Software Defined Network (SDN) --- Mininet, OVS, P4 switch, POX, RYU Learning Guide

Miniet,P4中文影音講解 (YouTube,內容持續更新中，百度雲盤,提取碼: d6wg**)**

[lab experiment]

A fast reverse proxy to help you expose a local server behind a NAT or firewall to the internet (2020/2/20 done)
SSH Tunnel (2018/12/27 done)
Static routing vs Dynamic Routing (Quagga: RIPv2) (2020/04/07 done)
Simple rtsp server and proxy example (2020/4/9 done) Extended work (2020/4/13 done)
How to add a node and a link at runtime? (2020/4/29 done)
Linux virtual server (NAT and DR) and HAProxy (2020/5/28 done)
One PC Two interfaces: one for send and the other for receive (2020/6/11 done)
Minievents: A mininet Framework to define events in mininet networks (2020/7/24)

[P4 Switch] Code can be downloaded from here.

P4-utils is an extension to Mininet that makes P4 networks easier to build, run and debug.
When I taught my students, the students recorded what I have taught. The voice and video quality may not be good. For reference only. (video, 2020/05-2020/06)

1. Basic examples. (2020/4/25 done)
2. L3 Routing (2020/4/25 done)
3. Simple Controller to populate routing tables for any topology. (2020/4/25 done)
4. Monitor Link BW (2020/4/30 done)
5. reboot p4 switch (2020/5/3 done)
6. Measure one way delay (2020/5/3 done)
7. Send to CPU (2020/5/7 done)
8. Implement the PacketIn function to install the routing rules (2020/5/14 done)
9. ECMP (Equal Cost MultiPath) Test (2020/5/15 done)
10. implement the idea in “A Zero Flow Entry Expiration Timeout P4 Switch” (2020/5/16 done)
11. Anti TCP SYN Port Scan (2020/5/30 done) (2020/6/11 updated)
12. H.264 RTP video streaming over P4 networks (2020/6/12 done)

13. Using SVM to classify whether the traffic is normal or malicious ICMP attack (2020/6/27 done)
14. Mininet-wifi(p4) roaming (2020/7/14 done)
15. Simple controller 2 to handle the failed link and re-calculate the routing table (2021/3/20 done)
16. Simple controller 3 to handle the failed link (like fast failover) (2021/3/21 done)
17. Swap P4 program at runtime (2021/4/1 done)

My P4 VM (2017/11/19 done)

My First P4 program (2017/11/19 done)

P4_14 Spec: VM (2018/3/6 done)

P4 Switch: MAC Layer and IP Layer Forwarding (2018/3/7 done)
P4 Switch(p4-16): MAC Layer and IP Layer Forwarding (2018/8/9 done)
P4 switch(p4-16): source routing (Add the routing information at the ingress switch of network) (2018/8/13 done) method2 (2018/9/2 done)
P4 switch(p4-16): multipath transmission (2018/8/25 done) p4-multipath.zip
P4 switch(p4-16): monitor the queue length of different output interfaces (2018/8/26 done) method2 (2018/8/27 done)
P4 switch(p4-16): simple thrift controller to change the traffic transmission path (2018/8/29 done)
P4 switch(p4-16): video streaming (with priority queues) (2018/8/30 done)
P4 switch(p4-16): recirculate() example (2018/9/2 done)
P4 switch(p4-16): simple token bucket shaper (2018/9/8 done)
P4 switch(p4-16): P4 + OVS(openvswitch) mininet (2018/9/9 done)
P4 switch(p4-16): Broadcast (2018/9/9 done)
P4 switch(p4-16): meter example (2018/9/12 done)
P4 switch(p4-16): counter example->get link bandwidth consumption (2018/9/15 done)
P4 switch(p4-16): clone() and mirroring_add example (2018/9/18 done)
P4 switch(p4-16): lookahead, varbit example: count the number of http request (GET) and http response packets (2018/9/26 done)
P4 switch + mininet + docker host (2018/10/8 done)

