

# Repacket

## パケットを書き換えるCLIツール

学習駆動コース 坂井ゼミ 野崎愛

### 前提: パケットを書き換える？

**パケットとは？** Repacketが対象とするパケットはIPパケットです。データは通信線上で送受信される際に小さい単位に分割されます。分割されたデータと制御のためのヘッダ情報とを合わせた通信の単位を、パケットと呼んでいます。

**書き換える？** Repacketはパケットのヘッダ部を書き換えるツールです。例えば検証・実験用途で誤ったチェックサムを持つパケットを簡単に生成することができます。

### “パケット書き換え”の現状・目標

| 現状           | 目標                   |
|--------------|----------------------|
| <b>手軽さ</b>   | 限られた人材がコーディング        |
| <b>セキュアさ</b> | パケットを書き換えるという行為に不安あり |
| <b>高速さ</b>   | ハードウェア(NIC)の高速化      |
|              | ハードウェアを活かすソフトウェア     |

→ 目標: **手軽に・セキュアに・高速に**パケットを書き換えられるツール

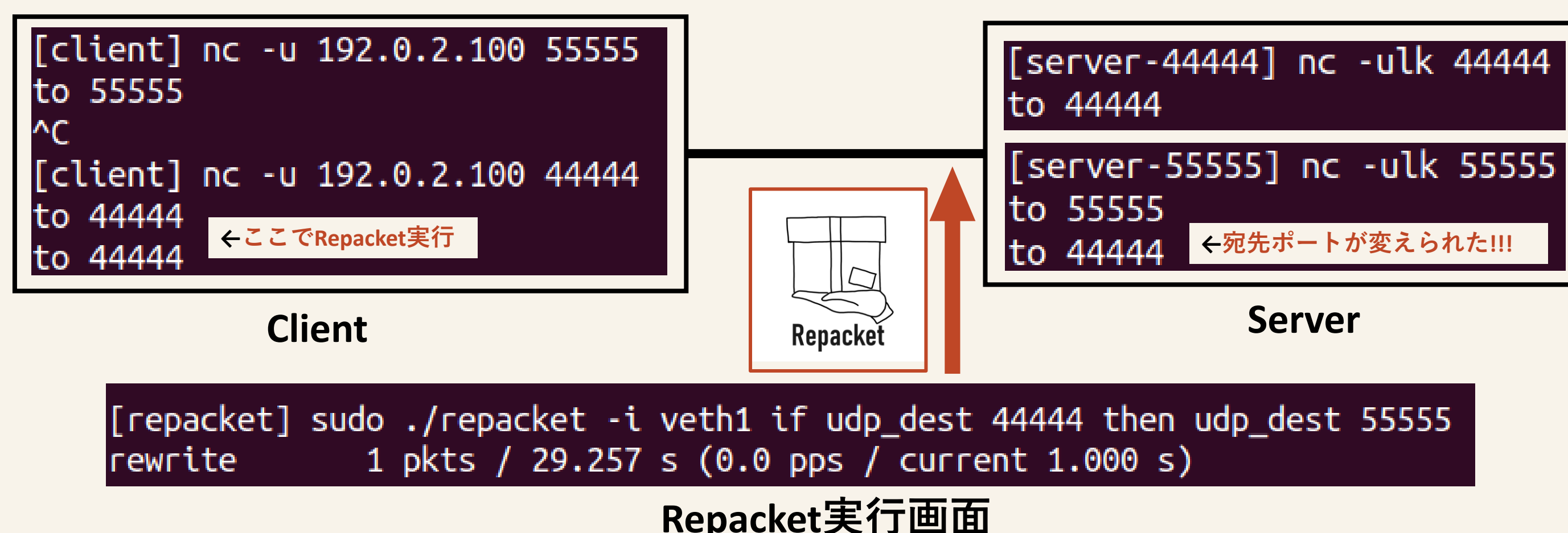
## Repacket = Rewrite + packet

## Linux上で動作しXDPをバックエンドに CLI操作1行でパケットのヘッダを書き換えるツール

### 使用方法

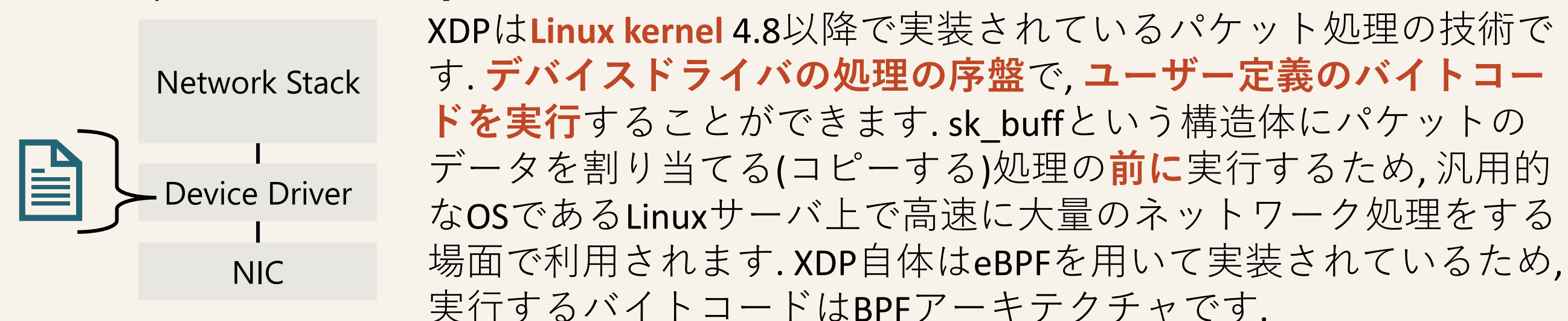
**repacket [-i interface] if [expression] then [expression]**  
[expression]はIP/TCP/UDP/ICMPの計39パラメータ・複数入力に対応  
(ex. ip\_ttl, tcp\_syn, udp\_check, icmp\_code)

