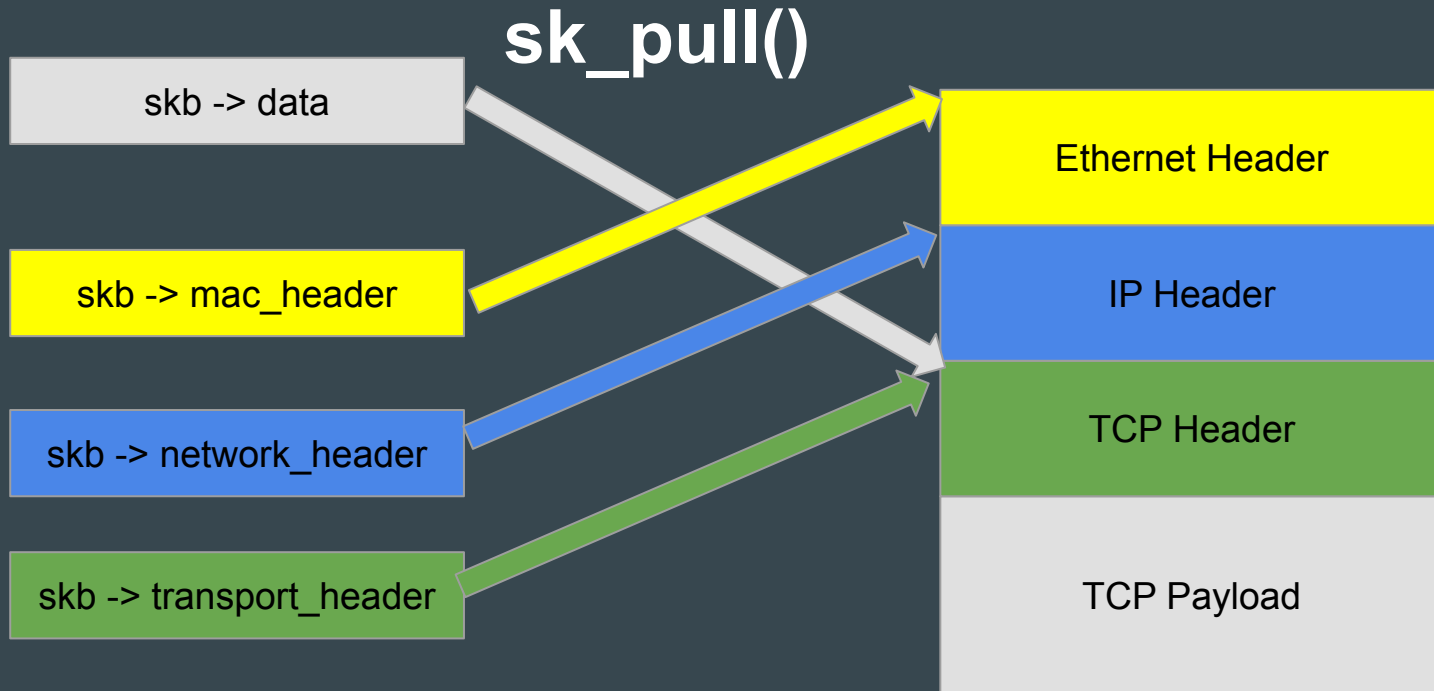
Packet process with sk_buff (1)



Packet process with sk_buff (2)



Packet process with sk_buff (3)



