

SAVI for Mixed Address Assignments Scenario



Jun Bi, Guang Yao, Joel M. Halpern, Eric Levy-Abegnoli

draft-bi-savi-mix-02

IETF 79, Beijing

Nov.9, 2010

Outline

- Purpose of this document
- Problem statement
- Potential solution for binding set-up
- Potential solution for binding
Removal
- A SAVI device architecture design
- Summary and next step

Purpose of this document



Purpose of MIX-Mode Document

- Current SAVI solutions cover solutions in each single address assignment mechanism (**AAM**)
 - SAVI-DHCP: stateful address assigned by DHCP
 - SAVI-FCFS: SLAAC address
 - SAVI-SEND: CGA with/without certificate
 - Static address: Manually configuration by admin (stated in savi-framework)
- In practice, a network may deploys multiple coexisting AAMs
 - DHCP/SLAAC/CGA/Static...

Purpose of savi-mixed Document

- Each SAVI solutions must be enabled at SAVI device in mixed scenario
- But address assigned by one AAM may conflict with other AAMs
- To resolve co-existence issues in each savi solution draft is not a good way, then a new savi document is needed to handle collisions in mixed scenario – as agreed at IETF 78 SAVI WG meeting

Problem Statement

Problem Statement

- If different savi solutions can bind different addresses, there will be no conflict.
 - So the best solution is not allowing overlap in address block of each AAM, but it can not cover all user's cases in reality
- If there is overlap in allowed address space, then we may have two problems
 - Different SAVI solutions request binding the **same address** at **different binding anchors**
 - Different SAVI solutions request binding the **same address** at the **same binding anchor**

Two Major Conflicts

- Different SAVI solutions request binding the **same address** at **different binding anchors**
 - The key issue is how to set up unique binding entry
 - A **preference level** based solution is proposed
- Different SAVI solutions request binding the **same address** at the **same binding anchor**
 - The key issue is how to **remove a binding entry properly** due to lifetime difference, etc.
 - A solution based on separating the bind entry in control plane and filtering entry in data plane – filtering entry only be removed when all savi solutions remove the binding entries

Potential Solution for Problem 1

Solution for Problem 1: Same address, Different Binding Anchor

- A preference level based solution for binding collision **in set-up procedure**
- The granularity of preference level is at Address Assignment Source (**AAS**)
 - the role who “generates” address
- AAS catalog
 - **Host-self**: slaac, CGA generated by host without certificate
 - **DHCP server**: stateful DHCP (including CGA by DHCP)
 - **PKI**: CGA/plain address with certificate
 - **Administrator**: static address manually configured by administrator (including static DHCP)

A preference level based solution

- A preference of binding is based on the preference of AAS
 - In general, more administration evolved AAS leads to higher priority when bind set-up collision happens
 - Admin>PKI>DHCP server>host-self
 - The savi device can be configured to have DHCP and host-self have the same preference level, then **FBFU** (First Bound First Uses)

A preference level based solution

- The sequence of candidate bindings needs be considered in the solution
 - Conflict happens when a candidate binding violate an existing binding
- For host to give up address if candidate binding preference to existing binding, by using NA send by savi device. This for both single device and multiple devices cases

AAS Preference with sequence

Existing Candidate \	Host-self	DHCP	PKI	Admin
Host-self	scope of savi-slaac and savi-send (Interaction should be considered)	Prefer DHCP	Prefer PKI	Prefer Admin
DHCP	default DHCP, or FBFU if config same priority	scope of savi-dhcp	Prefer PKI	Prefer Admin
PKI	Prefer PKI	Prefer PKI	scope of savi-send	Prefer Admin
Admin	Prefer Admin	Prefer Admin	Prefer Admin	Prefer candidate binding

How it works for multiple savi devices scenario

- For 4 preference levels of AAS, let's say
 - AAS4(statci)
 - AAS3(PKI)
 - AAS2(DHCP)
 - AAS1 (Host-self)
- If the admin configure AAS2 and AAS1 has the same preference level, then we only have AAS4, AAS3, and AAS1 (DHCP and Host-self), then the following description at AAS2 will be not performed at savi device

How it works for multiple savi devices scenario

- DAD/ARP proxy and manually configuration based mechanism could be considered
 - The “Detection state” and “Live state” in SAVI device removed from savi-dhcp is one design example

How it works for multiple savi devices scenario

- When a candidate binding comes, the savi device will
 - act as DAD-proxy only for AAS3 and AAS2. This DAD will be only sent to multicast group of savi devices (not host, because we can only change the savi device behavior, not the host)
 - Meanwhile, savi device block the DAD from host for address from AAS3 and AAS2, to avoid a remote host with lower preference AAS claims the conflict

How it works for multiple savi devices scenario

- When a candidate binding comes, the savi device will
 - Savi-device won't do DAD-proxy for candidate binding from AAS4, it just be bound immediately, the administrator will make sure it's not conflict with other address manually
 - Savi-device won't do DAD-proxy for candidate binding from AAS1, because the host will do it. Since it's the lowest preference, any conflict will stop the host using this address.

