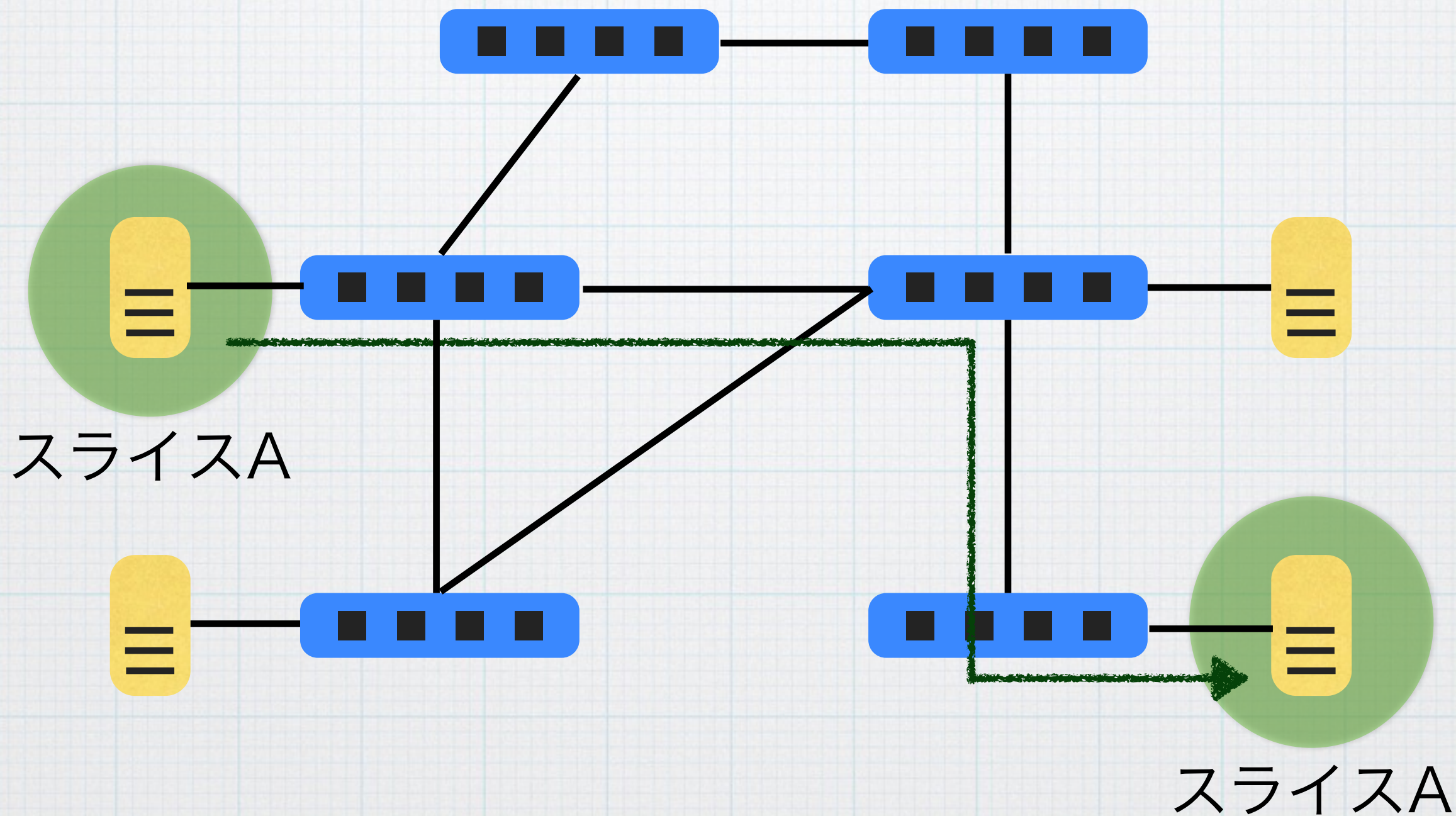


仮想ネットワーク (スライス機能)

