

# Workshop OpenFlow

## 4. Rate limiter (traffic metering)

---

Akbari Indra Basuki

Pusat Penelitian Informatika, LIPI

# Daftar Materi

---

Topik	Keterangan
Basic forwarding	Dasar-dasar OpenFlow
Routing & Monitoring	Program Controller: Shortest-path routing Monitor node and link status Integrasi NetworkX dan matplotlib
Packet Filtering (Firewall + Web Interface)	Program Controller: Bloom Filter, Flask
Load balancing	Group bucket and group tables Round robin load balancing Main-backup path protection
Rate limiting	Meter table
Stateless vs Stateful data plane Stateful data plane	Jenis data plane dalam memproses paket OpenState SDN Arp handling Port Knocking

# Rate limiter (meter table)

---

Bagaimana operator membatasi kecepatan jaringan untuk setiap pengguna?

- Menggunakan meter table berdasarkan alamat IP pengguna

OpenFlow mendukung pembatasan kecepatan dan bandwidth menggunakan Meter table

Hanya berjalan pada userswitch (bukan OpenVSwitch)

Jenis pembatasan kecepatan:

- Drop limiter
- DscpRemark limiter

Acuan pembatasan:

- Ukuran paket: OFPMF\_KBPS
- Jumlah paket: OFPMF\_PKTPS
- Paket burst: OFPMF\_BURST
- Statistik paket: OFPMF\_STATS

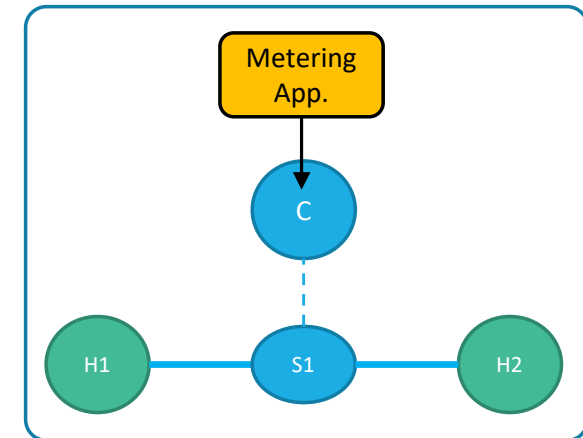
# Studi kasus – Speed limiter dengan meter table

## Skenario:

- Switch S1 dapat membatasi kecepatan pengiriman paket dari Host 1 ke Host 2
- Kecepatan pengiriman ditentukan 100 Kbps

## Implementasi:

- Switch 1: Meter table SELECT dengan mode DropLimiter berdasarkan ukuran paket (OFPMF\_KBPS)
- Controller menginstall flow rule secara statis ketika switch terkoneksi



# Source code

```
@set_ev_cls(ofp_event.EventOFPSwitchFeatures, CONFIG_DISPATCHER)
def switch_features_handler(self, ev):
    msg = ev.msg
    dp = msg.datapath
    ofproto = dp.ofproto

    #pasang band meter
    bands = [dp.ofproto_parser.OFPMeterBandDrop(rate=100, burst_size=10)]
    req = dp.ofproto_parser.OFPMeterMod(datapath=dp, command=ofproto.OFPMC_ADD, flags=ofproto.OFPMF_KBPS, meter_id=1, bands=bands)

    dp.send_msg(req)
    print "band meter rule installed"

    #atur packet yang dikenai aturan meter
    match = dp.ofproto_parser.OFPMatch(in_port=1, eth_type=0x0800, ipv4_src="10.0.0.1")
    actions = [ dp.ofproto_parser.OFPACTIONOutput(2)]
    inst = [ dp.ofproto_parser.OFPInstructionActions(ofproto.OFPIT_APPLY_ACTIONS, actions), dp.ofproto_parser.OFPInstructionMeter(meter_id=1)]
    mod = dp.ofproto_parser.OFPFlowMod(datapath=dp, table_id=0, priority=10, match=match, instructions=inst)
    dp.send_msg(mod)
    print "flow meter rule installed"

    #default: semua paket IP tanyakan ke controller (priority=0)
    match = dp.ofproto_parser.OFPMatch(eth_type=0x0800)
    actions = [dp.ofproto_parser.OFPACTIONOutput(ofproto.OFPP_CONTROLLER, ofproto.OFPCML_NO_BUFFER)]
    self.add_flow(dp, match, actions)
```

