



DPL Nspect

Table of contents

Introduction

DPL Nspect Execution

Connecting to a Host

Command Output

DPL Nspect Commands and Flags

Help Flags

Output Control Flags

Filter Control Flags

Control Flags for Connecting to BlueField

Misc Flags

DPL Nspect Commands

System-info

Devices

Counters

Tables

Query

Different Output Option Flags

Graph (requires the --low-level flag)

Illustration

Commands and Flags Summary

This page describes the DPL Nspect tool, which is used to inspect DPL programs loaded onto the NVIDIA® BlueField® DPU.

Introduction

DPL Nspect is a CLI tool with several key use cases:

- Viewing the dynamic state of DPL program components, including tables, entries, keys, actions, and counters
- Recording a debug session and generating an archive that can be opened in the DPL Debugger
- Dumping the state of a DPL program into an archive for offline analysis

DPL Nspect Execution

Running `dpl_nspect .sh` without arguments displays the basic usage information. For a more detailed usage guide, use the `-h` or `--help` argument. This argument also provides command-specific usage details (e.g., `dpl_nspect .sh [command] -h`).

Connecting to a Host

DPL Nspect requires a connection to a local or remote host where the `dpl_nspect_server` is running. The host address and port can be configured using the environment variable. For example: `DPL_NSPECT_SERVER_ADDRESS=127.0.0.1:9560`.

Command Output

The output format of executed commands varies depending on the command. In some cases, the output can be modified using additional arguments to enable integration with automated scripts.

DPL Nspect Commands and Flags

The `dpl_nspect` offers various commands for interacting with the DPL program and BlueField:

- `system-info` – Displays BlueField system information, including hardware model, OS name and version, and OFED version
- `devices` – Lists all configured devices on BlueField, including their ID, name, virtual devices, and the loaded DPL program name

Note

Virtual devices are software abstractions representing real or emulated hardware components. They provide a standardized interface for applications to interact with BlueField resources, regardless of the underlying physical hardware. For example:

- Virtual network interfaces (VNIs) – Logical representations of physical NICs, allowing multiple virtual networks to share a single physical interface
 - Virtual processing units (VPUs) – Software-defined processing units within BlueField that can be allocated to applications for tasks like packet processing or security offloading
 - Virtual storage devices – Software-defined representations of physical storage, offering flexibility in managing and allocating storage resources
- `counters` – Displays details and values of DPL counters, including the associated table, offset, and value
 - `tables` – Lists the P4 tables of DPL programs loaded on BlueField, detailing their keys, actions, counters, and references to the program's source location
 - `query` – Lists P4 table entries, showing keys, actions, and the priority of each key set

- `graph` – Generates a DPL pipeline graph using DOT language, which can be rendered with DOT-compatible visualization tools
- `hw-steering` – Displays a hardware steering rules dump, providing a low-level view of hardware entries
- `debug` – starts a debug session

Help Flags

To display details of `dpl_nspect` options usage use any of the following flags:

- `help`, `-h`, `--help` – Displays the help message and exits

Output Control Flags

The following optional flags can be used independently to modify command output format and behavior:

- `-ll`, `--low-level` (default: false) – Displays low-level internal information (DOCA HWS implementation)
- `-ni`, `--non-interactive` (default: false) – Disables interactive output mode. Interactive mode outputs data in chunks similar to the Linux `less` command, making it easier to analyze large outputs. Disabling this mode may be necessary when running `dpl_nspect` commands in automated scripts.
- `-nh`, `--no-hints` – Disables hints for additional information
- `--csv` (default: false) – Outputs command results in CSV format
- `--json` (default: false) – Outputs command results in JSON format
- Flags for the `dpl_nspect counters` command:
 - `--top-packets TOP_PACKETS` (default: false) – Displays the top entries with the highest packet counter values

- `--top-bytes TOP_BYTES` (default: false) – Displays the top entries with the highest byte counter values
- Flags for the `dpl_nspect_graph` counters command:
 - `--url` (default: false) – Encodes the output as a URL that can be directly pasted into a web browser for quick rendering using online visualization tools
 - `--type` – Specifies the type of graph to generate

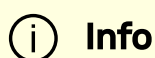
Filter Control Flags

The following flags allow filtering of data based on specific criteria:

- `-d DEVICE_IDS, --device_ids DEVICE_IDS` (default: None) – Specifies a comma-separated list of device IDs to filter, e.g., `1000,2000`. If not specified, data is retrieved for all devices.
- `--table_ids TABLE_IDS` (default: `[]`) – Filters results by specific table IDs
- `--table_names TABLE_NAMES` (default: `[]`) – Filters results by specific table names
- `--counter-names COUNTER_NAMES` (default: `[]`) – Filters results by specific counter names
- `--indices INDICES` (default: `[]`) – Filters results by specific entry indices

Control Flags for Connecting to BlueField

- `-a ADDRESS, --address ADDRESS` – Specifies the `dpl_nspect_server` socket address in the format `[IPv4/IPv6][:port]`.