P4 switch + DPI example (2018/10/17 done)
P4 switch + DPI example 2 (2018/10/25 done)
P4 Switch: P4-DNS (2018/12/9 done)
My Implementation for "High-Speed Data-Plane Packet Aggregation and Disaggregation by P4 Switches" (2019/6/18 done)
P4 Switch(P4-16): GRE TUNNEL (2019/6/21 done)
P4 Switch(P4-16): anti-Port Scan (2019/6/25 done)
P4 Switch(P4-16): Port Knocking(2019/6/26 done)
P4 Switch(P4-16): Two different P4 switches (2019/7/2 done)
P4 Swtich(P4-16): anti-Sync Attack (2019/7/3 done)
How to run p4 bmv2 docker in mininet? (2019/12/30 done)
Using Linux (Ubuntu) VM as a P4 software switch (2020/6/29 done)

P4runtime
P4runtime: dynamically change the transmission paths (2018/7/24 done)

(load balancer for http service)
Connection Hash Load Balancer for P4 switch (2018/4/18 done, 2020/5/21 updated)
Round-Robin Load Balancer for P4 switch (2018/6/27 done, 2020/5/21 done) P4 switch(p4-16) round-robin load balancer with 4 servers (2018/12/26)
P4 Switch(p4-16): random load balancer (2019/7/17 done, 2020/5/21 updated)
P4 Switch(p4-16): weighted round robin load balancer (2019/7/17 done, 2020/5/21 updated)
Dynamic Round Robin Load Balancer (2019/12/21 done) version2: with fault detection (2020/2/25 done)
Performance evaluation (2019/12/21 done)
First implementation of my MOST project (MOST 108-2221-E-507-005-) (2019/12/21 done)
The second version of implementation for my MOST project (MOST 108-2221-E-507-005-) (2020/4/29 done)

Mininet + K3S (2020/1/15 done)

[mininet-Q&A]

1. How to run the command for mininet host in the root namespace ? (2021/3/14 done)
2. Count packets on openvswitch (2021/3/14 done)
3. Mininet-p4 3 QA (for my youtube video) (2021/4/30 done)

[**My Lab— with Pox Controller**]

Lab 1: Create a network and run a simple performance test (2013/8/2 done)

Lab 2: Create a simple network and use a POX controller to contrl the behaviors of switch (2013/8/2 done)

Lab 3: Use "ovs-vsctl" command to directly control open vswitch (2013/8/7 done)

Lab 4: Advanced "ovs-vsctl" usage examples (2013/8/7 done)

Lab 5: Dynamically change the network parameters—change link delay (2013/8/7 done)

Lab 6: A simple controller (2013/8/7 done)

Lab 7: Measure the Path Loss Rate (2013/8/16 done)

Lab 8: Measure the Latency (2013/11/10 done)

Lab 9: Limit the bandwidth (2014/02/17 done)

Lab 10: Test features request/reply (2014/02/17 done)

Lab 11: Dynamically change the forwarding rules (2014/02/18 done)

Lab 12: Mininet Host talking to the Host on NCTUNS (2014/03/12)

Lab 13: Using Bellman-Ford to find a shortest path (2014/06/26)
For

Mininet Random Topology Generator (2016/01/22 done)

Generate Multiple Paths with equal cost (2016/02/29 done)

How to get the total number of openflow packets that are transmitted between controller and switches? (2016/8/25 done)

Using FNSS (Fast Network Simulation Setup) to build network topology for mininet emulation (2016/9/13 done)
Mininet Host Talk to Real PC (2016/10/12 done)
Router in mininet (2017/1/29 done)

Path Recovery when link is down (2017/7/23 done)
Simple NAT (2017/9/20 done)
A Mininet host in one VM talks to anther host in another VM via GRE tunnel (2018/2/23 done)
How to monitor the queue length in the mininet ? (2018/3/2 done)

[**QoS**]
Metering Function (2018/3/21 done)
Metering Function and Queueing (2018/3/21 done)

