Neuromorphic Hardware in Outer Space: Software Defined Networking Executed on an In-Orbit Loihi Spiking Processor

Tarek Taha Md. Nayim Rahman Chris Yakopcic Ricardo Lent

University of Houston *rlent@central.uh.edu*

UNIVERSIT

HOUSTON

Brisk Computing, LLC *ttaha@ieee.org*

June 20, 2023







- Cognitive network,
- □Software-defined Networking (SDN),
- □Network architecture and implementation,
- □Neuromorphic processor as cognitive agent (Intel's Loihi)
- Testbed experimental results and energy calculation
- □Implementation in outer space



□In cognitive networking, networking issues are dealt with autonomously by observing and collecting information from the environment and making appropriate decisions to achieve a higher level of automation.

- □Cognitive network enabled devices adapt to changes in the network environment or user demand without human intervention.
- Cognitive networking improves the performance and efficiency of the network.

Applications:

- > Space exploration missions where the data transmissions occur over long unreliable channels
- Selecting new network paths, or managing the allocation and deallocation of computing resources

Software-Defined Networking (SDN)



□SDN uses software-based controller or application programming interfaces to control the communication of network hardware infrastructure and data traffic.

Benefits:

➤More flexibility and control on the network operation.

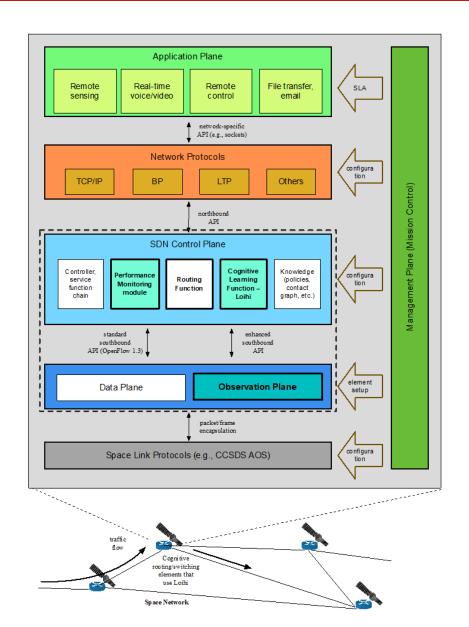
≻Customizability of the network operation.

➢ Robust security.

Cognitive Network Architecture

Designed network architecture:

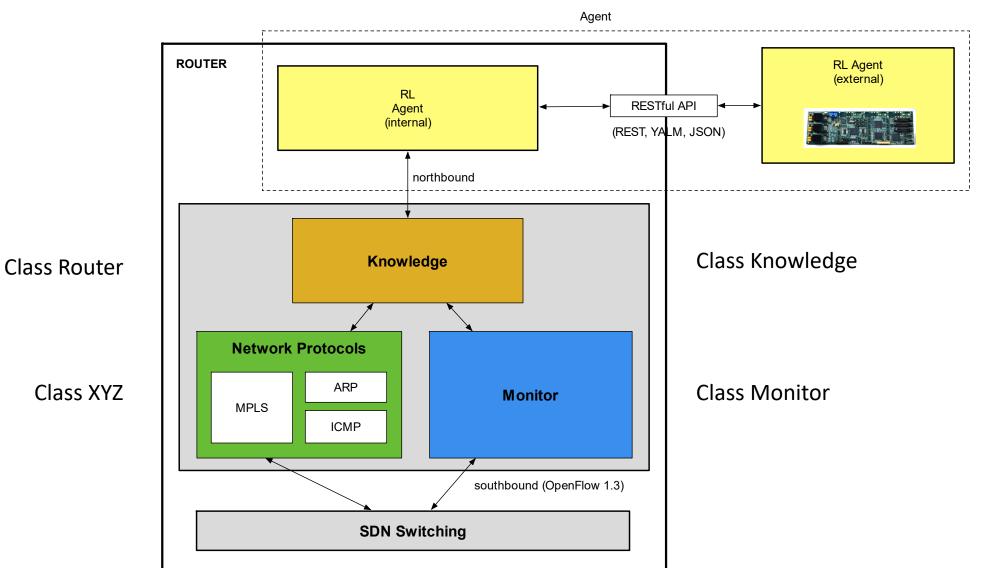
- ➢ Based on SDN
- Dynamic environment
- ➤Supports dynamic routing management
- Continually maps active flows to paths, which change according to an assigned goal and the state of the links





Software Architecture





Class Agent



<pre>#routing: qlearning routing: random #routing: shortestpath #routing: shortestpath</pre>
<pre>#routing: static graph: s22: s23: 3 s24: 1 s25: 2</pre>
s23: s22: 2 s21: 3 s26: 1 s24: 4 s21: s23: 3
s24: 1 # s25: 2 s24: s22: 3 s23: 1 s21: 2 s25: s22: 3
s27: 1 # s21: 2
hostIp: s27: 192.168.101.27 s26: 192.168.108.26
lerIp: s23: 192.168.108.23 s25: 192.168.101.25
path: s26: s27: - s26 - s23 - s21 - s25 - s27 s27: s26: - s27 - s25 - s27 - s25 - s22 - s23 - s23 - s26

□ Architecture defined around the *Ryu* controller and MPLS (Multiprotocol label switching)

□ Routing is handled by a virtual network function (VNF)

Flows are defined as a 5-tuple (IP src, IP src_port, IP dst, IP dst_port, protocol)

The MPLS network core consists of LER and LSR nodes

- Label-based, no IP
- Ingress LER maps a new flow to the ingress label
- Labels are distributed along the selected path
- Egress LER removes the label and forwards out the IP packets