高宮 安仁 @yasuhito

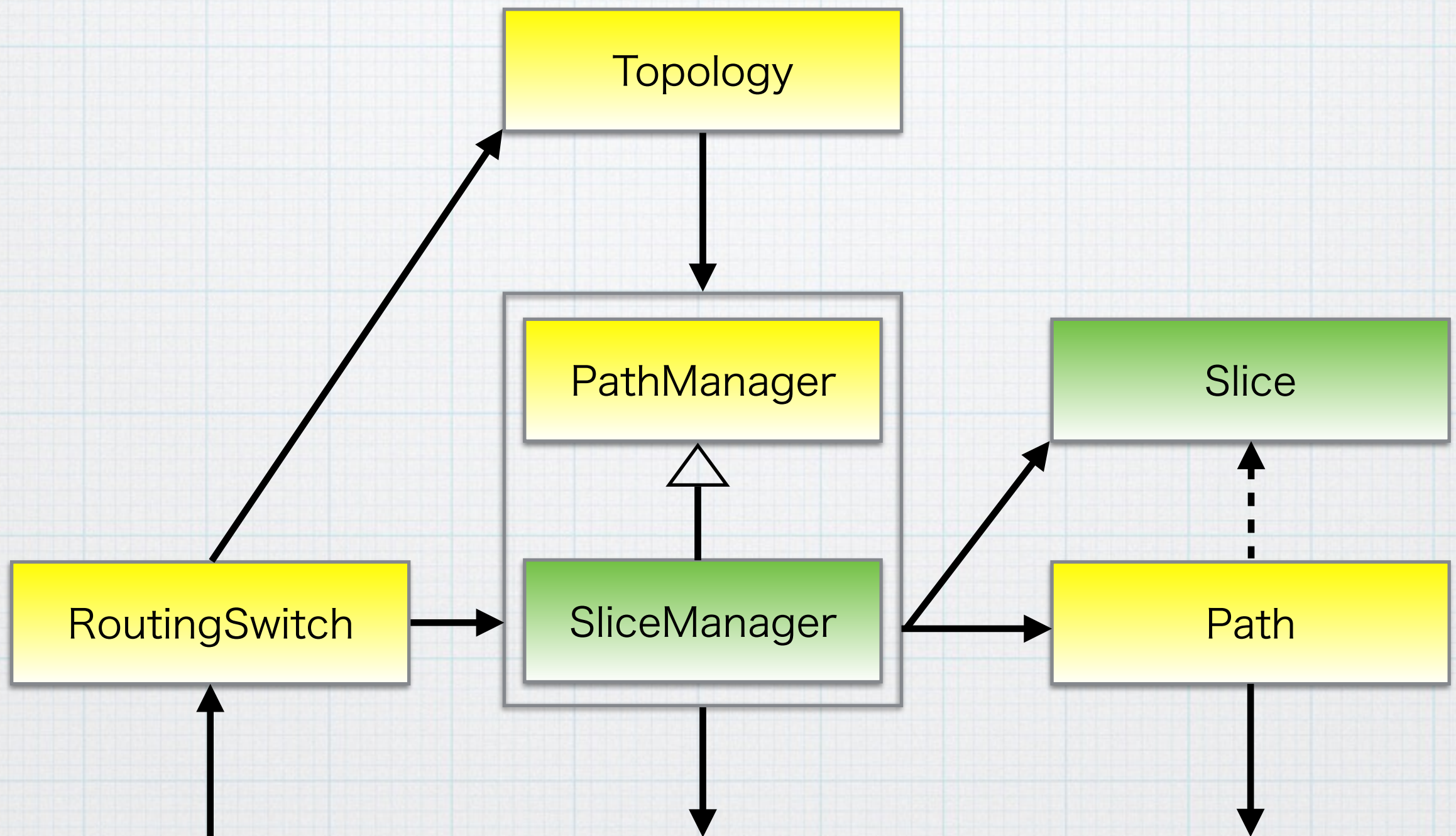
ネットワークスライス



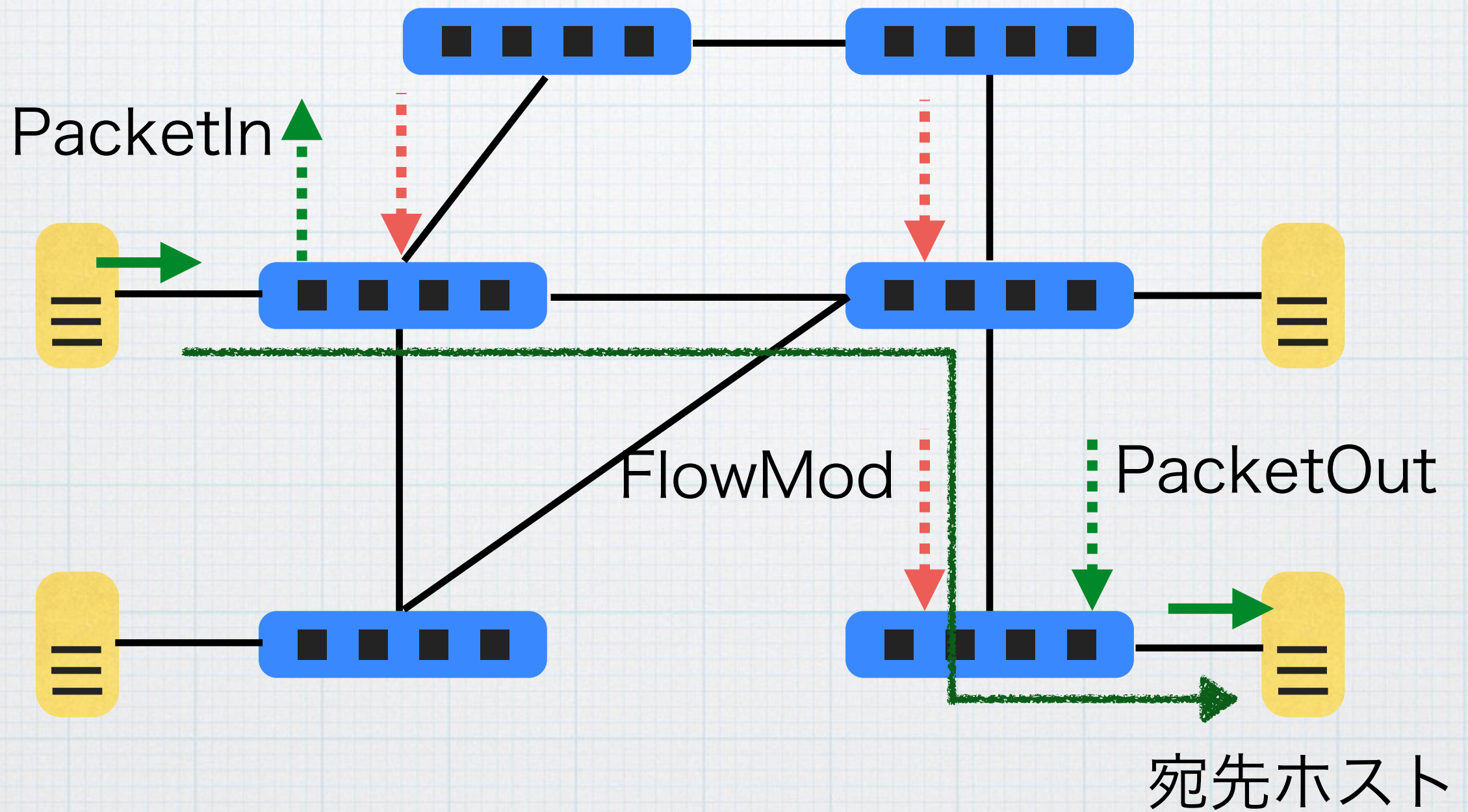
コントローラに必要な機能は？



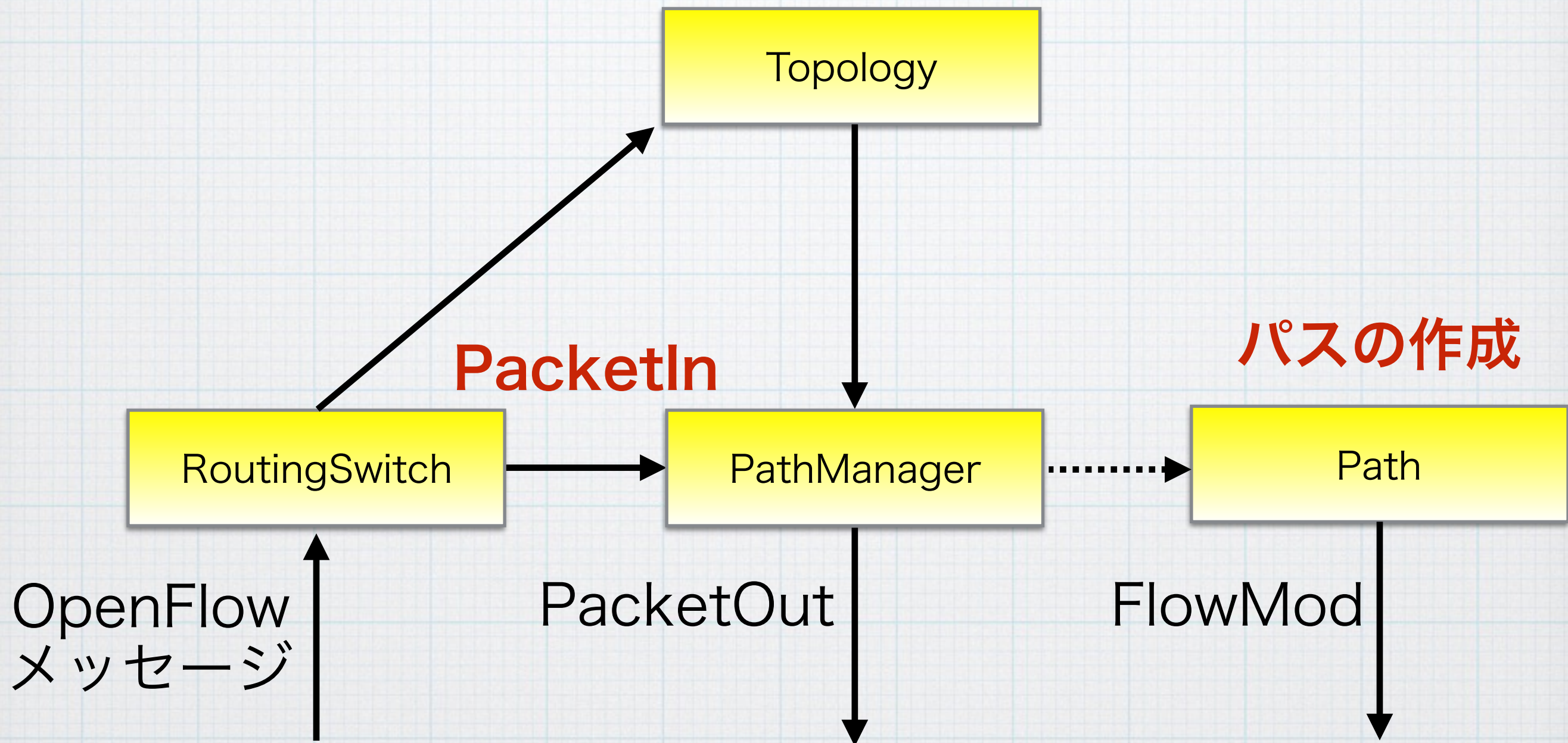