#### UDP宛先ポートを書き換えるデモ



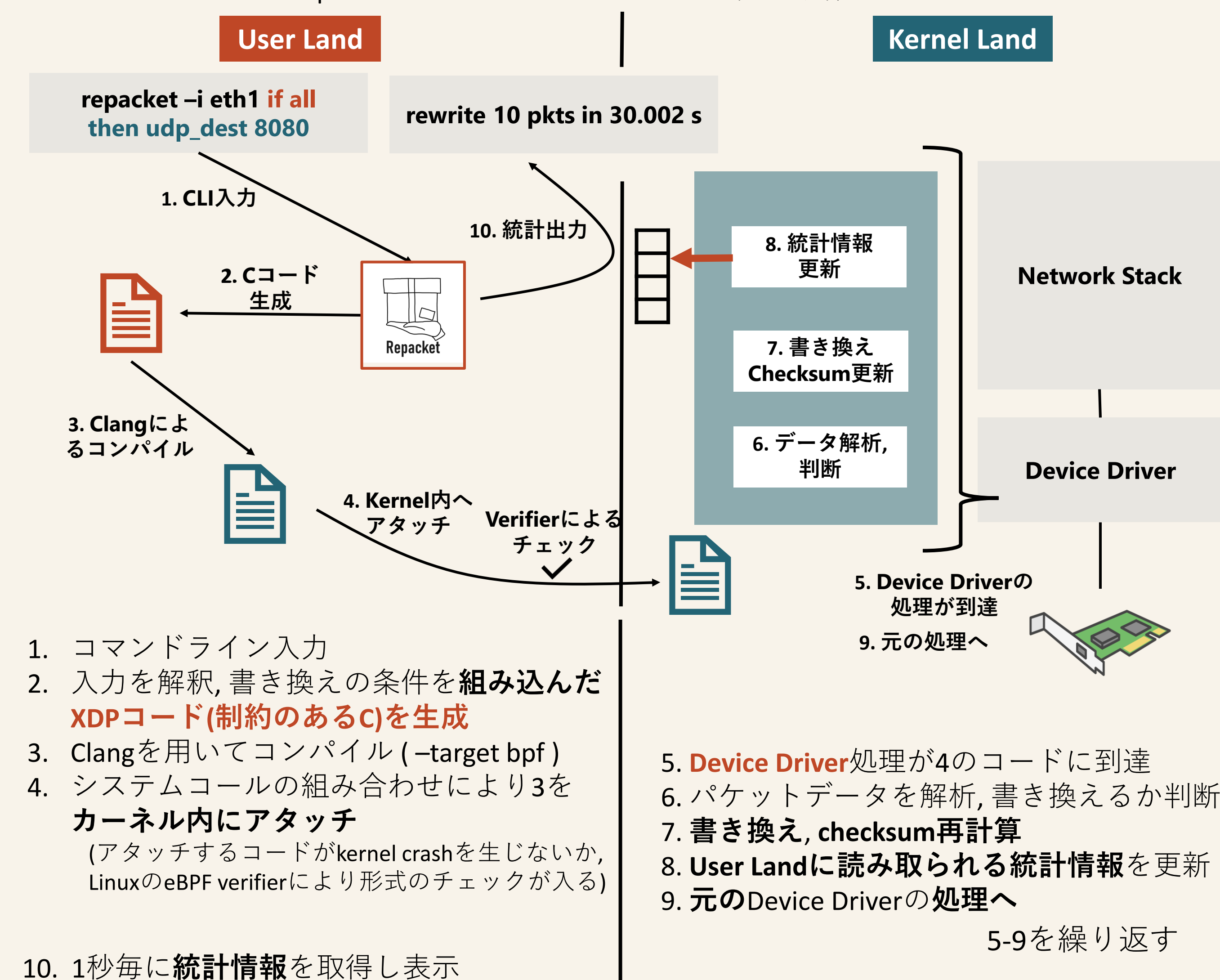
### XDPとは？

RepacketはXDP(eXpress Data Path)を利用してパケットを書き換えています。



### 設計

RepacketはUser Land・Kernel Land両側で動作します。



## 性能

### 測定環境

Traffic Generator/Repacket machine (同一マシン)  
■ CPU: AMD Ryzen5 5600G, 6core, 12th, 4.4GHz  
■ MEM: 32GB  
