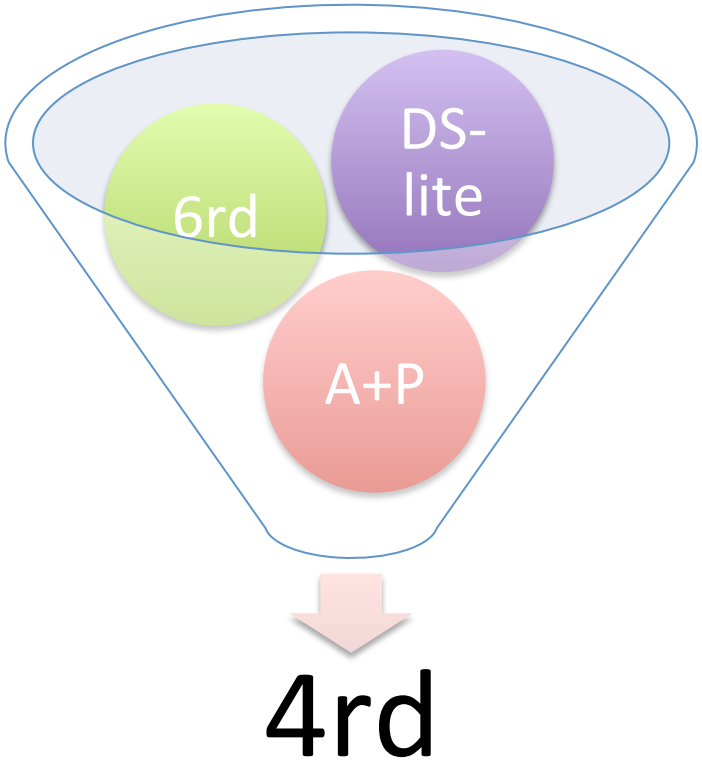
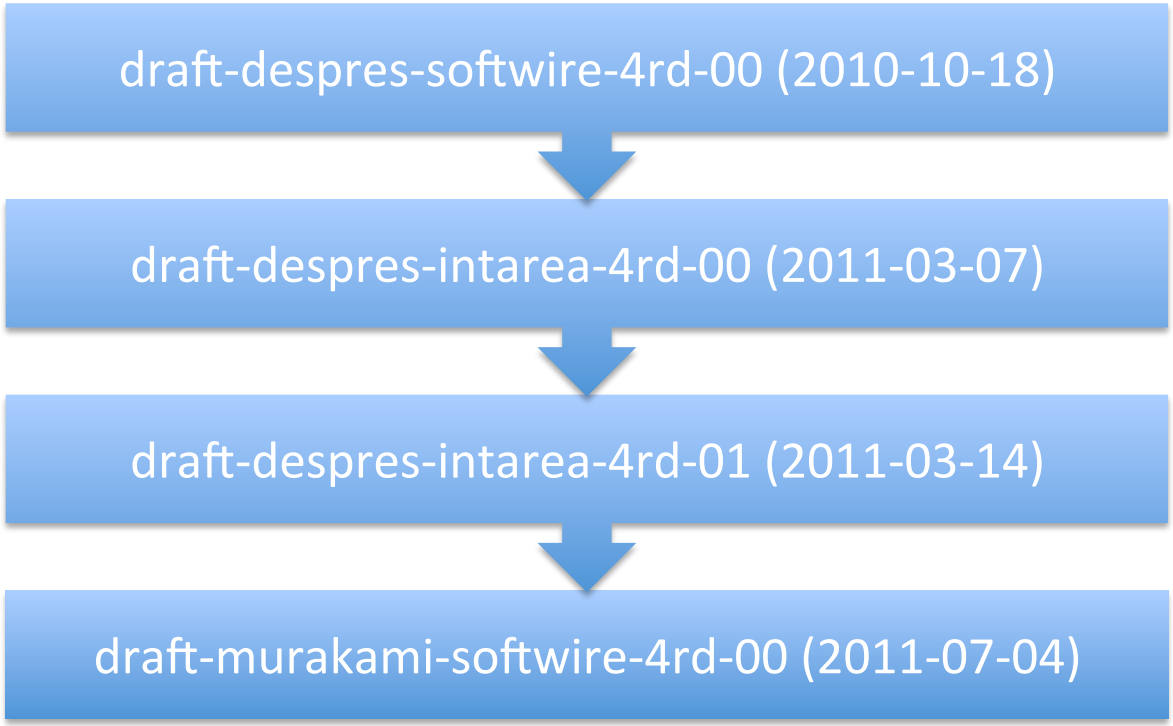
4rd @ IETF 81

draft-murakami-softwire-4rd-00

(Satoru Matsushima / Tetsuya Murakami / Ole Trøan)

