

# Intel<sup>®</sup> Data Plane Development Kit - IPv4 Fragmentation Sample Application

User Guide

---

*April 2012*

**Intel Confidential**



INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

A "Mission Critical Application" is any application in which failure of the Intel Product could result, directly or indirectly, in personal injury or death. SHOULD YOU PURCHASE OR USE INTEL'S PRODUCTS FOR ANY SUCH MISSION CRITICAL APPLICATION, YOU SHALL INDEMNIFY AND HOLD INTEL AND ITS SUBSIDIARIES, SUBCONTRACTORS AND AFFILIATES, AND THE DIRECTORS, OFFICERS, AND EMPLOYEES OF EACH, HARMLESS AGAINST ALL CLAIMS COSTS, DAMAGES, AND EXPENSES AND REASONABLE ATTORNEYS' FEES ARISING OUT OF, DIRECTLY OR INDIRECTLY, ANY CLAIM OF PRODUCT LIABILITY, PERSONAL INJURY, OR DEATH ARISING IN ANY WAY OUT OF SUCH MISSION CRITICAL APPLICATION, WHETHER OR NOT INTEL OR ITS SUBCONTRACTOR WAS NEGLIGENT IN THE DESIGN, MANUFACTURE, OR WARNING OF THE INTEL PRODUCT OR ANY OF ITS PARTS.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined". Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or go to: <http://www.intel.com/design/literature.htm>.

Any software source code reprinted in this document is furnished for informational purposes only and may only be used or copied and no license, express or implied, by estoppel or otherwise, to any of the reprinted source code is granted by this document.

Code Names are only for use by Intel to identify products, platforms, programs, services, etc. ("products") in development by Intel that have not been made commercially available to the public, i.e., announced, launched or shipped. They are never to be used as "commercial" names for products. Also, they are not intended to function as trademarks.

Intel and the Intel logo are trademarks of Intel Corporation in the U.S. and/or other countries.

\*Other names and brands may be claimed as the property of others.

Copyright © 2012, Intel Corporation. All rights reserved.



## Contents

---

|   |   |
|---|---|
| <b>1.0 Introduction</b> .....             | 4 |
| 1.1 Documentation Roadmap .....           | 4 |
| <b>2.0 Overview</b> .....                 | 4 |
| <b>3.0 Building the Application</b> ..... | 5 |
| <b>4.0 Running the Application</b> .....  | 5 |

## Revision History

---

| Date          | Revision | Description                      |
|---------------|----------|----------------------------------|
| March 2012    | 1.0      | Updates for software release 1.2 |
| December 2011 | 0.7      | Initial version of document      |



## 1.0 Introduction

This application is a simple example of packet processing using the Intel® Data Plane Development Kit (Intel® DPDK). The application does L3 forwarding with IPv4 packet fragmentation. The MBUF and PMD libraries have been modified to allow the fragmentation of a packet without copying the packet's data.

### 1.1 Documentation Roadmap

The following is a list of Intel® DPDK documents in suggested reading order:

- **Release Notes:** Provides release-specific information, including supported features, limitations, fixed issues, known issues and so on. Also, provides the answers to frequently asked questions in FAQ format.
- **Getting Started Guide:** Describes how to install and configure the Intel® DPDK software; designed to get users up and running quickly with the software.
- **Programmer's Guide:** Describes:
  - The software architecture and how to use it (through examples), specifically in a Linux\* application (linuxapp) environment
  - The content of the Intel® DPDK, the build system (including the commands that can be used in the root Intel® DPDK Makefile to build the development kit and an application) and guidelines for porting an application
  - Optimizations used in the software and those that should be considered for new development

A glossary of terms is also provided.

- **API Reference:** Provides detailed information about Intel® DPDK functions, data structures and other programming constructs.
- **Sample Application User Guides:** A set of guides, each describing a sample application that showcases specific functionality, together with instructions on how to compile, run and use the sample application.

## 2.0 Overview

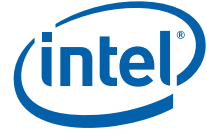
The application demonstrates the use of zero-copy buffers for packet fragmentation. The initialization and run-time paths are very similar to those of the L2 forwarding application (see the *Intel® DPDK L2 Forwarding Sample Application User Guide* for more information). This guide highlights the differences between the two applications.

