Unified IPv6 Transition Framework With Flow-based Forwarding

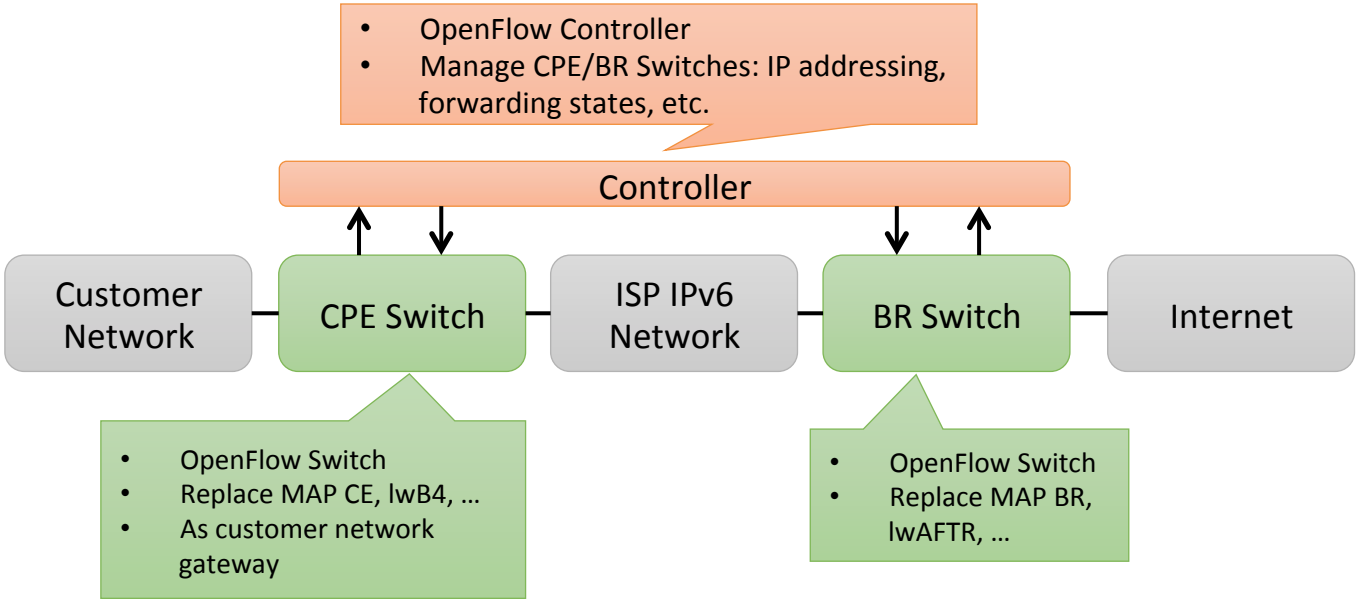
draft-cui-softwire-unified-v6-framework-00
Presenter: Cong Liu

Motivation

- There has been many softwire transition mechanisms
  - Generally look the same, with differences on: addressing, provisioning, address sharing policy, etc.
- This work is trying to discover a “unified” approach for softwire mechanism
  - Use existing methods, currently based on openflow
  - Unify softwire provisioning
  - Unify forwarding devices

Introduction

- Mainly focus on IPv4 over IPv6 tunneling scenario
- Replace routers (CPE & BR) with OpenFlow switches
  - Keep other devices in ISP network unchanged
- Centralized controller to manage provisioning & forwarding rule



Device configuration

- Before connect to the controller, each switch is configured with:
  - An IPv6 address/prefix
  - Controller's IPv6 address, port, etc.
- CPE Switches require automatic configuration
  - Be compatible with RFC7084: Support DHCPv6 PD
  - Controller Information: DHCPv6 or NETCONF (?)