[**MiniNAM: A Network Animator for Mininet**]
MiniNAM (2018/4/17 done)

[**Pyretic + Mininet**] Read this before you run pyretic (Tips)

1. Routing (2015/6/23 done) Dijkstra's Algorithm (2015/7/5 done)

2. Flow Monitor (2015/6/24 done) Throughput Measurement

3. Switch Monitor (2015/6/28 done) Bandwidth Consumption Measurement

4. Change the path when packet transmission (2015/6/28 done)

5. Find a path with maximum capacity with Dijkstra's Algorithm (2015/7/7 done)

6. Find all shortest paths (2015/7/9 done)

7. Yen's K-Shortest Paths (2015/7/18 done)

8. Select one from all shortest paths with round robin method (2015/7/21 done)

9. Reduce the time of find a route (2015/7/21 done)

10. multiple controller: pyretic + pox (2015/7/22 done)

11. For my reader's question (2015/11/4 done)

12. Multipath Dijkstra Algorithm & ECMP (2016/3/21 done)

13. Available Bandwidth BasedDijkstra's Algorithm (2016/5/10 done)

14. Fat Tree Topology (with Dijkstra's Algorithm) (2016/5/17 done)

[RYU + mininet]

1. Shortest path + OpenFlow13 (2015/12/21 done)
2. Multipath transmission using RYU (2015/12/22 done)
3. Dijktra Algorithm implementation in RYU (2016/9/21 done)
4. Add a meter in RYU (2016/9/21 done)
5. SDN experiment using Estinet RT188T switches (2016/12/28 done)
6. How to create a simple network using switches that support openflow version 1.3 ? (2017/1/7 done)
7. How to apply a meter in a user-level switch? (2017/6/11 done)
8. Ingress Policing + Queue (2017/6/19 done)
9. Find the maximum capacity path with Dijkstra's Algorithm in accordance with current network status (2017/6/20 done)
10. Rest API (2020/4/28 done)

[RYU + mininet + NFV]

Monitor function (2017/1/29 done)

Firewall , Rate Limit (2017/5/7 done)

Simple SDN/NFV example (2017/6/19 done)

[VND and mininet-wifi]

1. @ramonfontes videos (VND: Very Useful Tool) (mini-wifi: allows the using of both WiFi Stations and Access Points) (Video Demo)

Mininet-WiFi: SDN emulator supports WiFi networks mininet-wifi-discuss

wireshark experiment: lab1:Beacon Analysis (Chinese) lab2:Authentication (Chinese) lab3:Association (Chinese)

2. coursera: SDN

3. mininet-wifi + pyretic: test1 (the controller allows the wired host to ping sta1, but not sta2) (2015/11/10 done)

4. mininet-wifi: test2 (one wireless stations with two physical interfaces. Each interface connects to different APs) (2015/11/17 done)

5 mininet-wifi: test3 (single path transmission vs. multiple paths transmission) (2015/11/17 done)

6. openflowPacketFilter.py (2016/4/20 done)

7. in-band control (2016/4/21 done)

[Video transmission evaluation over mininet]

myEvalSVC-Mininet (1) (2015/01/30 modified)

myEvalSVC-Mininet (2) (2015/01/30 done)

VLC over SDN (2015/2/1 done)

myEvalSVC-Mininet (3) (2015/02/24 done)

ffmpeg streaming (2016/10/7 done) Use ffmpeg to encode the video and do the streaming ffmpeg rtp streaming (2018/6/13 done)

Mulitpath Transmission for Improved Video Delivery over software Defined Wired and Wireless Networks (2016/11/2 done) (Chinese version)

HEVC video transmission over mininet (2016/11/16 done)

ffmpeg transcoding (2017/4/7 done)

Preferential detour of unimportant data stream in software defined networks for improving video transmission quality (2018/4/1 done, in Chinese)

Using Kodo Library (Network Coding) to provide Enhanced Video QoS (2018/12/29 done)