Expected network graph and IP mappings are given:

- At boot time (YALM file)
- Dynamically updated through a REST interface
- Additional information is acquired dynamically
- Become part of the *knowledge*

Agent Base Class



The agent is decoupled from the network operation

- Greatly simplifies the implementation
- Does not require to be aware of the actual paths
- Just need to observe the routing costs to make decisions
- □ Agent is called passing the flow ID *f* and number of paths |P(f)| (*flow_init*)
- □ The agent decides the path index (*flow_route*)
- □ MPLS router implements the selected path

```
class AgentBase():
    #def __init__(self, *args, **kwargs):
    # Agent-specific functions
    def flow_init(self, flowID, nbr_action, costL):
        pass
    def flow_route(self, flowID):
        pass
    def flow_cost(self, flowID, actionL, costL):
        pass
    def flow_close(self, flowID):
        pass
    # Function exposed to the router
    #def getPath(self, rtrName, srcName, dstName):
    # Utility function used by the router
    def getCost(self, fromNode, toNode):
        return 1.0
```



Link costs are dynamically evaluated by the Monitor. The information becomes part of the knowledge:

rlent@s30:~/devel/sdn_exper/Exper1\$ curl localhost:8080/switch13/data
{"s22": {"s24": {"delay": 0.001650300464313984, "cnt": 10}, "s25": {"delay": 0.00162362134605784
47, "cnt": 10}, "s23": {"delay": 0.3096590902378288, "cnt": 9}}, "s23": {"s22": {"delay": 0.3378
689080892093, "cnt": 10}, "s21": {"delay": 0.07774153303166254, "cnt": 10}, "s24": {"delay": 0.1
3889510999074178, "cnt": 9}}, "s21": {"s23": {"delay": 0.0015240369358162878, "cnt": 9}, "s24":
{"delay": 0.001529543478949547, "cnt": 9}}, "s24": {"s22": {"delay": 0.0015023722411098483, "cnt": 9}}, "s24": {"delay": 0.0015023722411098483, "cnt": 9}}, "s24": {"s22": {"delay": 0.0015023722411098483, "cnt": 9}}, "s24": {"s22": {"delay": 0.0015023722411098483, "cnt": 9}}

The path cost (negative reward) of the active flows is evaluated after a link cost change

The agent is informed of the affected flows and new path costs (*flow_cost*)

The router periodically attempts to modify the path of the active flows (*flow_route*)

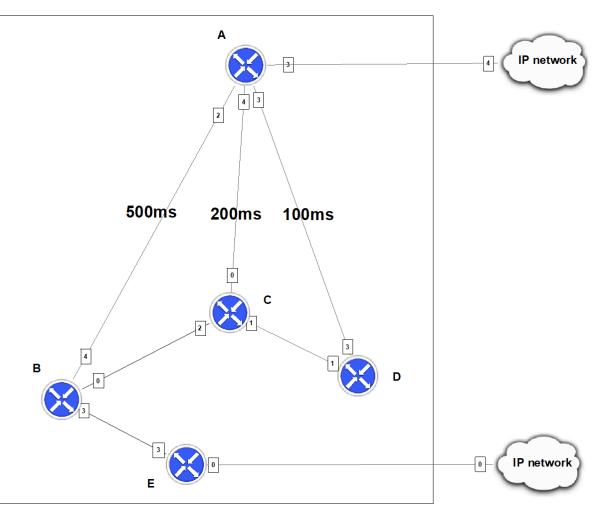
Experimental Setup



A high-throughput satellite (HTS) system was emulated in the University of Houston's laboratory facilities.

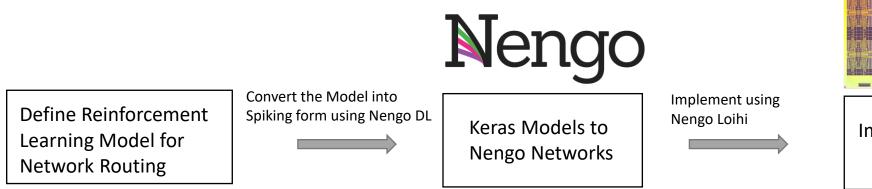
One satellite router (node A) connected to four ground stations and routers (nodes B, C, D, E).

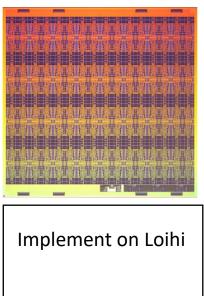
- □ The ground router E is connected to an external IP network. The router implementation was deployed on nodes A, B, C, D, and E with nodes A and E operating as LER and the rest of nodes as LSR.
- One way propagation delay was introduced both for the downlink and the uplink using Linux's Traffic Control (TC).
- □ The delays for the links A-B, A-C, and A-D were configured with 500ms, 200ms, and 100ms.
- A YAML interface (a network configuration file) was developed to pass the network topology to the controller.



Reinforcement Learning on Intel Loihi

- □ Reinforcement learning method learns
 - by interacting with its environment and
 - receiving rewards or punishment from its interactions.
- **Q**-learning is one of the basic working principle of Reinforcement learning.
- Q-learning tries to learn an optimal action-selection policy for any given finite Markov Decision Process.
- □ It is used in game playing, control systems, operations research, multi-agent systems, and so on.







Intel's Neuromorphic Processor (Loihi)



Each Loihi Chip consists of 3 synchronous x86 cores and 128 neuromorphic cores.

