

ネットワークトポロジを 検出しよう

高宮 安仁 @yasuhito

Hello
World

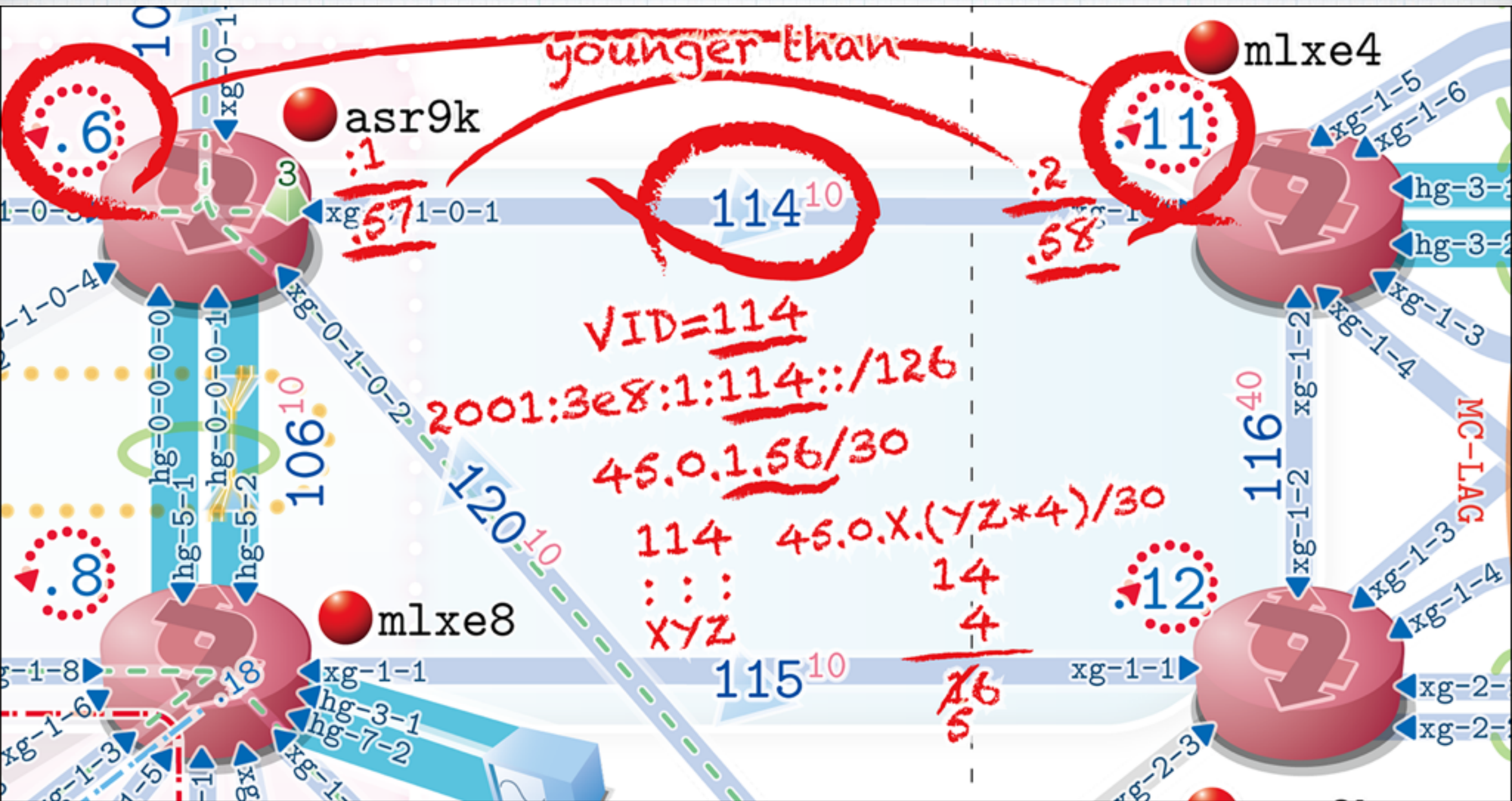
スイッチ
ルータ

トポロジ
ディスカバリ

ルーティング
スイッチ

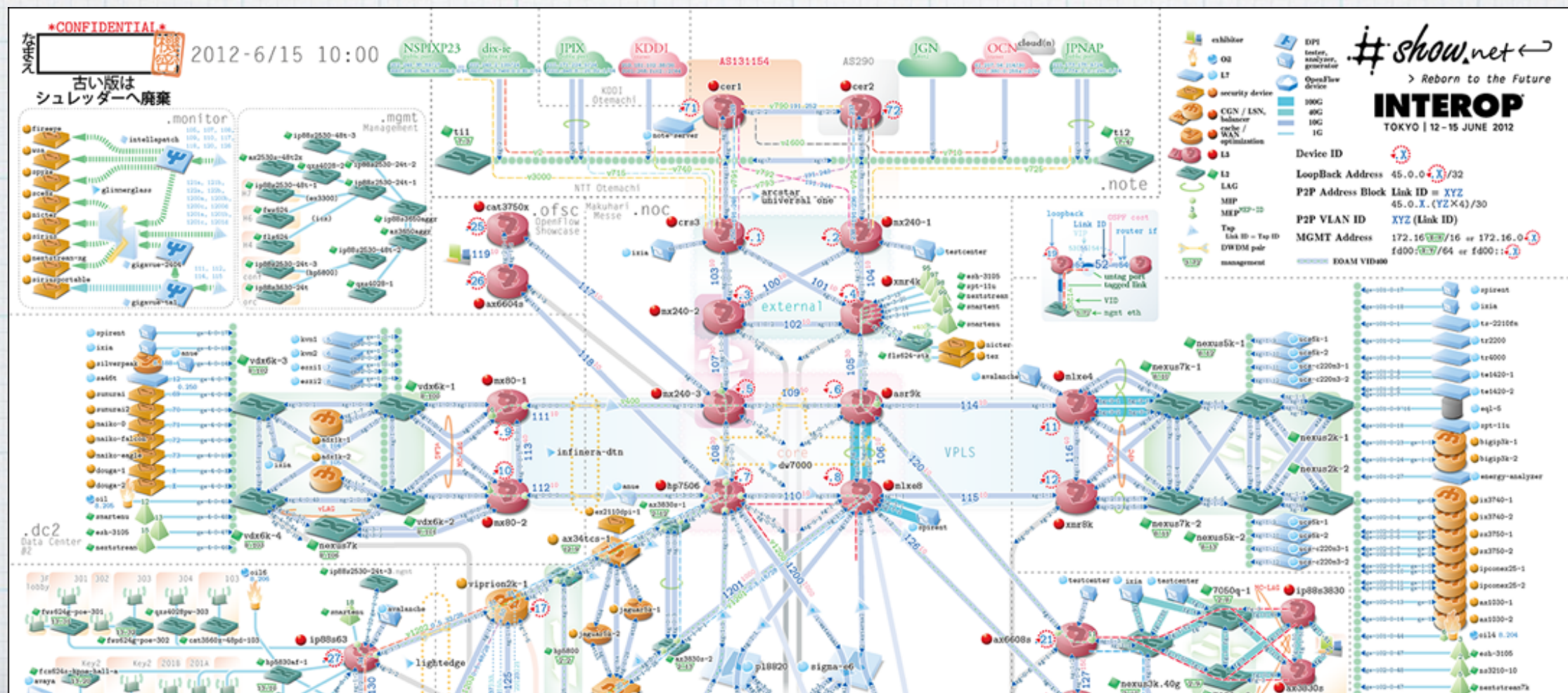
仮想NW





トポロジ

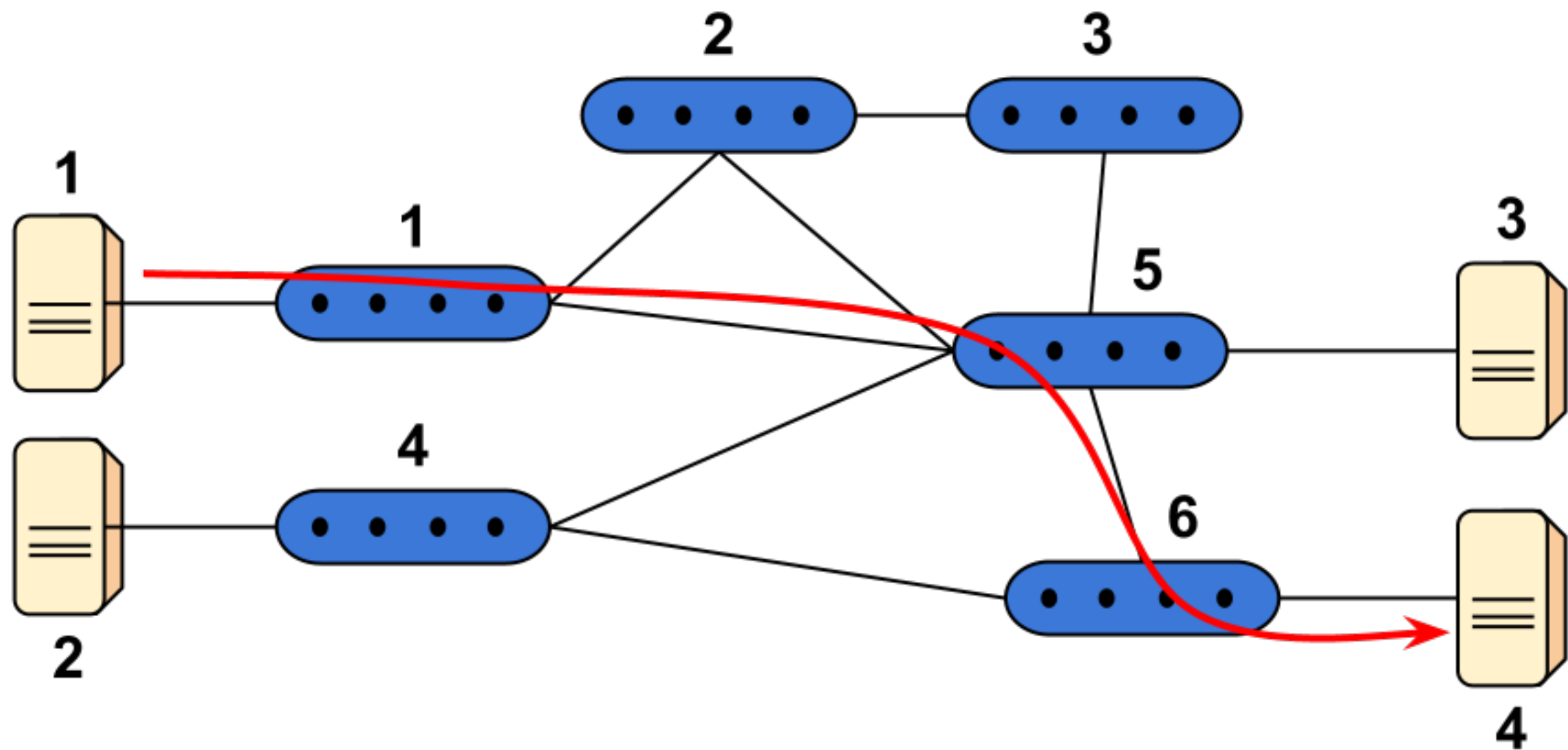