How to multicast video using VLC ? (2019/01/22 done) Multicast Ping Test (2019/01/23)

[My SDN vmware image]

VM (login name and password: mininet/mininet)

dockernet VM (When the dockernet.ova.rar file has been downloaded, rename the file as dockernet.ova. Import this file into your virtualbox or vmware.)

useful mininet script files

[My Talking Slide and Other Related Labs]

Software Defined Network (SDN) experiment using Mininet and POX Controller

Lab 1: basic mininet operations

Lab 2: manually control the switch

Lab 3: move the rules to the POX controller

Lab 4: set different forwarding rules for each switch in the controller

Lab 5: set traffic to different output queues (QoS issue) (2014/01/12 done)

Lab 6: FlowVisor (2014/01/21 done)

Lab 7: Multiple Tables Test (2014/01/25 done)

Lab 8: IP Load Balancer (2014/1/27 done)

Lab 9: Traffic measurement (2014/09/28 done) Traffic measurement 2 (IP -> IP with mask --> TCP/UDP/ICMP) (2014/11/11 done)

Lab 10: Duplicate Packets (2015/1/26 done)

Lab 11:Bridge remote mininets using VXLAN (2015/1/27 done)

Lab 12:Using l2_multi to find a shortest path (2015/1/29 done) l2_multi.py (implement Floyd-Warshall algorithm: find the shortest paths between all pairs of vertices. Refer to GeeksforGeeks for more information)

Lab 13:Using l2_bellmanford to find a shortest path (2015/2/3 done) (Bellman-Ford algorithm: computes the shortest paths from a single source to all of the other vertices)

Lab 14:Traffic Volume Control (2015/3/16 done)

Lab 15: A host with two interfaces (2015/4/9 done) Ring Topology (2015/4/18 done) Three Hosts (2015/05/07 done) switch host switch host2   switch host3 (2015/5/11 done) application layer routing (2015/5/11 done) 0629Test (2015/6/29 done)

Lab 16: IPv6 example (2015/5/8 done)

Lab 17: IPv4 GRE Tunnel (2015/5/18 done)

Lab 18: Ingress Rate Limit (2015/5/27 done)

Lab 19: Stochastic Switch using Open vSwitch in Mininet (2015/7/31 done)

Lab 20: Performance evaluation of UDP flow transmission: single path vs. multiple paths (2015/9/4 done)

Lab 21:How to use iperf over mininet? (2015/9/15 done)
iperf Question (2018/3/20 done) performance measurement (get the packet delay) (2018/4/8 done)

Lab 22:How to use bonding to increase the throughput? (2015/9/23 done)

Lab 23: Mininet Operations (2015/9/29 done: configure host as a router; DHCP; NAT; GRE tunnels) VLAN (2015/9/29 done) Wireshark Monitor Results (2015/9/30 done) Configure a Linux Bridge as a Hub (2015/10/27 done)

Lab 24: Spanning Tree Protocol (2015/10/27 done)

Lab 25:MPTCP Test (multipath tcp) (2016/12/17 done)

Lab 26: arpsoofing behavior (2017/1/7 done)

Lab 27:Test Fast-Failover Group in OpenFlow 1.3 (2017/12/29 done)

Lab 28:Test Select Group in OpenFlow 1.3 (2018/1/1 done)

Lab 29: Test Group Chaining in OpenFlow 1.3 (2018/1/2 done)

Lab 30: Openvswitch Port Mirroring (2019/2/13 done)

Lab 31: Intrusion detection using Suricata in SDN (2019/2/23 done)

Lab 32: Network Bonding + MPTCP (2019/7/9 done)

[**mininet + ns3 for SDN and WIFI simulations**: 2014/09/06 done]

virtual machine: download (This work is based on https://github.com/mininet/mininet/wiki/Link-modeling-using-ns-3 . I have added the lxde for graphical mode operation. I also added some examples under /home/mininet/examples/ns3 for SDN and WIFI simulations. Please download the ova file. You can use VirtualBox: Import Appliance or Vmware to use this vm. Note: when you have downloaded the mininet-ns3.ova.rar, please directly change the file name to mininet-ns3.ova.)