■ Kernel: 5.13.0-27-generic  
■ NIC1: Mellanox ConnectX-3 (10GbE)  
■ NIC2: Mellanox ConnectX-4 (10GbE)

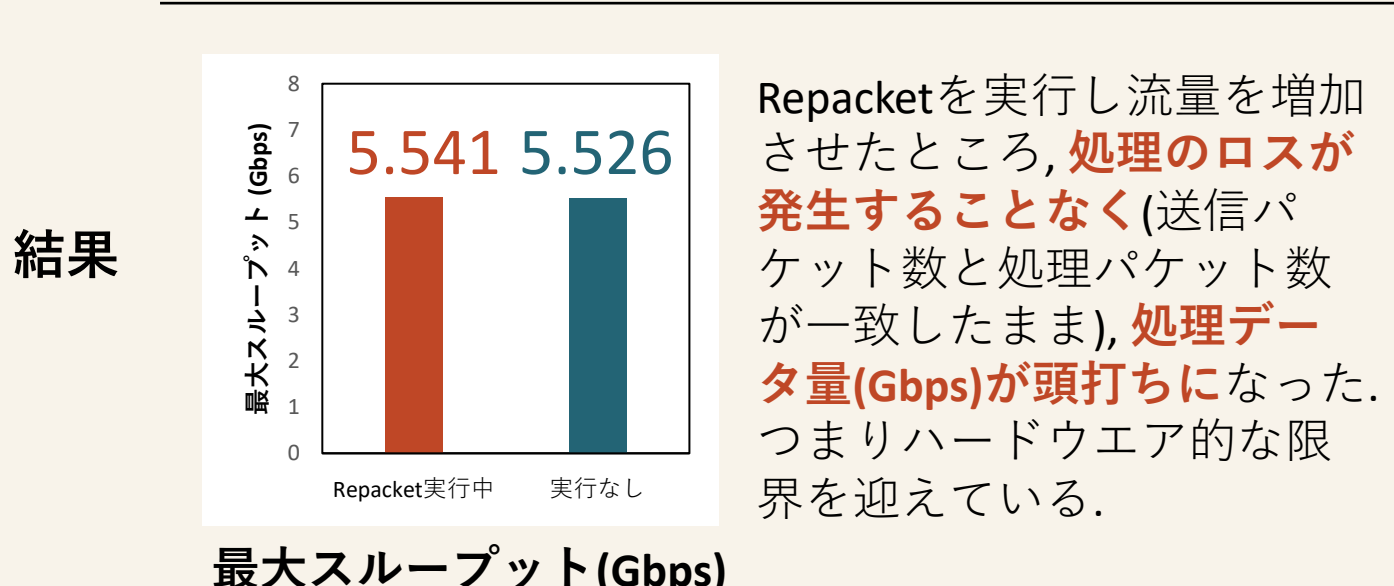


### 処理能力 (Gbps)

Repacketの要件でもあるハードウェアを活かすソフトウェアであることを確かめる。

1. Repacketを実行  
2. iperf3でパケットを発生させ、パケットロスせず処理できる最大スループットを探す  
3. 3回平均を取得

比較: Repacket最大スループット, iperf3で得られる最大帯域 (Gbps)