Forwarding Configuration

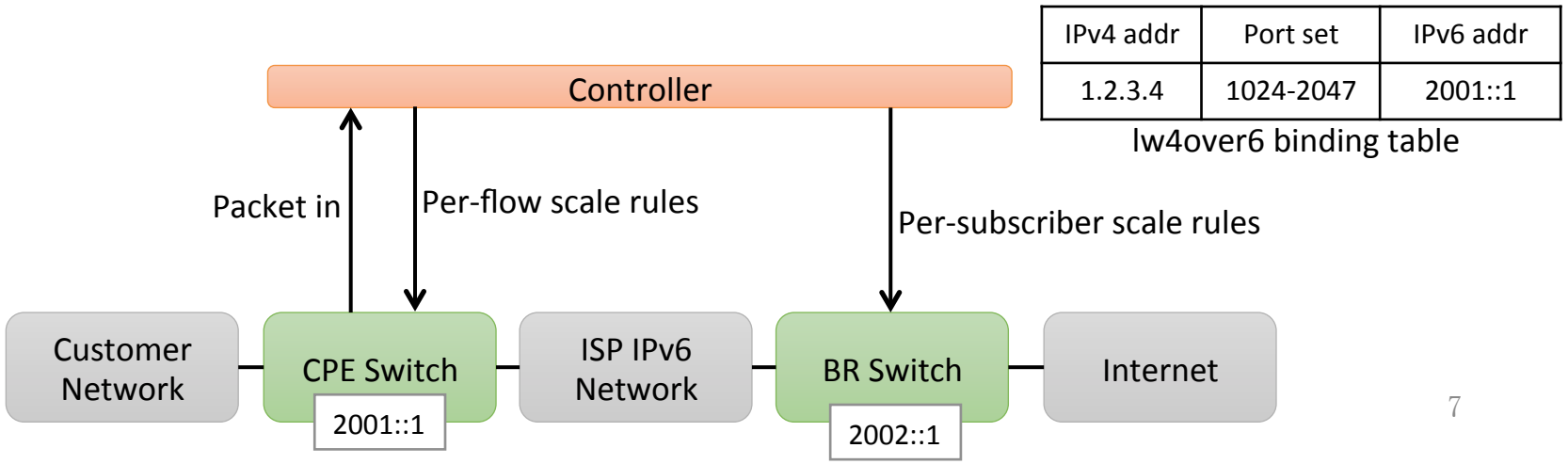
- Use Openflow-style forwarding rule for switches
  - Rule format: Match – Action
- Softwire information are represented by forwarding rules, do not need DHCPv6-based provisioning
  - BR Address: Destination address of CPE's tunnel encapsulation action
  - IPv4 address and PSID: Matching conditions of BR's downstreaming rules, values of set-field actions (to implement NAT44)

Requirements for Switches

- On top of OpenFlow switch
- Action:
  - Both CPE&BR: Support IPv6 tunneling encapsulation / decapsulation actions
- Match:
  - BR Switch: Support match field masking for ports
    (BR Switch can then treat all traffic to the same IPv4 address + port set as a single flow)

Example: 4over6

- Controller preserves IPv4 addr+PSID for each CPE
  - MAP style: calculate from CPE's IPv6 prefix
  - Lw4o6 style: dynamic allocated
- BR Switch forwarding rules:
  - IPv6 tunneling encapsulation / decapsulation rule for each CPE
- CPE Switch forwarding rules:
  - IPv6 tunneling encapsulation / decapsulation rule for all flows
    - Mesh mode: variable tunnel destination address for each destination
  - NAT rule for each flow (re-write IPv4 address and port)



IPv4 addr	Port set	IPv6 addr
1.2.3.4	1024-2047	2001::1

lw4over6 binding table

NAT Fallback

- Allow switches to handle NAT locally
  - Implemented by a virtual interface or iptables
  - Needs automatically configuration for external address and ports
- Keep the ability of controller based NAT
  - Switch could handle “important” flows to improve service quality
- Tradeoff: Flexibility V.S. Efficiency