- 初期構成は静的に与えることができる
 - しかし、障害や構成変更で動的に変わる
- トポロジの変化を検知できるという便利

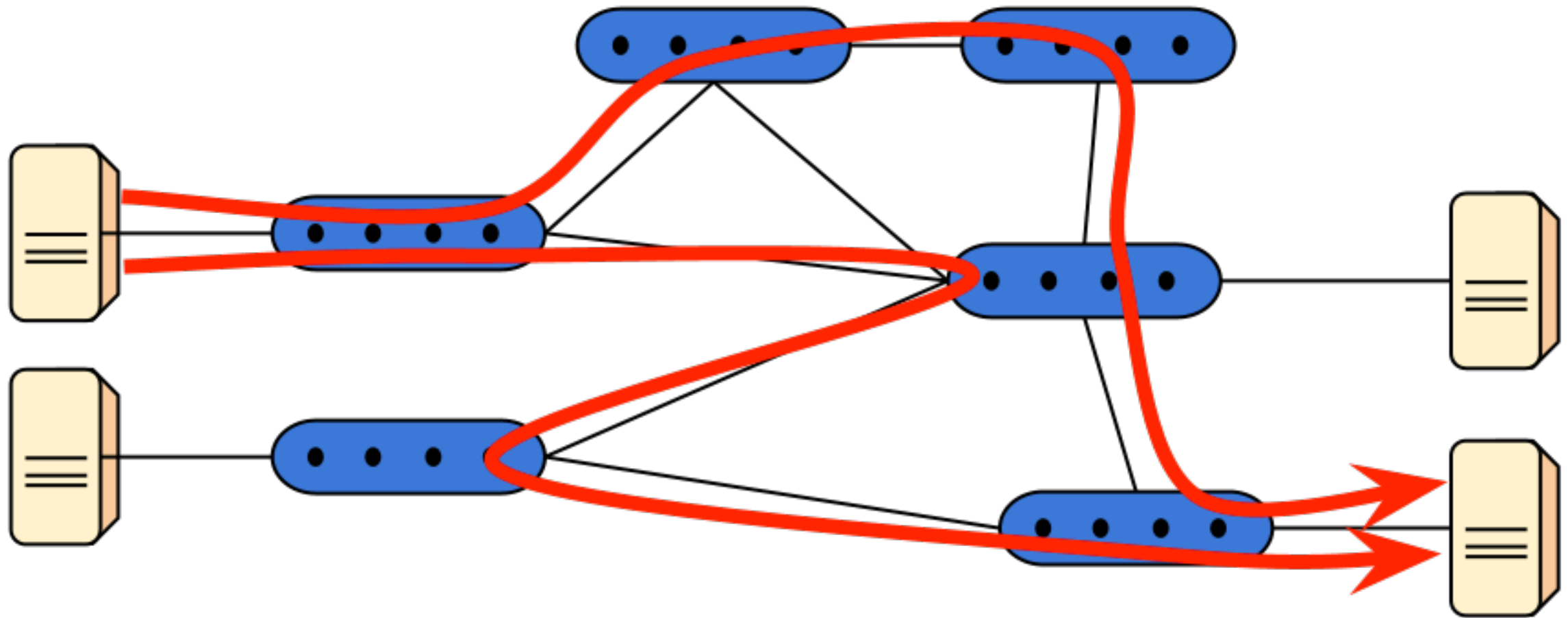


ルーティングスイッチ



フローの書き込み





- マルチパスで帯域をかせぐ/有効利用
- 最短経路以外でもルーティング可能

トポロジの検知

Link Layer Discovery Protocol (IEEE 802.1AB)