How it works for multiple savi devices scenario

- **How to reply** DAD-NA as DAD-proxy at remote savi device when it receives DAD from DAD-proxy
 - savi device will only reply for existing binding from AAS4, AA3
 - savi device won't reply for existing binding from AAS2, because the only reason it needs to reply is because there is a DAD-proxy DAD from AAS2, but the conflict within one AAS is out of the scope of this document
 - savi device won't reply for existing binding from AAS1, because the host will reply it (also, it's within the scope of savi for AAS1, not this mixed document)

How it works for multiple savi devices scenario

- **How to remove** existing binding at remote savi device when receives a DAD from DAD-proxy
 - The removal of existing binding from AAS4 is always done manually by administrator
 - The existing binding from AAS3 won't be removed (because PKI won't conflict)
 - The existing binding from AAS2 will be removed (because only AAS3 cause it, dhcp itself won't conflict)
 - The existing binding from AAS1 will be removed (because AAS3 and AAS2 have higher preference levels)

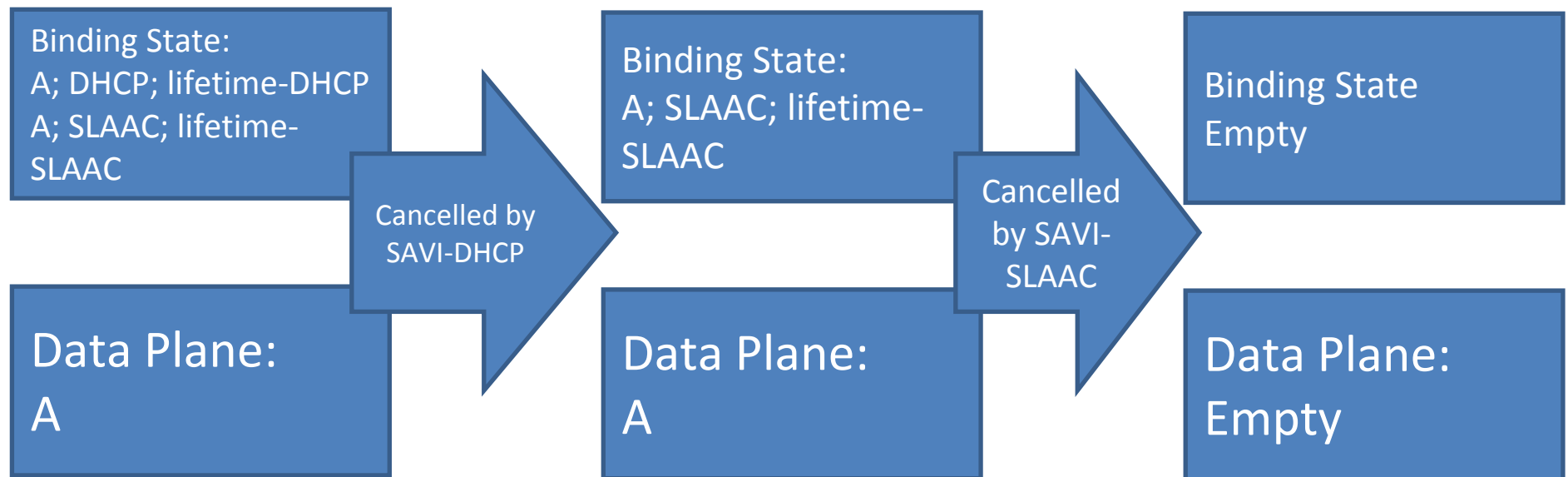
Potential Solution for Problem 2

Solution for Problem 2: Same address, Same Binding Anchor

- One binding entry may be set-up by different savi-solutions in mixed AAMs scenario
- The lifetimes of binding entry generated by different savi-solutions are generally different
- The problem: collision in binding removal is that one savi-solution may require to remove a binding entry when its lifetime expires, before another lifetime expires

