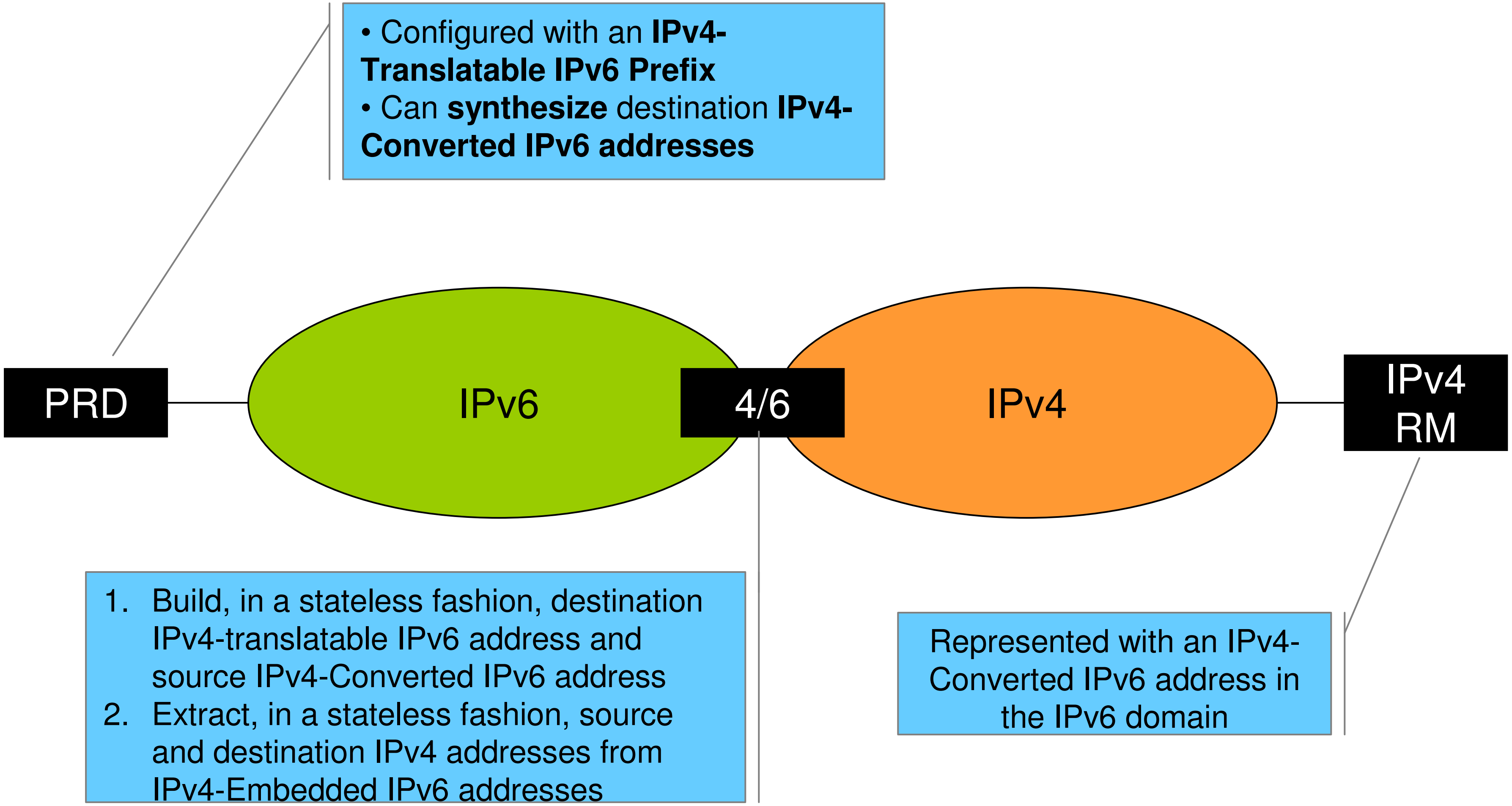
Analysis of Port Indexing Algorithms

draft-bsd-softwire-stateless-port-index-analysis

Softwire WG Interim Meeting-BEIJING, September 2011

M. Boucadair, N. Skoberne and
W. Dec

Terminology Reminder



- Configured with an **IPv4-Translatable IPv6 Prefix**
- Can **synthesize** destination **IPv4-Converted IPv6 addresses**

