PacketMill: Toward Per-Core 100-Gbps Networking

Alireza Farshin*, Tom Barbette*, Amir Roozbeh**, Gerald Q. Maguire Jr.*, Dejan Kostić*

* KTH Royal Institute of Technology  + Ericsson Research
A Story of Packet Delivery

NFV* Server

Packet Delivery

Translate

Driver

Metadata
- Length
- VLAN
- Data

* Network Function Virtualizations (NFV)
Metadata Is not Specialized for the Network Function

Metadata
Length
VLAN
Data

Another Translation

Framework Metadata
Length
VLAN
[NAT]
[DPI]
[Router]
Data

Not enough space or unnecessary fields

Remove
Add
Inefficient Metadata Management

Application

Packet Class*

Copy and Convert

Customized Format

Copy and Convert

Point and Cast

Overlay

rte mbuf

Packet Class*

DPDK Libraries

1

FastClick Model

Copy and Convert

rte mbuf

Generic Format

2

BESS Model

NIC Driver

Requires Two Copying

Metadata

Raw Packet

Vendor-Specific Format

Carries Unnecessary Fields

2021-04-12

* An Example Data Structure (used in FastClick)
Network Functions are a maze
Modular Packet Processing Frameworks
Implement a Chain of Network Functions

- DPI*
- NAT**
- Load Balancer
- Router
- IPSec
- Firewall

---

* Deep Packet Inspection (DPI)
** Network Address Translator (NAT)
Modular Packet Processing Frameworks
Create a Chain of Mazes
PacketMill mitigates these problems, increasing the performance and efficiency of the current software & hardware when processing packets.
A metadata management model called **X-Change** that enables DPDK-based applications to use customized data structures instead of `rte_mbuf`

**No need to translate**
**X-Change Prevents any Extra Operations**

- Exchanges data structures with DPDK PMD*

**Others:**
- Uses fewer in-flight buffers
- Avoid allocating/releasing buffers

---

* Poll Mode Driver (PMD): user space DPDK drivers
X-Change uses LTO** to inline the functions

```c
/* X-Change Implementation for Default DPDK */
void xchg_set_vlan_tci(struct xchg* pkt, uint16_t vlan_tci) {
    ((struct rte_mbuf*)pkt)->vlan_tci = vlan_tci;
}

/* X-Change Implementation for Custom Buffers */
void xchg_set_vlan_tci(struct xchg* pkt, uint16_t vlan_tci) {
    SET_VLAN_ANNO((Packet*)pkt, vlan_tci);
}
```
A metadata management model called **X-Change** that enables DPDK-based applications to use customized data structures instead of `rte_mbuf`.

- No need to translate
- Provides better order

**A**

- Tracks GetElementPtrInst (GEPI) Instructions
- Reorders the application-specific data structure
- Fix the GEPI Instructions

**B**

Uses LLVM optimization passes to reorder data structures in the IR* (LLVM bit code)
A metadata management model called **X-Change** that enables DPDK-based applications to use customized data structures instead of *rte_mbuf*

**No need to translate**

**Uses LLVM optimization passes to reorder data structures in the IR* (LLVM bit code)**

**Provides better order**

**Uses/Embeds the available information in the configuration file to perform source-code modifications**

**Simplifies the maze**
PacketMill generates a customized binary for a given chain of network function by performing whole-stack optimizations.

Currently supports:
FastClick & Mellanox PMD (mlx5)
PacketMill Workflow

Describing the Network Function

Input Config File

Click-based Router
PacketMill Workflow

Describing the Network Function

Input Config File → Modifies Source Code → Embeds the Constants and the Graph → Fast Click Source

Whole-Program IR

Compiles with LTO* → IR Bitcode

XCHG* + DPDK

Link → Opt. IR → Link

Customized Binary

Modifies IR Bitcode

IR Bitcode

Opt. IR

Whole-Program IR

Compiles with LTO*

XCHG* + DPDK

Input Config File

Modifies Source Code

Embeds the Constants and the Graph

Fast Click Source

Customized Binary

Link

Opt. IR

Whole-Program IR

Compiles with LTO*

XCHG* + DPDK

Input Config File

Modifies Source Code

Embeds the Constants and the Graph

Fast Click Source

Whole-Program IR

Compiles with LTO*

XCHG* + DPDK

Opt. IR

Customized Binary

Link

Whole-Program IR

Compiles with LTO*

XCHG* + DPDK

Opt. IR

Customized Binary

Link

Whole-Program IR

Compiles with LTO*

XCHG* + DPDK

Opt. IR

Customized Binary

Link

Whole-Program IR

Compiles with LTO*

XCHG* + DPDK

Opt. IR

Customized Binary

Link

Whole-Program IR

Compiles with LTO*

XCHG* + DPDK

Opt. IR

Customized Binary

Link
PacketMill Provides the Right Format & Order for Metadata and Minimizes the Framework Footprint

Better Cache Locality

Right Format
Better Order

NF Metadata
[Router]
[DPI]
Data
[NAT]
VLAN

Packet
Evaluation

1. Impact of Code Optimizations
2. X-Change vs. Existing Metadata Management Models
3. Impact of Workload/Trace
4. Sophisticated Network Functions
5. Multicore Network Functions
X–Change vs. Existing Metadata Management Model

Throughput stops increasing

Throughput (Gbps) vs. Processor Frequency (GHz)

- X-Change
- Overlaying
- Copying

Higher is better

1x Mellanox Connect-X 5

CPU Core

100 Gbps

~33% increase in Throughput

~21% increase in Throughput
X–Change is the Only Model Capable of Forwarding Packets at >100 Gbps

200 Gbps

2x Mellanox Connect-X 5

Higher is better

Total Throughput (Gbps)

Processor Frequency (GHz)

X-Change
Overlaying
Copying

~17 Gbps
PacketMill Forwards Packets Faster than State-of-the-Art Frameworks

![Graph showing throughput vs packet size]

- Approx. 45% Faster

Higher is better
PacketMill Shifts the Knee of Throughput vs. Latency Curve

A router is forwarding a real campus trace with one core at different rates

Lower and More Right is Better
Conclusion

• Mitigating code inefficiencies and improving metadata management makes it possible to process packets on commodity hardware at higher rates.

• PacketMill achieves a better performance compared to other packet processing frameworks.

• PacketMill forwards at >100 Gbps with one core being fed with two NICs.

• Check out our paper for more information.

[QR code linking to the paper]

[aliireza/packetmill]

[packetmill.io]
Thanks for watching

Do not hesitate to contact us if you have any questions.

farshin@kth.se and barbette@kth.se