Socket() buffer

- **Socket() is for application only (user space)**

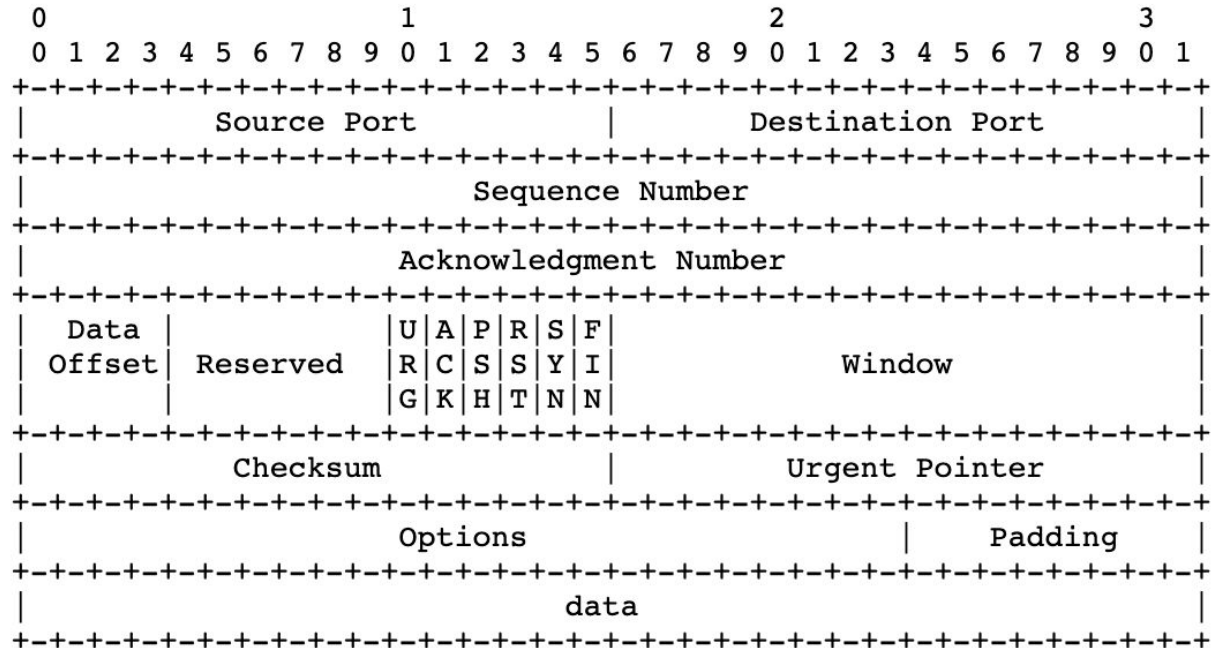
Socket() buffer Lab

**問題二：protocol stack, 是
通過一個 kernal thread 的
方式運行的嗎？**

問題三：最大的連線數是不是受限於 65535 個 ports ？

TCP Header

TCP Header Format



TCP Header Format

TCP fields you must know

- Source/Destination port max value = $2^{16} - 1 = 65536 - 1$ (because value start at 0)
- Sequence/Acknowledgment Number is to inform the sending host that the transmitted data was received successfully.
- ACK, SYN, FIN flags

問題四：“如何改善 NAT 性能”的效能思考

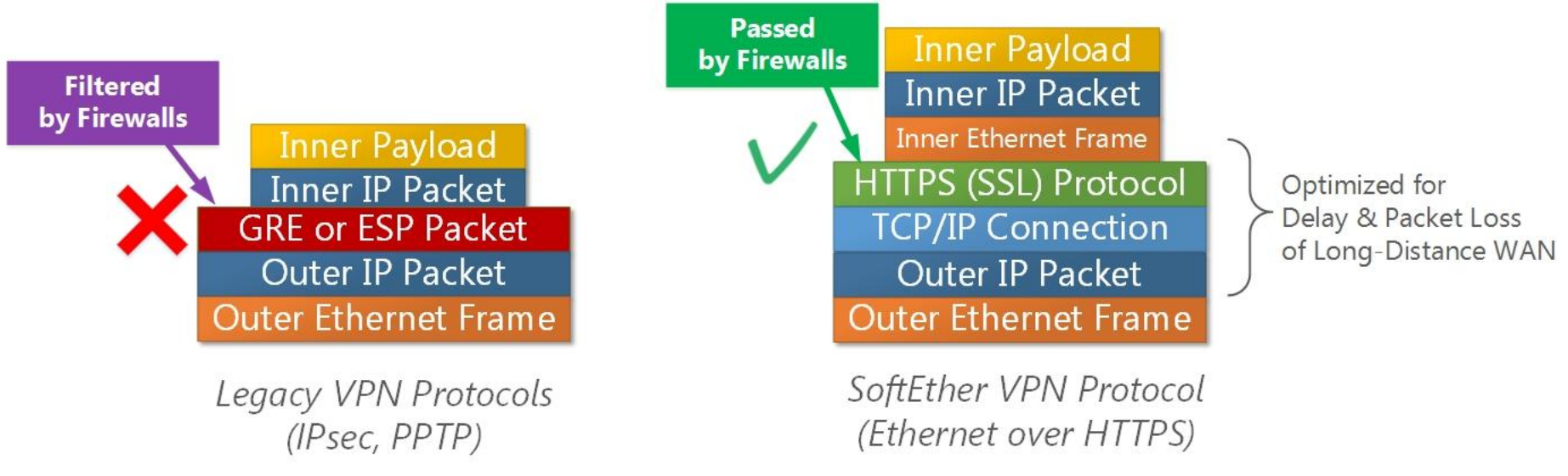
MASQUERADE!?