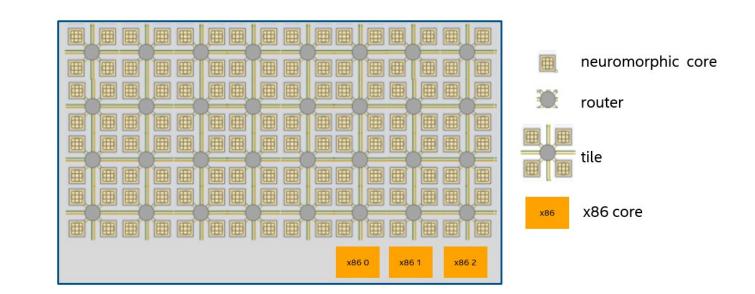
Loihi is available in several Size:

- Kapoho Bay (1 or 2 chips)
- > Wolf Mountain (4 chips),
- Nahuku (32 chips),
- Pohoiki Beach (64 chips),
- Pohoiki Springs (764 chips)

□ Programming Language:

≻ NxSDK,

- Nengo (Deep learning),
- ➢ Slayer (Deep learning),
- > LAVA



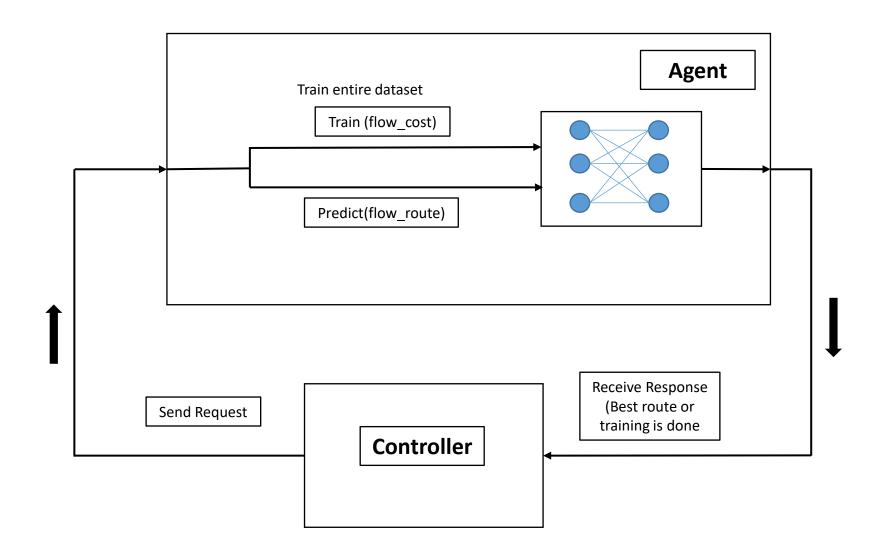
Loihi Kapoho Bay on Laptop at Brisk



Kapoho Bay is a USB stick form factor that incorporates 1 or 2 Loihi chips.
 With 2 chip Kapoho Bay has 256 neuromorphic cores with 262,144 neurons and 260,000,000 synapses.







Dataset Processing



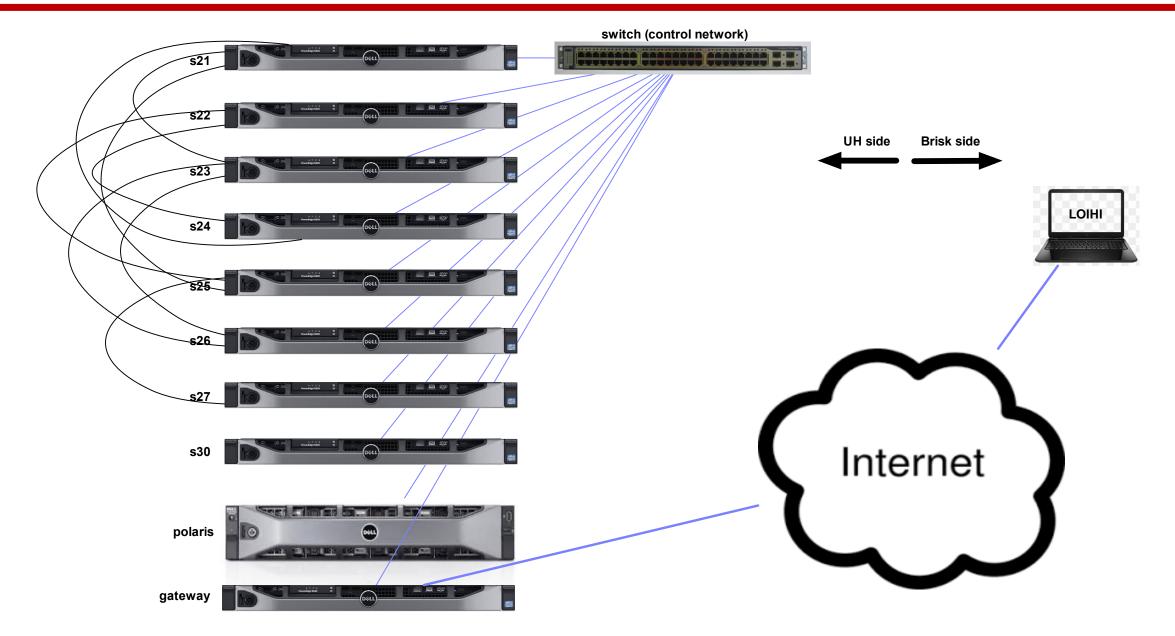
- The input state of each training sample was constructed using the flow ID and the previous timestamps received reward.
- \Box At first, a unique binary 4×4 matrix was created for each flow ID by converting it into a 16bit string using the equation: (round(flow id/2¹⁰)*10000).