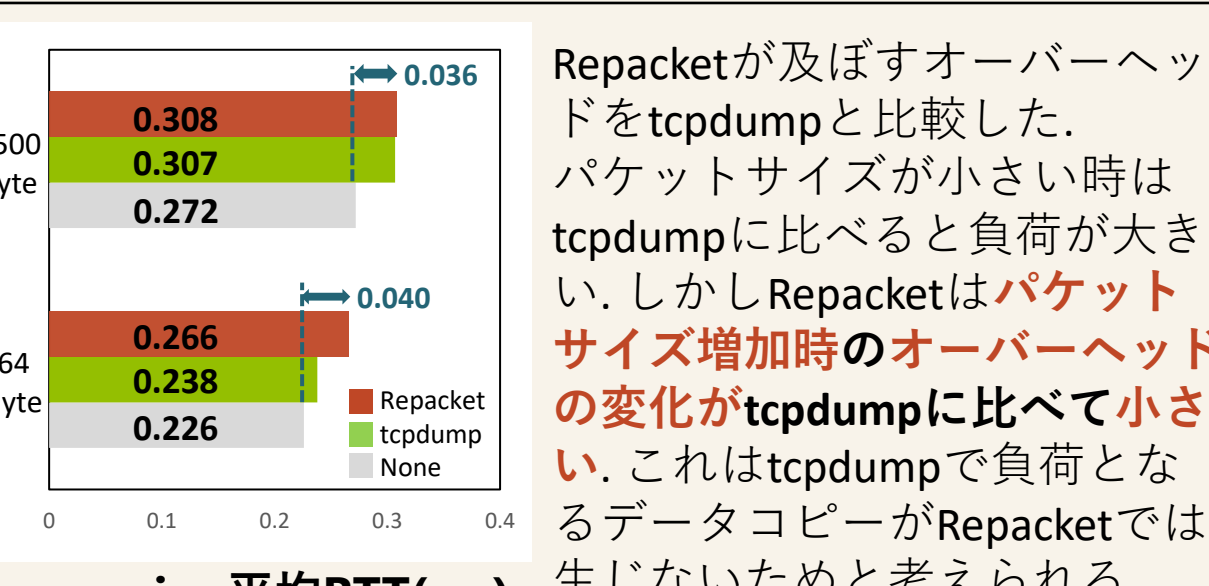


### Repacket実行による他パケット処理への影響

Repacketが高速に動くことだけでなく、他のネットワーク処理を犠牲にしているか確かめる。

1. {Repacket実行, tcpdump実行, なにもしない}  
2. ping12パケットの平均RTTを記録  
3. 更に10回平均を取得  
4. 64byteと1500byteの2パターンで実施

なし, tcpdump, Repacket (ms)



## こだわり

### XDPコード生成

カーネル内で動作するコードをユーザーのCLI入力毎に生成。入力毎に変化する条件をパケット処理前にコードに組み込むことで、パケット処理時のKernel Land/User Land間やり取りコストを減らす。

### パラメータの細かさ

書き換えるオプションとして設定するパラメータを細かく対応。  
Ex. IPのType of Service: ip\_dscp, ip\_ecnと指定可能  
Ex. TCPのReserved: tcp\_res, tcp\_ns, tcp\_cwr, tcp\_ecnと指定可能

### テストの充実

特に無効なCLI入力, 生成コードが正しく動作するかに注力。関数レベルの単体テスト, 実際にパケットを流すE2Eテストを実施。

103 tests from 10 test suites ran. 103 tests.

## 制作までの過程

「システムコールとは？」状態から始まり、XDPに出会い、パケット処理に興味を持ち、Repacketの制作が始まりました。学習駆動コースの名の通り学習することで新たに制作のアイデアが湧き、制作する過程で学習が進むサイクルを繰り返しました。

5月 応募テーマはBGPデーモン  
7月 XDPに出会い、カーネル内パケット処理に興味  
8月 XDPを用いたルータの作成  
10月 XDPでパケットキャプチャ+パケットトレース  
11月 パケットを書き換えるRepacket

```

-----8-----16-----24-----32
[v 4|h15|tos 0|totlen 84|
id 58873|NA|offset 0|
ttl 64|proto 1|csum 53419|
src 192.0.2.1|
dst 192.0.2.2|
|type 8|code 0|csum 35671|
id 6|seq 9|

```

パケットキャプチャのfriendly dump機能