1. Build, in a stateless fashion, destination IPv4-translatable IPv6 address and source IPv4-Converted IPv6 address
2. Extract, in a stateless fashion, source and destination IPv4 addresses from IPv4-Embedded IPv6 addresses

Represented with an IPv4-Converted IPv6 address in the IPv6 domain

Rationale

- The main goal of -00 if to **understand** and **analyze** the various port indexing schemes proposed so far

- Then, hopefully be able to **compare** them against a set of criteria

- The current version of the draft uses a set of **properties to characterize** each algorithm but no comparison is provided yet

Rationale (Cont’d)

- Two aspects need to be analyzed separately (// or sequential)
  - **Port set indexing scheme**
  - **Address format** to embed the port information in an IPv4-translatable IPv6 address/prefix
    - *Port Set Index is embedded in an IPv4-translatable IPv6 prefix*
    - *Port Number or Port Set Index is embedded in a IPv4-translatable IPv6 address*

Comparison Criteria

- **Not covered in -00**

- **“Requirements for Extending IPv6 Addressing with Port Sets” (draft-boucadair-softwire-stateless-requirements) can be as a starting point**
  - *Port utilization efficiency*
  - *Ability to accommodate various address sharing ratios*
  - *Support of differentiated port sets*
  - *Compliant with RTP/RTCP applications*
  - *Ability to assign 0-1023 to a given user*
  - *Etc.*

Address Format Properties

- **Several properties are defined in -00**
  - *Domain Prefix64 Flexibility*
    - *Ability to support PREFIX64s of different lengths*
  - *IPv4 traffic isolation*
    - *Ability to distinguish between IPv4-embedded IPv6 traffic and native IPv6 traffic*
  - *Encode Routing Bits in 64 bits*
    - *Ability to encode all routing bits in 64*
    - *Applicable only for the scenario in which IPv4-translatable IPv6 prefix is used also for native IPv6 communications*

Address Format Properties

Property	Description
Complexity:	*Reflects the complexity level of understanding the algorithm and the expected complexity to configure an implementation*
Address Sharing Ratio:	*Number of users sharing the same IPv4 address*
Number of ports in a Port-Set:	*Number of assigned ports*
Minimal Sharing Ratio:	*Minimum number of users able to share the same IPv4 address*
Maximal Sharing Ratio:	*Maximum number of users able to share the same IPv4 address*
Guessing Complexity of a Valid Port:	*Level of complexity to guess a valid port within the assigned port set*
Guessing Complexity of the whole Port-Set:	*Level of complexity to guess the whole assigned port set*
Excluded ports:	*Indicates whether ports are from the assigned port set. This provides a hint about the efficiency of the port set algorithm*
Support of 0-1023 port range:	*Ability to assign 0-1023 range to a given user*
Differentiated Port Sets (Bound to the same IP address):	*Capability to assign port sets of different sizes to customers assigned with the same IPv4 address*
Differentiated Port Sets (Network Level):	*Capability to assign port sets of different sizes to customers attached to the same network*
Compliance with RTP/RTCP:	*Compatibility with RTP/RTCP applications*

NOTES

1. **In each analyzed port derivation algorithm, an attacker may implement a redirection loop to detect a significant amount of allowed ports**
   - **For all monotonously scattered schemes, the whole Port Set may be deduced by extrapolation …**
   - **… while this is not applicable for contiguous port ranges because no information about port bounds is leaked in the IPv4-translatable IPv6 address)**
2. **Identifying the whole port set may be seen as a “risk” to identify a given host**
3. **Excluding ports may be seen as a waste of port**