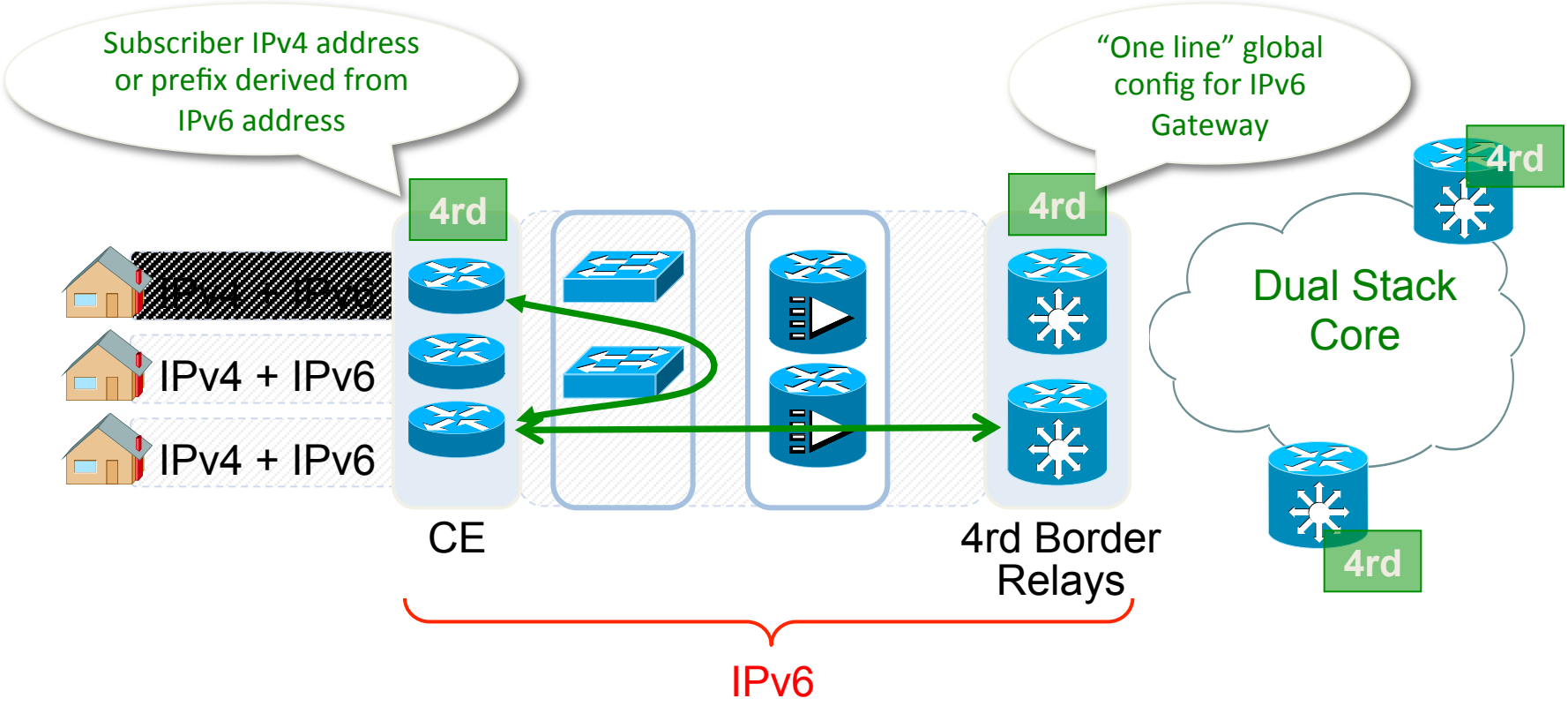


Draft history



Motivation draft: draft-operators-softwire-stateless-4v6-motivation-02
Applicability statement: draft-sun-intarea-4rd-applicability-01
DHCPv6 option: draft-mrugalski-dhc-dhcpv6-4rd-00

4rd in One Slide



- Native dual-stack IP service to the Subscriber
- Simple, stateless, automatic IPv4-in-IPv6 encap and decap functions
- IPv4 traffic automatically follows IPv6 Routing