There are two key differences from the L2 Forwarding sample application:

- The first difference is that the IPv4 Fragment sample application makes use of indirect buffers.
- The second difference is that the forwarding decision is taken based on information read from the input packet's IPv4 header.

The Longest Prefix Match (LPM) table is used to store/lookup an outgoing port number, associated with that IPv4 address.

By default, input frame sizes up to 9.5 KB are supported. Before forwarding, the input IPv4 packet is fragmented to fit into the "standard" Ethernet\* v2 MTU (1500 bytes).



## 3.0 Building the Application

To build the application:

1. Go to the sample application directory:

```
export RTE_SDK=/path/to/rte_sdk
cd ${RTE_SDK}/examples/ipv4_frag
```

2. Set the target (a default target is used if not specified). For example:

```
export RTE_TARGET=x86_64-default-linuxapp-gcc
```

See the *Intel® DPDK Getting Started Guide* for possible RTE\_TARGET values.

3. Build the application:

```
make
```

## 4.0 Running the Application

The LPM object is created and loaded with the pre-configured entries read from a global `l3fwd_route_array` table. For each input packet, the packet forwarding decision (that is, the identification of the output interface for the packet) is taken as a result of LPM lookup. If the IP packet size is greater than default output MTU, then the input packet is fragmented and several fragments are sent via the output interface.

Application usage:

```
./build/ipv4_frag [EAL options] -- -p PORTMASK [-q NQ]
```

where:

- `-p PORTMASK` is a hexadecimal bitmask of ports to configure
- `-q NQ` is the number of queue (=ports) per lcore (the default is 1)

To run the example in linuxapp environment with 2 lcores (2,4) over 2 ports(0,2) with 1 RX queue per lcore:

```
./build/ipv4_frag -c 0x14 -n 3 -- -p 5
EAL: coremask set to 14
EAL: Detected lcore 0 on socket 0
EAL: Detected lcore 1 on socket 1
EAL: Detected lcore 2 on socket 0
EAL: Detected lcore 3 on socket 1
EAL: Detected lcore 4 on socket 0
...
Initializing port 0 on lcore 2... Address:00:1B:21:76:FA:2C, rxq=0 txq=2,0 txq=4,1
done: Link Up - speed 10000 Mbps - full-duplex
Skipping disabled port 1
Initializing port 2 on lcore 4... Address:00:1B:21:5C:FF:54, rxq=0 txq=2,0 txq=4,1
done: Link Up - speed 10000 Mbps - full-duplex
Skipping disabled port 3
Adding route 0x640a0000 / 16 (2)
Adding route 0x64140000 / 16 (2)
...
L3FWD: entering main loop on lcore 4
L3FWD: -- lcoreid=4 portid=2
L3FWD: entering main loop on lcore 2
L3FWD: -- lcoreid=2 portid=0
```



To run the example in linuxapp environment with 1 lcore (4) over 2 ports(0,2) with 2 RX queues per lcore:

```
./build/ipv4_frag -c 0x10 -n 3 -- -p 5 -q 2
```

To test the application, flows should be set up in the flow generator that match the values in the `l3fwd_route_array` table.

The default `l3fwd_route_array` table is:

```
struct l3fwd_route l3fwd_route_array[] = {
    {IPv4(100,10,0,0), 16, 2},
    {IPv4(100,20,0,0), 16, 2},
    {IPv4(100,30,0,0), 16, 0},
    {IPv4(100,40,0,0), 16, 0},
};
```

For example, for the input IPv4 packet with destination address: 100.10.1.1 and packet length 9198 bytes, seven IPv4 packets will be sent out from port #2 to the destination address 100.10.1.1: six of those packets will have length 1500 bytes and one packet will have length 318 bytes.

Unlike the L3 Forwarding sample application, no NUMA support is provided in the IPv4 Fragmentation sample application.

Refer to the *Intel® DPDK Getting Started Guide* for general information on running applications and the Environment Abstraction Layer (EAL) options.

§ §