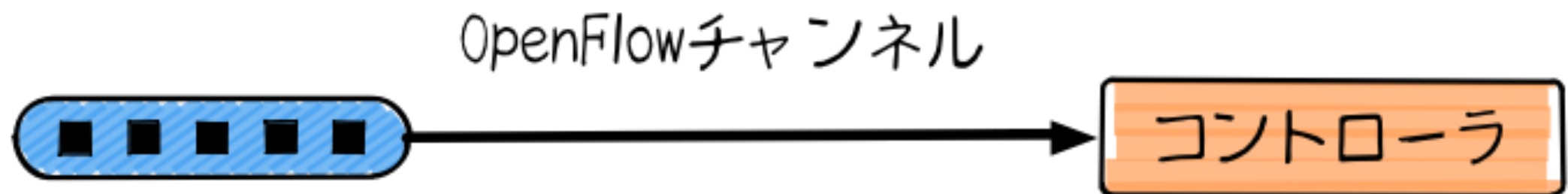


OpenFlow 1.3 プロトコル入門

高宮安仁 @yasuhito

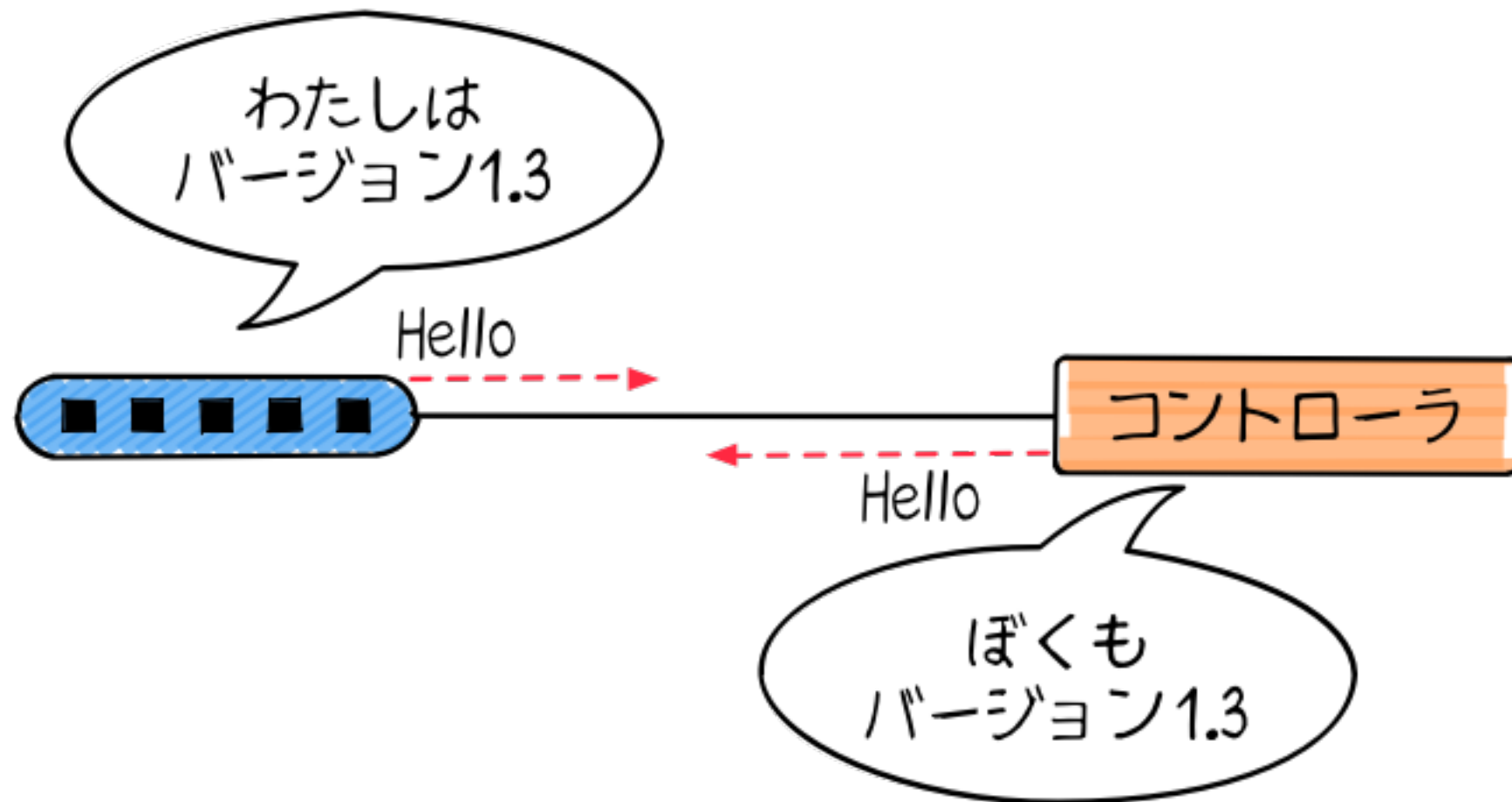
OpenFlow1.0と1.3の 共通仕様

スイッチ・コントローラ間の接続