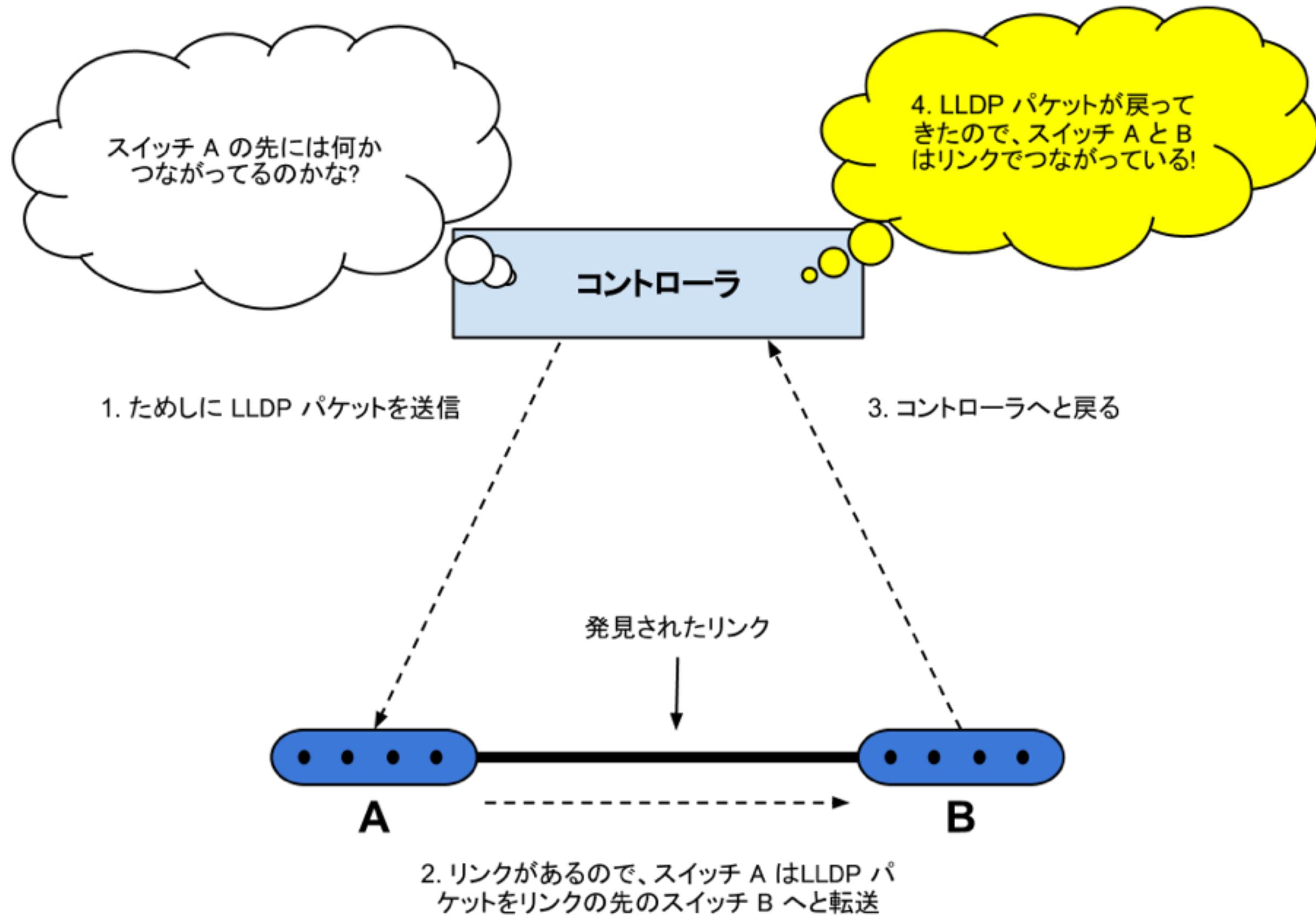
データリンク層の接続を検出するための

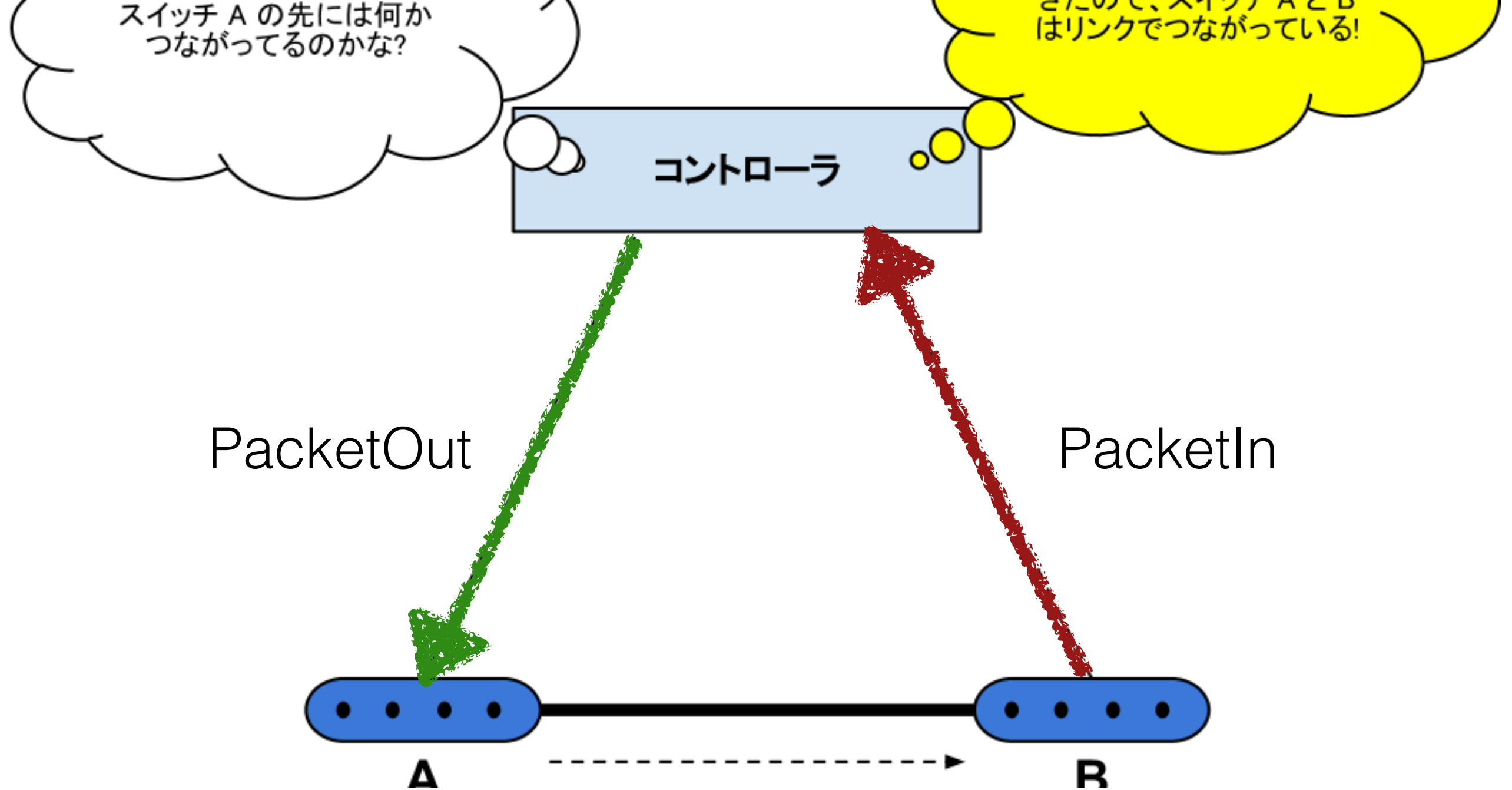
一般的なプロトコル

dpid



Destination MAC	Source MAC	Ethertype	Chassis ID TLV	Port ID TLV	Time to live TLV	Optional TLVs	End of LLDPDU TLV
01:80:c2:00:00:0e, or 01:80:c2:00:00:03, or 01:80:c2:00:00:00	Station's address	0x88CC	Type=1	Type=2	Type=3	Zero or more complete TLVs	Type=0, Length=0