Info

For IPv6 addresses, if a port is specified, the address must be enclosed in square brackets. Examples:

- `ipv6:[2607:f8b0:400e:c00::ef]:443`
- `ipv6[::]:1234`

- This parameter is mandatory, unless the environment variable `DPL_NSPECT_SERVER_ADDRESS` is set with the address and port. For example:

```
DPL_NSPECT_SERVER_ADDRESS=127.0.0.1:9560
```

- `--timeout TIMEOUT` (default: `0`) – Specifies the connectivity timeout (optional)

Misc Flags

- `--version`, `-V` – Displays the program's version number and exits

DPL Nspect Commands

System-info

```
dpl_nspect system-info
```

| DESCRIPTION | VALUE |
|--------------|--------|
| DOCA Version | 2.10.0 |

| DESCRIPTION | VALUE |
|-----------------------------|---|
| Debug Schema Version | Device ID: 1000, doctype: /dplp4c/devtools/base/v6, target_doctype: /dplp4c/devtools/docca/v1 |
| Linux OS version | Ubuntu 22.04 LTS |
| OFED Version | MLNX_OFED_LINUX-24.01-0.1.7.0 |
| DPL Runtime Service version | Not Found |
| RDMA Core Version | 99999P4.2410m1nx54-1.P4.109.gec00b3c78ef6 |

Devices

```
dpl_nspect devices
```

| ID | NAME | PCI ADDRESS | PORT ID | NAME | NUMBER | DPL PROGRAM | FW VERSION | HW MODEL |
|------|------|--------------|---------|---------------|--------|-------------|------------|--|
| 1000 | br9 | 0000:65:00.0 | 0 | enp101s0f0np0 | 50 | alpha2_d2 | 32.38.0588 | BlueField-3 integrated ConnectX-7 network controller |
| | | | 4 | eth3 | 5 | | | |
| | | | 1 | eth0 | 2 | | | |
| | | | 2 | eth1 | 3 | | | |
| | | | 3 | eth2 | 4 | | | |

Counters

dpl_nspect counters

DEVICE: 1000, NV_COUNTER: alpha2_drop2.decap.decap_counter, SCOPE: alpha2_drop2, TYPE: Pkts/Bts, SIZE: 4, IN USE: 3

| IDX | P4 | ENTRY OFFSET | VALUE |
|-----|-------------|--------------|----------|
| 0 | decap_table | 1 | (2, 140) |
| 1 | decap_table | 2 | (0, 0) |
| 2 | decap_table | 0 | (0, 0) |

DEVICE: 1000, NV_COUNTER: alpha2_drop2.encap.encap_counter, SCOPE: alpha2_drop2, TYPE: Pkts/Bts, SIZE: 4, IN USE: 3

| IDX | P4 | ENTRY OFFSET | VALUE |
|-----|-------------|--------------|--------|
| 0 | encap_table | 0 | (0, 0) |
| 1 | encap_table | 2 | (0, 0) |
| 2 | encap_table | 1 | (0, 0) |

DEVICE: 1000, NV_DIRECT_COUNTER: alpha2_drop2.gtp.gtp_counter, SCOPE: alpha2_drop2, TYPE: Pkts/Bts, SIZE: 128, IN USE: 3

| IDX | P4 | ENTRY OFFSET | VALUE |
|-----|-----------|--------------|--------|
| 0 | gtp_table | 2 | (0, 0) |
| 1 | gtp_table | 1 | (0, 0) |
| 2 | gtp_table | 0 | (0, 0) |

The value is in the following format: (number of packets, number of bytes)

Tables

dpl_nspect tables

| DEVICE: 1000 | | | | | | |
|------------------|--------------------------------|---|---|------------------------------|---------|------------------|
| ID | TABLE | KEYS | ACTIONS | COUNTER | ENTRIES | SOURCES |
| 449 637 66 | alpha2_drop2.decap.decap_table | std_meta.ingress_port headers.gre.\$valid\$ headers.vxlan.\$valid\$ | alpha2_drop2.decap.send_to_port alpha2_drop2.decap.decap_gre alpha2_drop2.decap.decap_vxlan NoAction | | 4 | alpha2_d2.p4:235 |
| 415 214 46 | alpha2_drop2.encap.encap_table | std_meta.ingress_port headers.ipv4.src_addr | alpha2_drop2.encap.send_to_port alpha2_drop2.encap.encap_gre alpha2_drop2.encap.encap_vxlan NoAction | | 4 | alpha2_d2.p4:173 |
| 343 181 61 | alpha2_drop2.gtp.gtp_table | std_meta.ingress_port headers.gtpv1.teid | alpha2_drop2.gtp.send_to_port alpha2_drop2.gtp.drop | alpha2_drop2.gtp.gtp_counter | 3 | alpha2_d2.p4:102 |

Query

dp1_nspect query

DEVICE: 1000, TABLE: alpha2_drop2.gtp.gtp_table, ID: 34318161, SOURCE: alpha2_d2.p4:102

| PRI O | P4 KEY | VALU E | MAS K | TYP E | P4 ACTION |
|----------|---|------------|----------|------------------------|---|
| 0 | std_meta.ingress_port headers.gtpv1.teid | 0x0 0x1 | | exac t exac t | alpha2_drop2.gtp.send_to_port(port=0x1) |
| 0 | std_meta.ingress_port headers.gtpv1.teid | 0x1 0x1 | | exac t exac t | alpha2_drop2.gtp.send_to_port(port=0x0) |
| 0 | ALWAYS_HIT | | | | alpha2_drop2.gtp.drop() |

DEVICE: 1000, TABLE: alpha2_drop2.encap.encap_table, ID: 41521446, SOURCE: alpha2_d2.p4:173

| PRI O | P4 KEY | VALUE | MASK | TYPE | P4 ACTION |
|----------|--|-----------------------|----------------|----------------------|--|
| 3 | std_meta.ingress_port headers.ipv4.src_addr | 0x0 0x10000 000 | 0xff000 000 | exact terna ry | alpha2_drop2.encap.send_to_port(port=0x4) |
| 2 | std_meta.ingress_port headers.ipv4.src_addr | 0x0 0x20000 000 | 0xff000 000 | exact terna ry | alpha2_drop2.encap.encap_gre(port=0x2, src_mac=0x112233445566, dst_mac=0x10203040506, sip=0x1010101, dip=0x2020202, key=0xabcd) |
| 1 | std_meta.ingress_port | 0x0 | 0xff000 000 | exact | alpha2_drop2.encap.encap_vxlan(|

DEVICE: 1000, TABLE: alpha2_drop2.encap.encap_table, ID: 41521446, SOURCE: alpha2_d2.p4:173

| | | | | | |
|---|-----------------------|------------|--|---------|---|
| | headers.ipv4.src_addr | 0x20000000 | | ternary | port=0x3, src_mac=0x112233445566, dst_mac=0x10203040506, sip=0x1010101, dip=0x2020202, vni=0xabcd) |
| 0 | ALWAYS_HIT | | | | NoAction() |

DEVICE: 1000, TABLE: alpha2_drop2.decap.decap_table, ID: 44963766, SOURCE: alpha2_d2.p4:235

| PRI O | P4 KEY | VALU E | MAS K | TYP E | P4 ACTION |
|-------|---|-------------------|-------|-------------------------|--|
| 0 | std_meta.ingress_port headers.gre.\$valid\$ headers.vxlan.\$valid\$ | 0x3 0x0 0x1 | | exact exact exact | alpha2_drop2.decap.decap_vxlan(port=0x0, src_mac=0xaabbccddeeff, dst_mac=0x112233445566) |
| 0 | std_meta.ingress_port headers.gre.\$valid\$ headers.vxlan.\$valid\$ | 0x4 0x0 0x0 | | exact exact exact | alpha2_drop2.decap.send_to_port(port=0x0) |
| 0 | std_meta.ingress_port headers.gre.\$valid\$ headers.vxlan.\$valid\$ | 0x2 0x1 0x0 | | exact exact exact | alpha2_drop2.decap.decap_gre(port=0x0) |
| 0 | ALWAYS_HIT | | | | NoAction() |

Different Output Option Flags

- `--csv`

```
dpl_nspect --csv tables
```

```
"DEVICE: 1000"  
"ID", "TABLE", "KEYS", "ACTIONS", "COUNTER", "ENTRIES", "SOURCE"  
"44963766", "alpha2_drop2.decap.decap_table", "std_meta.ingress_port  
headers.gre.$valid$  
headers.vxlan.$valid$","alpha2_drop2.decap.send_to_port  
alpha2_drop2.decap.decap_gre  
alpha2_drop2.decap.decap_vxlan  
NoAction","","4","alpha2_d2.p4:235"  
"41521446", "alpha2_drop2.encap.encap_table", "std_meta.ingress_port  
headers.ipv4.src_addr","alpha2_drop2.encap.send_to_port  
alpha2_drop2.encap.encap_gre  
alpha2_drop2.encap.encap_vxlan  
NoAction","","4","alpha2_d2.p4:173"  
"34318161", "alpha2_drop2.gtp.gtp_table", "std_meta.ingress_port  
headers.gtpv1.teid","alpha2_drop2.gtp.send_to_port  
alpha2_drop2.gtp.drop","alpha2_drop2.gtp.gtp_counter","3","alpha2_d2
```

- `--json`

```
dpl_nspect --json tables
```

```
[  
  [  
    {  
      "DEVICE": 1000  
    },  
    {
```

```

"ID" : 44963766 ,
"TABLE" : "alpha2_drop2.decap.decap_table",
"KEYS" : [
    "std_meta.ingress_port",
    "headers.gre.$valid$",
    "headers.vxlan.$valid$"
],
"ACTIONS" : [
    "alpha2_drop2.decap.send_to_port",
    "alpha2_drop2.decap.decap_gre",
    "alpha2_drop2.decap.decap_vxlan",
    "NoAction"
],
"COUNTER" : "",
"ENTRIES" : 4,
"SOURCE" : {
    "filename" : "/tmp/p4src/alpha2_d2.p4",
    "line" : 235,
    "column" : 11
}
},
{
"ID" : 41521446,
"TABLE" : "alpha2_drop2.encap.encap_table",
"KEYS" : [
    "std_meta.ingress_port",
    "headers.ipv4.src_addr"
],
"ACTIONS" : [
    "alpha2_drop2.encap.send_to_port",
    "alpha2_drop2.encap.encap_gre",
    "alpha2_drop2.encap.encap_vxlan",
    "NoAction"
],
"COUNTER" : "",
"ENTRIES" : 4,

```

```

    "SOURCE": {
      "filename": "/tmp/p4src/alpha2_d2.p4",
      "line": 173,
      "column": 11
    }
  },
  {
    "ID": 34318161,
    "TABLE": "alpha2_drop2.gtp.gtp_table",
    "KEYS": [
      "std_meta.ingress_port",
      "headers.gtpv1.teid"
    ],
    "ACTIONS": [
      "alpha2_drop2.gtp.send_to_port",
      "alpha2_drop2.gtp.drop"
    ],
    "COUNTER": "alpha2_drop2.gtp.gtp_counter",
    "ENTRIES": 3,
    "SOURCE": {
      "filename": "/tmp/p4src/alpha2_d2.p4",
      "line": 102,
      "column": 11
    }
  }
]
]

