if_ovpn

OpenVPN Data Channel Offload

netgate

OpenVPN

- VPN _(No, really!)

- Originally developed by James Yonan

  - First release May 13th, 2001

- p2p, client/server

- pre-shared key, certificate or username/password based authentication

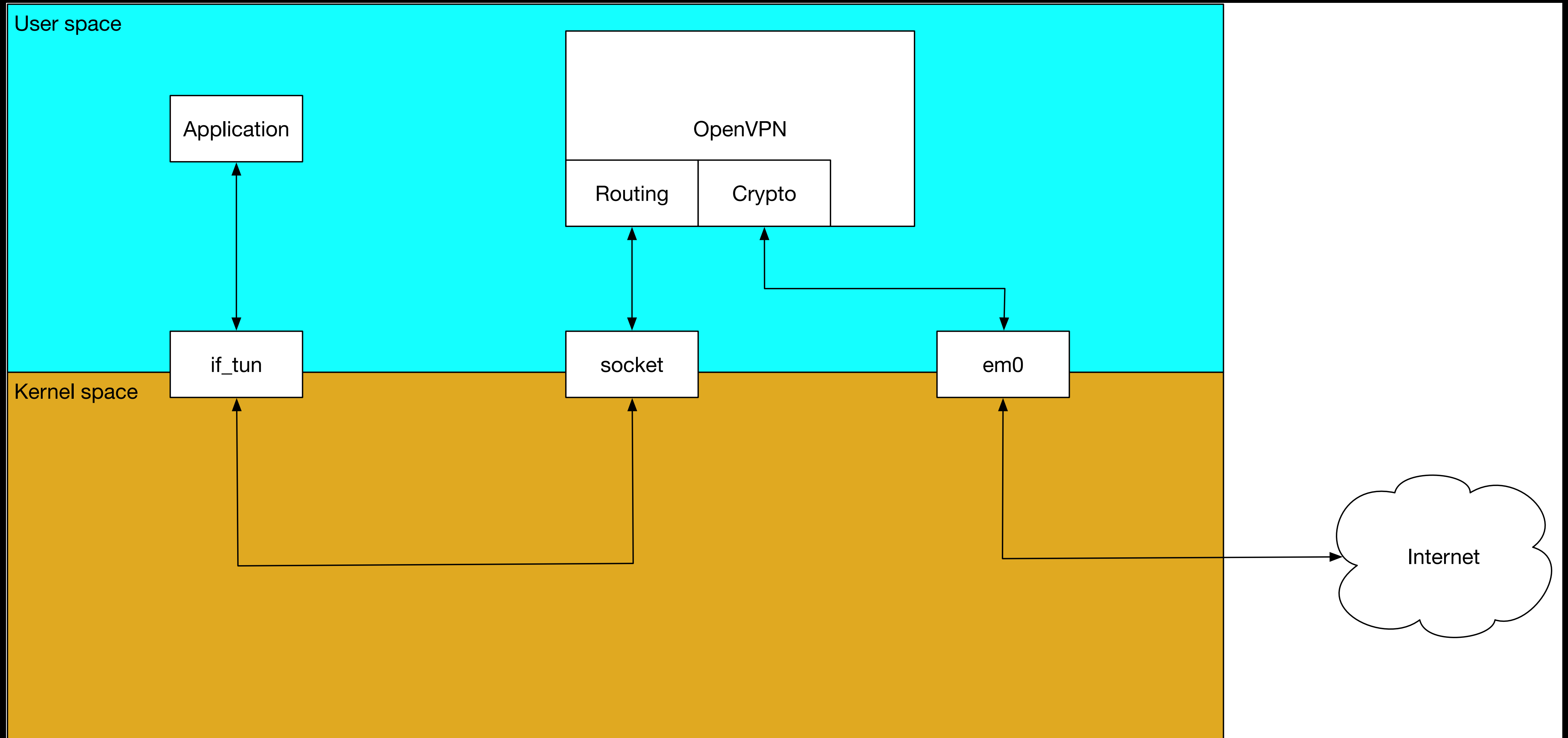- Windows, Linux, macOS, Android, AIX, FreeBSD, OpenBSD, DragonflyBSD, …

