daPIPE

Data Plane Incremental Programming Environment

Mario Baldi

Data Center Switching Group

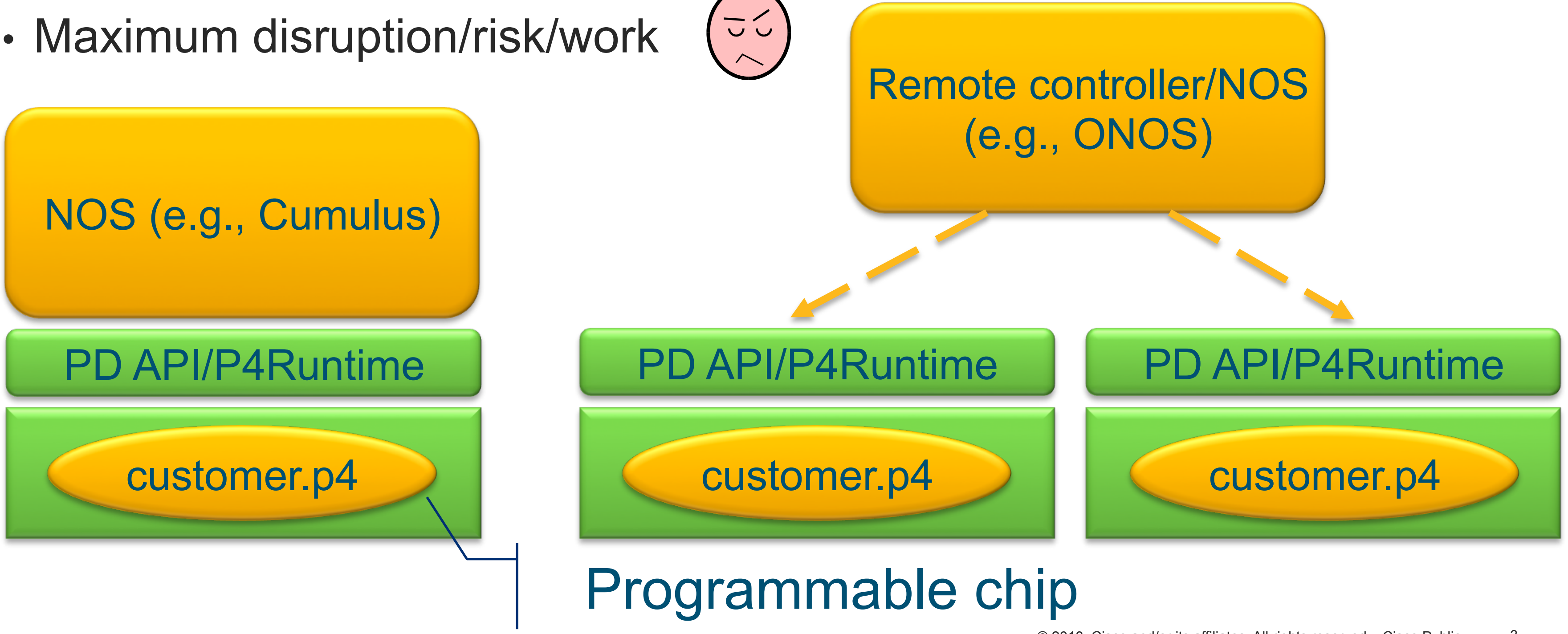
Programmable switches:

What are deployment options?

Whitebox Deployment

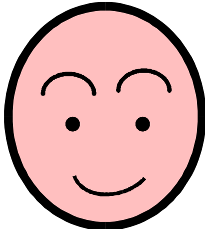
- 🟦 Platform vendor (Cisco)
- 🟩 Chip vendor (Barefoot)
- 🟨 Customer/open source

- Maximum flexibility ☺
- Maximum disruption/risk/work ☹



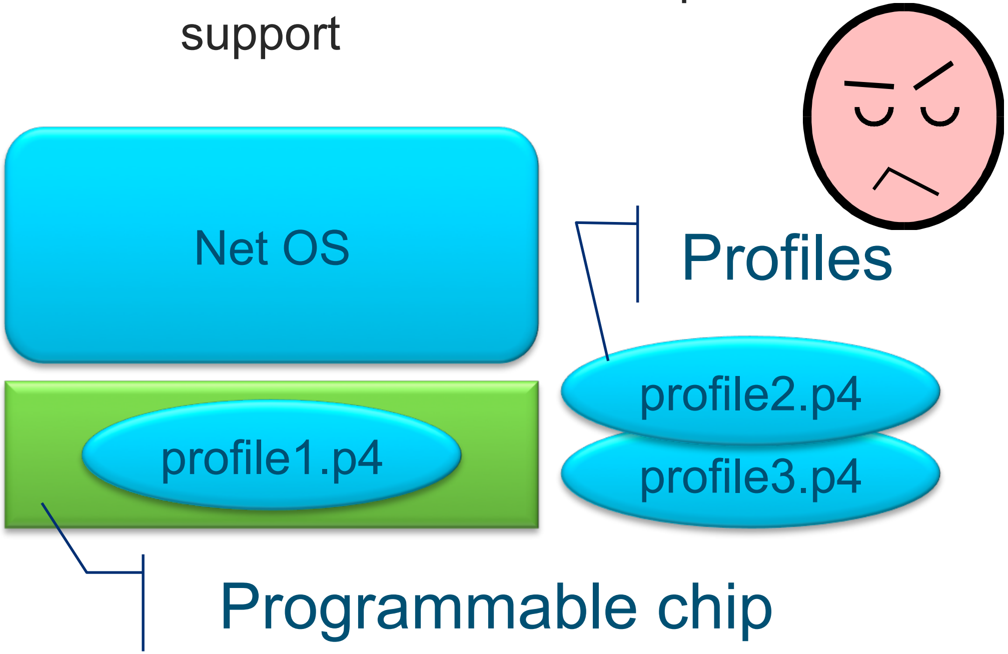
Turn-key Deployment

- Deployment as usual
  - Familiar features and interfaces
- Resource optimization
- Future proof
- Feature agility
- Streaming telemetry



🟦 Platform vendor (Cisco)
🟩 Chip vendor (Barefoot)
🟨 Customer/open source

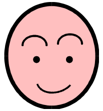
- No flexibility
  - No custom feature and protocol support



Hybrid Deployment

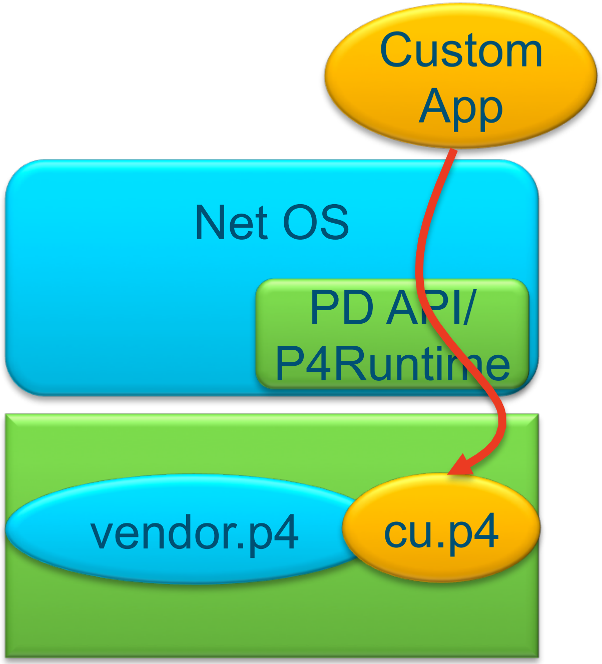
- 🟦 Platform vendor (Cisco)
- 🟩 Chip vendor (Barefoot)
- 🟧 Customer/open source/

- Best of breed



- Deployment as usual
  - Familiar features and interfaces
- Minimum development effort
  - Leverage existing functions in building new features

Minimize disruption and risk!



Challenges

Do not break what works

- Vendor data plane code is well tested
- … and we don't want to need regression testing

Don't want to show, don't want to see

- Vendor code and custom code may be confidential
- Not practical to familiarize with a lot of vendor code to just write a few lines

Resource availability

- Still "limited" on current chips

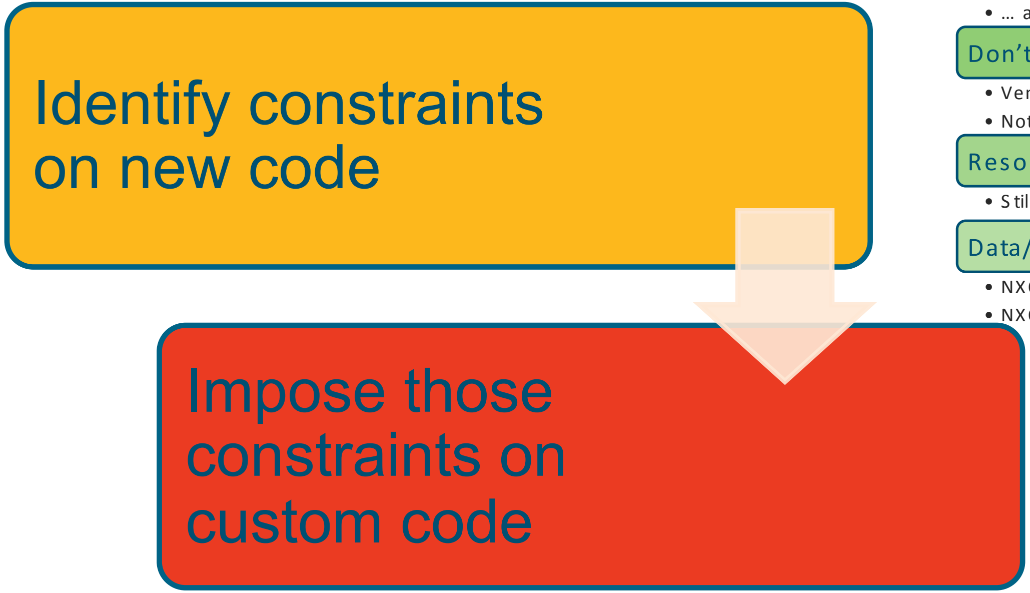
Data/control plane dependence

- Net OS should keep working
- Net OS should not be aware of custom data plane functions

In a nutshell, we need an explicit effort to support

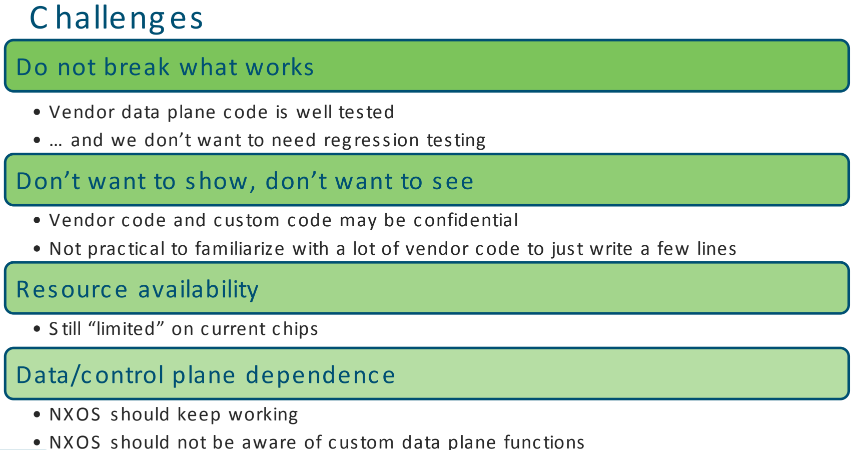
Incremental Programming

How can we address these challenges?