[Click me?!](#)

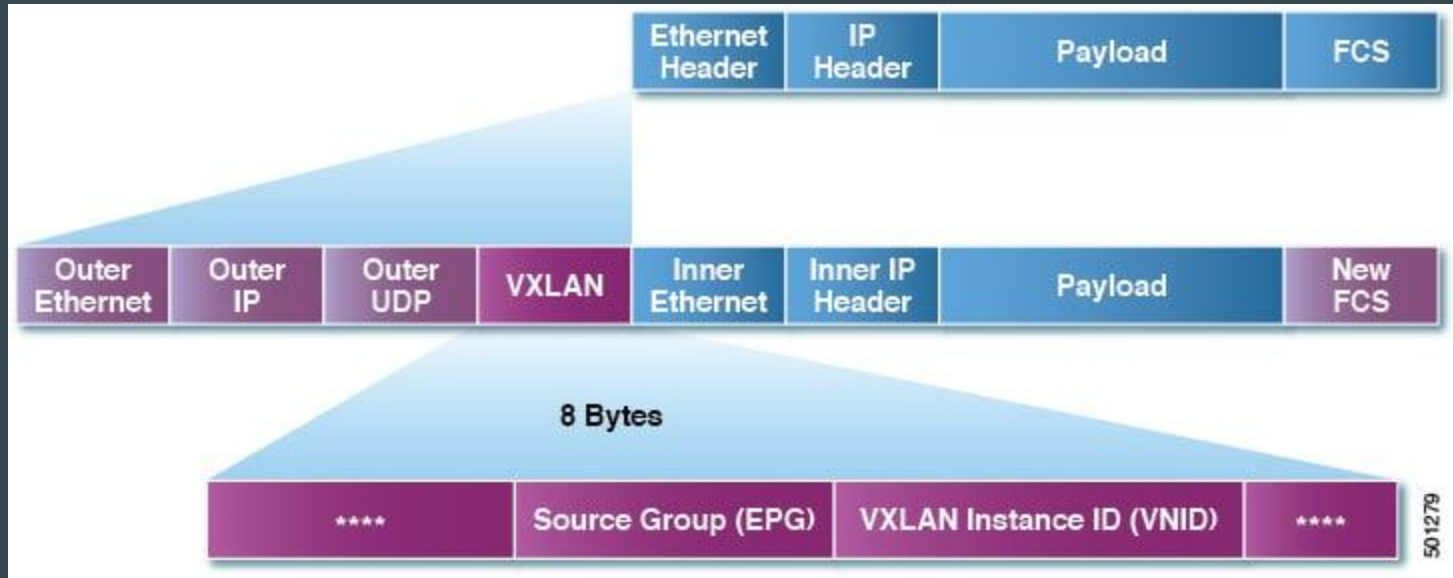
Masquerade

- It's a concept that enables you to hide real source/destination IP or port.
- Implement on NAT, VPN, VxLAN
 - NAT is modify source IP and port
 - VPN is add extra IP header at the head of packet
 - VxLAN is add extra IP header at the head of packet

