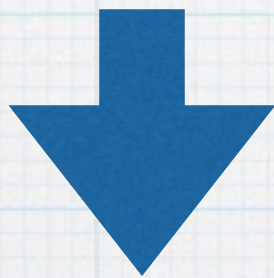


OpenFlow 1.3を
使いこなそう
(ルータを作ろう)

Hello
World

スイッチ
ルータ



トポロジ
ディスカバリ

ルーティング
スイッチ

仮想NW

レポート(グループ課題)

simple_routerをOpenFlow1.3にしよう

1.まずは単純に

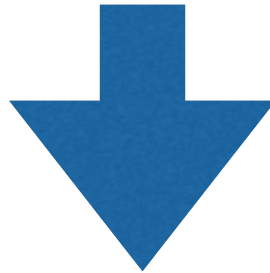
actions: [A1, A2, ...]

→ instructions: Apply.new([A1, A2, ...])

2.マルチプルテーブル化

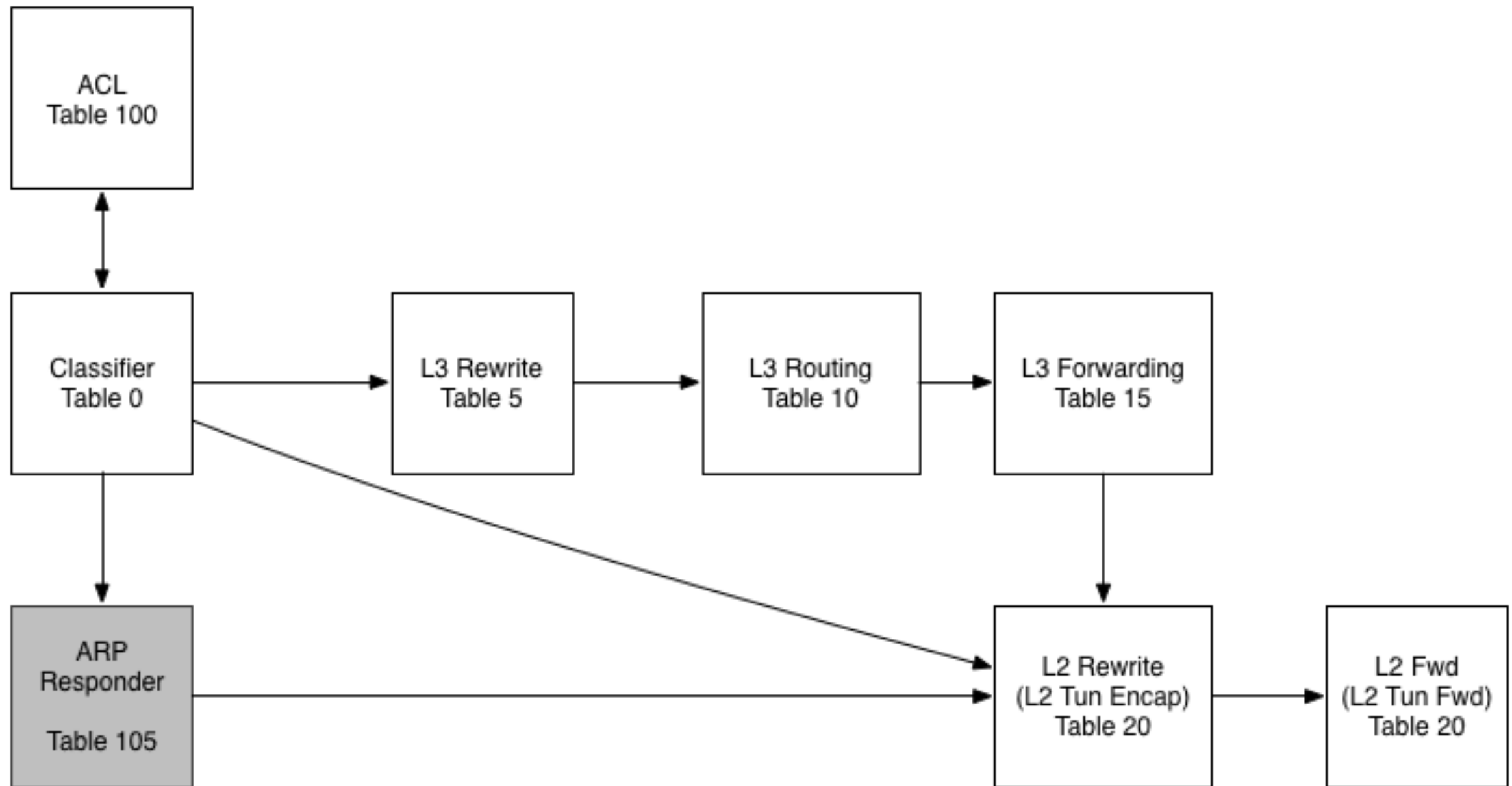
パケットやレイヤでテーブルを分ける

```
send_flow_mod_add(  
    datapath_id,  
    ...  
    actions: SendOutPort.new(1)  
)
```



```
send_flow_mod_add(  
    datapath_id,  
    ...  
    instructions: Apply.new(SendOutPort.new(1))  
)
```

実装イメージ



<http://dtucker.co.uk/hack/building-a-router-with-openvswitch.html>

simple_router (OpenFlow1.0)のコード


```
def packet_in(dpid, message)
  return unless sent_to_router?(message)
  case message.data
  when Arp::Request
    packet_in_arp_request dpid, message.in_port, message.data
  when Arp::Reply
    packet_in_arp_reply dpid, message
  when Parser::IPv4Packet
    packet_in_ipv4 dpid, message
  else
    logger.debug "Dropping unsupported packet type: #{message.data.inspect}"
  end
end
```

- テーブル0でパケットごとのテーブルに振り分け
- `packet_in_arp_request` でARPを処理

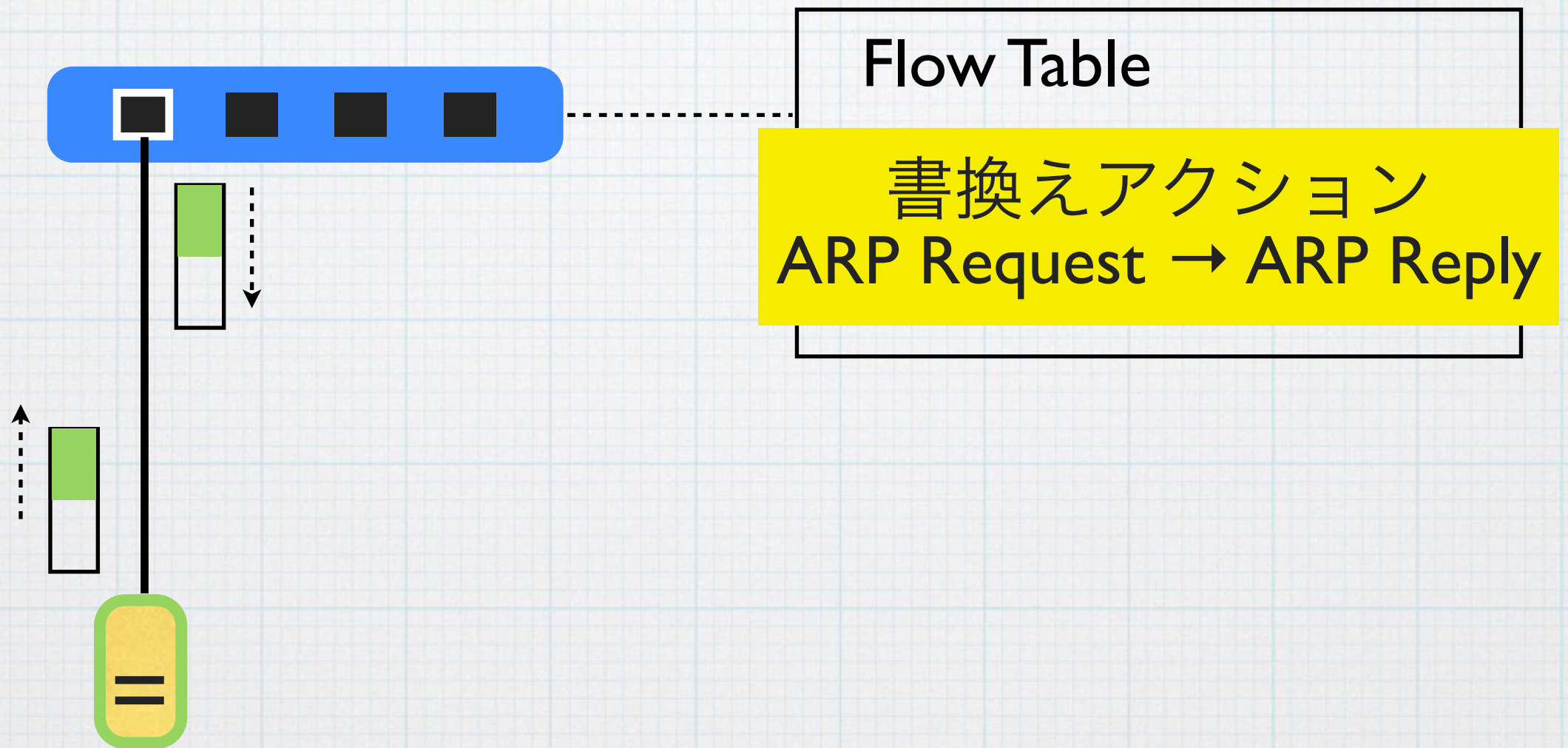
```
def packet_in_arp_request(dpid, in_port, arp_request)
  interface = @interfaces.find_by(port_number: in_port, ip_address: arp_request.target_protocol_address)
  return unless interface

  send_packet_out(
    dpid,
    raw_data: Arp::Reply.new(
      destination_mac: arp_request.source_mac,
      source_mac: interface.mac_address,
      sender_protocol_address: arp_request.target_protocol_address,
      target_protocol_address: arp_request.sender_protocol_address
    ).to_binary,
    actions: SendOutPort.new(in_port))
end
```

- ふつうはARP Replyを作って
send_packet_out
- 実はフローテーブルでできる!

ARPのフローテーブル処理

MACアドレス



ヒント:使えるアクション

- NiciraRegMove: フィールドのコピー
- Set 系アクション
 - SetArpOperation: ARP OP の書き換え
 - SetArpSenderHardwareAddress: ARP SHA
 - SetArpSenderProtocolAddress: ARP SPA
 - SetSourceMacAddress: source MAC

```
# イーサヘッダの source_mac_addressのデータを  
# → destination_mac_address にコピー  
NiciraRegMove.new(from: :source_mac_address,  
                   to: :destination_mac_address)
```

- フィールドAをフィールドBにコピー
- `NiciraRegMove.new(from: フィールドA,
 to: フィールドB)`

ARP Request の ARP OP を Reply に

SetArpOperation.new(Arp::Reply::OPERATION)

- ・ 指定したフィールドの値を書換え
- ・ Setフィールド名.new(新しい値)

レポート(再掲)

simple_routerをOpenFlow1.3にしよう

- 1.まずは単純に対応
- 2.マルチプルテーブル化

今日はARPのフローテーブル処理まで