きれいにコントローラを書くと



最短路パスを作る



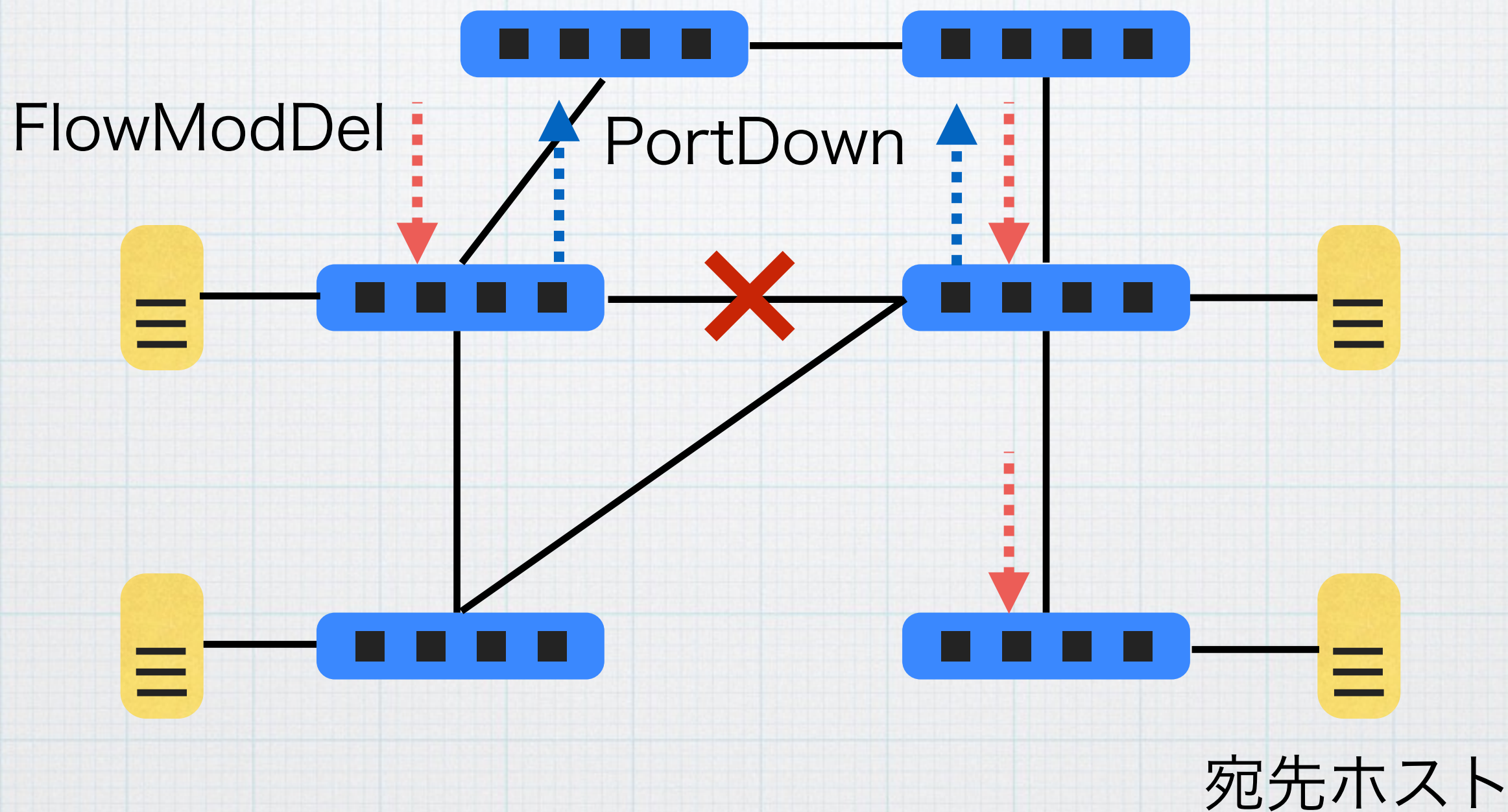
最短路パスを作る



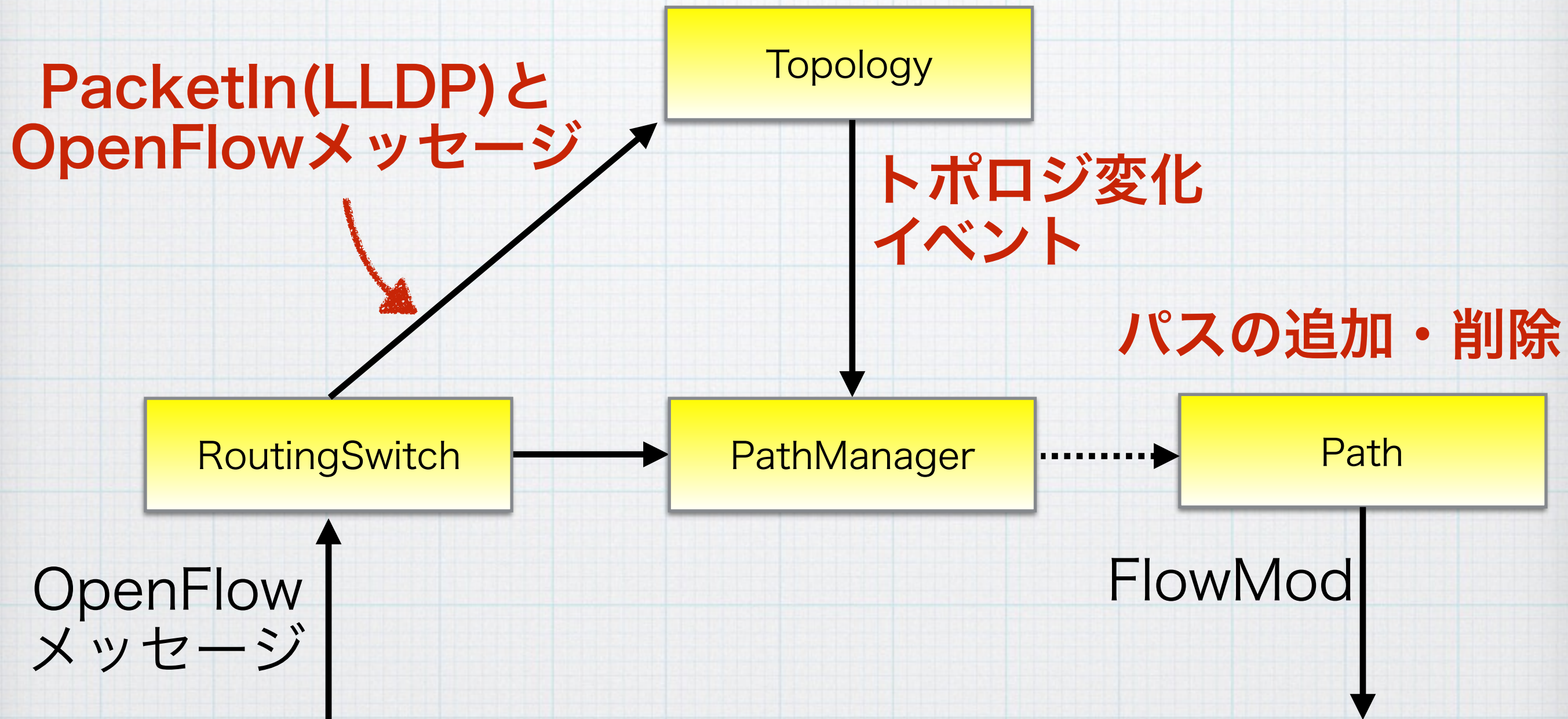
テク#1 処理のカプセル化

- `Path.create(path, packet_in)`
パスに沿って上流からFlowMod
- `Path.destroy(path)`
パスに沿って下流からFlowModDel
- `Path.all`
すでに作ったすべてのパスを返す
- `Path.find { |each| ... }`
すべてのパスから条件に合うものを検索