IPsec VPN & SSL VPN packet format



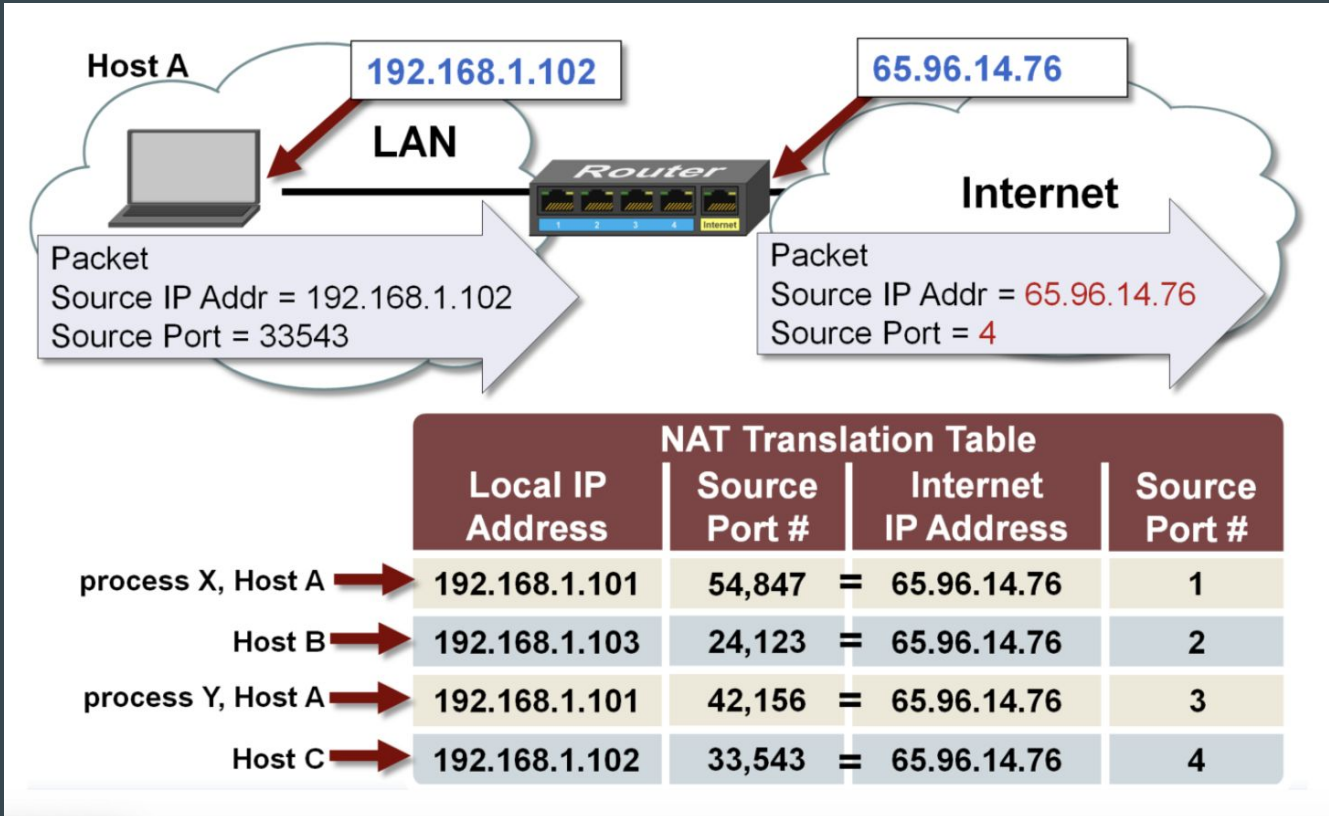
VxLAN packet format



NAT

- **private >> public,
public >> private,
private >> private**
- **90% of NAT is using SNAT. Some of DNAT is using on the firewall.**
- **NAT translation table saves every translation to track connection.**

SNAT



Questions

Questions

- **Did anyone use Socket() library before? How was that?**

Reference

- [DMA channel](#)
- [Circular buffer](#)
- [sk_buff \(The Linux Foundation wiki\)](#)
- [What is SKB in Linux kernel?](#)
- [Sockets and Socket-Buffer](#)
- [sk_buff 定義與操作](#)
- [struct sk_buff \(linux/include/linux/skbuff.h\)](#)
- [Socket Buffer Functions](#)

- [Doubly Linked List \(GeeksToGeeks\)](#)
- [Doubly Linked List \(Wikipedia\)](#)

- [Socket python library](#)
- [Sockets Tutorial](#)
- [Python sockets buffering](#)

- TCP packet format (RFC)
- Understanding TCP Sequence and Acknowledgment Numbers

- [Masquerade - Phantom of the Opera](#)
- [Linux Networking/IP Masquerade](#)
- [Linux VPN Masquerade HOWTO](#)
- [SoftEther VPN](#)
- [IPsec \(RFC\)](#)
- [VxLAN \(RFC\)](#)
- [NAT](#)
- [NAT Translation Table](#)