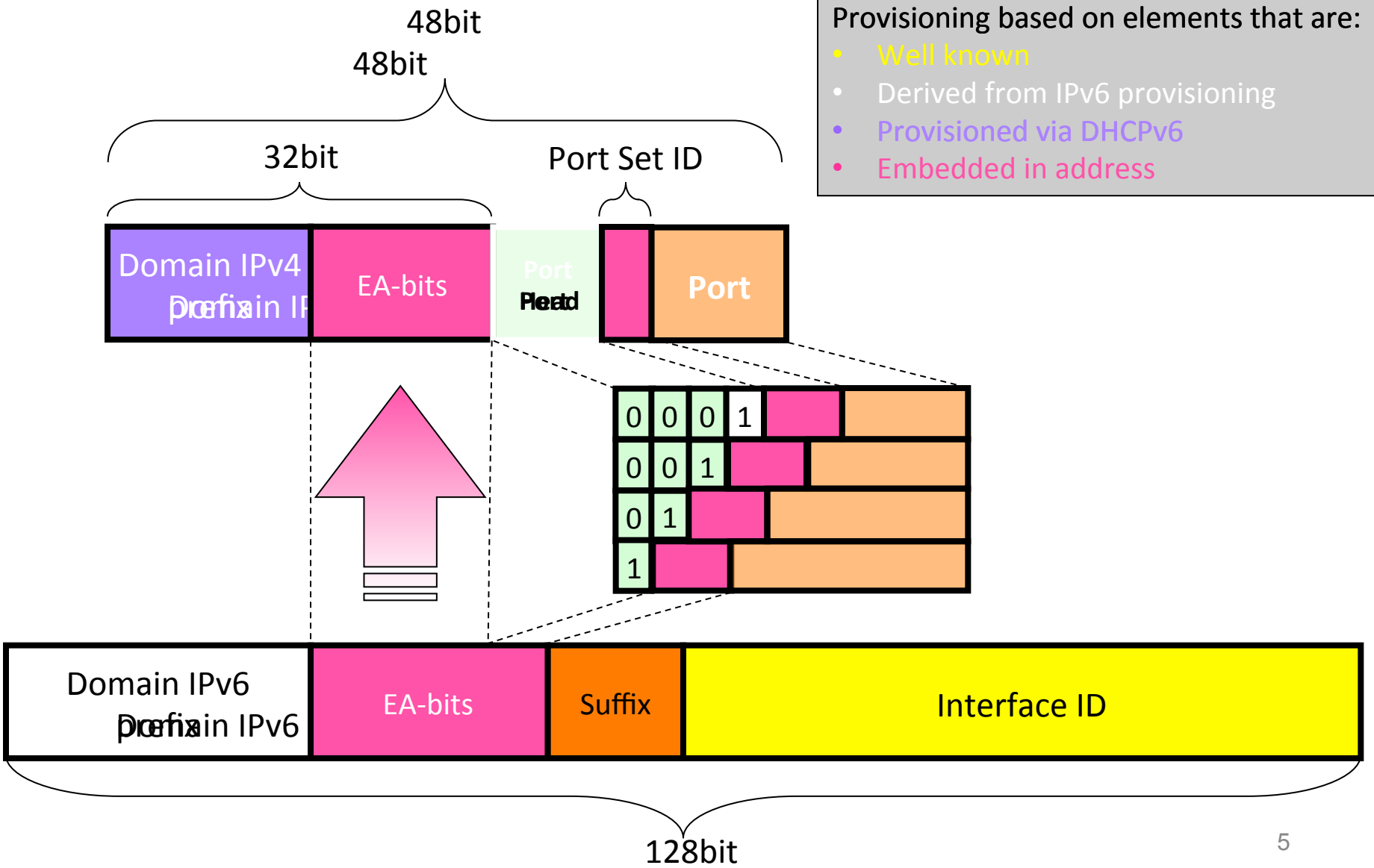
- Native dual-stack IP service to the Subscriber
- Simple, stateless, automatic IPv4-in-IPv6 encap

and

Comparison

4rd aka Stateless DS-lite	Stateful DS-lite
NBMA link	Hub and spoke tunnel
Algorithmic mapping between payload addresses and tunnel endpoint addresses	Configured tunnel with tunnel endpoints configured on AFTRs
Implicitly provisions IPv4 address, prefix or shared IPv4 address	No address provisioning
Scales by traffic	Scales by number of customers
Resiliency achieved by native routing, anycast and multiple BRs	Resiliency achieved by AFTRs never crashing ☺ (Proprietary HA solutions, probing…)
NAT44 on CPE (port restricted)	NAT44 on AFTR
Fixed port set allocated on customer provisioning. (A source port can be bound to many destinations)	Port set limited only by total amount of ports on AFTR
Increase of port space can be done with additional mapping rules or renumbering	Just add more IPv4 addresses on AFTR
4rd CE apps use private address	Private Address