Flow ID	Binarized representation
1	000000000001010
2	000000000010100

- □After that the information of the best previous route was appended with that matrix to convert the input state into an 8×8 matrix.
- □In the experiment, only three possible routes are considered.
- □ Flow IDs are divided into two categories: downlink ('s23') and uplink ('s25') where each category can have maximum 1024 flow IDs

Distributed Test Environment



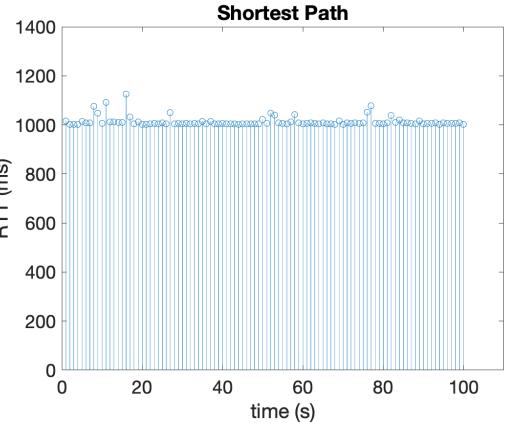


Results: Shortest Path

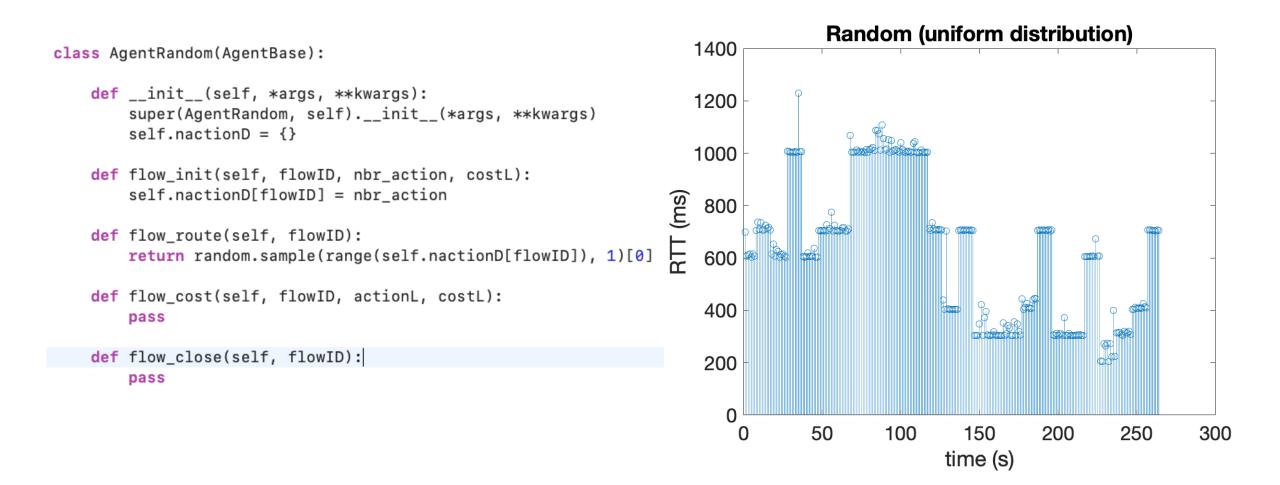


Test: end-to-end ping

- Link state test rate: 1 pps
- Route update rate: 0.1 route/s (max)







Neural Networks



Agent:

remote

remote

- NN agent consists of a local and a remote component
- Local component: interfaces with the Ryu controller and the proposed SDN/VNF architecture
- Remote component: implements NNs with Keras and Tensorflow
- RESTful communication between the 2 components

в

local

local

500ms

200ms

X

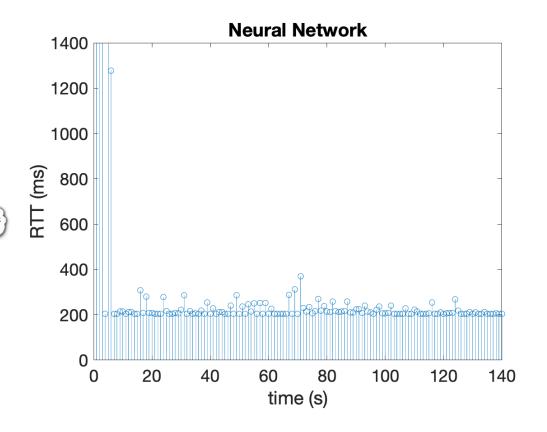
 \mathbf{X}

100ms

IP networ

IP network

0



Energy Calculation



Application ran for 10 continuous operation (each operation took 12 seconds).

Power was measured for 30 seconds while running the routing application.

 \Box The energy consumption reaches around 7.5 Joule (0.25W imes 30s).



- VDD: Neuro cores, embedded Lakemont CPU, mesh router, FPIO/PIO (programmed input-output) protocol logic, etc. (everything else that consumes power on the chip) are expressed as VDD and
- VDDM: VDDM shows the power consumption by the SRAM memories

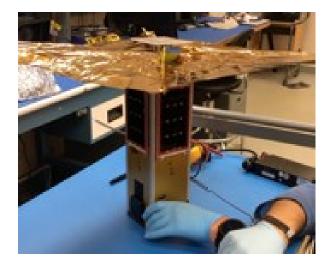
Space Exploration for Future Technology



- On January 13th, 2022, the TES-13 CubeSat launched on a Virgin Orbit launch vehicle flight and has recently successfully finished the successful Phase 1 Operations.
- The TES-13 is based on a 3U nano-sat design.
- □NASA launched a Loihi Kapoho Bay USB unit hosted by an Up Board in a CubeSat (TES-13) to test the machine learning algorithms on the Loihi in a space environment. Loihi based SDN application was selected for this launch.