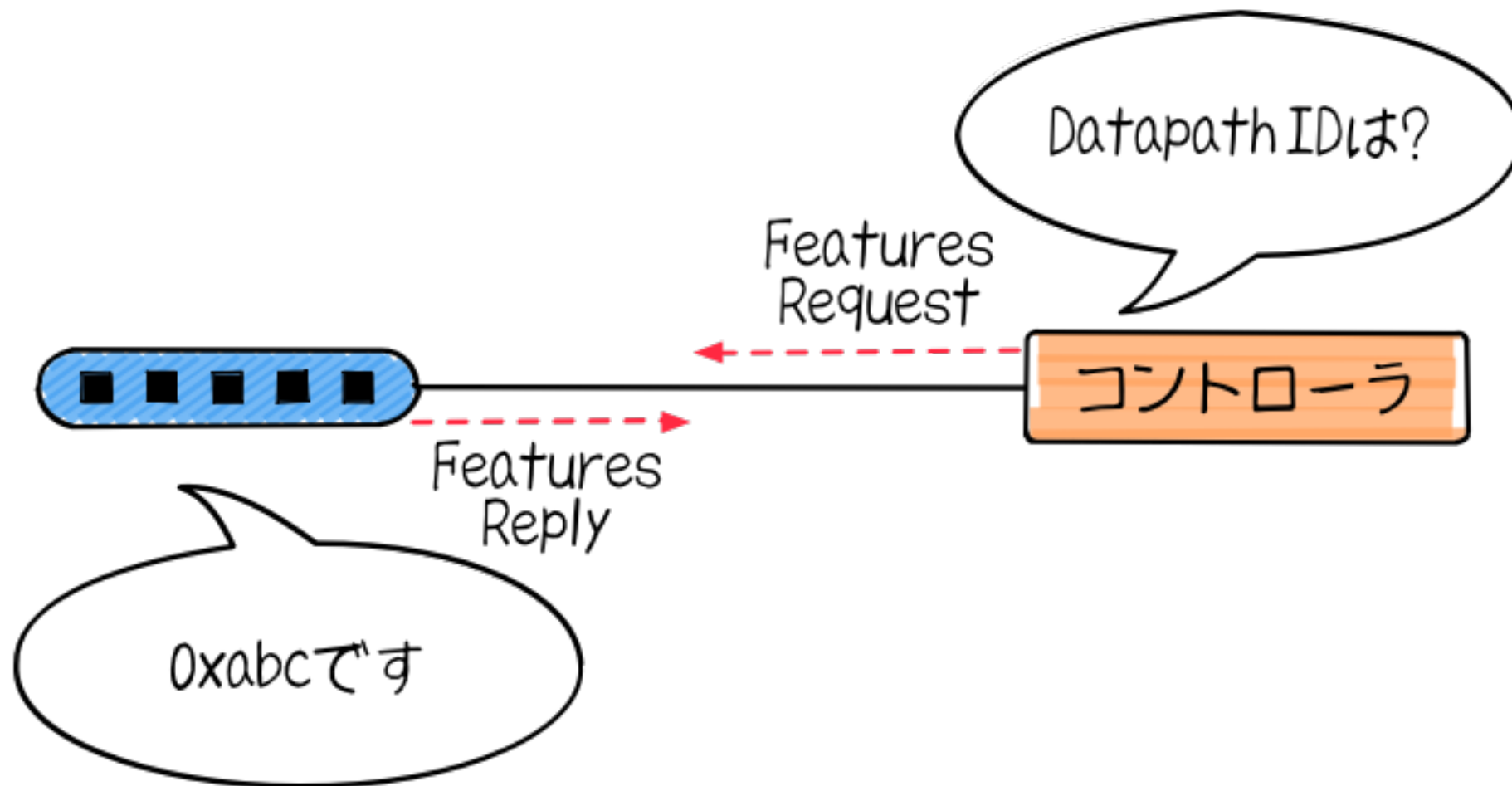
- ・ どちらから接続してもよい
- ・ ふつうはスイッチが接続しに行く

バージョンネゴシエーション

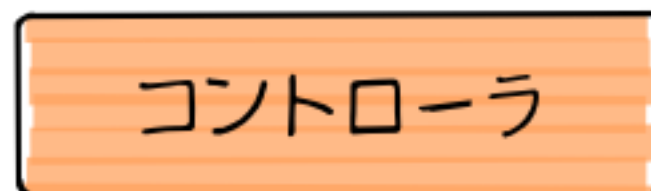


- Helloに自分のバージョンを入れて送る