パスの削除



トポロジの変化を検出する



テク#2 既存クラスの再利用 & オブザーバパターン

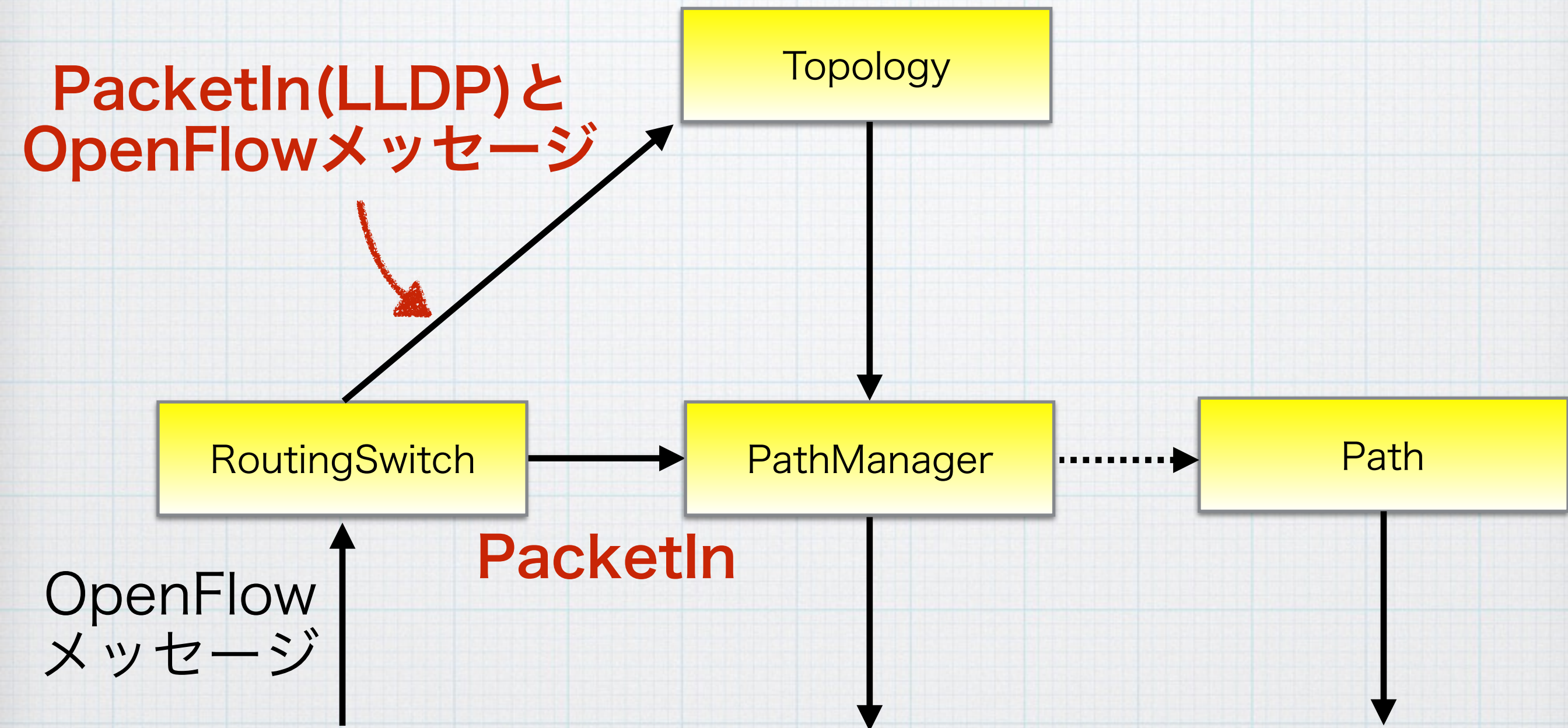
```
class RoutingSwitch < Trema::Controller
  # ...
  def start_topology
    TopologyController.new do |topology|
      topology.start
      topology.add_observer @path_manager
    end
  end
end
```

```
class PathManager < Trema::Controller
  # ...
  def delete_link(port_a, port_b, _topology)
    @graph.delete_link port_a, port_b
    Path.select { |each| each.link?(port_a, port_b) }.each(&:destroy)
  end
end
```

PathManagerで
Topologyからの
イベントを受け取る

リンク削除に
反応し、パスを消す

メッセージの転送



テク#3 ハンドラの委譲

```
# L2 routing switch
class RoutingSwitch < Trema::Controller
  extend Forwardable

  def_delegators :@topology, :switch_ready
  def_delegators :@topology, :features_reply
  def_delegators :@topology, :switch_disconnected
  def_delegators :@topology, :port_modify

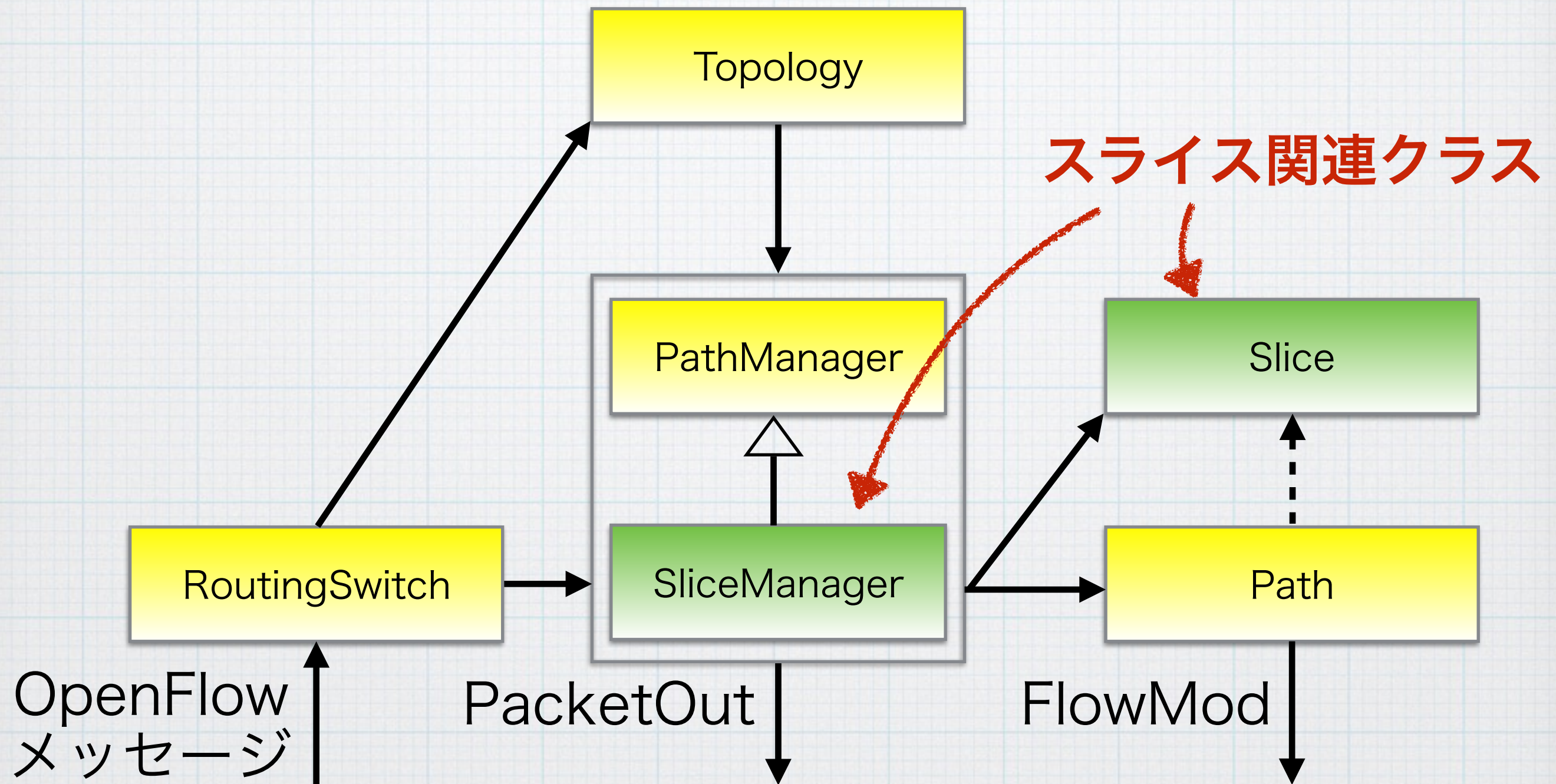
  def packet_in(dpid, message)
    @topology.packet_in(dpid, message)
    @path_manager.packet_in(dpid, message) unless message.lldp?
  end
```

Topologyへ委譲

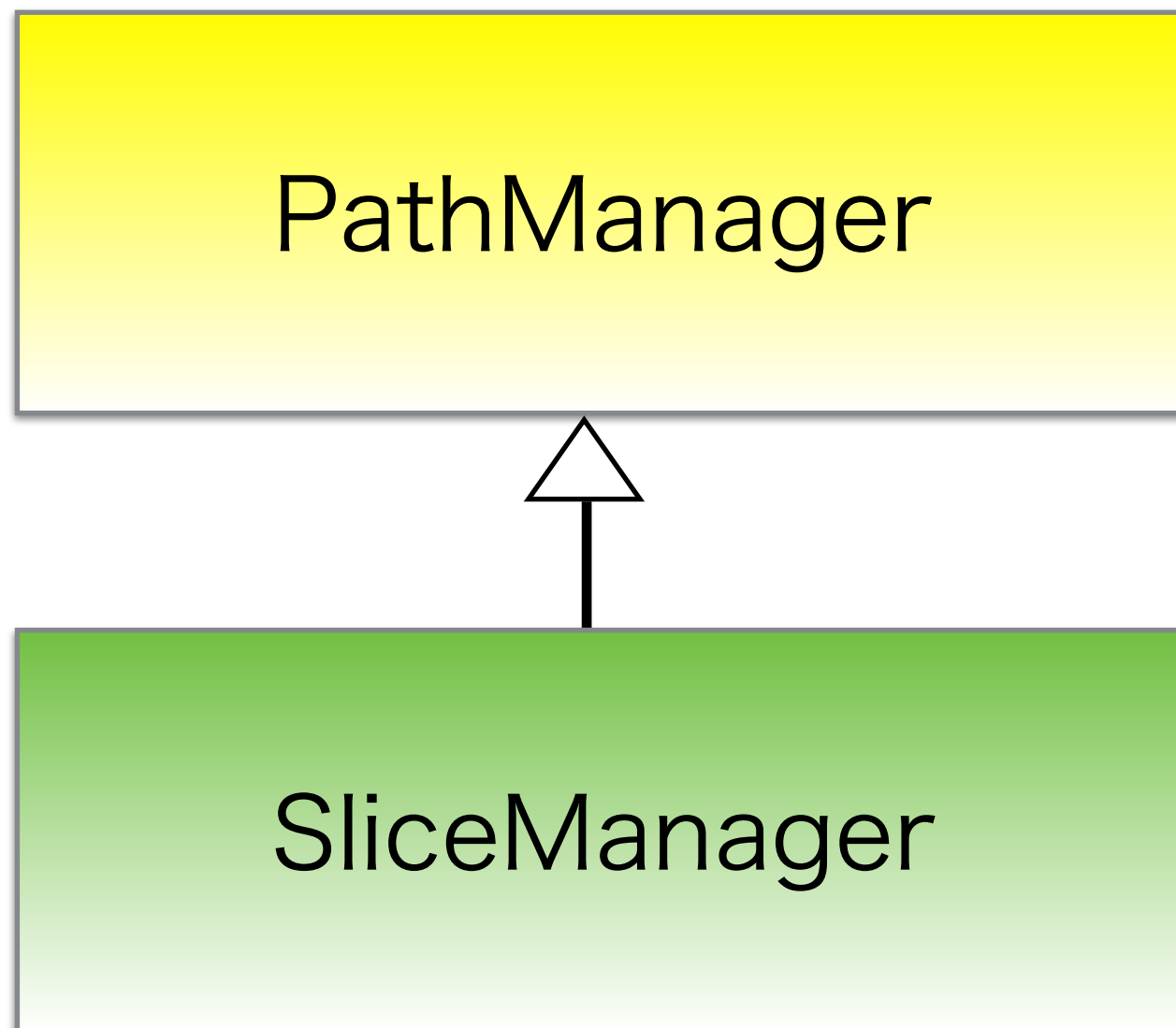


Topologyと
PathManagerへ委譲

スライス機能の追加

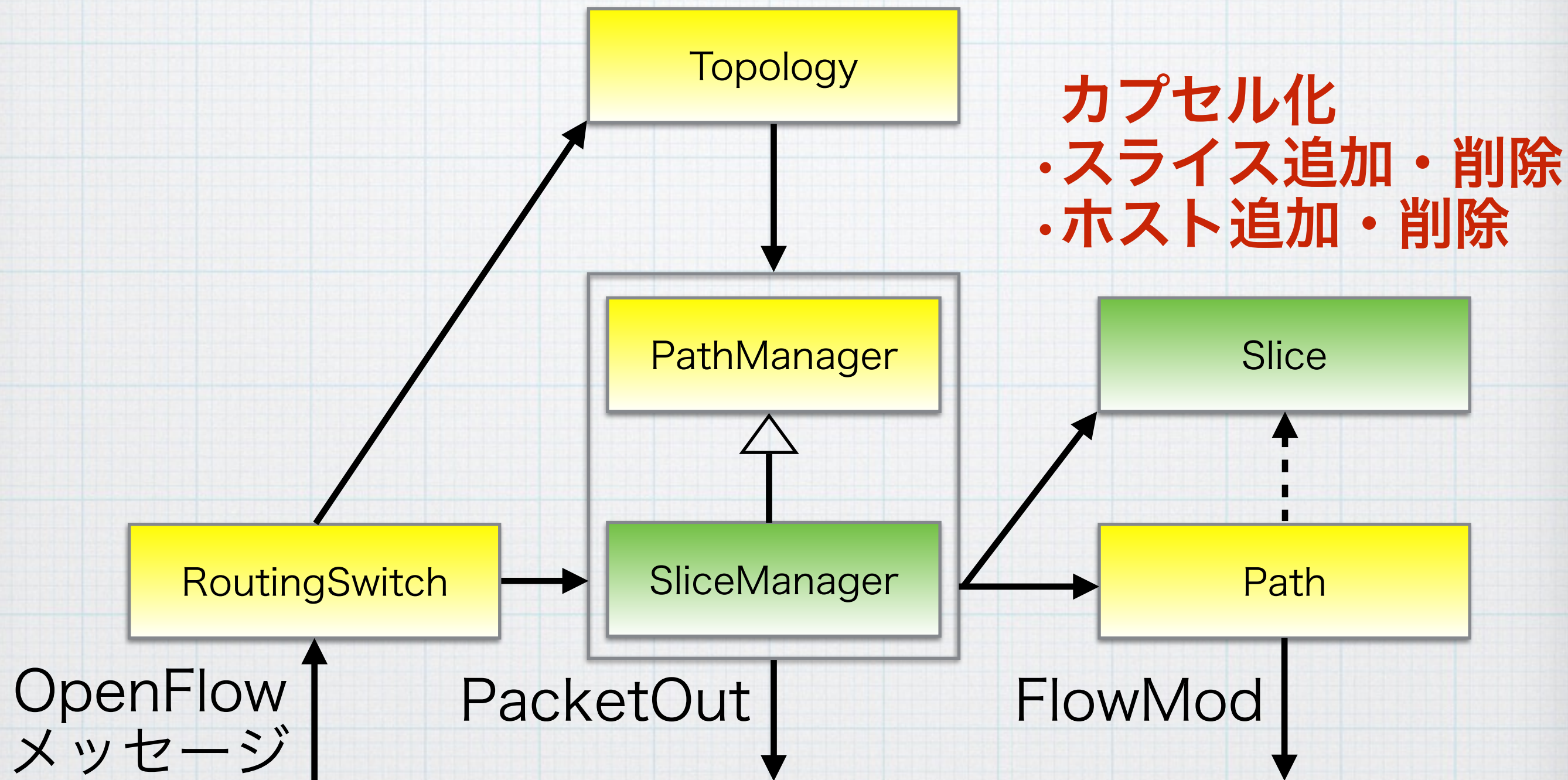


テク#4 継承とオーバーライド



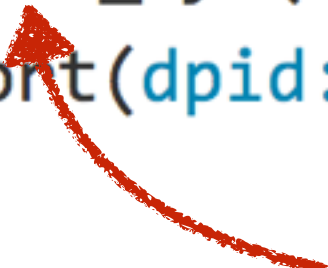
- **Topologyハンドラ**
トポロジグラフを作る
- **PacketInハンドラ**
最短路パスを作る