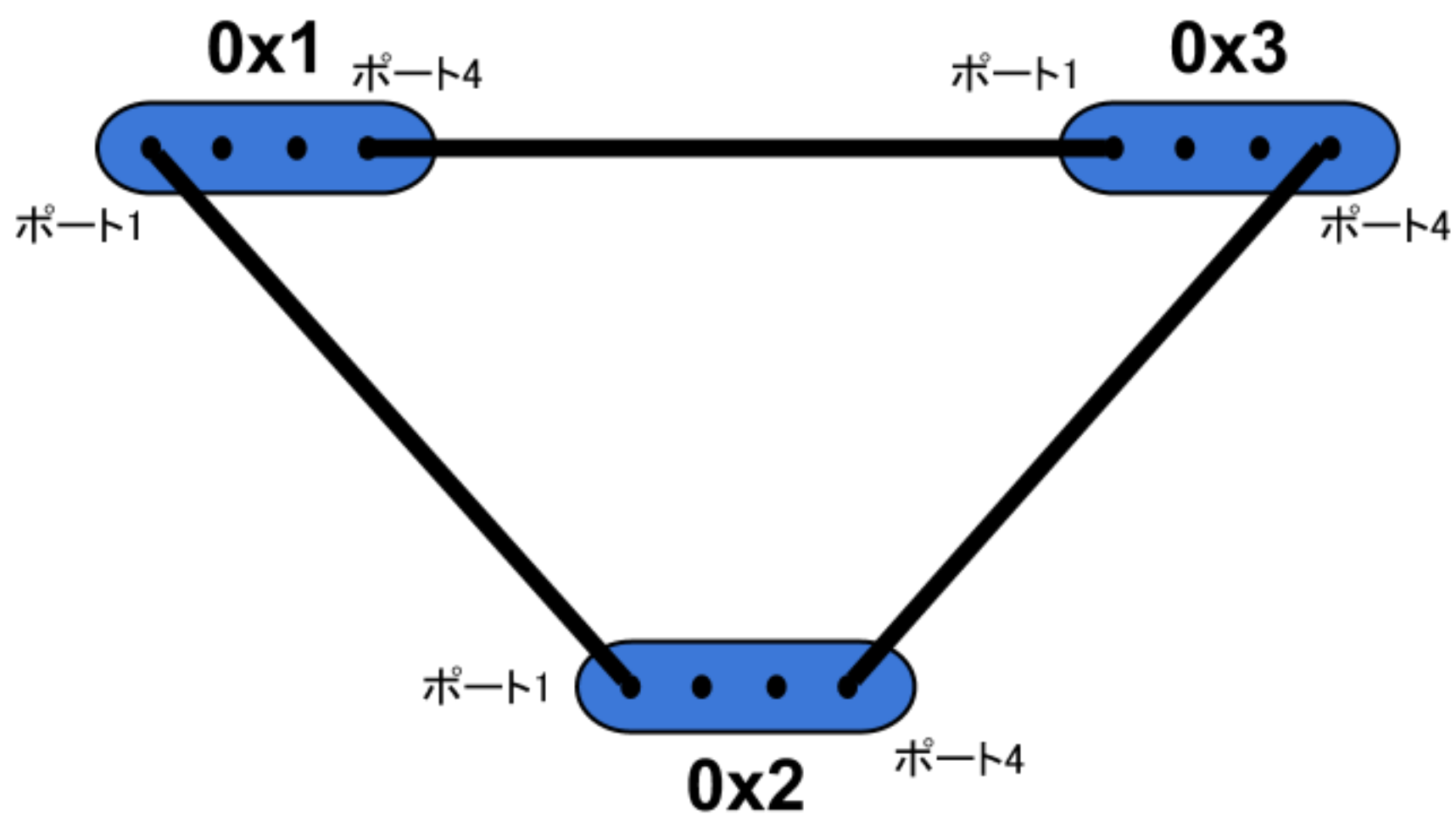
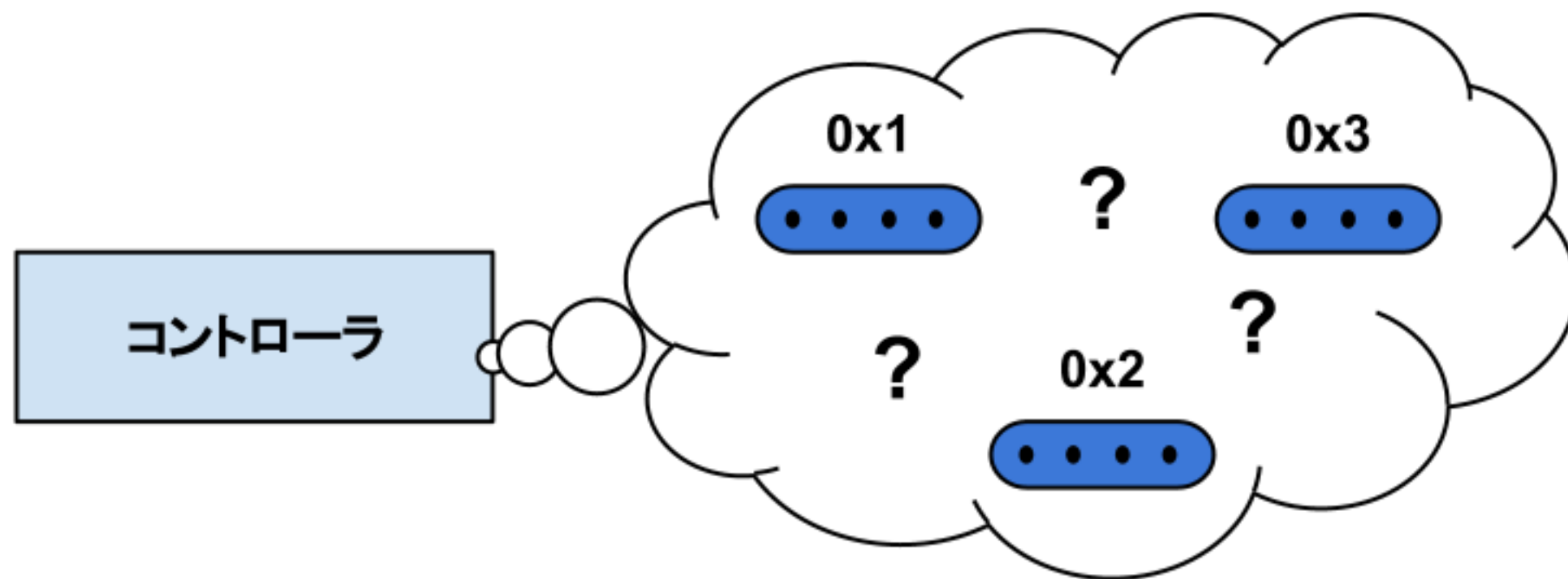