```

Graph (requires the --low-level flag)

- No extra flags

```
dpl_nspect -ll graph --type pipeline_low_level
```

```
Device 1000:
digraph hybrid_pipeline {
    graph [compound=true]
        // Legend
        subgraph cluster_legend {
            hal_table [label=<<B>HAL Table ID</B><BR/>
[Implements P4 Objects]> shape=diamond]
            legend_invis [label=legend_invis shape=point
style=invis width=0]
            p4_control [label="P4 Control" color=blue
shape=rectangle]
        }
        alpha2_drop2_start [label=start color=green
shape=none]
        legend_invis -> alpha2_drop2_start [style=invis]
        // Pipeline Stage: main
        subgraph cluster_alpha2_drop2 {
            graph [color=blue label=alpha2_drop2
shape=rectangle]
                24 [label=<<B>24</B><BR/>[alpha2_drop2]>
shape=diamond]
                alpha2_drop2_start -> 24
                48 [label=<<B>48</B><BR/>
[alpha2_drop2.gtp.gtp_table]> shape=diamond]
                DROP [label=DROP fontcolor=red shape=box]
                48 -> DROP
                alpha2_drop2_end [label=end color=red
shape=none]
                48 -> alpha2_drop2_end
                36 [label=<<B>36</B><BR/>[alpha2_drop2]>
shape=diamond]
```



```

36 -> 48
56 [label=<<B>56</B><BR/>
[alpha2_drop2.decap.decap_table]> shape=diamond]
56 -> alpha2_drop2_end
44 [label=<<B>44</B><BR/>[alpha2_drop2]>
shape=diamond]
44 -> 56
44 -> alpha2_drop2_end
32 [label=<<B>32</B><BR/>[alpha2_drop2]>
shape=diamond]
32 -> 44
32 -> alpha2_drop2_end
52 [label=<<B>52</B><BR/>
[alpha2_drop2.encap.encap_table]> shape=diamond]
52 -> 32
52 -> alpha2_drop2_end
40 [label=<<B>40</B><BR/>[alpha2_drop2]>
shape=diamond]
55 [label=<<B>55</B><BR/>
[alpha2_drop2.encap.encap_table]> shape=diamond]
40 -> 55
40 -> 32
40 -> alpha2_drop2_end
28 [label=<<B>28</B><BR/>[alpha2_drop2]>
shape=diamond]
28 -> 40
28 -> alpha2_drop2_end
36 -> 28
36 -> alpha2_drop2_end
24 -> 36
24 -> alpha2_drop2_end
}
}

```

- `--url` – generates an HTTP URL link to the program's control flow graph, in dot format, rendered using <https://dreampuf.github.io/GraphvizOnline/>.

```
dpl_nspect -ll graph --url --type pipeline_low_level
```



- Generate the program's parser graph in dot format:

```
dpl_nspect -ll graph --type parser
```

```
Device 1000:
// Parser packet_parser
digraph packet_parser {
graph [concentrate=true splines=compound]
// Legend
subgraph cluster_legend {
fixed_state [label="Fixed State" style=filled]
```

```

legend_invis [label=legend_invis shape=point style=invis
width=0]
flex_state [label="Flex State"]
default_transition_note [label="All states have an hidden transition to
accept\nEdges are annotated with their match value." shape=none]
legend_invis -> default_transition_note [style=invis]
}
start [label=start color=green shape=point]
default_transition_note -> start [style=invis]
nv_parse_ethernet [label=ethernet id=nv_parse_ethernet
style=filled]
start -> nv_parse_ethernet
nv_parse_vlan [label=vlan id=nv_parse_vlan style=filled]
nv_parse_ethernet -> nv_parse_vlan [label="0x8100" color=gray
id="nv_parse_ethernet-nv_parse_vlan"]
nv_parse_ipv4 [label=ipv4 id=nv_parse_ipv4 style=filled]
nv_parse_ethernet -> nv_parse_ipv4 [label="0x800" color=gray
id="nv_parse_ethernet-nv_parse_ipv4"]
nv_parse_ipv6 [label=ipv6 id=nv_parse_ipv6 style=filled]
nv_parse_ethernet -> nv_parse_ipv6 [label="0x86dd" color=gray
id="nv_parse_ethernet-nv_parse_ipv6"]
nv_parse_mpls [label=mpls id=nv_parse_mpls style=filled]
nv_parse_ethernet -> nv_parse_mpls [label="0x8847" color=gray
id="nv_parse_ethernet-nv_parse_mpls"]
nv_parse_geneve [label=geneve id=nv_parse_geneve
style=filled]
nv_parse_inner_ipv4 [label=inner_ipv4 id=nv_parse_inner_ipv4
style=filled]
nv_parse_geneve -> nv_parse_inner_ipv4 [label="0x800"
color=gray id="nv_parse_geneve-nv_parse_inner_ipv4"]
nv_parse_inner_ipv6 [label=inner_ipv6 id=nv_parse_inner_ipv6
style=filled]
nv_parse_geneve -> nv_parse_inner_ipv6 [label="0x86dd"
color=gray id="nv_parse_geneve-nv_parse_inner_ipv6"]
nv_parse_inner_mpls [label=inner_mpls id=nv_parse_inner_mpls
style=filled]

```

```
nv_parse_geneve -> nv_parse_inner_mpls [label="0x8847"  
color=gray id="nv_parse_geneve-nv_parse_inner_mpls"]  
nv_parse_inner_ethernet [label=inner_ethernet  
id=nv_parse_inner_ethernet style=filled]  
nv_parse_geneve -> nv_parse_inner_ethernet [label="0x6558"  
color=gray id="nv_parse_geneve-nv_parse_inner_ethernet"]  
nv_parse_gre [label=gre id=nv_parse_gre style=filled]  
nv_parse_gre -> nv_parse_inner_ipv4 [label="0x800" color=gray  
id="nv_parse_gre-nv_parse_inner_ipv4"]  
nv_parse_gre -> nv_parse_inner_ipv6 [label="0x86dd" color=gray  
id="nv_parse_gre-nv_parse_inner_ipv6"]  
nv_parse_gre -> nv_parse_inner_mpls [label="0x8847" color=gray  
id="nv_parse_gre-nv_parse_inner_mpls"]  
nv_parse_nvgre_vsid [label=nvgre_vsid id=nv_parse_nvgre_vsid  
style=filled]  
nv_parse_gre -> nv_parse_nvgre_vsid [label="0x6558" color=gray  
id="nv_parse_gre-nv_parse_nvgre_vsid"]  
nv_parse_icmp [label=icmp id=nv_parse_icmp style=filled]  
nv_parse_icmpv6 [label=icmpv6 id=nv_parse_icmpv6  
style=filled]  
nv_parse_inner_vlan [label=inner_vlan id=nv_parse_inner_vlan  
style=filled]  
nv_parse_inner_ethernet -> nv_parse_inner_vlan [label="0x8100"  
color=gray id="nv_parse_inner_ethernet-nv_parse_inner_vlan"]  
nv_parse_inner_ethernet -> nv_parse_inner_ipv4 [label="0x800"  
color=gray id="nv_parse_inner_ethernet-nv_parse_inner_ipv4"]  
nv_parse_inner_ethernet -> nv_parse_inner_ipv6 [label="0x86dd"  
color=gray id="nv_parse_inner_ethernet-nv_parse_inner_ipv6"]  
nv_parse_inner_ethernet -> nv_parse_inner_mpls [label="0x8847"  
color=gray id="nv_parse_inner_ethernet-nv_parse_inner_mpls"]  
nv_parse_inner_icmp [label=inner_icmp id=nv_parse_inner_icmp  
style=filled]  
nv_parse_inner_icmpv6 [label=inner_icmpv6  
id=nv_parse_inner_icmpv6 style=filled]  
nv_parse_inner_ipv4 -> nv_parse_inner_icmp [label="0x1"  
color=gray id="nv_parse_inner_ipv4-nv_parse_inner_icmp"]
```

```

nv_parse_inner_tcp [label=inner_tcp id=nv_parse_inner_tcp
style=filled]
nv_parse_inner_ipv4 -> nv_parse_inner_tcp [label="0x6"
color=gray id="nv_parse_inner_ipv4-nv_parse_inner_tcp"]
nv_parse_inner_udp [label=inner_udp id=nv_parse_inner_udp
style=filled]
nv_parse_inner_ipv4 -> nv_parse_inner_udp [label="0x11"
color=gray id="nv_parse_inner_ipv4-nv_parse_inner_udp"]
nv_parse_inner_ipv6 -> nv_parse_inner_icmpv6 [label="0x3a"
color=gray id="nv_parse_inner_ipv6-nv_parse_inner_icmpv6"]
nv_parse_inner_ipv6 -> nv_parse_inner_tcp [label="0x6"
color=gray id="nv_parse_inner_ipv6-nv_parse_inner_tcp"]
nv_parse_inner_ipv6 -> nv_parse_inner_udp [label="0x11"
color=gray id="nv_parse_inner_ipv6-nv_parse_inner_udp"]
nv_parse_inner_mpls1 [label=inner_mpls1
id=nv_parse_inner_mpls1 style=filled]
nv_parse_inner_mpls -> nv_parse_inner_mpls1 [label="0x0"
color=gray id="nv_parse_inner_mpls-nv_parse_inner_mpls1"]
nv_parse_inner_mpls_next_header
[label=nv_parse_inner_mpls_next_header
id=nv_parse_inner_mpls_next_header style=filled]
nv_parse_inner_mpls -> nv_parse_inner_mpls_next_header
[label="0x1" color=gray id="nv_parse_inner_mpls-nv_parse_inner_mpls_next_header"]
nv_parse_inner_mpls2 [label=inner_mpls2
id=nv_parse_inner_mpls2 style=filled]
nv_parse_inner_mpls1 -> nv_parse_inner_mpls2 [label="0x0"
color=gray id="nv_parse_inner_mpls1-nv_parse_inner_mpls2"]
nv_parse_inner_mpls1 -> nv_parse_inner_mpls_next_header
[label="0x1" color=gray id="nv_parse_inner_mpls1-
nv_parse_inner_mpls_next_header"]
nv_parse_inner_mpls3 [label=inner_mpls3
id=nv_parse_inner_mpls3 style=filled]
nv_parse_inner_mpls2 -> nv_parse_inner_mpls3 [label="0x0"
color=gray id="nv_parse_inner_mpls2-nv_parse_inner_mpls3"]
nv_parse_inner_mpls2 -> nv_parse_inner_mpls_next_header
[label="0x1" color=gray id="nv_parse_inner_mpls2-

```

```

nv_parse_inner_mpls_next_header" ]
nv_parse_inner_mpls4 [label=inner_mpls4
id=nv_parse_inner_mpls4 style=filled]
nv_parse_inner_mpls3 -> nv_parse_inner_mpls4 [label="0x0"
color=gray id="nv_parse_inner_mpls3-nv_parse_inner_mpls4"]
nv_parse_inner_mpls3 -> nv_parse_inner_mpls_next_header
[label="0x1" color=gray id="nv_parse_inner_mpls3-
nv_parse_inner_mpls_next_header" ]
nv_parse_inner_mpls4 -> nv_parse_inner_mpls_next_header
[label="0x1" color=gray id="nv_parse_inner_mpls4-
nv_parse_inner_mpls_next_header" ]
nv_parse_inner_mpls_next_header -> nv_parse_inner_ipv4
[label="0x4" color=gray id="nv_parse_inner_mpls_next_header-nv_parse_inner_ipv4"]
nv_parse_inner_mpls_next_header -> nv_parse_inner_ipv6
[label="0x6" color=gray id="nv_parse_inner_mpls_next_header-nv_parse_inner_ipv6"]
nv_parse_inner_vlan -> nv_parse_inner_ipv4 [label="0x800"
color=gray id="nv_parse_inner_vlan-nv_parse_inner_ipv4"]
nv_parse_inner_vlan -> nv_parse_inner_ipv6 [label="0x86dd"
color=gray id="nv_parse_inner_vlan-nv_parse_inner_ipv6"]
nv_parse_inner_vlan -> nv_parse_inner_mpls [label="0x8847"
color=gray id="nv_parse_inner_vlan-nv_parse_inner_mpls"]
nv_parse_ipsec_esp [label=esp id=nv_parse_ipsec_esp
style=filled]
nv_parse_ipv4 -> nv_parse_inner_ipv4 [label="0x4" color=gray
id="nv_parse_ipv4-nv_parse_inner_ipv4"]
nv_parse_ipv4 -> nv_parse_inner_ipv6 [label="0x29" color=gray
id="nv_parse_ipv4-nv_parse_inner_ipv6"]
nv_parse_ipv4 -> nv_parse_icmp [label="0x1" color=gray
id="nv_parse_ipv4-nv_parse_icmp"]
nv_parse_tcp [label=tcp id=nv_parse_tcp style=filled]
nv_parse_ipv4 -> nv_parse_tcp [label="0x6" color=gray
id="nv_parse_ipv4-nv_parse_tcp"]
nv_parse_udp [label=udp id=nv_parse_udp style=filled]
nv_parse_ipv4 -> nv_parse_udp [label="0x11" color=gray
id="nv_parse_ipv4-nv_parse_udp"]

```

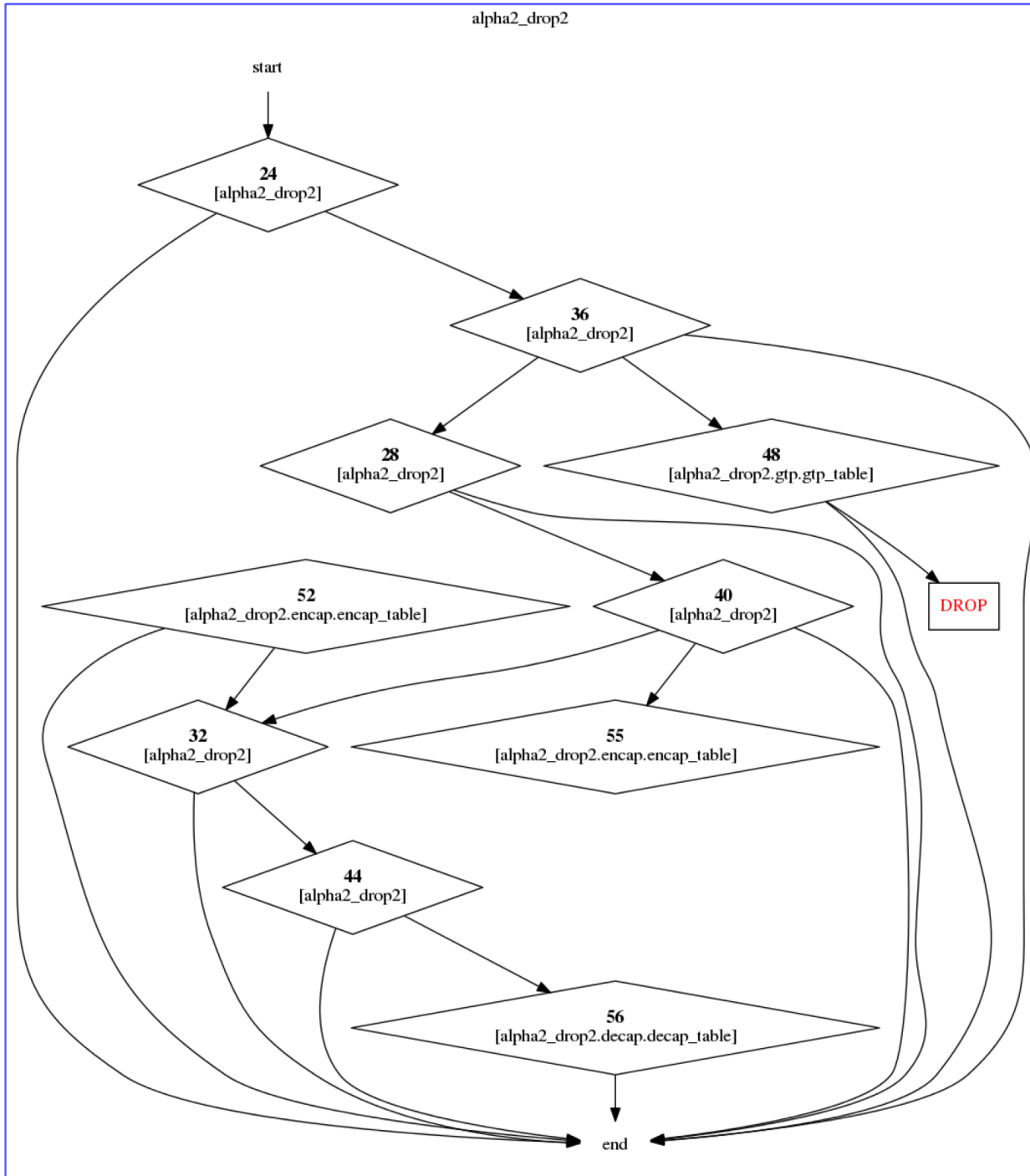
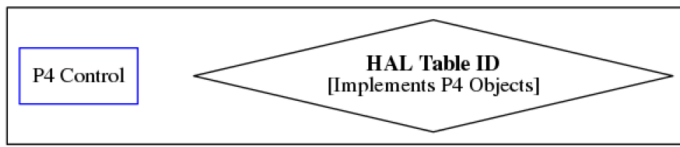
```

nv_parse_ipv4 -> nv_parse_gre [label="0x2f" color=gray
id="nv_parse_ipv4-nv_parse_gre"]
nv_parse_ipv4 -> nv_parse_ipsec_esp [label="0x32" color=gray
id="nv_parse_ipv4-nv_parse_ipsec_esp"]
nv_parse_ipv6 -> nv_parse_inner_ipv4 [label="0x4" color=gray
id="nv_parse_ipv6-nv_parse_inner_ipv4"]
nv_parse_ipv6 -> nv_parse_inner_ipv6 [label="0x29" color=gray
id="nv_parse_ipv6-nv_parse_inner_ipv6"]
nv_parse_ipv6 -> nv_parse_icmpv6 [label="0x3a" color=gray
id="nv_parse_ipv6-nv_parse_icmpv6"]
nv_parse_ipv6 -> nv_parse_tcp [label="0x6" color=gray
id="nv_parse_ipv6-nv_parse_tcp"]
nv_parse_ipv6 -> nv_parse_udp [label="0x11" color=gray
id="nv_parse_ipv6-nv_parse_udp"]
nv_parse_ipv6 -> nv_parse_gre [label="0x2f" color=gray
id="nv_parse_ipv6-nv_parse_gre"]
nv_parse_ipv6 -> nv_parse_ipsec_esp [label="0x32" color=gray
id="nv_parse_ipv6-nv_parse_ipsec_esp"]
nv_parse_mpls1 [label=mpls1 id=nv_parse_mpls1 style=filled]
nv_parse_mpls -> nv_parse_mpls1 [label="0x0" color=gray
id="nv_parse_mpls-nv_parse_mpls1"]
nv_parse_mpls_next_header [label=nv_parse_mpls_next_header
id=nv_parse_mpls_next_header style=filled]
nv_parse_mpls -> nv_parse_mpls_next_header [label="0x1"
color=gray id="nv_parse_mpls-nv_parse_mpls_next_header"]
nv_parse_mpls2 [label=mpls2 id=nv_parse_mpls2 style=filled]
nv_parse_mpls1 -> nv_parse_mpls2 [label="0x0" color=gray
id="nv_parse_mpls1-nv_parse_mpls2"]
nv_parse_mpls1 -> nv_parse_mpls_next_header [label="0x1"
color=gray id="nv_parse_mpls1-nv_parse_mpls_next_header"]
nv_parse_mpls3 [label=mpls3 id=nv_parse_mpls3 style=filled]
nv_parse_mpls2 -> nv_parse_mpls3 [label="0x0" color=gray
id="nv_parse_mpls2-nv_parse_mpls3"]
nv_parse_mpls2 -> nv_parse_mpls_next_header [label="0x1"
color=gray id="nv_parse_mpls2-nv_parse_mpls_next_header"]
nv_parse_mpls4 [label=mpls4 id=nv_parse_mpls4 style=filled]