Identify constraints on new code

Impose those constraints on custom code



Challenges

Do not break what works

- Vendor data plane code is well tested
- … and we don’t want to need regression testing

Don’t want to show, don’t want to see

- Vendor code and custom code may be confidential
- Not practical to familiarize with a lot of vendor code to just write a few lines

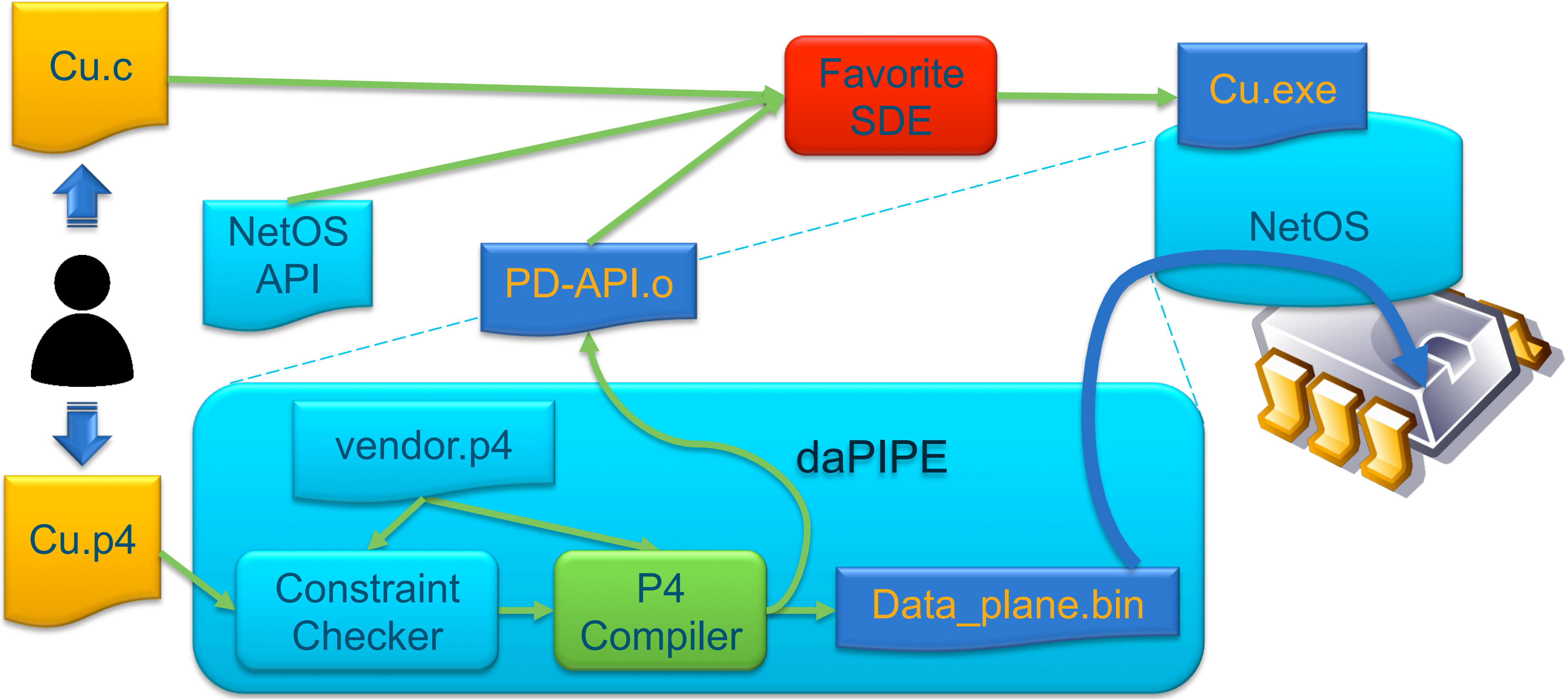
Resource availability

- Still “limited” on current chips

Data/control plane dependence

- NXOS should keep working
- NXOS should not be aware of custom data plane functions

Customer Programming Workflow



Cu.c

Favorite SDE

Cu.exe

NetOS API

PD-API.o

NetOS

vendor.p4

daPIPE

Cu.p4

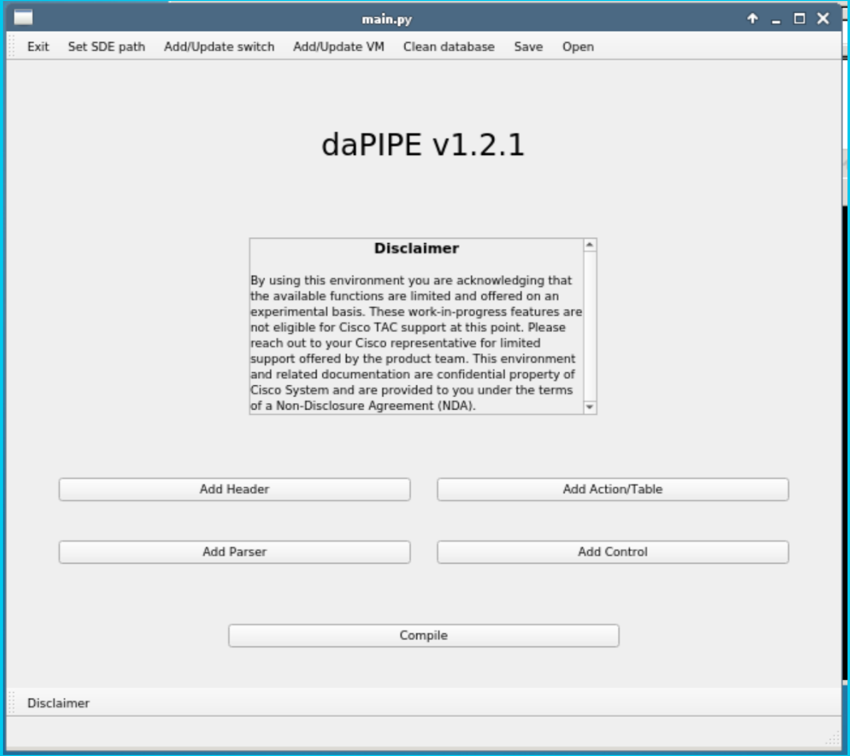
Constraint Checker

P4 Compiler

Data_plane.bin

daPIPE

Data Plane Incremental Programming Environment

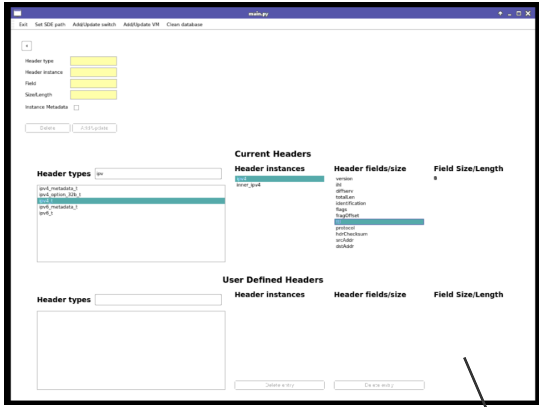


Disclaimer

By using this environment you are acknowledging that the available functions are limited and offered on an experimental basis. These work-in-progress features are not eligible for Cisco TAC support at this point. Please reach out to your Cisco representative for limited support offered by the product team. This environment and related documentation are confidential property of Cisco System and are provided to you under the terms of a Non-Disclosure Agreement (NDA).