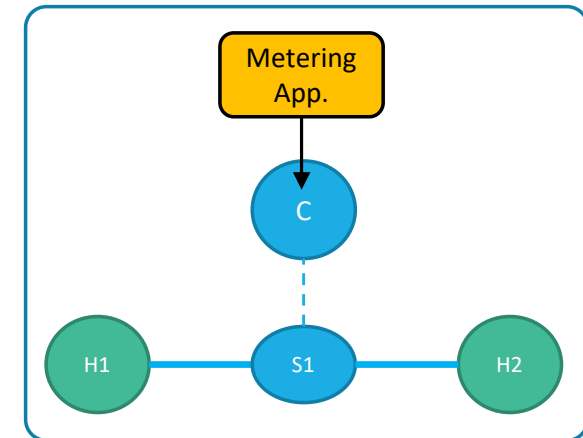
Metering per-ukuran paket 100 Kbps

Kirim ke meter table/limiter sebelum paket di forward paket ke Host 2

# Langkah pengujian

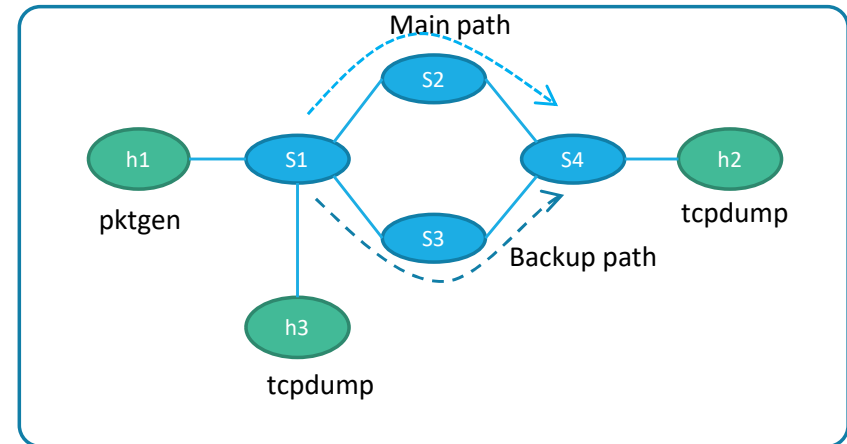
## Pengujian:

- Jalankan aplikasi controller (simpleMeter.py), meter 100 Kbps  
`Ryu-manager simpleMeter.py`
- Jalankan mininet, topologi: simpleTopo.py  
`Sudo python simpleTopo.py`
- H2 menjalankan iperf server  
`Iperf -s -i 1 -u`
- H1 menjalankan iperf client, throughput 1000 Kbps (1 Mbps)  
`Iperf -c 10.0.0.2 -u -b 1000k -t 10`  
Ctrl+C untuk mengakhiri program tcpdump
- Amati perbedaan throughput yang dapat dihasilkan oleh switch tanpa (simpleswitch.py) dan dengan metering (simplemeter.py)
  - Apakah throughput keduanya sama (1 Mbps) atau menurun seperti meter table (100 Kbps)?



# Latihan: Variasi jenis meter

1. Ubah kode untuk mengimplementasikan jenis metering berdasarkan Jumlah paket (OFPMF\_PKTPS)
2. Ubah mode dari drop limiter menjadi mode DscpRemark (OFPMeterBandDscpRemark)



# Daftar pustaka

---

1. [https://ryu.readthedocs.io/en/latest/ofproto\\_v1\\_3\\_ref.html](https://ryu.readthedocs.io/en/latest/ofproto_v1_3_ref.html)
2. [http://csie.nqu.edu.tw/smallko/sdn/ryu\\_meter.htm](http://csie.nqu.edu.tw/smallko/sdn/ryu_meter.htm)