- お互いに共通バージョンがあったら成功

DatapathIDの取得



- スイッチのスペックとdpidを取得
- これでようやく switch_readyが起こる



TCPで接続

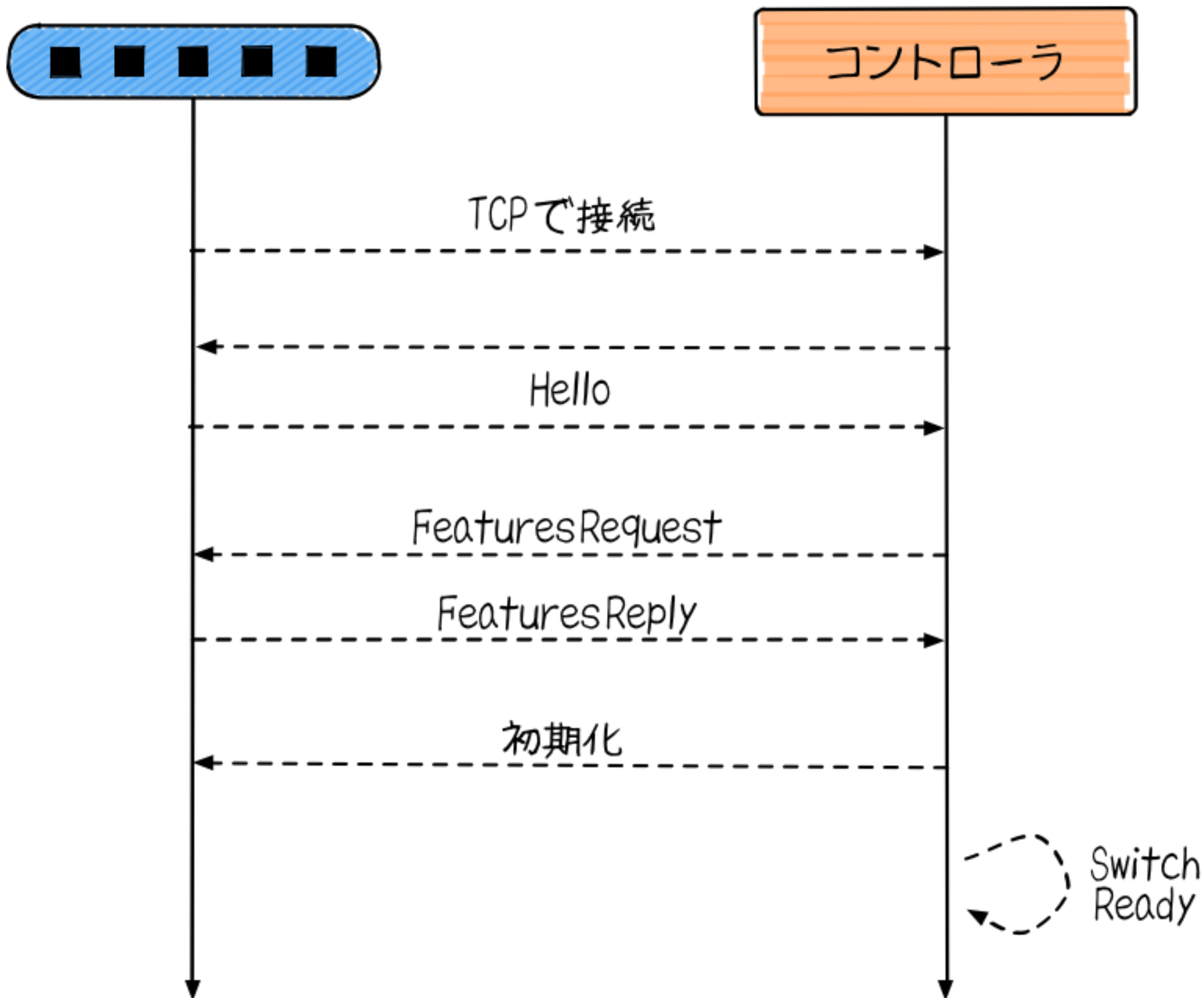
Hello

FeaturesRequest

FeaturesReply