Support developers and streamline their task (while enforcing constraints)

Components of the Solution



Header type

Header instance

Field

Size/Length

Instance Metadata □

Current Headers

Header types ipv

ipv4_metadata_t
ipv4_option_32b_t
ipv4_t
ipv6_metadata_t
ipv6_t

Header instances

ipv4
inner_ipv4

Header fields/size

version
ihl
diffserv
totalLen
identification
flags
fragOffset
ttl
protocol
hdrChecksum
srcAddr
dstAddr

Field Size/Length

8

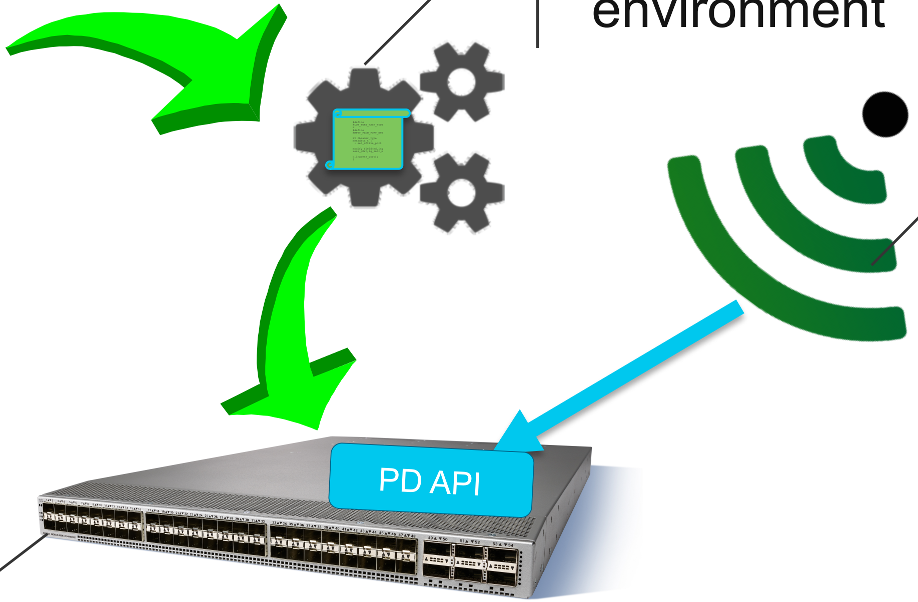
User Defined Headers

Header types

Header instances

Header fields/size

Field Size/Length



daPIPE build environment

Control program

daPIPE Graphical User Interface

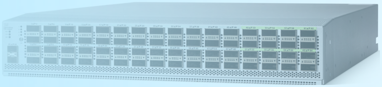
Nexus 34180YC

Nexus 3400 Programmable Switch Family

Shipping

48p 10/25Gb/s SFP + 6p 40/100Gb/s QSFP
Nexus 34180YC

Committed

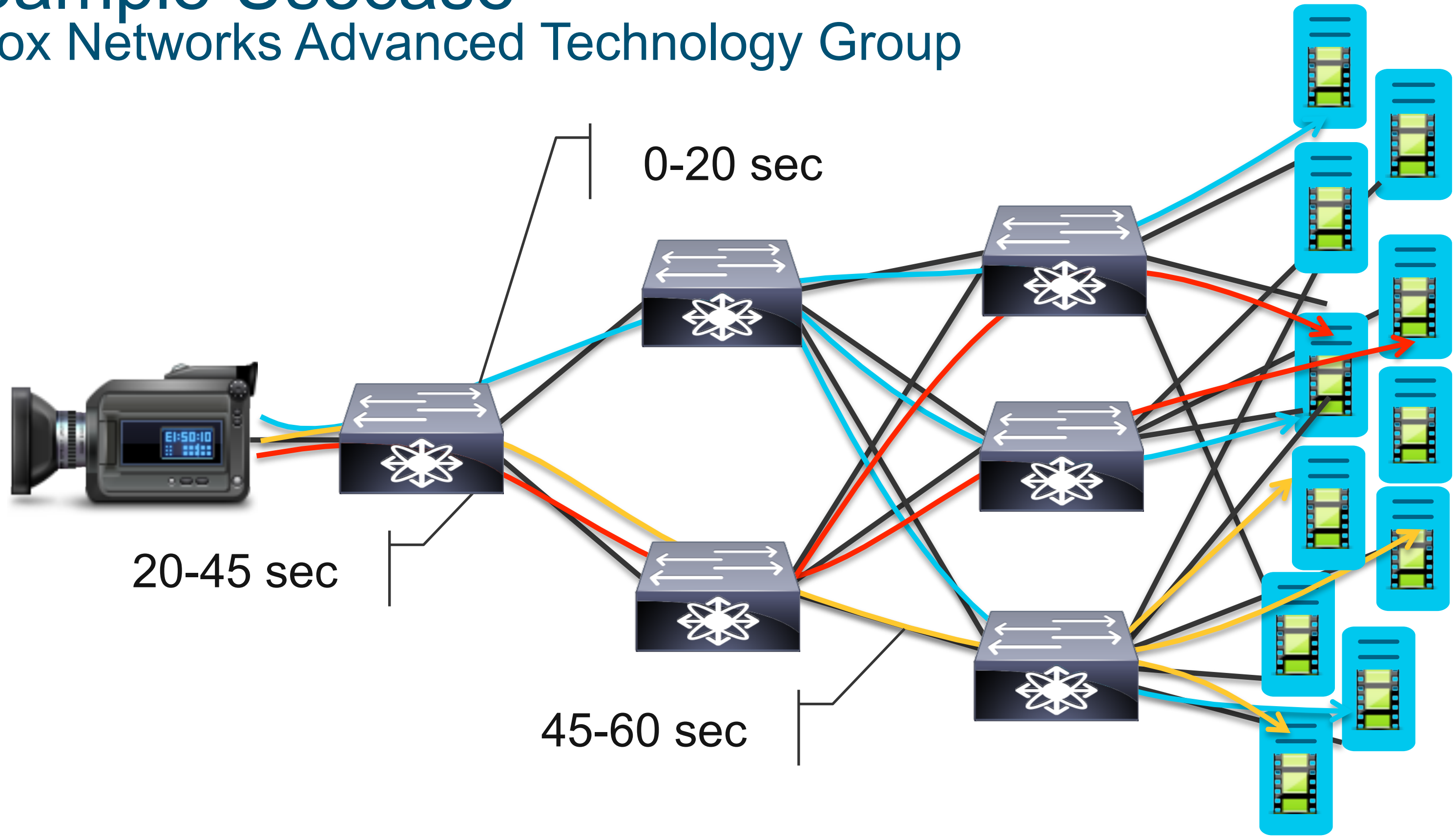


64p 40/100Gb/s QSFP
Nexus 3464C

- Based on Tofino 1 by Barefoot
- 1.8/6.4 Tb/s aggregated switching capacity
- Flexible port configuration and multiple profiles for addressing different feature and scale requirements
- Inband Network Telemetry (INT) support

daPIPE in Action

Sample Usecase
Fox Networks Advanced Technology Group



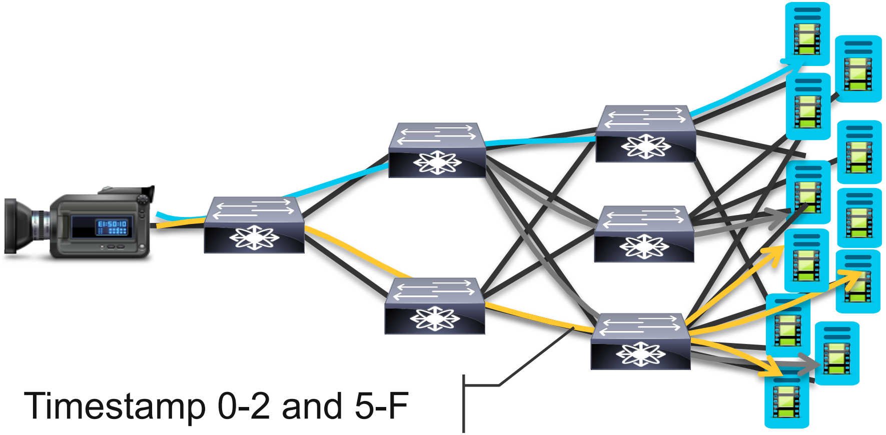
https://github.com/FOXNEOAdvancedTechnology/ts_switching_P4

Specification

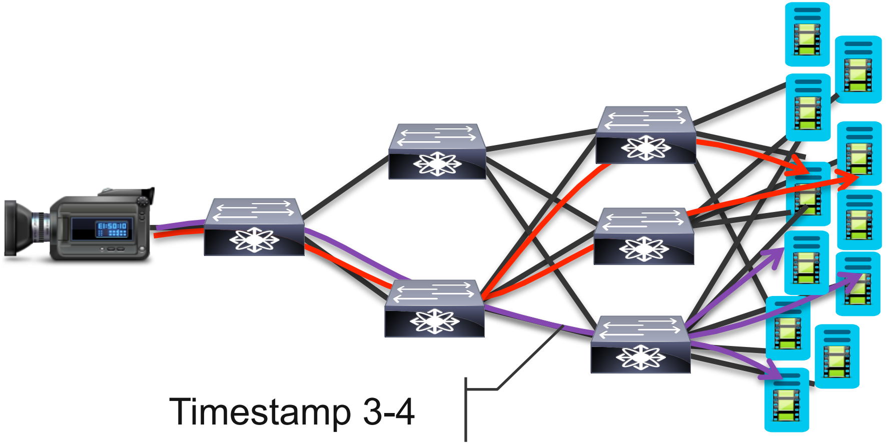
- A switch shall forward packets based on the RTP timestamp they contain

- If sent to 239.1.1.1, change destination address to 239.3.3.3 when RTP timestamp is
  - Between 0 and 2
  - Between from 5 and F

- If sent to 239.2.2.2, change destination address to 239.3.3.3 when RTP timestamp is
  - Between 3 and 4



Timestamp 0-2 and 5-F

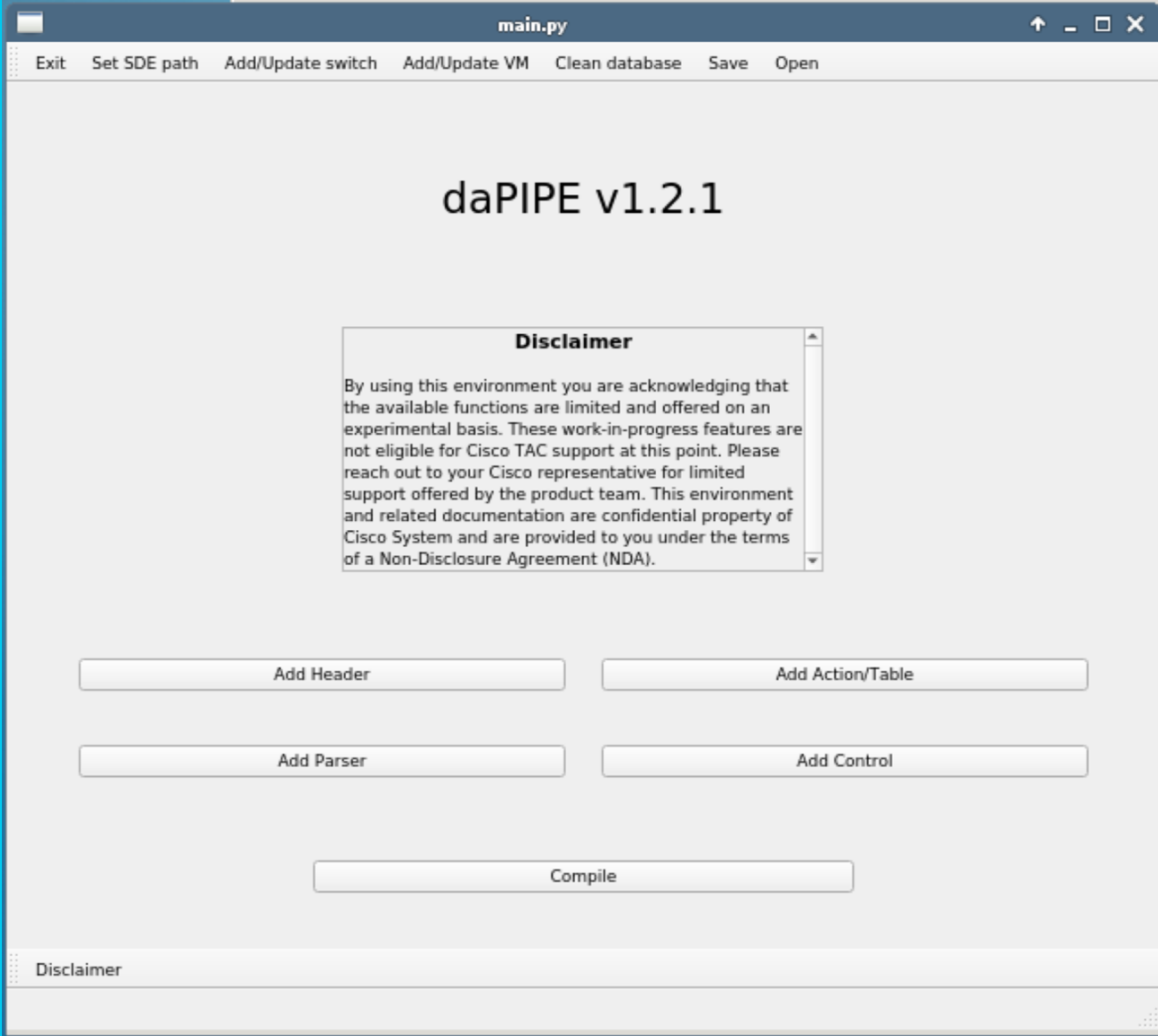


Timestamp 3-4

Development Workflow

- Browse available (stock) metadata
- Define custom headers and metadata
- Specify parser(s) and their hook(s) in existing (stock) parsers
- Define custom tables and actions
- Specify control flow
- Compile and load on chip
- Develop control plane functionalities

Main window

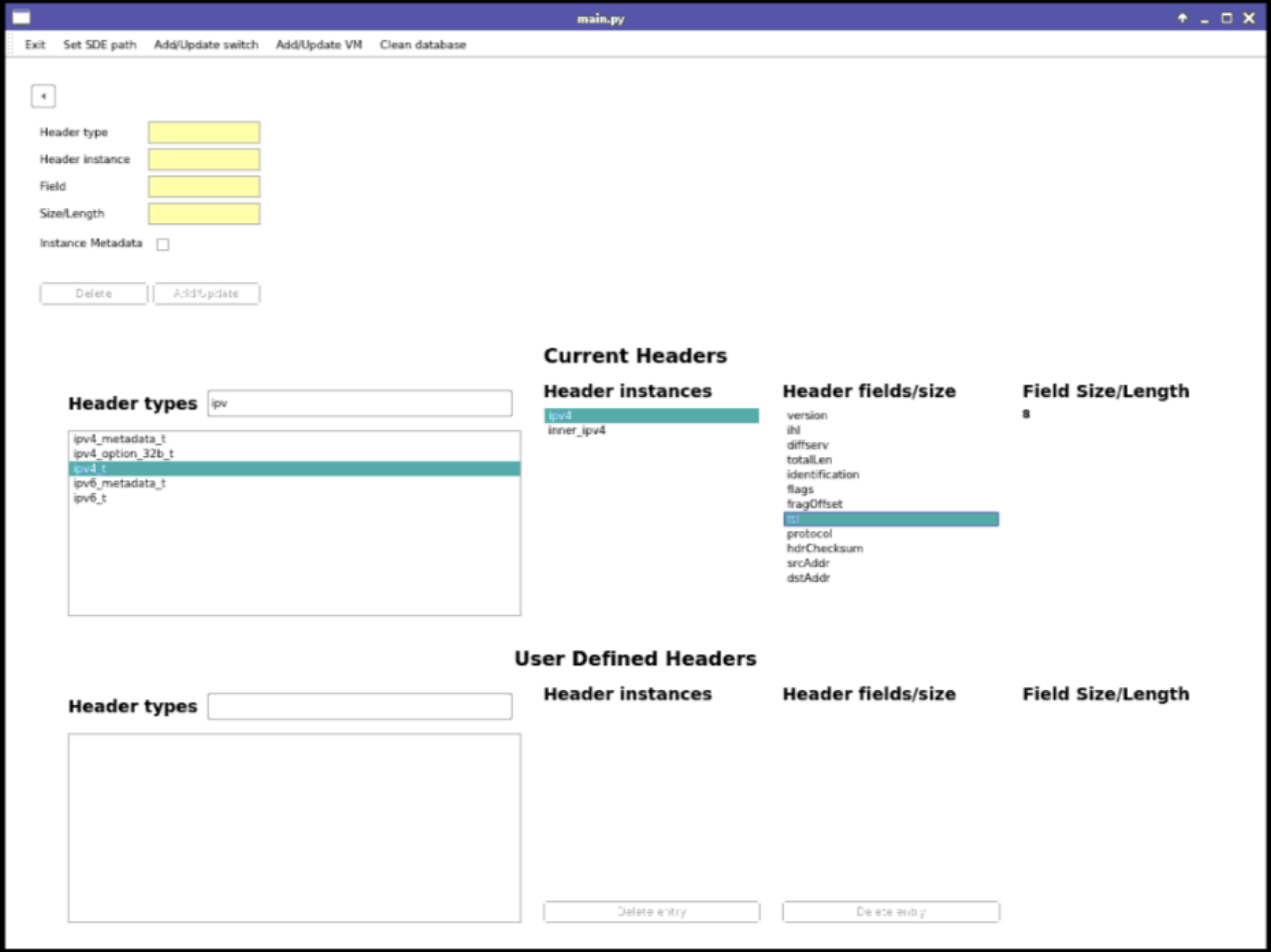


daPIPE v1.2.1

Disclaimer

By using this environment you are acknowledging that the available functions are limited and offered on an experimental basis. These work-in-progress features are not eligible for Cisco TAC support at this point. Please reach out to your Cisco representative for limited support offered by the product team. This environment and related documentation are confidential property of Cisco System and are provided to you under the terms of a Non-Disclosure Agreement (NDA).

Existing header view



Current Headers

Header types: ipv

ipv4_metadata_t
ipv4_option_32b_t
ipv4_t
ipv6_metadata_t
ipv6_t

Header instances

ipv4
inner_ipv4

Header fields/size

version
ihl
diffserv
totalLen
identification
flags
fragOffset
ttl
protocol
hdrChecksum
srcAddr
dstAddr

Field Size/Length

8

User Defined Headers

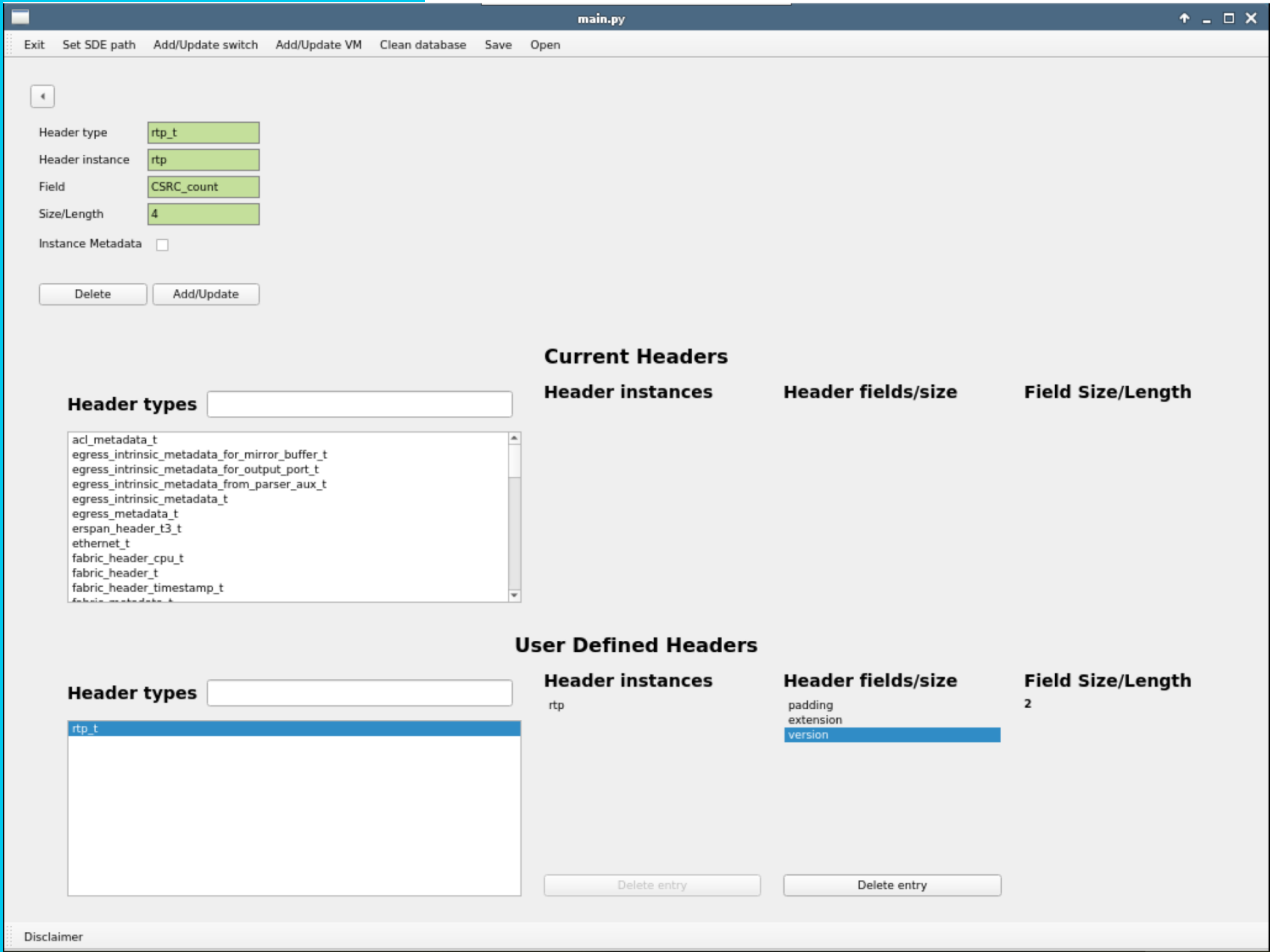
Header types

Header instances

Header fields/size

Field Size/Length

Adding RTP header



Header type: rtp_t

Header instance: rtp

Field: CSRC_count

Size/Length: 4

Instance Metadata □

Current Headers

Header types

acl_metadata_t
egress_intrinsic_metadata_for_mirror_buffer_t
egress_intrinsic_metadata_for_output_port_t
egress_intrinsic_metadata_from_parser_aux_t
egress_intrinsic_metadata_t
egress_metadata_t
erspan_header_t3_t
ethernet_t
fabric_header_cpu_t
fabric_header_t
fabric_header_timestamp_t

Header instances

Header fields/size

Field Size/Length

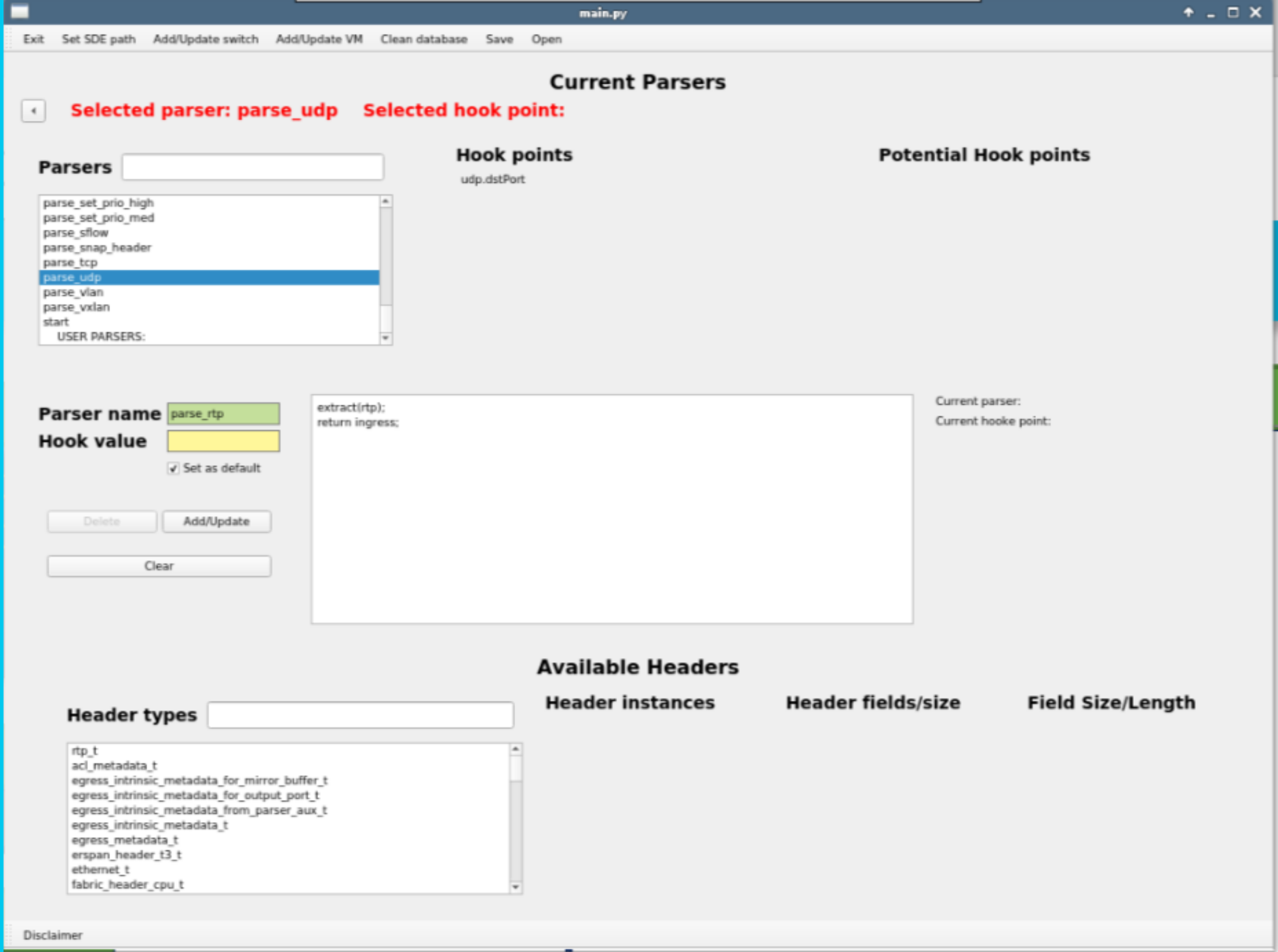
User Defined Headers

Header types

rtp_t

Header instances	**Header fields/size**	**Field Size/Length**
rtp	padding extension version	2

Adding RTP parser



Current Parsers

Selected parser: parse_udp Selected hook point:

Parsers

parse_set_prio_high
parse_set_prio_med
parse_sflow
parse_snap_header
parse_tcp
parse_udp
parse_vlan
parse_vxlan
start
USER PARSERS:

Hook points

udp.dstPort

Potential Hook points

Parser name: parse_rtp

Hook value:

☑ Set as default

```
extract(rtp);
return ingress;
```

Current parser:

Current hooke point:

Available Headers

Header types

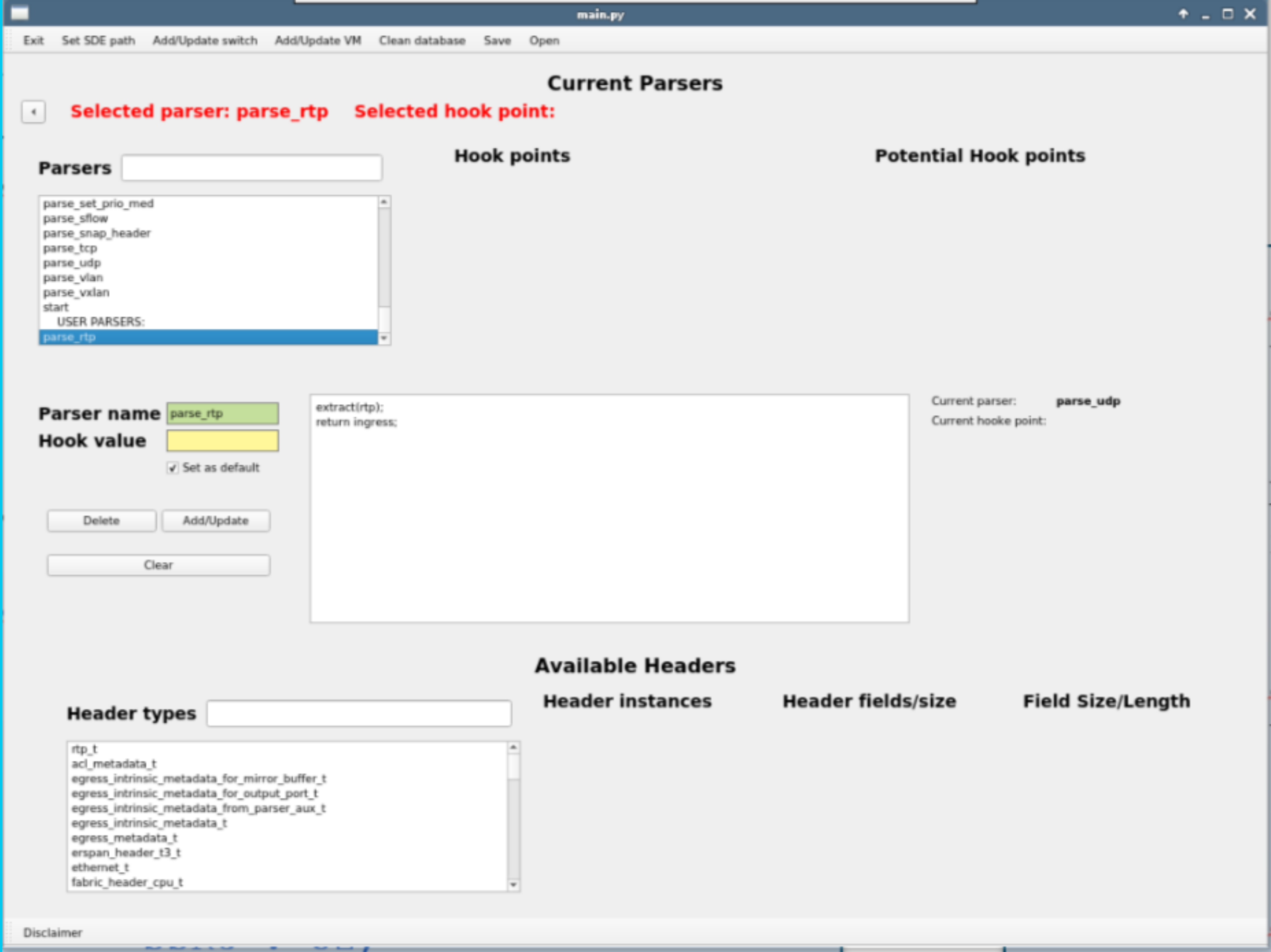
rtp_t
acl_metadata_t
egress_intrinsic_metadata_for_mirror_buffer_t
egress_intrinsic_metadata_for_output_port_t
egress_intrinsic_metadata_from_parser_aux_t
egress_intrinsic_metadata_t
egress_metadata_t
erspan_header_t3_t
ethernet_t
fabric_header_cpu_t

Header instances

Header fields/size

Field Size/Length

New parser added



Selected parser: parse_rtp Selected hook point:

parse_set_prio_med
parse_sflow
parse_snap_header
parse_tcp
parse_udp
parse_vlan
parse_vxlan
start
USER PARSERS:
parse_rtp

Parser name: parse_rtp

Hook value:

☑ Set as default

```
extract(rtp);
return ingress;
```

Current parser: parse_udp

Current hooke point:

rtp_t
acl_metadata_t
egress_intrinsic_metadata_for_mirror_buffer_t
egress_intrinsic_metadata_for_output_port_t
egress_intrinsic_metadata_from_parser_aux_t
egress_intrinsic_metadata_t
egress_metadata_t
erspan_header_t3_t
ethernet_t
fabric_header_cpu_t

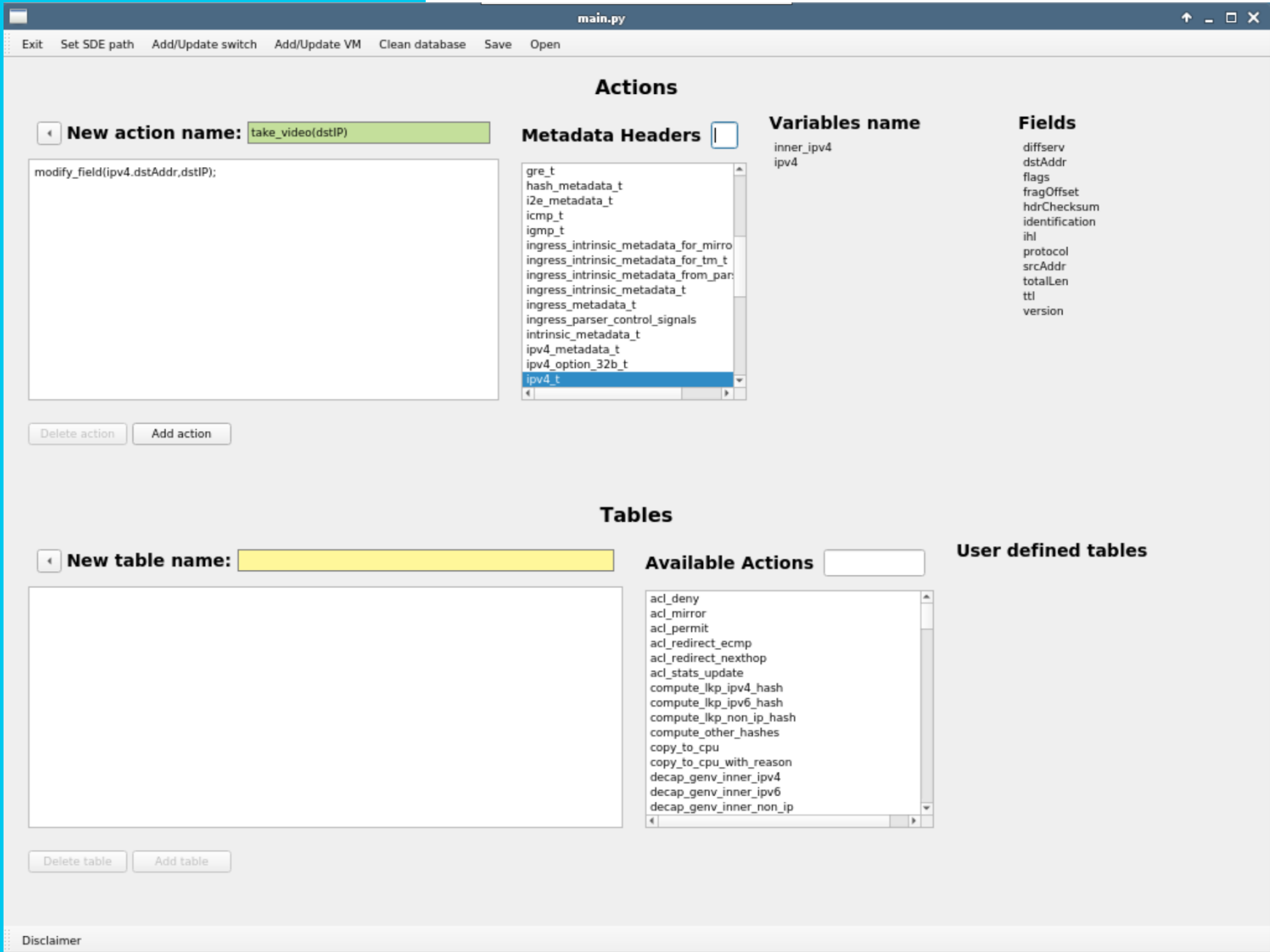
Resulting Parsing Code

Stock code
Custom code
Autom. code

```
...
header_type ethernet_t {
    fields {
        dstAddr : 48;
        srcAddr : 48;
        etherType : 16;
    }
}
header ethernet_t ethernet;
...
header_type rtp_t {
    fields {
        version : 2;
        padding : 1;
...
        sequence_number : 16;
        timestamp : 32;
        SSRC : 32;
    }
}
header rtp_t rtp;
...
```

```
...
parser parse_ethernet {
    extract(ethernet);
    return select(latest.etherType)
{
        ETHERTYPE_IPV4 : parse_ipv4;
        default: ingress;
   }
}
parser parse_udp {
    extract(udp);
    return parse_rtp;
}
...
parser parse_rtp {
    extract(rtp);
    return ingress;
}
...
```

Add action



New action name: take_video(dstIP)

```
modify_field(ipv4.dstAddr,dstIP);
```

Metadata Headers

- gre_t
- hash_metadata_t
- i2e_metadata_t
- icmp_t
- igmp_t
- ingress_intrinsic_metadata_for_mirro
- ingress_intrinsic_metadata_for_tm_t
- ingress_intrinsic_metadata_from_par:
- ingress_intrinsic_metadata_t
- ingress_metadata_t
- ingress_parser_control_signals
- intrinsic_metadata_t
- ipv4_metadata_t
- ipv4_option_32b_t
- ipv4_t

Variables name

- inner_ipv4
- ipv4

Fields

- diffserv
- dstAddr
- flags
- fragOffset
- hdrChecksum
- identification
- ihl
- protocol
- srcAddr
- totalLen
- ttl
- version

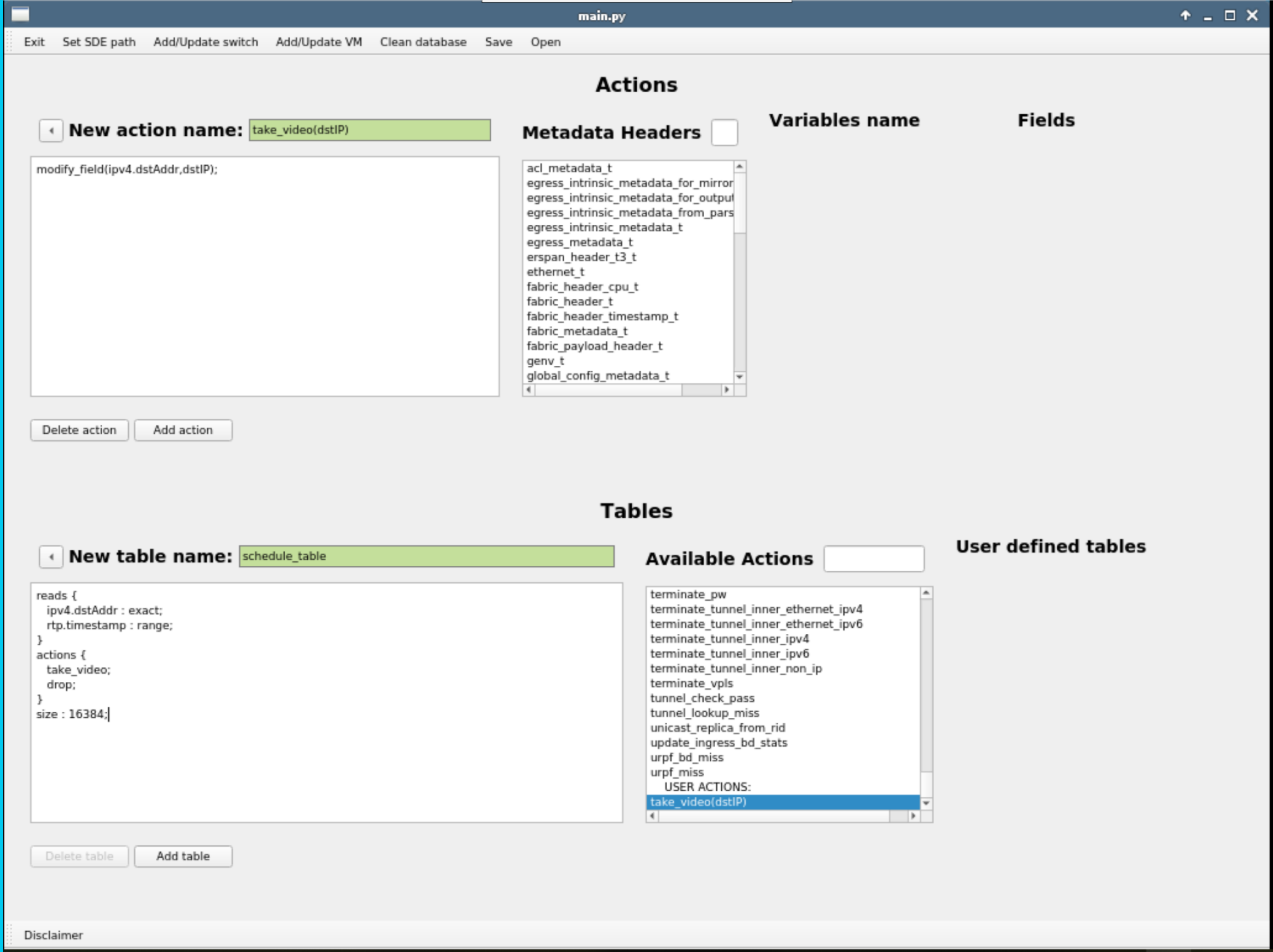
New table name:

Available Actions

- acl_deny
- acl_mirror
- acl_permit
- acl_redirect_ecmp
- acl_redirect_nexthop
- acl_stats_update
- compute_lkp_ipv4_hash
- compute_lkp_ipv6_hash
- compute_lkp_non_ip_hash
- compute_other_hashes
- copy_to_cpu
- copy_to_cpu_with_reason
- decap_genv_inner_ipv4
- decap_genv_inner_ipv6
- decap_genv_inner_non_ip

User defined tables

Adding a table



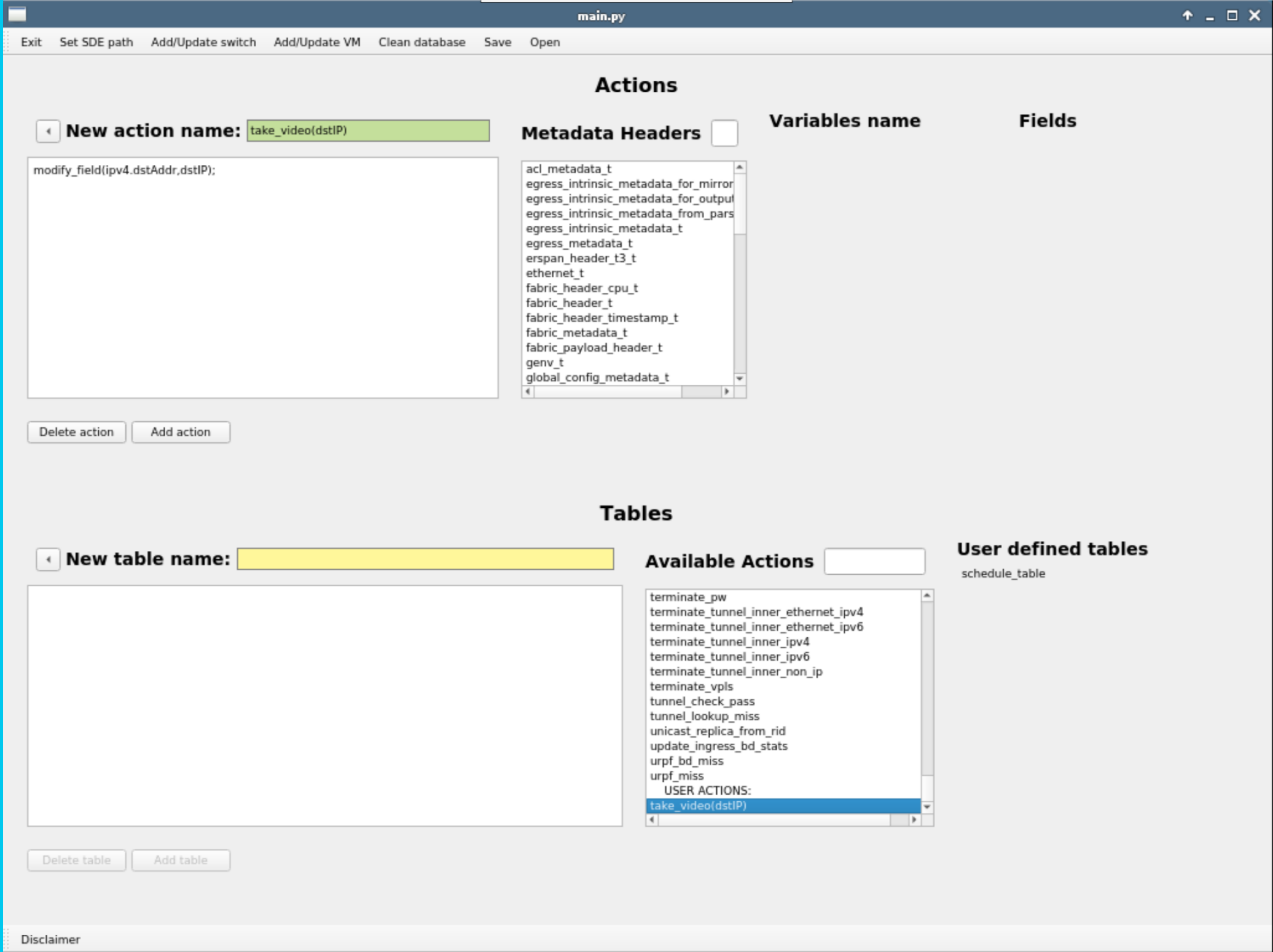
New action name: take_video(dstIP)

```
modify_field(ipv4.dstAddr,dstIP);
```

New table name: schedule_table

```
reads {
   ipv4.dstAddr : exact;
   rtp.timestamp : range;
}
actions {
   take_video;
   drop;
}
size : 16384;
```

New table available



Actions

New action name: take_video(dstIP)

```
modify_field(ipv4.dstAddr,dstIP);
```

Metadata Headers

acl_metadata_t
egress_intrinsic_metadata_for_mirror
egress_intrinsic_metadata_for_output
egress_intrinsic_metadata_from_pars
egress_intrinsic_metadata_t
egress_metadata_t
erspan_header_t3_t
ethernet_t
fabric_header_cpu_t
fabric_header_t
fabric_header_timestamp_t
fabric_metadata_t
fabric_payload_header_t
genv_t
global_config_metadata_t

Variables name

Fields

Tables

New table name:

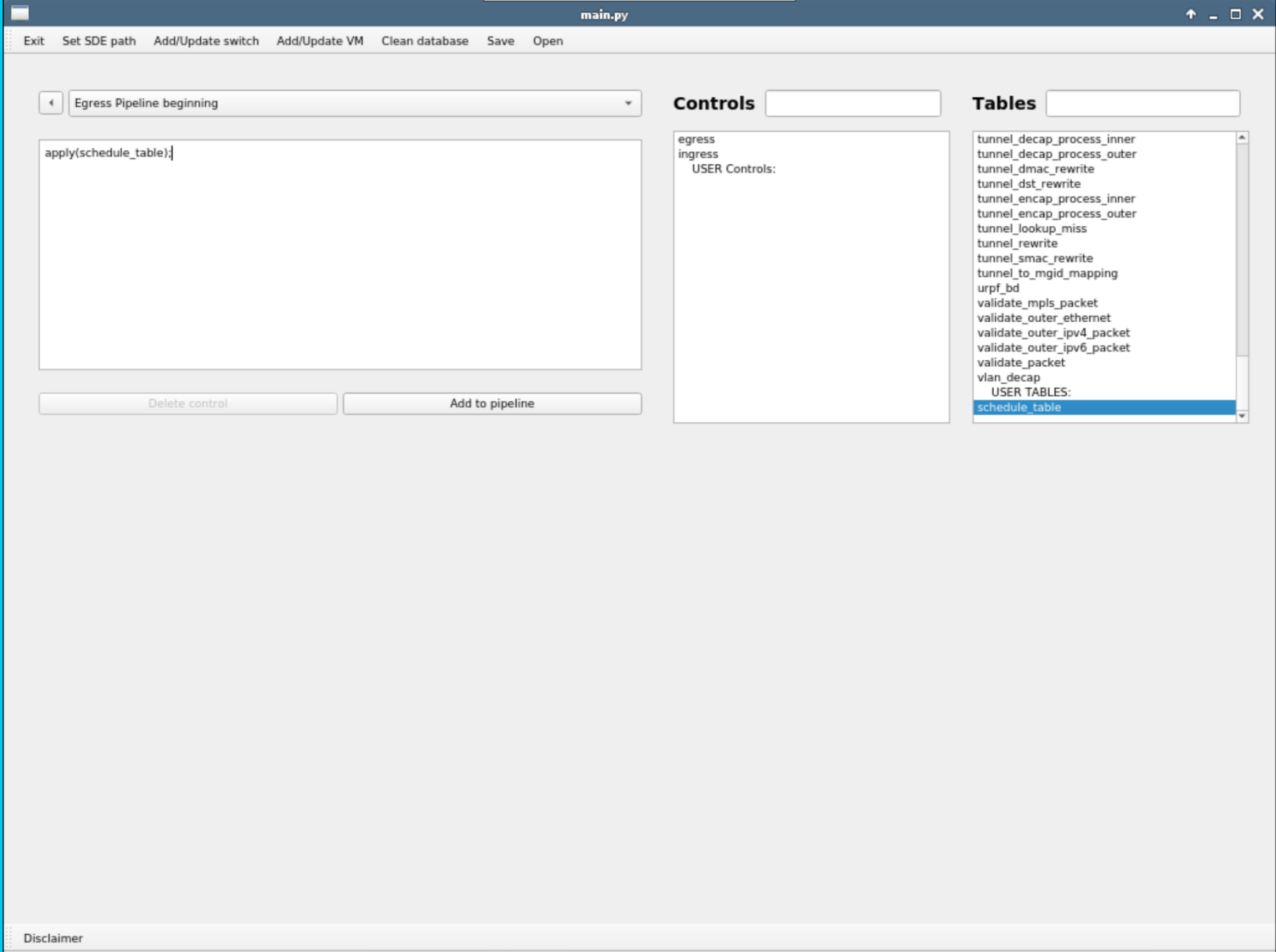
Available Actions

terminate_pw
terminate_tunnel_inner_ethernet_ipv4
terminate_tunnel_inner_ethernet_ipv6
terminate_tunnel_inner_ipv4
terminate_tunnel_inner_ipv6
terminate_tunnel_inner_non_ip
terminate_vpls
tunnel_check_pass
tunnel_lookup_miss
unicast_replica_from_rid
update_ingress_bd_stats
urpf_bd_miss
urpf_miss
USER ACTIONS:
take_video(dstIP)

User defined tables

schedule_table

Define control flow

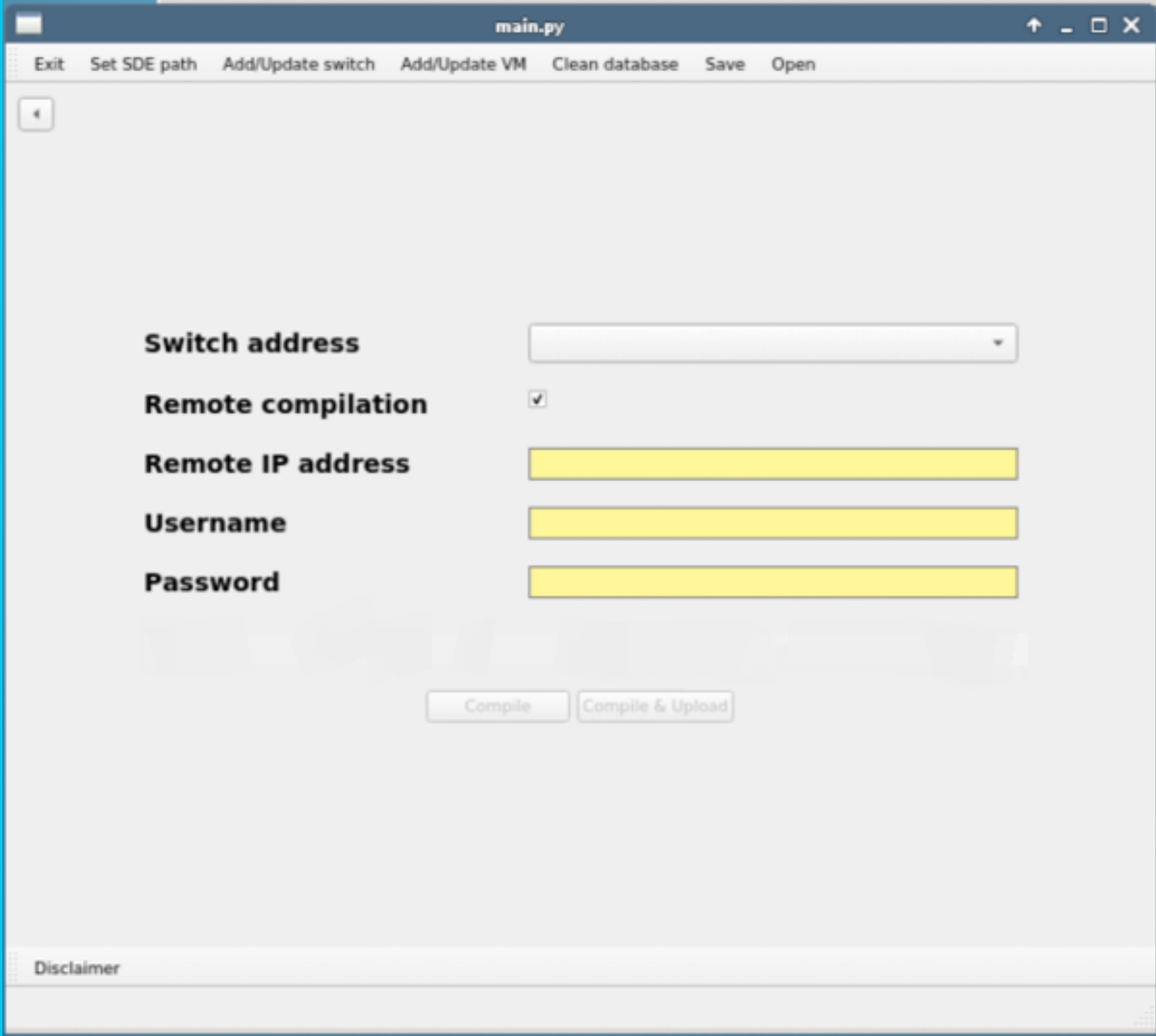


```
apply(schedule_table);
```

egress
ingress
USER Controls:

tunnel_decap_process_inner
tunnel_decap_process_outer
tunnel_dmac_rewrite
tunnel_dst_rewrite
tunnel_encap_process_inner
tunnel_encap_process_outer
tunnel_lookup_miss
tunnel_rewrite
tunnel_smac_rewrite
tunnel_to_mgid_mapping
urpf_bd
validate_mpls_packet
validate_outer_ethernet
validate_outer_ipv4_packet
validate_outer_ipv6_packet
validate_packet
vlan_decap
USER TABLES:
schedule_table

Compile and upload to switch



Switch address

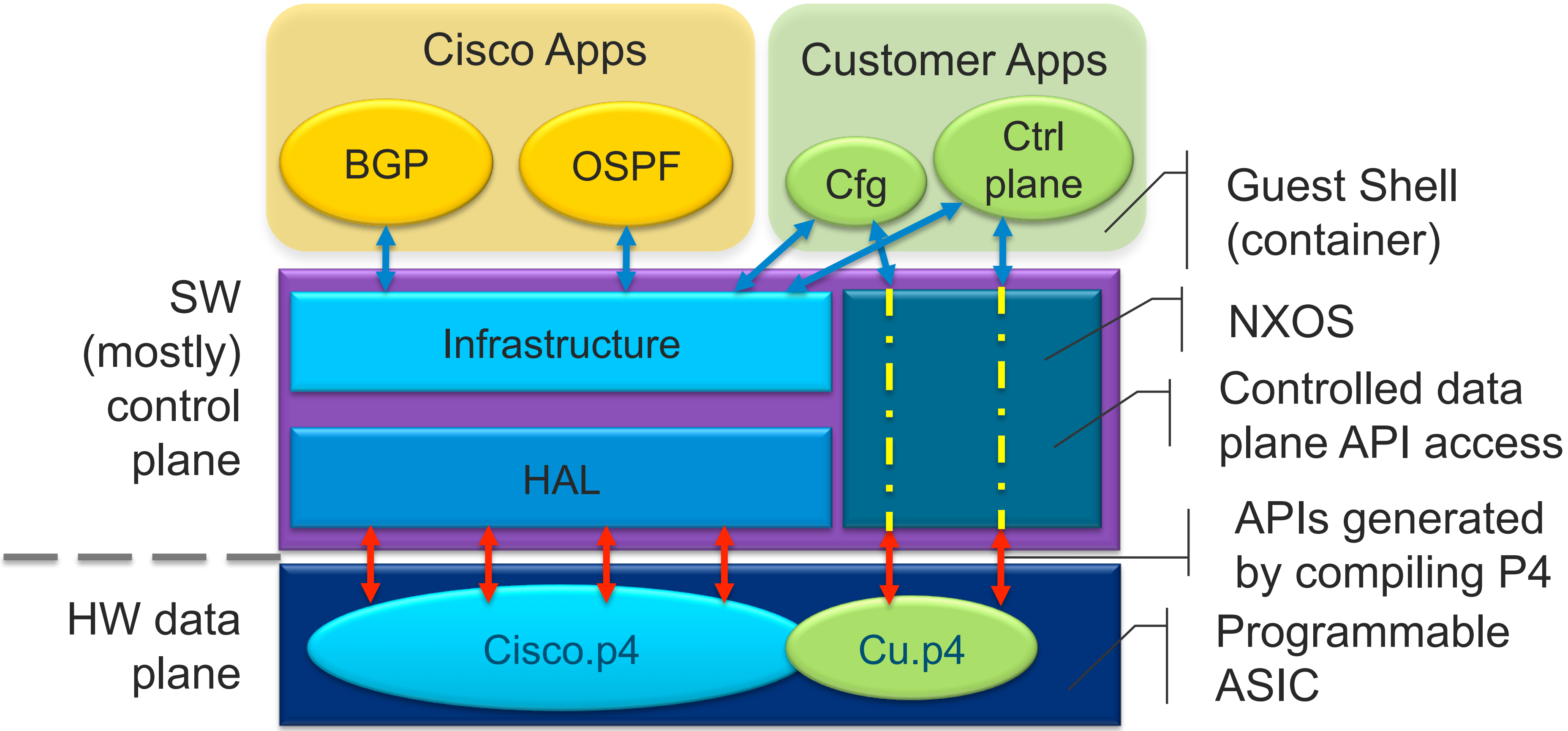
Remote compilation ☑

Remote IP address

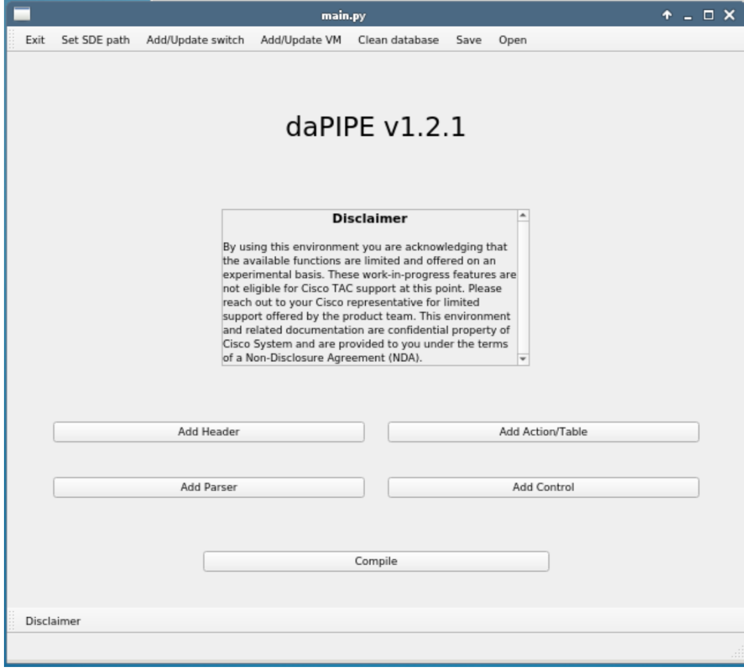
Username

Password

Control Plane



In summary



Disclaimer

By using this environment you are acknowledging that the available functions are limited and offered on an experimental basis. These work-in-progress features are not eligible for Cisco TAC support at this point. Please reach out to your Cisco representative for limited support offered by the product team. This environment and related documentation are confidential property of Cisco System and are provided to you under the terms of a Non-Disclosure Agreement (NDA).

- daPIPE enables incremental programming
  - Cisco NX34xxx so far
  - Not platform specific
    - Any platform, any NetOS
- Developer can focus just on new features
  - Does not need to work on common features
  - Can leverage existing functions
- No need to deal with the complexity of stock P4 code
- Constrained changes ensure stock feature and NetOS integrity
- It does not address any possible use case, but it addresses many

Interested in giving it a try?

Get in touch with me (mariobal@cisco.com) …

… and be willing to deal with the imperfections of something new