1. Example 1:

h1(10.0.0.1)<---wifi-network-a--->[AP1--- s3 (open vswitch)---AP2]<---wifi-network-b--->h2(10.0.0.2) wifi-network-a and wifi-network-b: 802.11a or 802.11b

2. Example 2:

h1(192.168.0.2)--wireless link--h0 (wireless router:192.168.0.1 and 10.0.01)-- s0(OVS)--h3(10.0.0.2)

h2 (192.168.0.3)--wireless link--

3. Example 3:

h1(192.168.0.1)--wireless link—s0 (AP + open vswitch)--wired link--h3(192.168.0.3)

h2 (192.168.0.2)--wireless link--

4. Example 4: QoS example (2014/10/02 done)

h1(192.168.0.1)--wireless link (EDCA supported)--s0 (AP + open vswitch)--wired link--h3(192.168.0.3)

h2 (192.168.0.2)--wireless link(EDCA supported)--

5. Example 5: Bellmanford + Host 1 --wired link-- s3(open vswitch) -- (WIFIBridgeLink) -- s4(open vswitch) -- (WIFIBridgeLink) -- s5(open vswitch) --wired link-- Host 2 (2015/3/16 done)

6. How to run wifiroaming.py in Opennet? (2015/5/26 done)

[中文影音介紹]

環境安裝 (2016/3/13 done) 基本操作一 (2016/3/13 done) 基本操作二 (2016/3/14 done) dockernet基本操作(2016/3/14 done) iperf3 gnuplot操作 (2016/3/14 done)

橋接器與集線器 (2016/3/15 done) 路由器與靜態路由 (2016/3/16 done) dockernet:動態路由 (2016/3/21 done)

Mininet Host Talk to VirtualBox XP VM (2016/3/20 done) VirtualBox VM talks to another VM via mininet network (2016/3/22 done)

NAT與DHCP(2016/3/27 done) DHCP-HELPER (2016/3/27 done)

IPv6簡介1 (2016/3/28 done) IPv6簡介2 (2016/3/28 done) GRE Tunnel (2016/4/11 done) DNS server (2016/4/11 done) PPTP server (2016/4/13 done) ip rule的使用(2016/4/14 done)

Mac Address Table Overflow Attack (2015/4/24 done) DHCP masquerade Attack(2016/4/26 done) Man in the Middle Attack:ettercap (2016/4/27 done) Man in the Middle Attack:bettercap (2016/5/15 done)

我的專題生: 蘇小民 CSIE NQU CTF

SDN:

vnd與 openvswitch基本操作 (2016/3/17 done) mininet與pox controller (2016/3/20 done) openvswitch basic operations (2016/5/15 done)

[Linux Bonding]

Fault Tolerance Test (2016/11/2 done) (Chinese version)

Increase Throughput Test (2016/11/2 done) (Chinese version)

[**Tun/Tap**]

How to send traffic through tunnel? (2017/5/18 done)

[**Security**]

DOS attack (2018/3/16)

[**VPN**]

Let Private or Public host talk to the server in the Private networks (2018/3/20 done)

[**References**]

POX: https://openflow.stanford.edu/display/ONL/POX+Wiki

RYU SDN Framework: http://osrg.github.io/ryu/ RYU Controller Tutorial: http://sdnhub.org/tutorials/ryu/

Mininet: http://mininet.org/

https://github.com/littlepretty/VirtualTimeForMininet (VT-Mininet: Virtual Time Enabled Mininet for SDN Emulation)

OpenFlow: http://archive.openflow.org/wk/index.php/OpenFlow_Tutorial

NetVis-Making Network Visualization Easy: http://www.cs.toronto.edu/~yujiali/proj/netvis.html