The TES-13 was launched on the Virgin Orbit Launcher 1

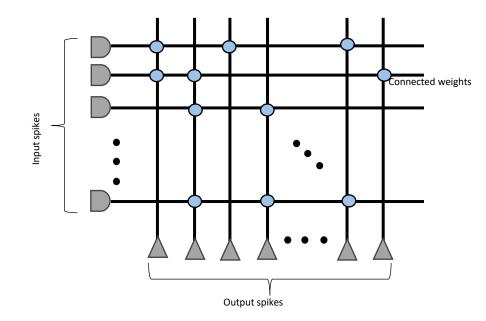


The TES-13 in preparation for integration into the dispenser

Modified SDN Implementation



- Nengo-DL (deep learning library of Nengo) could not run on the Up Board due to the limited supported libraries (e.g., vector assembly instruction is not supported) by the Pentium processor on Up Board.
- As a result, a new classification-based spiking feed-forward network (a two-layer perceptron network) was developed for the Loihi using Intel's NxSDK that performed the core functionality of the routing.
- □ In the network, when the dot product of the input signals and the weights of a given neuron exceeds the predefined threshold, then the neuron will send a spike to the following layer.
- □ The networks weights were optimized in MATLAB and stored in the Loihi neuromorphic cores.



Modified SDN Implementation



□ This simplified version had no learning (it predicted the best route all the times).

- Due to memory limitation, instead of writing all the neurons firing patterns, sum of all the spikes generated by the input neurons were stored for each execution. The final output of this approach is an 87-character string of numbers that would be transmitted to earth from the CubeSat to indicate successful execution.
- This output with spaces added in to clarify what the different fields are is shown below:

Output spike

8 2 1011101111 100 2 1101111110 001 2 1101101110 010 8 2 1011101111 100 2 110111110

001 2 1101101110 010 8

- First '8' shows the program ran until the first Loihi call.
- **Q** 2 number of input neurons that spiked
- □ 1011101111 spiking pattern of 10 middle layer neurons
- □ 100 spiking pattern of 3 output neurons
- **Q** 2 start of a repeating pattern for the next execution
- □ The second '8' shows that the program completed the multicore Loihi calls (the first 3 runs).
- □ The third '8' shows all lines of code were executed without error.

Conclusion



Implemented SDN application on neuromorphic hardware in space.

Future work:

Execute developed learning enabled SDN application into space.



Backup

Loihi Connection



Connected to server (laptop)

(python3_venv_clone) (loihi_env_clone) brisk@brisk-ils:~/Documents/AnacondaExamples/server_network_nengo\$ lsusb -t Bus 04.Port 1: Dev 1, Class=root_hub, Driver=xhci_hcd/4p, 10000M Bus 03.Port 1: Dev 1, Class=root hub, Driver=xhci hcd/2p, 480M Bus 02.Port 1: Dev 1, Class=root hub, Driver=xhci hcd/8p, 10000M Bus 01.Port 1: Dev 1, Class=root_hub, Driver=xhci_hcd/16p, 480M | Port 4: Dev 2, If 0, Class=Hub, Driver=hub/4p, 480M Port 3: Dev 8. If 1. Class=Vendor Specific Class. Driver=, 480M Port 3: Dev 8, If 0, Class=Vendor Specific Class, Driver=, 480M Port 1: Dev 4, If 0, Class=Vendor Specific Class, Driver=, 480M Port 1: Dev 4, If 1, Class=Vendor Specific Class, Driver=, 480M Port 4: Dev 9, If 0, Class=Vendor Specific Class, Driver=, 480M Port 2: Dev 6, If 0, Class=Vendor Specific Class, Driver=, 480M Port 2: Dev 6, If 1, Class=Vendor Specific Class, Driver=, 480M Port 5: Dev 3, If 1, Class=Human Interface Device, Driver=usbhid, 12M Port 5: Dev 3, If 2, Class=Human Interface Device, Driver=usbhid, 12M Port 5: Dev 3, If 0, Class=Human Interface Device, Driver=usbhid, 12M Port 10: Dev 5, If 2, Class=Human Interface Device, Driver=usbhid, 12M Port 10: Dev 5, If 0, Class=Human Interface Device, Driver=usbhid, 12M Port 10: Dev 5, If 1, Class=Human Interface Device, Driver=usbhid, 12M Port 14: Dev 7, If 0, Class=Wireless, Driver=btusb, 12M | Port 14: Dev 7, If 1, Class=Wireless, Driver=btusb, 12M (python3_venv_clone) (loihi_env_clone) brisk@brisk-ils:~/Documents/AnacondaExamples/server_network_nengo\$ `python3 -c "import nxsdk; print(nxsdk.__path__[0])")bin/x86/kb/lakemont_driver --test-fpio Using Kapoho Bay serial number 434

test_fpio_loopback: chain=0 chips=2 blocks=1000 time=10436us => 1.53315Mb/s

(python3_venv_clone) (loihi_env_clone) brisk@brisk-ils:~/Documents/AnacondaExamples/server_network_nengo\$