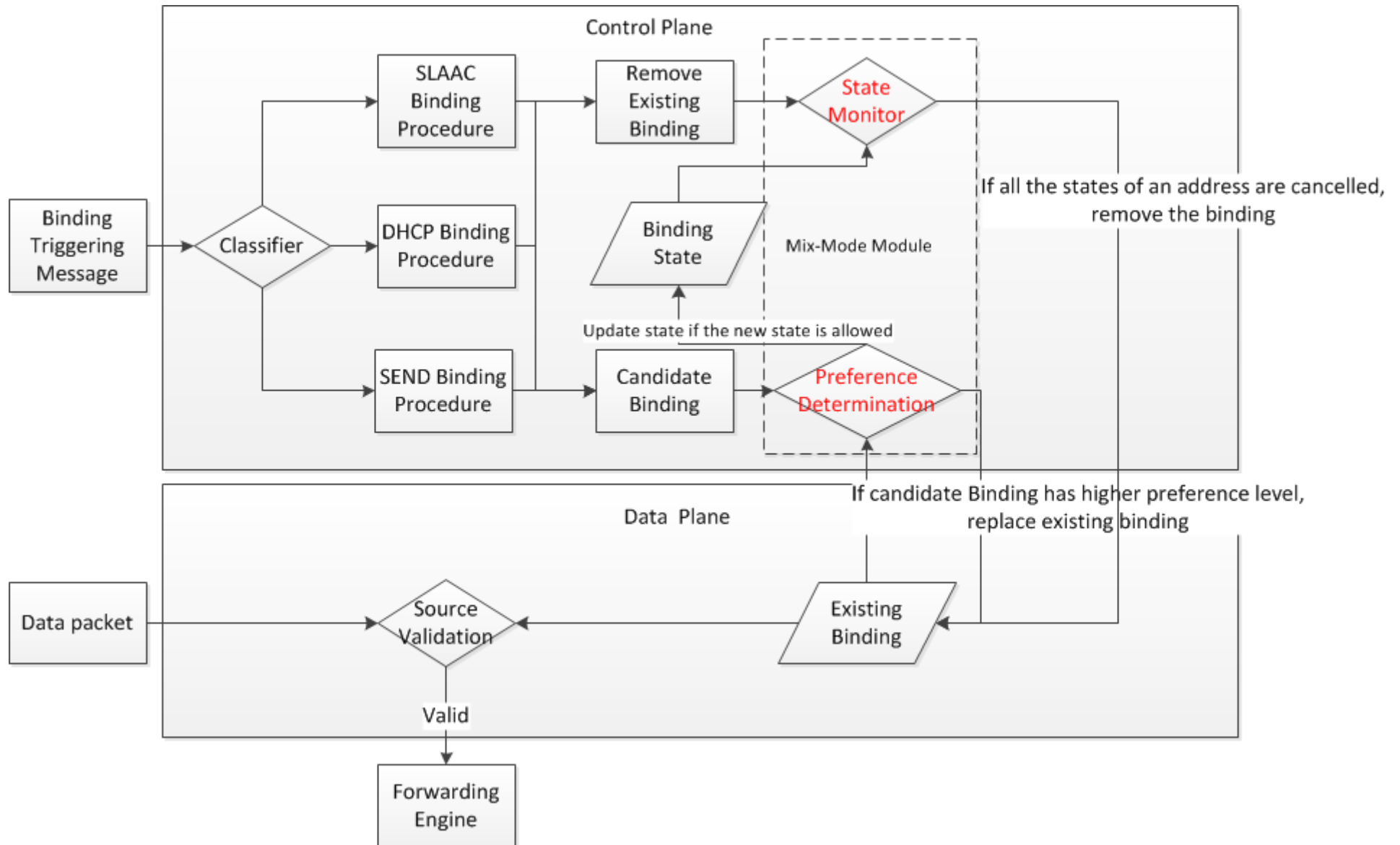
Proposed Solution

- Keep multiple binding state entries set up by different savi-solutions for one address in control plane
- Remove the filtering entry in data plane only when all the binding entries are removed



SAVI Device Architecture Design for mixed mode

SAVI Device Architecture Design for mixed mode



Summary and Next Step

Summary

- Problems in savi-mixed scenario
 - Different SAVI solutions request binding the same address at different binding anchors – set-up
 - Different SAVI solutions request binding the same address at the same binding anchor – removal
- Proposed Solutions
 - Problem1: AAS preference level based solution
 - Problem2: Remove the filtering entry in data plane **only if** all the binding states entries in control plan are cancelled.
- A savi device architecture design reference

Next Step

- Can we adopt this document as a WG draft ?
 - Though the solutions still need enhancement, does this document provide a base for a WG future work ?

Thank you very much!

Existing problem

- It is not enough for only the SAVI device with candidate binding to know the conflict
- The SAVI device with the existing binding must also get known the conflict to determine whether to delete an existing binding or not
- Unfortunately, there is no good idea currently.
 - Maybe new protocol must be designed