REPRODUCING NETWORK RESEARCH: https://reproducingnetworkresearch.wordpress.com/gallery/

ovs-ofctl: http://openvswitch.org/cgi-bin/ovsman.cgi?page=utilities%2Fovs-ofctl.8

ofpeck: http://archive.openflow.org/wk/index.php/Ofpeck

OpenFlow Experiment in Real-Time Internet Edutainment: http://users.ecs.soton.ac.uk/drn/ofertie/

Sflow provides a means for exporting truncated packets, together with interface counters.
http://blog.sflow.com/2013_05_01_archive.html
http://blog.sflow.com/2014/04/mininet-integrated-hybrid-openflow.html

Windy’ s Software Defined Networks: http://windysdn.blogspot.tw/

Tech and Trains(MiniEdit): http://gregorygee.wordpress.com/

OpenDayLight:1) http://sdnhub.org/tutorials/opendaylight/
2)http://www.opendaylight.org/announcements/2014/02/opendaylight-delivers-open-source-software-enable-software-defined-networking

Link modeling using ns3: https://github.com/mininet/mininet/wiki/Link-modeling-using-ns-3

OpenNet: A Simulator for Software-Defined Wireless Local Area Network (built on top of mininet and ns-3)

SDNAP (SDN ASSOCIATED PRESS)

roan’ s Blog : openvswitch setup

hwchiu’ s Blog : Multipath routing with Group table at mininet

http://www.muzixing.com/ (multipath and QoS application on RYU) http://dtucker.co.uk/hack/building-a-router-with-openvswitch.html (building a router with open vswitch)

https://github.com/netgroup/wmSDN (wireless mesh software defined network)

https://github.com/OFERTIE/ofsoftswitch13-testing (test environment for openflow 1.3)

https://github.com/alexcraig/GroupFlow (multicast over SDN with openflow)

http://dtucker.co.uk/hack/building-a-router-with-openvswitch.html (building a router with Open vSwitch)

https://github.com/saeenali/openvswitch/wiki/Stochastic-Switching-using-Open-vSwitch-in-Mininet (Stochastic Switching using Open vSwitch in Mininet)

SDN Fun! (Working with Networks/Graphs in Python)

OpenFlow & Mininet (written in Chinese)

http://sdntutorials.com/sdn-resources/ (SDN Resources)

https://wiki.onosproject.org/display/ONOS/ONOS+for+Newcomers (ONOS for Newcomers)

MaxiNet (Distributed Emulation of Software-Defined Networks: it extends the Mininet emulation environment to span the emulation across several physical machines. This allows to emulate very large software-defined networks.)

miniNExt (it is an extension layer that makes it easier to build complex networks in Mininet. It includes: Routing engines, Servers, Connectivity components, NAT, and Network Management components.)

SDNLAB (包含mininet, openvswitch, openflow, opendayligh, onons, SDN 工具系列實驗)

SDN-tutorial (github,中文)

SDN-Work (Takeshi’s SDN開發筆記)

Awesome SDN (A awesome list about Software Defined Network)

Learning Network Programming (Researching switched networks programming approaches and solutions): OPENFLOW POX

Open-Source Routing and Network Simulation: mininet OpenDaylight

https://github.com/muzixing/ryu/tree/master/ryu/app (many useful ryu applications) https://github.com/OpenState-SDN/ryu/wiki (OpenState-SDN/ryu: MAC Learning, Port Knocking, Server Load Balancing, DDoS mitigation)

https://github.com/xinguard/XinUI (XinUI is a simple UI for Ryu SDN Controller)

https://github.com/cstracy/Multicast_SDN (An Ryu application of multicast)

https://github.com/warsang/-L3_Ryu_Fakeway_ECMP_MPTCP (implementing QoS on an Openflow enabled network through use of MPTCP and ECMP)

https://www.rootusers.com/how-to-configure-network-teaming-in-linux/ (How to Configure Network Teaming in Linux)

Compile openvswitch v2.7.0 on Ubuntu 16.04.2 LTS

Interesting use cases of RyU controller and OVS

[**Network Function Virtualization**]

https://netlab.dcs.gla.ac.uk/projects/glasgow-network-functions

https://github.com/UofG-netlab/gnf-dockerfiles

http://sb.tmit.bme.hu/mediawiki/index.php/ESCAPEv1#Overview

https://github.com/hsnlab/escape

[**Docker**]

1. dockernet (Extends Mininet API to use Docker containers as Mininet hosts.)