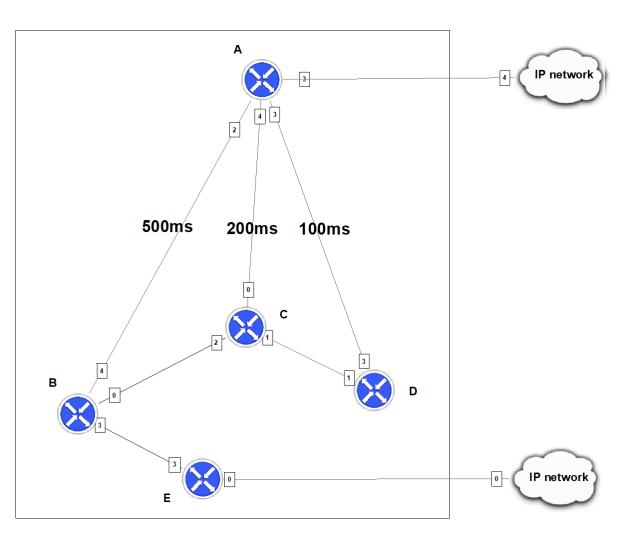
Output from Loihi



1612379556.617951 root DEBUG Entered run_network_onLoihī method:::::: [keras_server_mem.py:179 run_network_onLoihi]	
Build finished in 0:00:00	
# Optimizing graph: creating signals	0:00:001
612379556.642543 root DEBUG partitions [tensor_graph.py:1066 create_signals]	
1612379556.642638 root DEBUG	
_	
Optimization finished in 0:00:00	
Construction finished in 0:00:00	
INFO:DRV: Connecting to 127.0.0.1:42221	
1612379557.024463 DRV INFO Connecting to 127.0.0.1:42221 [host_coordinator.py:75 connect]	
INFO:DRV: Host server upDone 0.205	
1612379557.026053 DRV INFO Host server upDone 0.20s [nxlogging.py:266 timedContextLogging]	
INF0:DRV: Encoding axons/synapsesDone 1.34ms 1612379557.028196 DRV INFO Encoding axons/synapsesDone 1.34ms [nxlogging.pv:266 timedContextLogging]	
INF0:DRV: Compiling Embedded snipsDone 0.12s 1612379557.150447 DRV INFO Compiling Embedded snipsDone 0.12s [nxlogging.py:266 timedContextLogging]	
INFO:DRV: Booting upDone 2.95s	
1612379560.099556 DRV INFO Booting upDone 2.955 [nxlogging.py:266 timedContextLogging]	
INFO:DRV: Encoding probesDone 0.15ms	
1612375560.100240 DRV INFO Encoding probesDone 0.15ms [nxlogging.py:266 timedContextLogging]	
INFO:DRV: Transferring probesDone 1.24ms	
1612379560.119232 DRV INFO Transferring probesDone 1.24ms [nxlogging.py:266 timedContextLogging]	
INFO:DRV: Configuring registersDone 7.21ms	
1612379560.126871 DRV INFO Configuring registersDone 7.21ms [nxlogging.py:266 timedContextLogging]	
INFO:DRV: Transferring spikesDone 2.68ms	
1612379560.129928 DRV INFO Transferring spikesDone 2.68ms [nxlogging.py:266 timedContextLogging]	
INF0:DRV: ExecutingDone 0.09s	
1612379560.224451 DRV INFO ExecutingDone 0.09s [nxlogging.py:266 timedContextLogging]	
INF0:DRV: Processing timeseriesDone 0.79ms	
1612379560.225633 DRV INFO Processing timeseriesDone 0.79ms [nxlogging.py:266 timedContextLogging]	
INFO:DRV: Executor: 30 timestepsDone 0.11s	
1612379560.225747 DRV INFO Executor: 30 timestepsDone 0.11s [nxlogging.py:266 timedContextLogging]	
Predict Outcome_L is: [[0.49776891 0.54278774 0.62189235]]	
Predict Label_L is: [2]	
INFO:HST: Using Kapoho Bay serial number 434	
1612379560.236293 HST INFO Using Kapoho Bay serial number 434 [subprocess_logging_handler.py:36 run]	
INF0:HST: Args chip=0 cpu=0 /home/brisk/python3_venv_clone/lib/python3.5/site-packages/nxsdk/driver/compilers///temp/1612379557.0283532/launcher_chip0_lmt0.binchips=1remote-r 1612379560.236421 HST INF0 Args chip=0 cpu=0 /home/brisk/python3_venv_clone/lib/python3.5/site-packages/nxsdk/driver/compilers///temp/1612379557.0283532/launcher_chip0	
[1] = 1 - remote - relay=0 [subprocess logging handler.py:36 run]	inco.ben seereps
INPORTSI Lakemont driver	
1612379560.236482 HST INFO Lakemont driver[subprocess logging handler.py:36 run]	
INFORTS chipe opue halted, statuseoxo	
1612379560.236556 HST INFO chip=0 cpu=0 halted, status=0x0 [subprocess logging handler.py;36 run]	
1612379560.245826 root DEBUG Model output: 1 [keras server mem.py:10 flow route]	
1612379560.246007 root DEBUG Loihi output: [2] [keras_server_mem.py:411 flow_route]	