- **Topologyハンドラ(継承)**
- **packet_in(上書き)**
最短路パスを作る前に
スライス判定を追加



REST API の実装

```
post 'slices/:slice_id/ports' do
  rest_api do
    Slice.find_by!(name: params[:slice_id]).
      add_port(dpid: params[:dpid], port_no: params[:port_no])
  end
end
end
```

 Sliceクラスのメソッドを呼ぶだけ!

使える実装テクあれこれ

ハンドラの委譲・オブザーバ

- ・ コントローラを機能分割して連携
- ・ 既存部品（トポロジ）の再利用

クラス継承・カプセル化

- ・ スライス、REST API などの機能追加

課題 スライス機能の拡張 (2週間)

スライスの分割・結合

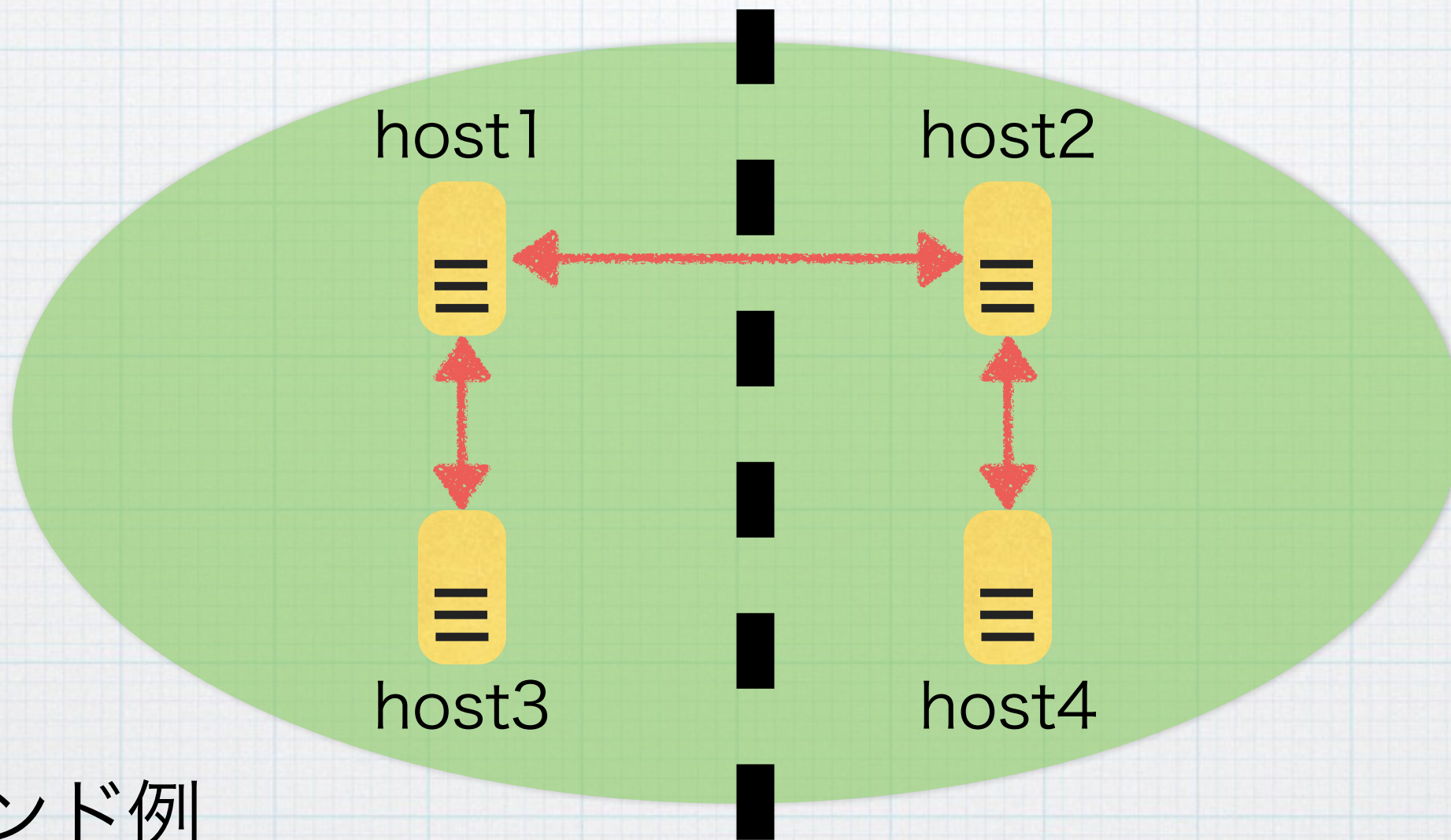
- ・ スライスの分割と結合機能を追加する

スライスの可視化

- ・ Graphvizでスライスの状態を表示

以上を実機スイッチで動かしてみよう

スライス分割コマンド



コマンド例

```
$ slice split slice_a
```

```
—into slice_b:host1,host3 slice_c:host2,host4
```