Next Step

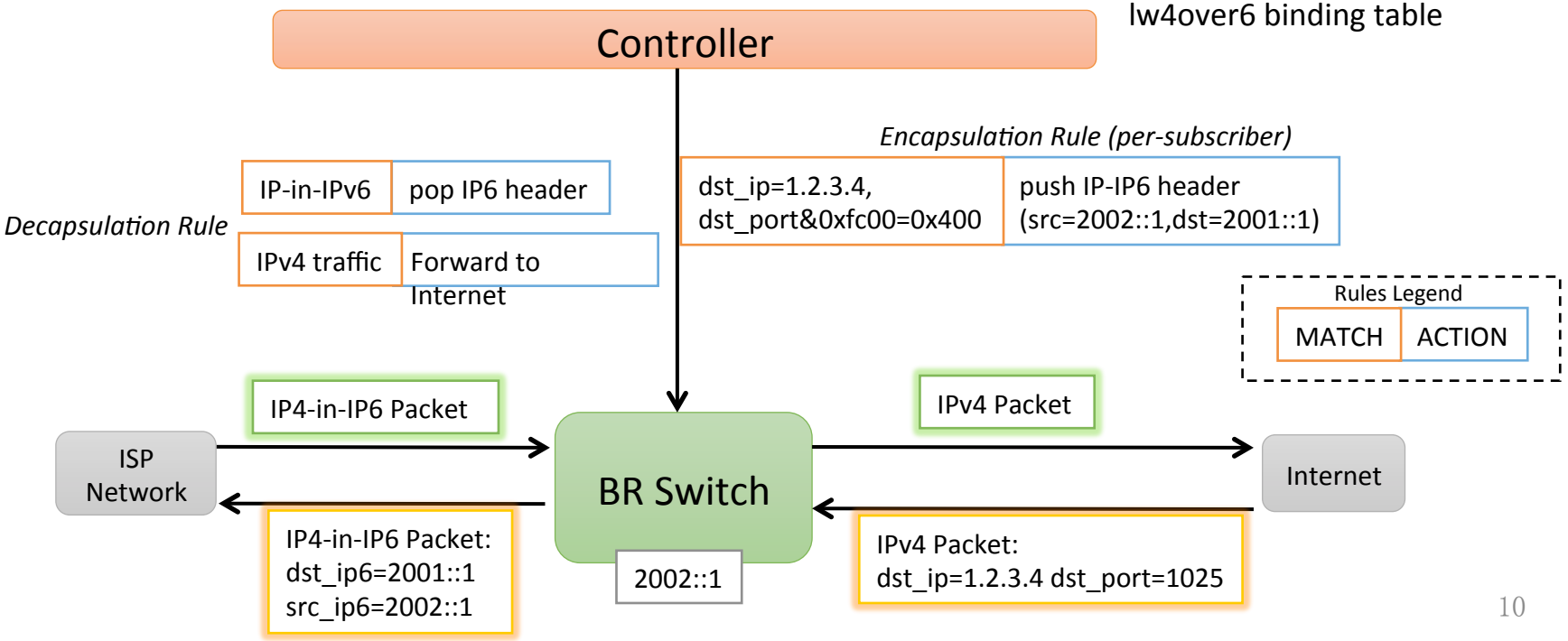
- Comments?
- Move forward in Softwire Workgroup?

Backup: 4over6
BR Forwarding Configuration

- For every binding entry: Controller installs forwarding rules in BR Switch (per-subscriber)
  - Decapsulation Rule: upstream to Internet
  - Encapsulation Rule: downstream to CPE

IPv4 addr	Port set	IPv6 addr
1.2.3.4	1024-2047	2001::1

lw4over6 binding table



Backup: 4over6 CPE Forwarding Configuration

- CPE Switch sends every initial packet of the same (source_ip, source_port) flow to controller
- Controller allocates available public IPv4 address+port, and installs forwarding rules in CPE Switch (per-flow)

Private IP	Private Port	Public IP	Public Port
192.168.1.2	30000	1.2.3.4	1025

NAT state table (for CPE 2001::1)

IPv4 addr	Port set	IPv6 addr
1.2.3.4	1024-2047	2001::1

lw4over6 binding table