Task 3: Integration and Validation

- Goal: To experimentally demonstrate a LOIHI-based SDN/VNF routing approach
- Implementation:
 - Ryu controller
 - Open VSwitch (OVS)
 - MPLS routing provided through a VNF
 - OVS nodes are virtualized with a 1:1 mapping of ports to physical interfaces
 - Hardware OpenFlow switches to be tested









	demo.mp4	
	rient — rient@s30: ~/devei/sdn_exper/Exper1 — ssh + ssh -Y rient@polaris — 80×20	icc21 - rlent@s23: ~ - ssh • ssh -Y rlent@polaris - 84×21
🖸 Terminal > 🔰 ぞ 🍕 🛱 🗧	ort=3 [router_mpls.py:168 setupMplsPath]	cookie=0x0, duration=68.614s, table=0, n_packets=21, n_bytes=1795, priority=0
brisk@brisk=ils: -/Documents/AnacondaExamples/server_network_nengo 🔷 💿 🧔	1612378749.186211 root ERROR [s22] ipPkt@LSR: SHOULD NOT BE REACHABLE	
	[switch13.py:309 _packet_in_handler]	rlent@s23:~\$ sudo ovs-ofctl dump-flows s23
brisk@brisk-ils:-/Documents/AnacondaExamples/server_network_hengo × brisk@brisk-ils:-/Documents/AnacondaExamples/server_network_nengo × 🖪		cookie=0x0, duration=29.960s, table=0, n_packets=4, n_bytes=392, priority=1,ic
s12378763.509172 root DEBUG partitions [tensor_graph.py:1066 create_signals]	on": 0}] [agent_keras.py:35 sendReg]	_port=enp0s8,nw_src=192.168.108.26,nw_dst=192.168.101.27 actions=push_mpls:0x8
512378763.509338 root DEBUG	<pre>>>>>> sendReg rcvd: [{"action_loihi": 0, "action": 0}]</pre>	ad:0xe93->0XM_OF_MPLS_LABEL[],output:enp0s10
		cookie=0x0, duration=21.445s, table=0, n_packets=4, n_bytes=408, priority=1,m
512378763.908684 root DEBUG Predict Label is: 0 [keras_server_mem.py:164 run_network]	Route update: identical path, so DO NOT implement	
512378763.908889 root DEBUG Entered run_network_onLoihi method:::::: [keras_server_mem.py:179 run_network_onLoihi]	1612378763.306176 root DEBUG flow_route [agent_keras.py:57 flow_route	_port=enp0s10,mpls_label=3734 actions=pop_mpls:0x0800,mod_dl_src:08:00:27:3d:7
ulid finished in 0:00:00 ptimizing graph: creating signals (0:00:00)		od_dl_dst:08:00:27:0f:9a:1f,output:enp0s8
12378763.941074 root DEBUG partitions [tensor_graph.py:1066 create_signals]	1612378763.306459 root DEBUG sendReq: [{'type': 'route', 'lerName': '	cookie=0x0, duration=68.968s, table=0, n_packets=21, n_bytes=1795, priority=0
612378763.941287 root DEBUG	<pre>s25', 'flowID': 2}] [agent_keras.py:33 sendReq]</pre>	ns=CONTROLLER:65535
[tensor_graph.py:1008_create_signals] tutuazion finished in 9:00:00	1612378768.128899 root INFO send ARP reply 08:00:27:3d:7d:58 192.168	rlent@s23:~\$ sudo ovs-ofctl dump-flows s23
onstruction finished in 0:00:00	.108.23 08:00:27:0f:9a:1f 192.168.108.26 [router_arp.py:51 handleArpPkt]	cookie=0x0, duration=30.299s, table=0, n_packets=5, n_bytes=490, priority=1,i
MF0:DRV: Connecting to 127.0.0.1:39861		_port=enp0s8,nw_src=192.168.108.26,nw_dst=192.168.101.27 actions=push_mpls:0x8
512378764.931515 DBV INFO Connecting to 127.0.0.1:39861 [host_coordinator.py:75 connect] 1970/1971 Host Server un book Server un book 9.015	on": 0}] [agent_keras.py:35 sendReq]	ad:0xe93->0XM_OF_MPLS_LABEL[],output:enp0s10
512378764.933357 DRV INFO Host server upDone 0.81s [nxlogging.py:266 timedContextLogging]	<pre>>>>>> sendReq rcvd: [{"action_loihi": 0, "action": 0}]</pre>	<pre>cookie=0x0, duration=21.784s, table=0, n_packets=5, n_bytes=510, priority=1,m</pre>
IFD:IRV: Compiling Embedded snipsDone 0.12s	Route update: identical path, so DO NOT implement	_port=enp0s10,mpls_label=3734 actions=pop_mpls:0x0800,mod_dl_src:08:00:27:3d:7
123787856.565099 DRV INFO Compliing Enbedded snlpsDone 0.125 [nxlogging.py:266 tixedContextLogging] TOTINY: Encoding axong SynapsesDone 1.55ms	1612378768.190132 root INFO send ARP reply 08:00:27:1f:8a:ff 192.168	od_dl_dst:08:00:27:0f:9a:1f,output:enp0s8
612378765.052874 DRV INFO Encoding axons/synapsesDone 1.55ms [nxlogging.py:266 timedContextLogging]	.101.25 08:00:27:3b:a9:17 192.168.101.27 [router_arp.py:51 handleArpPkt]	cookie=0x0, duration=69.307s, table=0, n_packets=21, n_bytes=1795, priority=0
IF0:DRV: Booting upDone 2.94s		ns=CONTROLLER:65535
51378787.998277 DRV	Desktop — rient@polaris: ~/devel/sdn exper/Exper1 — ssh • ssh •Y rient@polaris — 80x21	rlent@s23:~\$
512378767.998997 DRV INFO Encoding probesDone 0.20ms [nxlogging.py:266 timedContextLogging]	Desktop — rlent@polaris: ~/devel/sdn_exper/Exper1 — ssh • ssh -Y rlent@polaris — 80×21	