2. Running GUI apps with Docker (mytest)

3. 比較save, export對於映像檔檔操作差異

[**Pyretic**]

1. Python + Frenetic =Pyretic Source: https://github.com/frenetic-lang/pyretic

2. Composing Software-Defined Networks

[**Tun/Tap**]

http://backreference.org/2010/03/26/tuntap-interface-tutorial/ (Tun/Tap interface tutorial)

https://github.com/khuevu/http-tunnel (Tunnel tcp connection through http in Python)

http://vinllen.com/tun-tap/ (tun/tap 運行機制)

https://github.com/montag451/pytun (Linux Tun/Tap wrapper for python)

[**P4**]

https://github.com/p4lang/tutorials (p4 tutorial)

http://p4.org/wp-content/uploads/2017/05/p4_d2_2017_p4_16_tutorial.pdf (P4_16 Introduction)

http://www.maojianwei.com/2016/06/15/P4-Programming-Protocol-Independent-Packet-Processors/ (Chinese, P4: 編寫協議無關的包處理器)

https://github.com/p4lang/p4app (p4app)

https://github.com/nsg-ethz/p4-learning (Compilation of P4 exercises, examples, documentation, slides for learning or teaching)

https://github.com/cslev/p4extern (how extern functions should be implemented)

[**p4-security**]

http://www.sdn-anti-spoofing.net/ (Network Anti-Spoofing with SDN Data plane)

https://github.com/Emil-501/block.p4 (Using P4 to realize P4-based NFs)

https://github.com/sendendi/Early-DDoS-Detection-on-Stateless-Device (Dearly DDoS Detection on Stateless Device)

https://github.com/zhangmenghao/p4research (DDoS Mitigation Using Switching ASICs)

https://github.com/hiwang123/HappyFlowFriends (Cloud-based DoS protection) https://github.com/JJK96/P4-filtering (Filtering DDoS traffic using the P4 programming language)

[vxlan]

https://www.cnblogs.com/wipan/p/9220615.html

［**segment routing**］

https://github.com/netgroup/srv6-mininet-extensions

https://www.sdnlab.com/22842.html Linux SRv6实战：VPN、流量工程和服务链（第一篇）

https://www.sdnlab.com/22900.html Linux SRv6实战 服务链功能详解（第二篇）

https://www.sdnlab.com/23218.html Linux SRv6实战 （第三篇） 多云环境下Overlay(VPP) 和Underlay整合测试

https://www.sdnlab.com/23420.html Linux SRv6实战（第四篇）-“以应用为中心”的Overlay & Underlay整合方案

https://www.sdnlab.com/20500.html Linux下SRv6及Mininet IPv6安装配置发包测试

https://www.sdnlab.com/23390.html uSID： SRv6新范式

［**openvswitch + Docker**］

https://www.itread01.com/content/1544369064.html 基於openvswitch+Docker構建SDN網路測試環境 (使用ovs-docker進行構建)

https://www.twblogs.net/a/5b8ae2a22b71775d1ce9a1dc 用ovs-docker讓容器網絡支持Vlan隔離

［**bier**］

https://www.epizeuxis.net/index.php/topics/reliable-content-distribution/

https://bitbucket.org/wb-ut/p4-bfr/src/master/

https://github.com/uni-tue-kn/p4-bier

[**Contact Information**]

Dr. Chih-Heng Ke

Department of Computer Science and Information Engineering,
National Quemoy University, Kinmen, Taiwan

Email: smallko@gmail.com / smallko@nqu.edu.tw