初期化

Switch
Ready



OpenFlow1.3の利点

OpenFlow 1.0の問題点 I

Flow Table

dropルール1

dropルール2

dropルール3

..

書き換えルール1

書き換えルール2

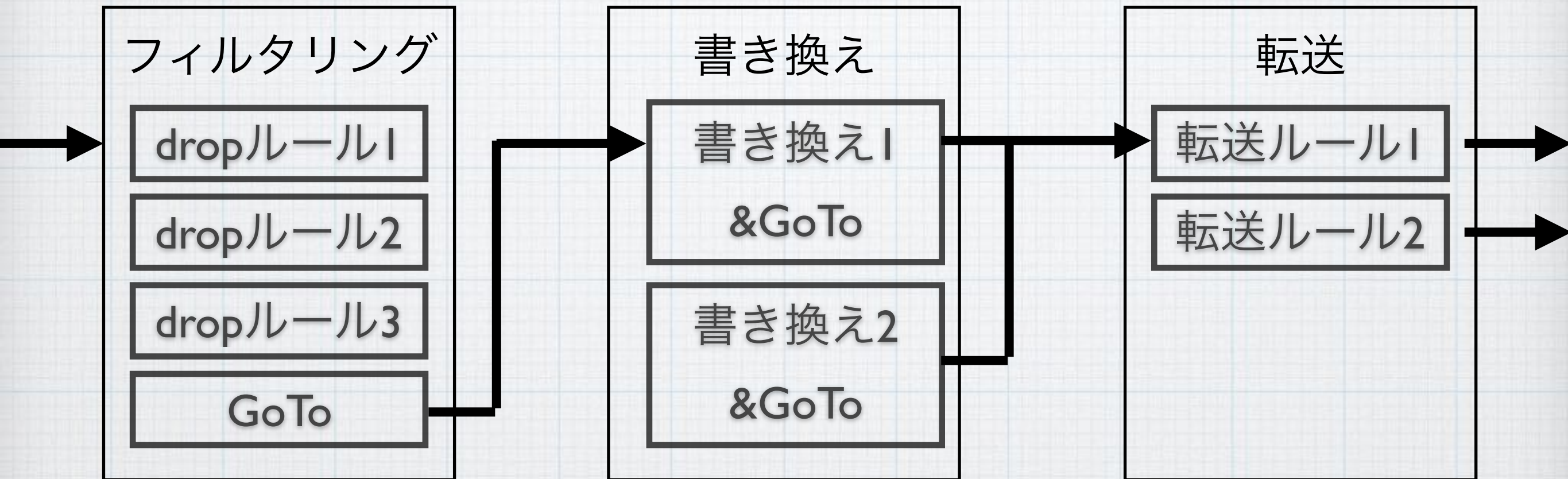
..