4RD : Stateless Address Mapping

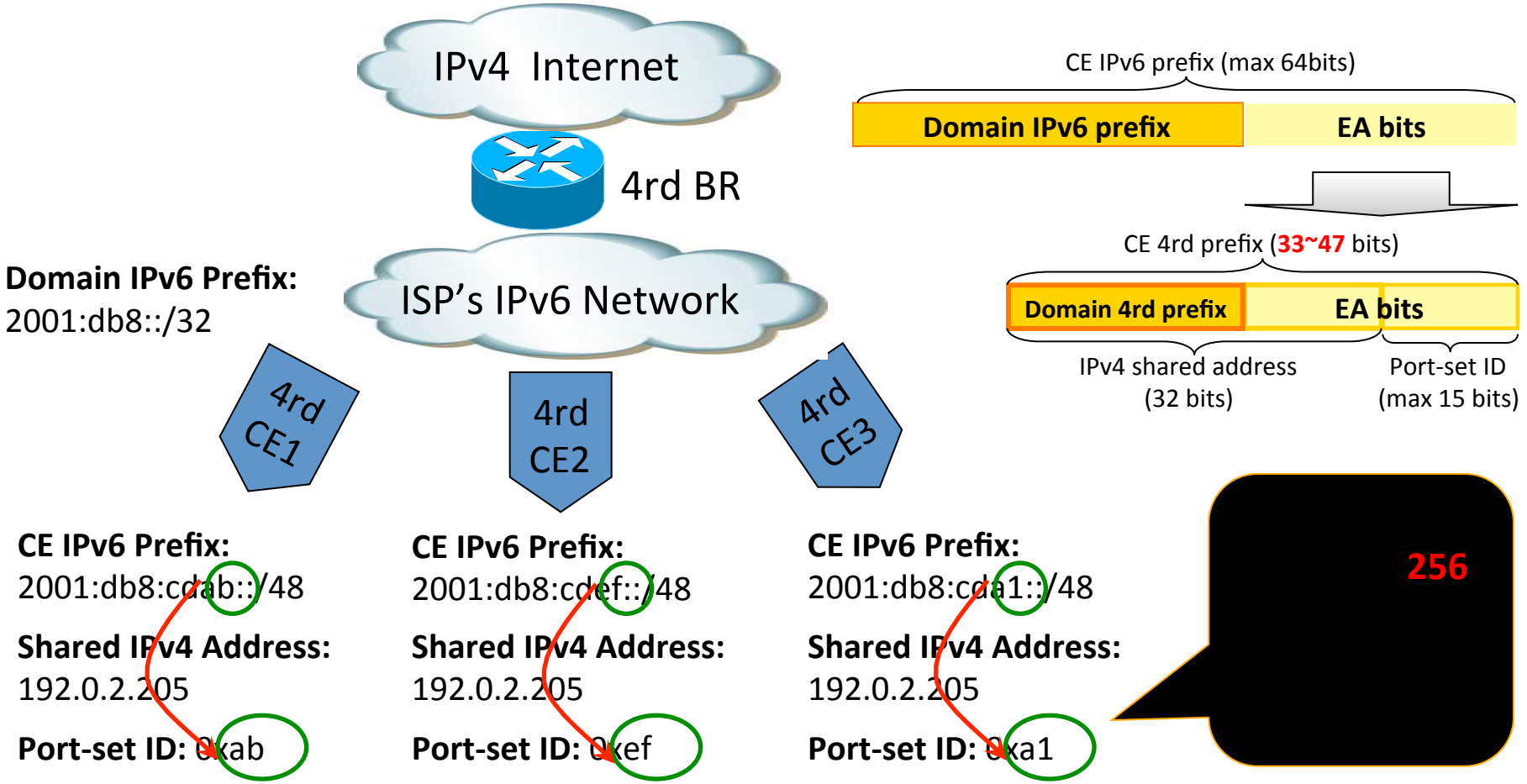


Provisioning based on elements that are:

- Well known
- Derived from IPv6 provisioning
- Provisioned via DHCPv6
- Embedded in address

IPv4 Address Sharing – Port-set ID deriving

If there is a remainder of EA bits, this part will be treated as Port-set ID.



Algorithm	Description	
Port sets with heads	0001 KKKK KKPP PPPP - 64 ports 001K KKKK KPPP PPPP - 128 ports 0001 KKKK KKPP PPPP - 64 ports 001K KKKK KPPP PPPP - 128 ports 01KK KKKK PPPP PPPP - 256 ports 1KKK KKKP PPPP PPPP - 512 ports	Consecutive ports
Prefix based	E.g: a /6 gives 1024 ports.	Consecutive
Modulo	P=j*N + K + 1024, for of j=0, 1, ..., (65536-1024-N)/N P=j*N + K + 1024, for of j=0, 1, ..., (65536-1024-N)/N K=((P-1024) % N) (P – Ports, N – Sharing ratio, K – port set ID) e.g	Scattered

Next: