NS4

Background

What is P4

P4, a high-level domain specific language (DSL), has attracted great attention from both academia and industry for its advantages of enabling operators to define behaviours of the programmable data plane

What is NS-4

NS4, a P4-driven network simulator supporting simulation of P4-enabled networks to address the problems existing in traditional simulators.

It consisting of data plane models integrated with ns-3, the state-of-the-art network simulator, and control plane models to interact with the P4 pipeline.

Problems in traditional network simulators

Traditional network simulators lack support for the programmable data plane. Existing software switches supporting P4 (such as bmv2 and pfpsim) have to run with a network emulator (such as mininet). A network emulator, in contrast to a network simulator, gets constrained by host resources, thus failing to simulate large-scale or ultra speed P4-enabled networks.

Challenges in integrating p4 to ns3

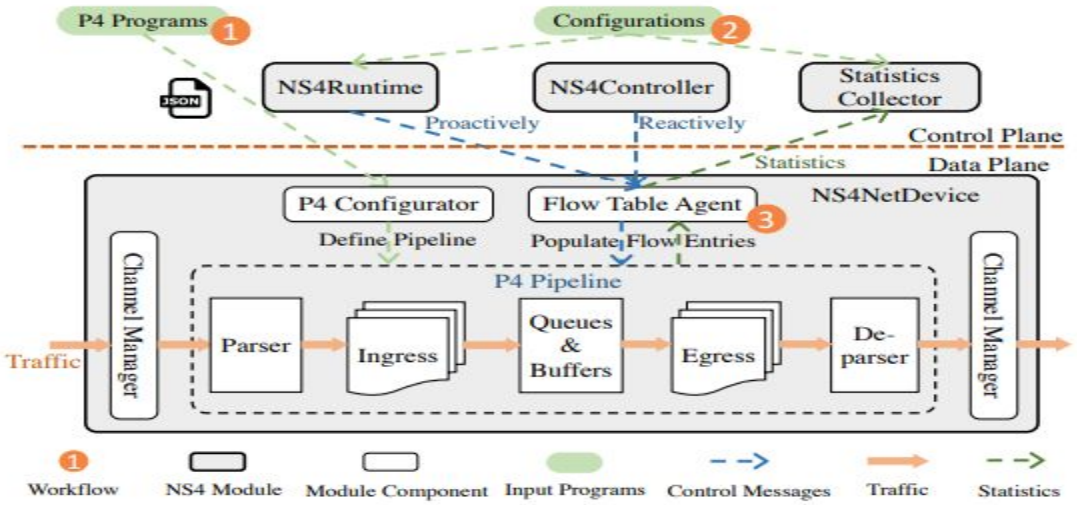
- Modelling behaviours of real P4 devices (e.g., Tofino and P4FPGA) based on ns-3 simulator
- Existing runtime tools such as P4Runtime and ONOS-BMv2 for controlling P4 programs cannot interact with the simulated P4 module since ns-3 is a discrete-event simulator. The runtime operations have to be transformed into discrete events to get conducted.
- Simulating multiple P4 devices in a network requires installing routing entries in every switch. Since configuring flow entries manually is laborious and error-prone, automatic population of flow entries is needed to simulate networks with numbers of P4 devices

DESIGN

Architecture of Ns4

NS4 is divided into a data plane half which contains a module for modelling a P4-enabled switch, NS4NetDevice, and a control plane half composed of modules for controlling the data plane.

Developers simulate a P4-enabled device by instantiating a NS4NetDevice, loading the P4 program, and populating flow entries from the control plane. To assure compatibility of NS4, NS4NetDevice can be connected to other NS4NetDevices or traditional network devices like routers or switches. The NS4NetDevice module works as a Netdevice in ns-3, and the control modules help operators configure the data plane program.



The workflow of simulating a P4 network are shown as numbers in Figure above

(1) Configure the behaviour of data plane by inputting compiled P4 programs to P4 pipeline configurator

(2) Create a control plane and configure the flow table operations and statistics collection tasks to be performed

(3) Build network topology, install applications and trigger the simulation. The control configurations are transformed into discrete events and get performed at the appointed time

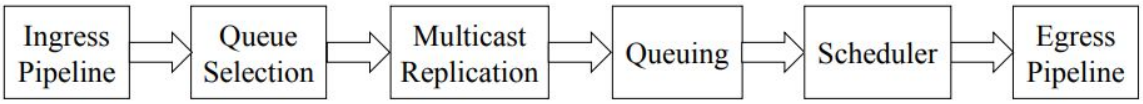
Programmable Data Plane of NS4

P4 Pipeline

P4 pipeline is the core of NS4NetDevice. To enable customization of the pipeline, a configurator and a flow table agent are abstracted in the module. On the initialization of the device, the configurator loads the compiled P4 program to configure the behaviour of pipeline. At runtime, the flow table agent can be called by control modules to add, modify, delete or query the entries of flow tables.

Queuing System.

There are multiple queues in the module NS4NetDevice, and each egress port of the device can be associated with several queues.



When a packet leaves the ingress pipeline, it is accompanied by a metadata which determines actions to be taken in a queuing system, such as the set of ports to which the packet will be sent. After that, packets are moved to the selected queues and wait to be scheduled

Control Plane of NS4

Discrete population.

The user configurations are loaded into control modules during device initialization and get parsed into discrete events by control modules. Every line in configurations corresponds to a discrete event, consisting of a timestamp, a device id, a description of the action type and the corresponding parameters, as shown in the table below

Table 2: NS4 event format.

Time stamp	Device ID	Command Type	Parameters

Table 3: NS4 flow table commands.

Command Type	Description	Parameters	Executor
table_set_default	Set default entry in a match table	<table name> => <action>	**NS4Runtime**
table_add	Add entry to a match table	<table name> <match fields> => <action>	
table_delete	Delete entry from a match table	<table name> <entry handle>	
table_delete_wkey	Delete entry using the match key	<table name> <match fields>	
table_modify	Modify entry in a match table	<table name> <entry handle> => <action>	
table_modify_wkey	Modify entry using the match key	<table name> <match fields> => <action>	
table_dump	Dump all entries in a match table	<table name>	**Statistics Collector**
counter_read	Read value(s) from counter	<counter name> <index | entry handle>	
counter_reset	Reset values for counter to 0	<counter name> <index | entry handle>	

Events of adding, modifying and deleting table entries are performed by NS4Runtime while other events of dumping the tables and reading the counters are performed by Statistics Collector. To record and schedule the events, NS4Runtime and Statistics Collector maintains an internal queue respectively. Events are arranged in the order of execution time in each module and get dequeued once the simulation time meets the appointed time.

Automatic population.

NS4Controller module to populate flow entries reactively during simulation. Since the flow tables in P4 devices are defined by P4 programs, the format of flow entries and name of the targeted table remain unknown to NS4Controller. Thus, operators need to pass the format and table name of each switch to NS4Controller before conducting a simulation. NS4Controller module is designed to be optional since it brings an extra burden on the system. Operators are able to choose whether to use the module according to the scale of the simulation network.