```

```
nv_parse_mpls3 -> nv_parse_mpls4 [label="0x0" color=gray
id="nv_parse_mpls3-nv_parse_mpls4"]
nv_parse_mpls3 -> nv_parse_mpls_next_header [label="0x1"
color=gray id="nv_parse_mpls3-nv_parse_mpls_next_header"]
nv_parse_mpls4 -> nv_parse_mpls_next_header [label="0x1"
color=gray id="nv_parse_mpls4-nv_parse_mpls_next_header"]
nv_parse_mpls_next_header -> nv_parse_ipv4 [label="0x4"
color=gray id="nv_parse_mpls_next_header-nv_parse_ipv4"]
nv_parse_mpls_next_header -> nv_parse_ipv6 [label="0x6"
color=gray id="nv_parse_mpls_next_header-nv_parse_ipv6"]
nv_parse_nvgre_vsid -> nv_parse_inner_ethernet [label="0x0"
color=gray id="nv_parse_nvgre_vsid-nv_parse_inner_ethernet"]
nv_parse_psp [label=psp id=nv_parse_psp style=filled]
parse_gtp [label=gtpv1 id=parse_gtp style=""]
nv_parse_udp -> parse_gtp [label="0x868" color=black
id="nv_parse_udp-parse_gtp"]
nv_parse_vxlan [label=vxlan id=nv_parse_vxlan style=filled]
nv_parse_udp -> nv_parse_vxlan [label="0x12b5" color=gray
id="nv_parse_udp-nv_parse_vxlan"]
nv_parse_vxlan_gpe [label=vxlan_gpe id=nv_parse_vxlan_gpe
style=filled]
nv_parse_udp -> nv_parse_vxlan_gpe [label="0x12b6" color=gray
id="nv_parse_udp-nv_parse_vxlan_gpe"]
nv_parse_udp -> nv_parse_geneve [label="0x17c1" color=gray
id="nv_parse_udp-nv_parse_geneve"]
nv_parse_udp -> nv_parse_inner_mpls [label="0x19eb" color=gray
id="nv_parse_udp-nv_parse_inner_mpls"]
nv_parse_udp -> nv_parse_ipsec_esp [label="0x1194" color=gray
id="nv_parse_udp-nv_parse_ipsec_esp"]
nv_parse_udp -> nv_parse_psp [label="0x3e8" color=gray
id="nv_parse_udp-nv_parse_psp"]
nv_parse_vlan -> nv_parse_ipv4 [label="0x800" color=gray
id="nv_parse_vlan-nv_parse_ipv4"]
nv_parse_vlan -> nv_parse_ipv6 [label="0x86dd" color=gray
id="nv_parse_vlan-nv_parse_ipv6"]
```


It is possible to render the graph to get the following picture:



Commands and Flags Summary

Some command flags are universally applicable to all commands, but others are applicable to only specific commands.

The following table outlines commands and applicable flags:

| | system-info | devices | counters | tables | query | graph | hw-steering | debug |
|-------------------|-------------|---------|----------|--------|-------|-------|-------------|-------|
| --address | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| --time-out | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| --non-interactive | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| --no-hints | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| --csv | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| --json | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| --device-ids | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| --low-level | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| --table-ids | | | ✓ | ✓ | ✓ | | | |
| --table_names | | | ✓ | ✓ | ✓ | | | |
| --counter-names | | | ✓ | | | | | |
| --indices | | | ✓ | | | | | |
| --top-packets | | | ✓ | | | | | |
| --top-bytes | | | ✓ | | | | | |
| --url | | | | | | ✓ | | |
| --type | | | | | | ✓ | | |
| --output-file | | | | | | | | ✓ |
| --debug-device-id | | | | | | | | ✓ |

Notice
This document is provided for information purposes only and shall not be regarded as a warranty of a certain functionality, condition, or quality of a product. NVIDIA Corporation (“NVIDIA”) makes no representations or warranties, expressed or implied, as to the accuracy or completeness of the information contained in this document and assumes no responsibility for any errors contained herein. NVIDIA shall have no liability for the

consequences or use of such information or for any infringement of patents or other rights of third parties that may result from its use. This document is not a commitment to develop, release, or deliver any Material (defined below), code, or functionality. NVIDIA reserves the right to make corrections, modifications, enhancements, improvements, and any other changes to this document, at any time without notice. Customer should obtain the latest relevant information before placing orders and should verify that such information is current and complete. NVIDIA products are sold subject to the NVIDIA standard terms and conditions of sale supplied at the time of order acknowledgement, unless otherwise agreed in an individual sales agreement signed by authorized representatives of NVIDIA and customer ("Terms of Sale"). NVIDIA hereby expressly objects to applying any customer general terms and conditions with regards to the purchase of the NVIDIA product referenced in this document. No contractual obligations are formed either directly or indirectly by this document. NVIDIA products are not designed, authorized, or warranted to be suitable for use in medical, military, aircraft, space, or life support equipment, nor in applications where failure or malfunction of the NVIDIA product can reasonably be expected to result in personal injury, death, or property or environmental damage. NVIDIA accepts no liability for inclusion and/or use of NVIDIA products in such equipment or applications and therefore such inclusion and/or use is at customer's own risk. NVIDIA makes no representation or warranty that products based on this document will be suitable for any specified use. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer's sole responsibility to evaluate and determine the applicability of any information contained in this document, ensure the product is suitable and fit for the application planned by customer, and perform the necessary testing for the application in order to avoid a default of the application or the product. Weaknesses in customer's product designs may affect the quality and reliability of the NVIDIA product and may result in additional or different conditions and/or requirements beyond those contained in this document. NVIDIA accepts no liability related to any default, damage, costs, or problem which may be based on or attributable to: (i) the use of the NVIDIA product in any manner that is contrary to this document or (ii) customer product designs. No license, either expressed or implied, is granted under any NVIDIA patent right, copyright, or other NVIDIA intellectual property right under this document. Information published by NVIDIA regarding third-party products or services does not constitute a license from NVIDIA to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property rights of the third party, or a license from NVIDIA under the patents or other intellectual property rights of NVIDIA. Reproduction of information in this document is permissible only if approved in advance by NVIDIA in writing, reproduced without alteration and in full compliance with all applicable export laws and regulations, and accompanied by all associated conditions, limitations, and notices. THIS DOCUMENT AND ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL NVIDIA BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF NVIDIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Notwithstanding any damages that customer might incur for any reason whatsoever, NVIDIA's aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms of Sale for the product. **Trademarks** NVIDIA and the NVIDIA logo are trademarks and/or registered trademarks of NVIDIA Corporation in the U.S. and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

© Copyright 2025, NVIDIA. PDF Generated on 04/24/2025