転送ルール1

転送ルール2

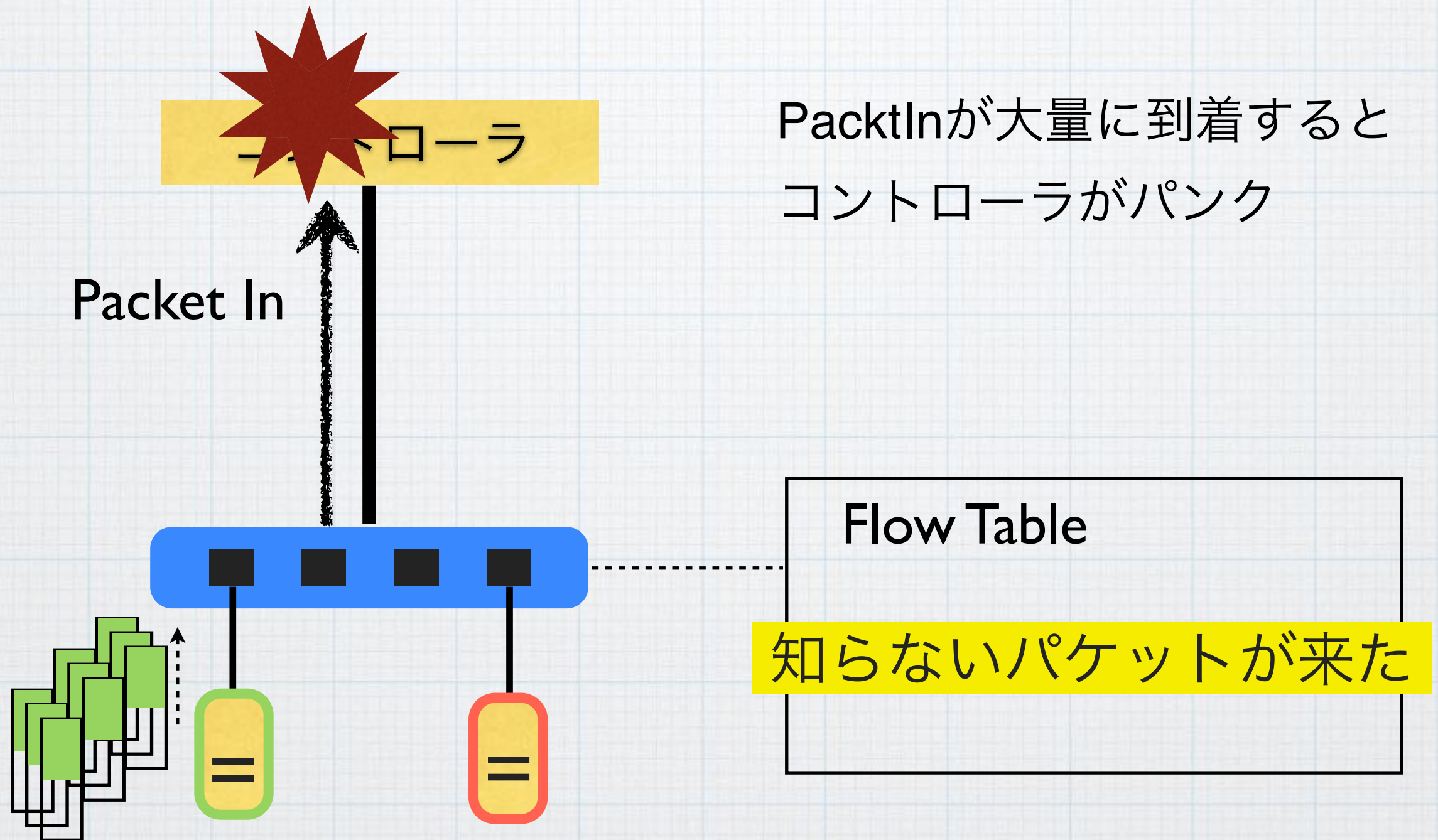
- OpenFlow 1.0 では、フローテーブルは一つだけ
- 役割の違うエントリが混在 & 混乱しがち