The problem

- It's slow

The problem
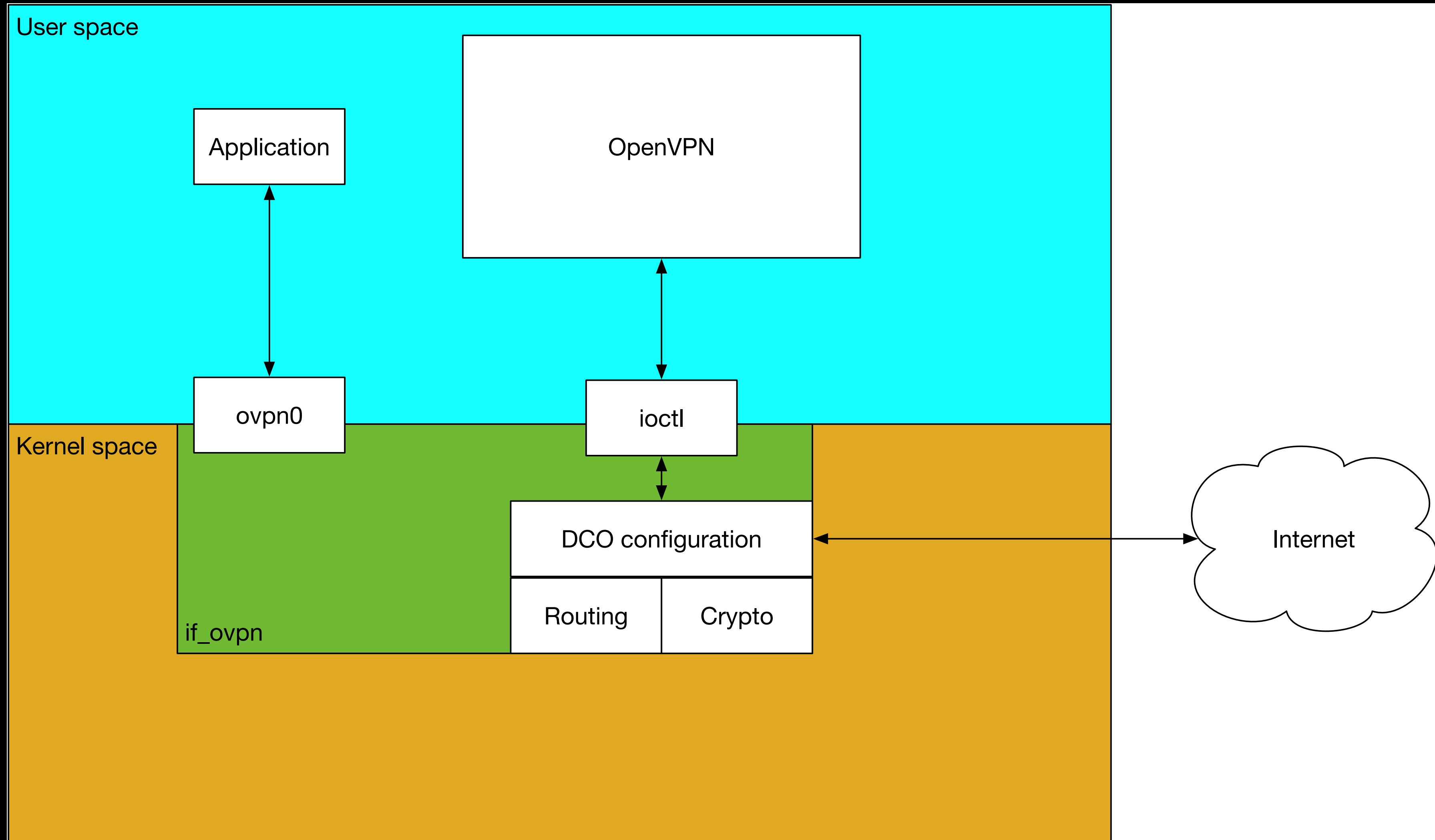
- It's slow

  - Because user space

The problem

User space

Application

OpenVPN

Routing | Crypto

if_tun

Kernel space

socket

em0

Internet

netgate

5

What is DCO

- Data Channel Offload

  - i.e. data channel in the kernel

- if_ovpn

- No more (extra) copying between user and kernel space

- Also, hardware accelerators!

  - (Although user space already got AES-NI, which is the one you have)

netgate

DCO

Limitations

- Only AES-GCM or ChaCha20/Poly1305

- No

    - Compression

    - Fragmention

    - Layer 2

    - Topologies other than subnet

    - Traffic shaping (in OpenVPN)

- OpenVPN used this as an opportunity for a clean break

    - Clients must be OpenVPN 2.4 or greater

netgate®

Considerations

- UDP vs. TCP

- Multiplexing socket

- Locking

- Ioctl

  - nvlist

- Routing

- Key rotation

- vnet

UDP vs. TCP

- OpenVPN supports tunnels over TCP

  - Because firewalls

- FreeBSD's if_ovpn is UDP-only

  - We didn't care enough about TCP

  - No equivalent to `udp_set_kernel_tunneling()` for TCP

netgate

Multiplexing

- Control channel (userspace) vs. data channel (kernel)

  - Share a single socket

  - fd passed to kernel during setup

  - Kernel passes unknown (i.e. control) packets to userspace

    - One of the few network stack modification required

netgate

Locking design

- rm_lock

  - Read-lock when processing incoming or outgoing packets

  - Write-lock for configuration changes

  - Exceptions

    - Counters

    - Replay protection

netgate.

ioctl

- Configuration interface

- Using nvlists for extensibility

- Linux uses netlink

  - We have netlink too

  - Now

    - Netlink landed after this work

Routing

- Tunnels are not a broadcast domain

  - Need to work out which peer to send to

  - Second routing lookup based on destination IP of the tunneled packet

  - `ovpn_route_peer()`

  - Special case

    - Only one client

netgate®

Key rotation

- Handled (mostly) by userspace

  - Negotiation (userspace)

  - Install new key with OVPN_NEW_KEY cmd

  - Switch over with OVPN_SWAP_KEYS

    - Each packet contains a key id

    - No traffic disruption

  - Remove old key with OVPN_DEL_KEY

netgate®

vnet

- Not a required feature

- But so, so good for testing

  - /usr/tests/sys/net/if_ovpn

netgate®

Performance

- Tested on a Netgate 4100

  - Intel® Atom® C3338R with QAT, 2-core @ 1.8 GHz

if_tun (AES-NI)	207.3 Mbit/s
DCO Software	213.1 Mbit/s
DCO AES-NI	751.2 Mbit/s
DCO QAT	1,064.8 Mbit/s

Where can I get this?

- OpenVPN 2.6.0

    - Released January 26th 2023

- FreeBSD 14.0

    - Also Linux and Windows

        - But you don't care

- pfSense+ 22.05

    - Released 27/06/2022

Thank You

Questions?

Kristof Provost
kp@FreeBSD.org
@kp@bsd.network

netgate®