MPD:10V: Transfering probesDome 1.39ms 13737056,02751 DRV I NFO Transfering probesDome 1.39ms [nxlogging.py:266 timedContextLogging]	\$27	Inent — rlent@s26: ~ — ssh • ssh -Y rlent@polaris — 84×20
Dizisiono.czz/i ow i info i ranisieri (ng probesone i isons [nixugy(ng,jy)zoo (ineuconcexiLogg(ng)] Melone: Configuring registersDone 6.98ms	SIOCADDRT: File exists	192.168.101.27 ping statistics
512378768.031642 DRV INFO Configuring registersDone 6.98ms [nxlogging.py:266 timedContextLogging]	rlent@polaris:~/devel/sdn_exper/Exper1\$ ls^C	1 packets transmitted, 0 received, 100% packet loss, time 0ms
الاتانية: Transferring spikesDone 3.35ns 13737858.037878 DRV I NFO Transferring spikesDone 3.35ns [nxlogging.py:266 timedcontextlogging]	rlent@polaris:~/devel/sdn_exper/Exper1\$ sh ^C	
Dice for the cost of the security and the cost of the security	rlent@polaris:~/devel/sdn_exper/Exper1\$ cat ini.sh	rlent@s26:~\$
512378768.128797 DRV INFO ExecutingDone 0.095 [nxlogging.py:266 timedContextLogging]	ssh s26 "ping -c1 192.168.108.23"	rlent@s26:~\$ ping -c 1 192.168.101.27
الاتانة: Processing timesertesDene 0.45ms 13378708.12940 RW INFO Processing timesertesDene 0.45ms [nxlogging.py:266 timedcontextlogging]	ssh s27 "ping -c1 192.168.101.25"	PING 192.168.101.27 (192.168.101.27) 56(84) bytes of data.
Processing times to the story of the story o	rlent@polaris:~/devel/sdn_exper/Exper1\$ sh ini.sh	64 bytes from 192.168.101.27: icmp_seq=1 ttl=64 time=2.85 ms
512378768.129581 DRV INFO Executor: 30 timestepsDone 0.11s [nxlogging.py:266 timedContextLogging]	PING 192.168.108.23 (192.168.108.23) 56(84) bytes of data.	
redict Outcome_L is: [[0.70975868 0.34358367 0.58251146]] redict LabeL_L is: [0]	64 bytes from 192.168.108.23: icmp_seq=1 ttl=255 time=2.38 ms	192.168.101.27 ping statistics
NFO:HST: Using Kapoho Bay serial number 434		1 packets transmitted, 1 received, 0% packet loss, time 0ms
\$12378768.140755 HST INFO Using Kapoho Bay serial number 434 [subprocess_logging_handler.py:36 run]	192.168.108.23 ping statistics	rtt min/avg/max/mdev = 2.857/2.857/2.857/0.000 ms
<pre>WPINSI: Args chtpd pub / home/brisk/python3.yem_clone/ltb/python3.s/site-packages/msski/jarver/compilers//./tenjsi1337874.934246/launcher_chtpd inte.binchtps1renote-relayed 1378788.14888 H5T INFO Args chtpd cupe / home/brisk/python3.yem/compilers//site-packages/fisski/jarver/compilers//.em/cite/jar27874.934426/launcher chtpd launcher chtpd Inte.binchtps 1378788.14888 H5T INFO Args chtpd cupe / home/brisk/python3.yem/cine/ltb/python3.yiste-packages/mski/jarver/compilers//.</pre>	1 packets transmitted, 1 received, 0% packet loss, time 0ms	rlent@s26:~\$ ping -c 10 192.168.101.27
1remote-relay=0 [subprocess_logging_handler.py:36 run]	rtt min/avg/max/mdev = 2.380/2.380/2.380/0.000 ms	PING 192.168.101.27 (192.168.101.27) 56(84) bytes of data.
<pre>FPD:HST: Lakemont driver FPD:HST: Lakemont driver.</pre>	PING 192.168.101.25 (192.168.101.25) 56(84) bytes of data.	64 bytes from 192.168.101.27: icmp_seq=1 ttl=64 time=1.98 ms
123787808.140954 H5T INFO Lakemont_driver[subprocess_logging_handler.py:36 run] Pr0:H5T chinos cpueñ halted, status=300	64 bytes from 192.168.101.25: icmp_seq=1 ttl=255 time=2.19 ms	64 bytes from 192.168.101.27: icmp_seq=2 ttl=64 time=1.71 ms
612378768.141006 HST INFO chip=0 cpu=0 halted, status=0x0 [subprocess_logging_handler.py:36 run]	and a second sec	64 bytes from 192.168.101.27: icmp_seq=3 ttl=64 time=1.72 ms
612378768.152217 root DEBUG Model output: 0 [keras server mem.py:410 flow route]	192.168.101.25 ping statistics	64 bytes from 192.168.101.27: icmp_seq=4 ttl=64 time=1.73 ms
513787686,15243 root DEBUG Lothioutput:[0][keras,server_nem.py:411 flow_noute] 97.705.52 - [0]/feb/2021 J.1559:281 "POT J. HTTP/.112 200 -	1 packets transmitted, 1 received, 0% packet loss, time 0ms	64 bytes from 192.168.101.27: icmp_seq=5 ttl=64 time=1.75 ms
612378768.152727 root DEBUG RPY: {"action_loih!": 0, "action": 0} [keras_server_men.py:337 do_POST]	rtt min/avg/max/mdev = $2.196/2.196/2.196/0.000$ ms	64 bytes from 192.168.101.27: icmp_seq=6 ttl=64 time=1.74 ms
ayim Rahman	rlent@polaris:~/devel/sdn_exper/Exper1\$	04 0y cc3 110m 152.100.101.27. 10mp_Scq=0 (01=04 01mc=1.74 mS