OpenFlow 1.3では

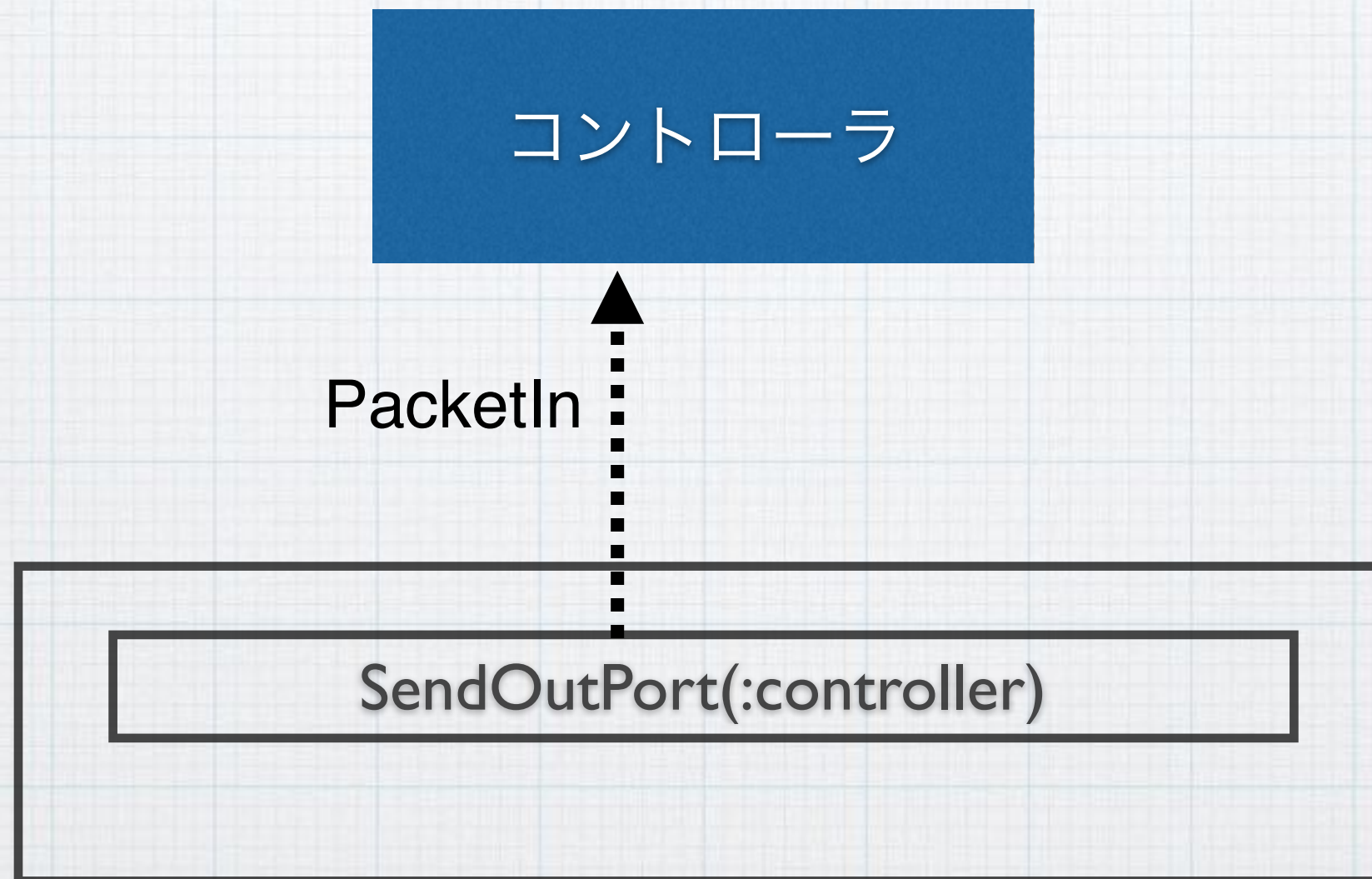


- 分けてスッキリ!
- 処理をパイプライン化できるので性能が向上することもある

OpenFlow 1.0の問題点 2



OpenFlow 1.3では



- 明示的に指定しない限り、PacketInは起こらない

OpenFlow1.3での変更点

アクションと インストラクション

アクション

- パケットの書き換えと転送 (OFI.0と同じ)
- OFI.3で種類が増えました (全40種類)

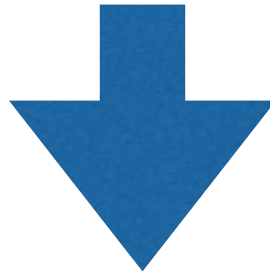
インストラクション

- テーブルの移動 (GoTo)
- アクションの実行方法の指定

```
send_flow_mod_add(  
    datapath_id,  
    table_id: 1,  
    idle_timeout: 0,  
    priority: 1,  
    match: Match.new,  
    instructions: GotoTable.new(2)  
)
```

- ・ 処理をテーブル2へ移行

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



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

アクションセット

[]

[A1]

[A1,A2]



- パケットに関連付いたアクション集合
- WriteActionsでアクションを追加
- GoToを含まないエントリにマッチしたら実行

まとめ

- switch_readyが起きるまで
- OpenFlow 1.3の利点
- アクションとインストラクション
- 実践的な使い方はルータの実装で説明