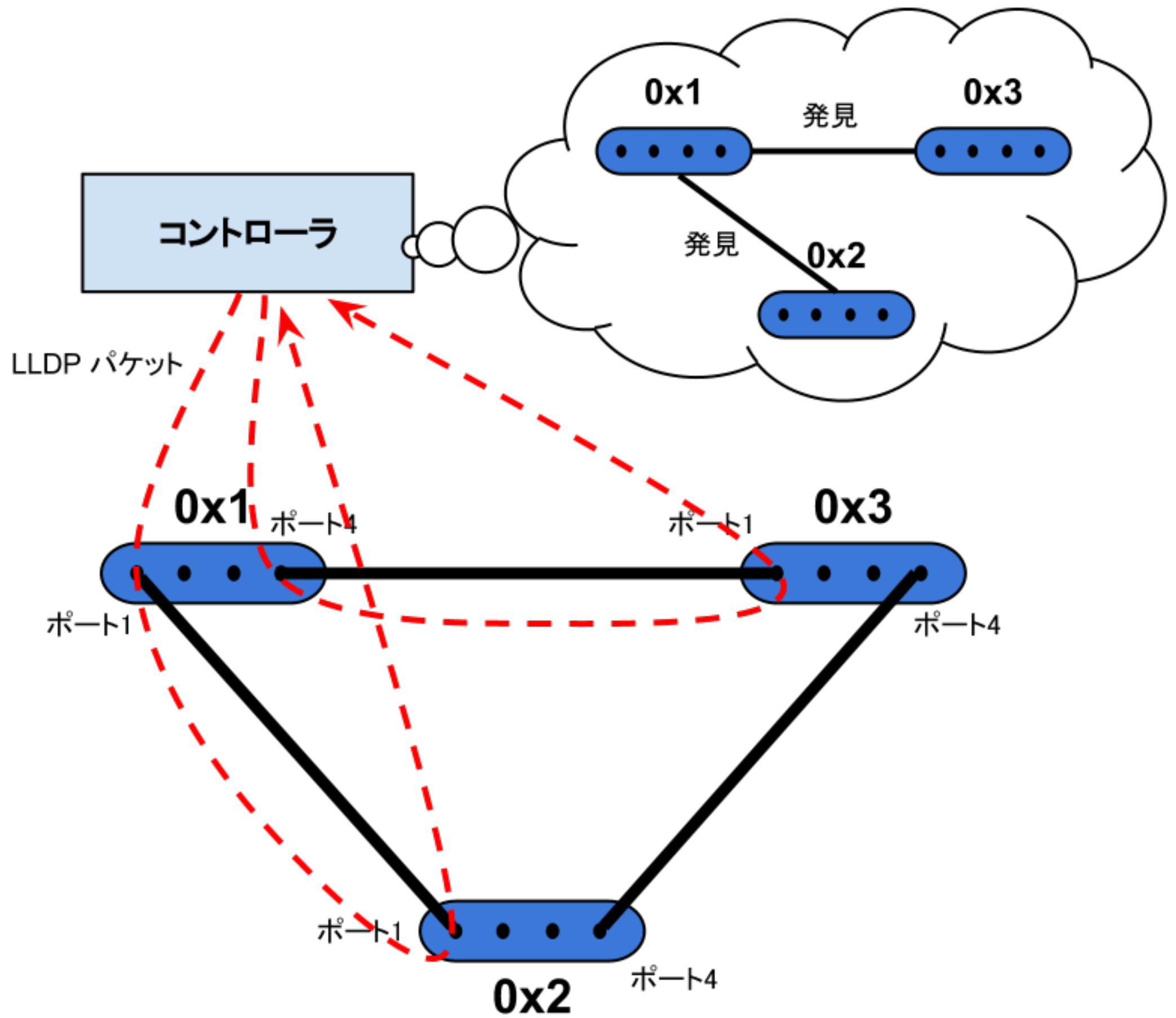


Packet In メッセージ
受信ポート = 1

LLDP パケット
スイッチ = 0x1, 送信ポート = 4

スイッチ A のポート 4 ⇔ スイッチ B のポート 1

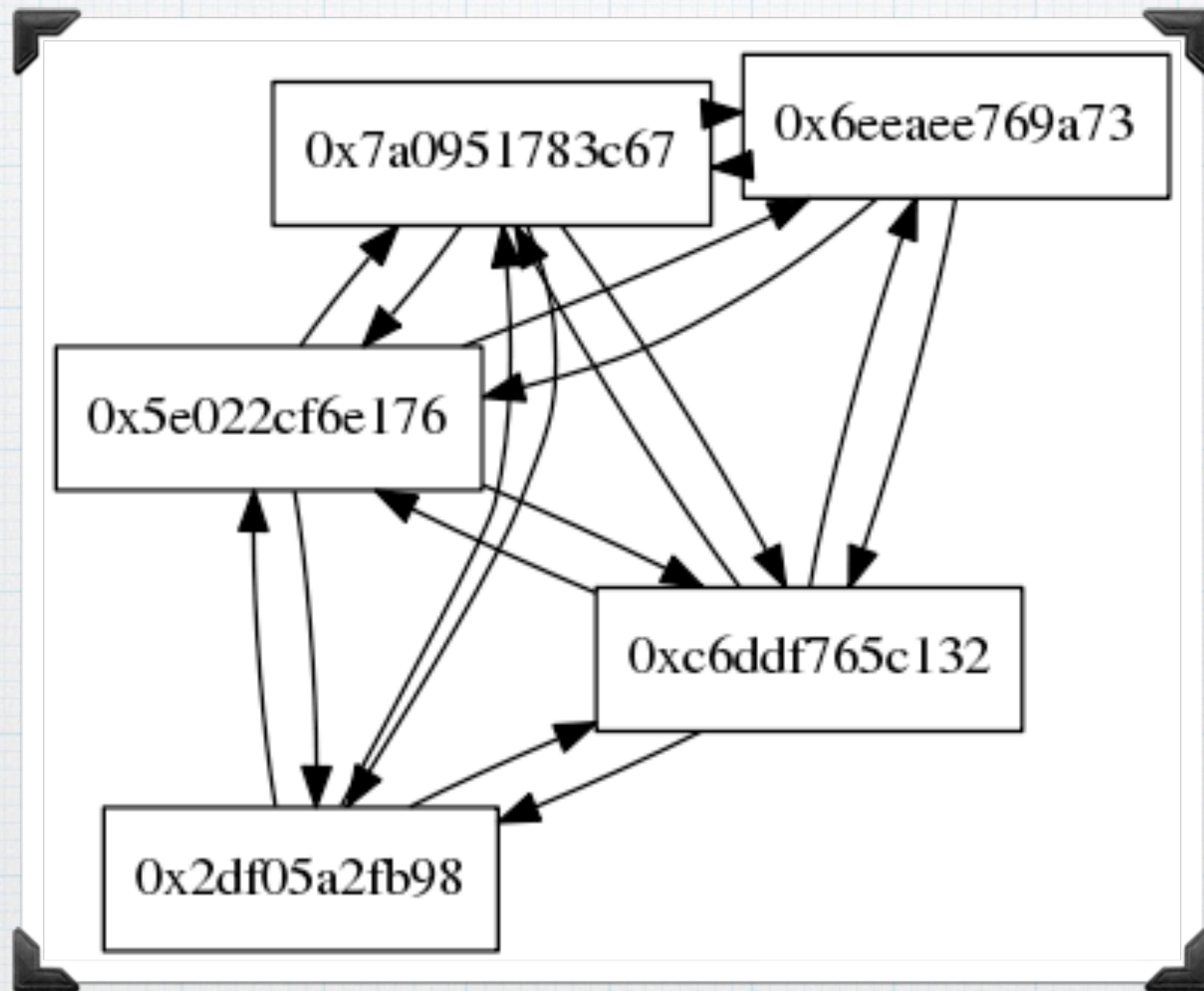




トポロジ実装例

- topology コントローラ
- トポロジの検知と表示 (テキスト/画像)を
Ruby コード約 300 行で実装
- 今回の課題は、このコードを元に改造すること

実行結果

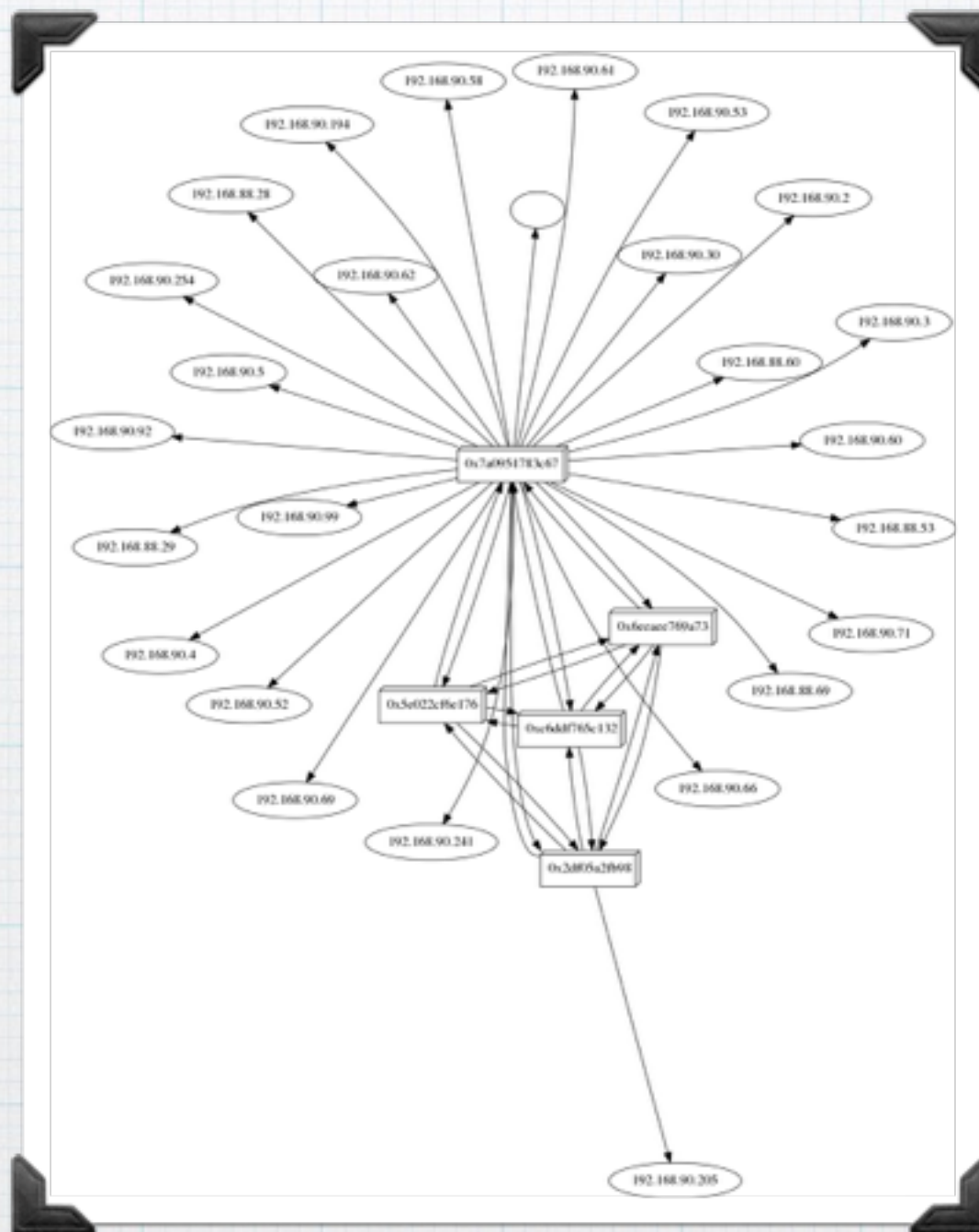


レポート課題

- topology を改造し、スイッチに加えてホストの接続関係を表示するコントローラを作る
- パケットを出したホストの IP アドレスを楕円で表示

※ パケットを出すには `trema send_packet`

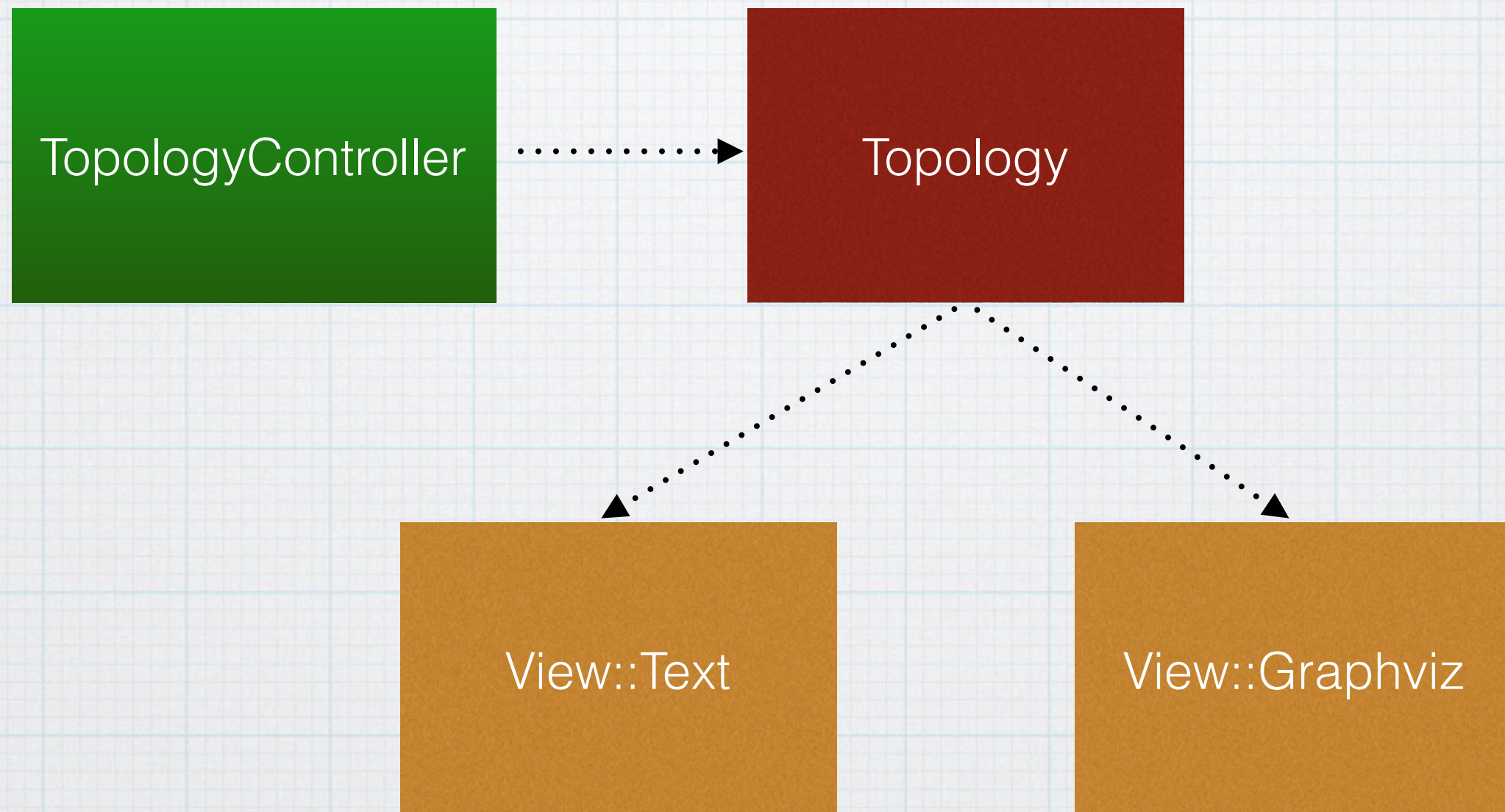
お手本: タンポポ



今日やること

1. topology をインストール
2. スイッチ 10 台程度の仮想ネットワークを作る
3. 実際にトポロジ画像が表示されることを確認
4. topology のコードを理解

topology の構造



コントローラ本体

トポロジ情報の
管理

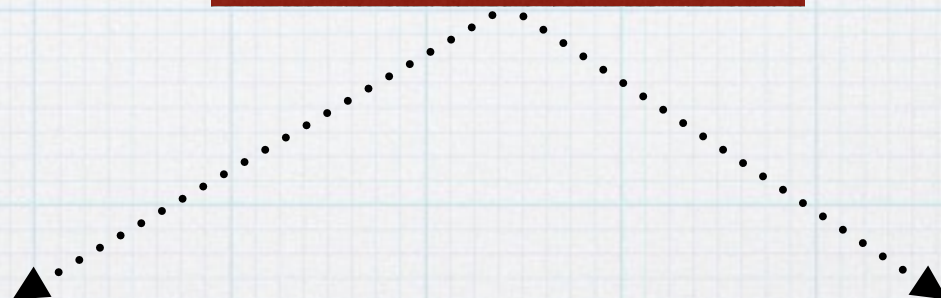
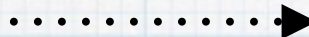
TopologyController

Topology

表示
(テキスト/画像)

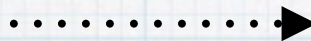
View::Text

View::Graphviz



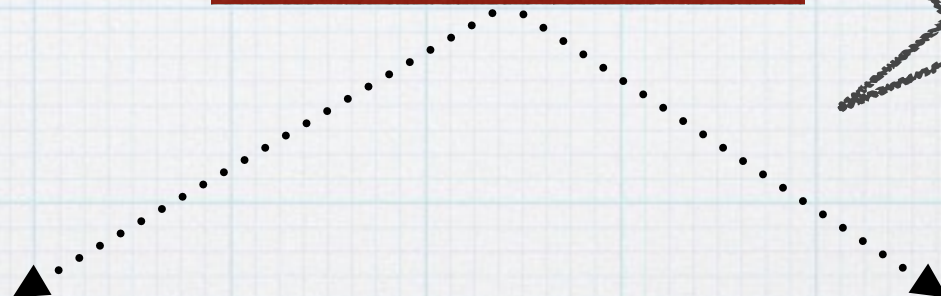
TopologyController

LLDP の PacketIn と
ポート状態の変化



Topology

トポロジの変化
を通知

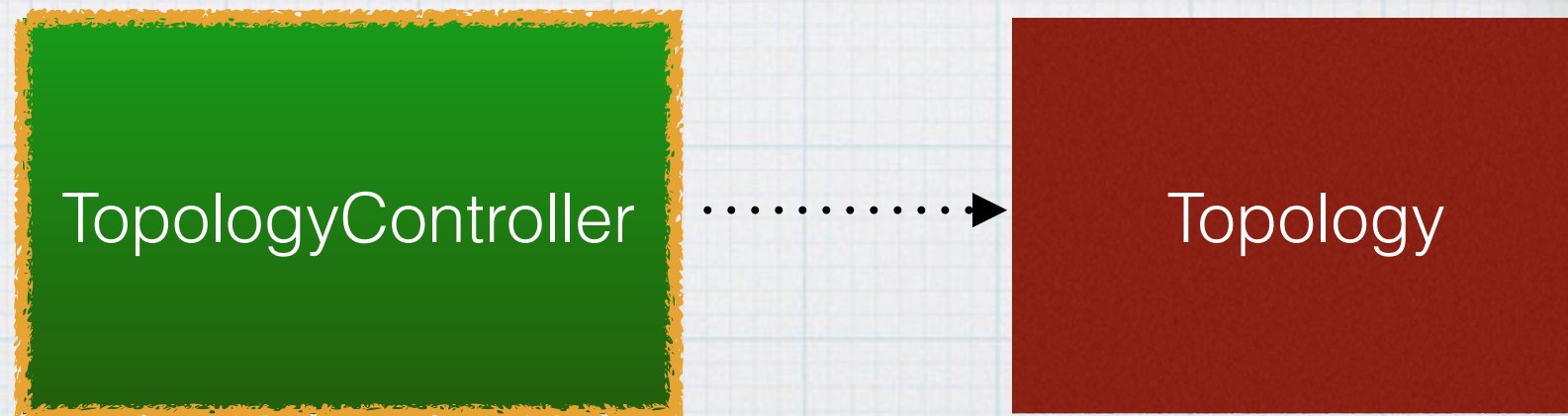


View::Text

View::Graphviz

TopologyController

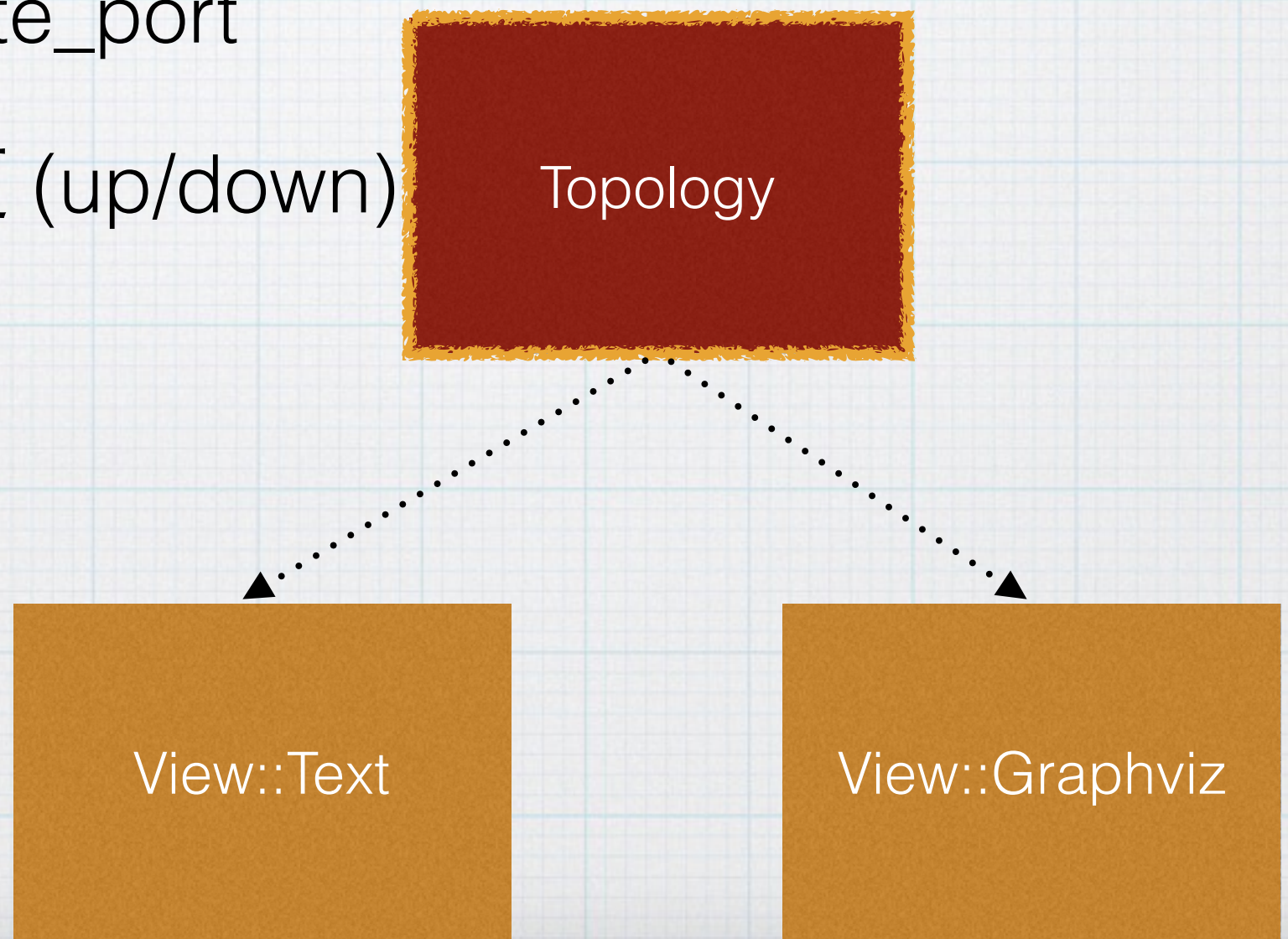
- スイッチのポート一覧を取得
FeaturesRequest, FeaturesReply
- ポート情報の更新
#port_status
- LLDP を定期的にばらまく
#flood_lldp_frames
- リンク情報の更新
#packet_in



Topology

Observer Pattern で変更を View に通知

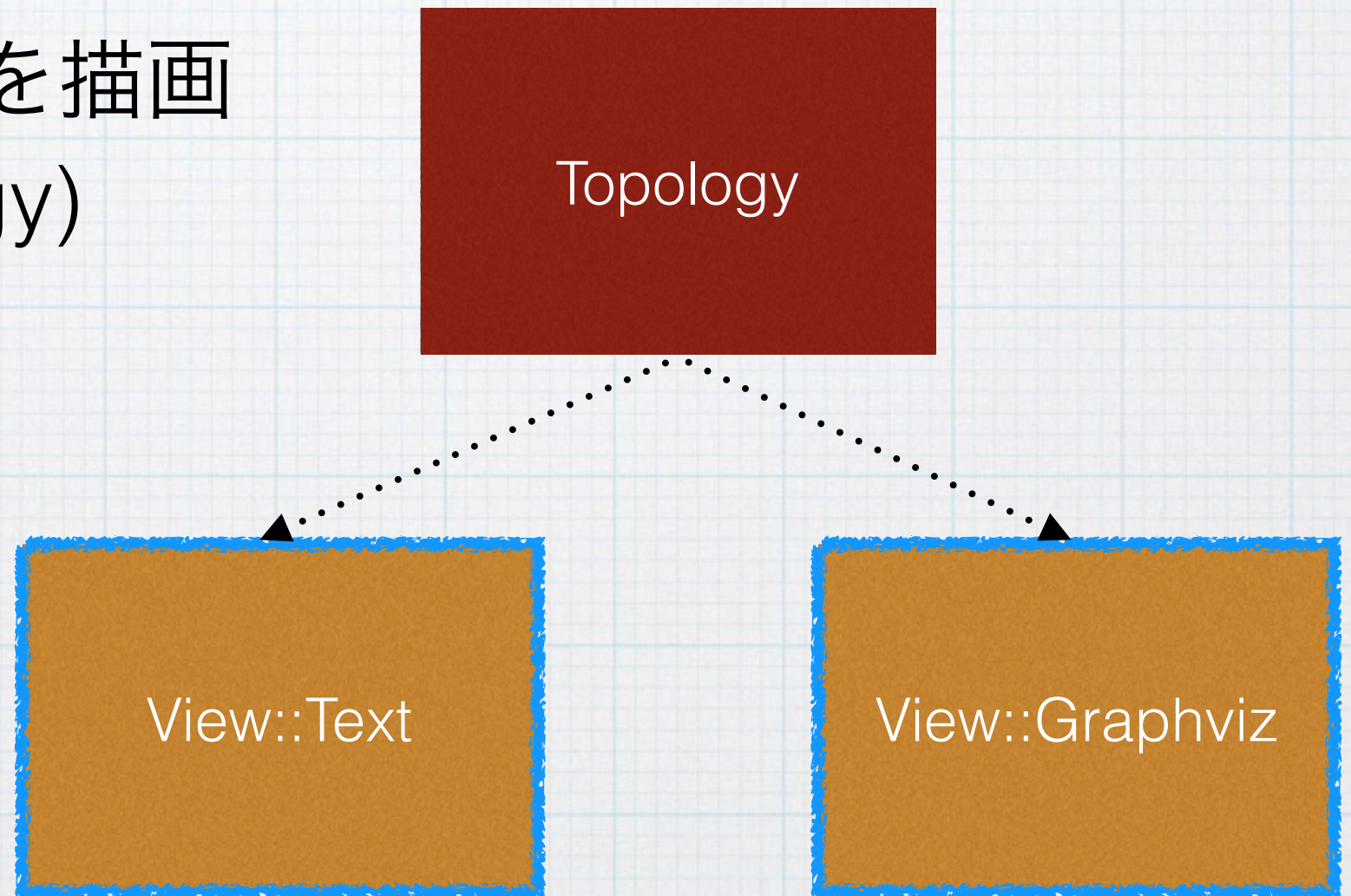
- ポートの追加と削除
#add_port, #delete_port
- ポート状態の変更 (up/down)
#update_port
- リンクの追加
#add_link_by



View::*

Observer Pattern で Topology の変化を監視

- トポロジの変化を描画
#update(topology)



LLDPのパーズと生成

ロード

```
require "pio"
```

LLDPのパーズ

```
Pio::Lldp.read(packet_in.data).dpid #=> 12345
```

生成

```
lldp = Pio::Lldp.new(:destination_mac => ...,  
                     :source_mac => ...)
```

送信

```
send_packet_out(dpid,  
                 :data => lldp.to_binary,  
                 ...)
```


今日やること(再掲)

1. topology をインストール
2. スイッチ 10 台程度の仮想ネットワークを作る
3. 実際にトポロジ画像が表示されることを確認
4. topology のコードを理解

レポート課題 (再掲)

- topology を改造し、スイッチに加えてホストの接続関係を表示するコントローラを作る
- パケットを出したホストの IP アドレスを楕円で表示