Analyzed Port Indexing Algorithms

- Only algorithms used for stateless 4/6 are covered so far
  1. I-D.boucadair-behave-ipv6-portrange (*portrange*)
  2. I-D.xli-behave-divi (*divi*)
     - divi-pd has been also documented
  3. I-D.murakami-softwire-4v6-translation (*murakami-4rd*)
  4. I-D.murakami-softwire-4rd (*murakami-4rd*)
  5. I-D.despres-softwire-4rd-addmapping (*despres-4rd*)
     - 00 version was complex while updated version is more simpler
     - It is as a variant of portrange

Analyzed Port Indexing Algorithms

Property	portrange	nc portrange	divi	murakmi-4rd	despres-4rd
Complexity:	*Low*	*Low*	*Medium*	*Medium*	*Low*
Address Sharing Ratio:	*1:2^(L-n-32)*	*1:2^(L-n-32)*	*1:N (1:2^E)*	*1:2^p*	*1:N (N up to 12)*
Number of ports in a Port-Set:	*2^(48-L+n)*	*2^(48-L+n)*	*2^(16-E)*	*Note (1)*	*2^(16-N) (N up to 12)*
Minimal Sharing Ratio:	*1:1*	*1:1*	*1:1*	*1:1*	*1:1*
Maximal Sharing Ratio:	*1:65536*	*1:65536*	*1:4096*	*1:32768*	*1:4096*
Guessing Complexity of a Valid Port:	*Low*	*Medium*	*Medium*	*Medium*	*Medium*
Guessing Complexity of the whole Port-Set:	*Medium*	*Low*	*Low*	*Medium*	*Low*
Excluded ports:	*None*	*None*	*0-1023*	*0-4095*	*None*
Support of 0-1023 port range:	*Supported*	*Not Supported*	*Not Supported*	*Not Supported*	*Not Supported*
Differentiated Port Sets (Bound to the same IP address):	*Supported*	*Supported*	*Not Supported*	*Not Supported*	*Supported (Note (3))*
Differentiated Port Sets (Network Level):	*Supported*	*Supported*	*Supported (Note (2))*	*Supported (Note (2))*	*Supported (Note (2))*
Compliance with RTP/RTCP:	*Supported*	*Not Supported*	*Not Supported*	*Supported*	*Supported*

- Note (1): See the formula in the I-D. For each additional bit beyond 12 bits of port-indexing (i.e., when the head is < 4 bits), the number of ports that cannot be used increases by a factor of 2 from the 4096 limit. Thus, for a 13 bit port-set-id, only ports above 8k can be used, ports above 16k for a 14 bit port-set-id, and for a 15 bit port-set-id, only ports above 32k can be used assigned, etc. The port usage efficiency with a 15 bit port-set id is 50%.
- Note (2): This can be supported if different BR are used
- Note (3): This can be supported if the destination port number is embedded in the IPv4-translatable IPv6 address

Misc

- Other algorithms have been proposed but their adaptation to a stateless 4/6 scheme would lead to a complex Port Indexing, e.g.-
  1. Generating Random Port Set and Non-Contiguous Port Range, e.g.,
     - ***Assign 64 Port Ranges with one single Port Mask****: e.g., if the Port Mask is set to 768 and the address is shared between 4 PRDs, 64 contiguous Port Ranges can be assigned to each PRD, there is always one within the span of the first 1024 well-known port values.*
     - ***Assign 128 Port Ranges with one single Port Mask****: e.g., if the Port Mask is set to 496 and the address is shared between 32 PRDs, 128 contiguous Port Ranges can be assigned to each PRD, each one with a length of 16 port values. The first two Port Ranges are both in the well-known ports span (i.e. 0-1023).*
     - Reference: draft-boucadair-pppext-portrange-option
  2. Dynamic Port set
     - Reference: draft-rqb-dynamic-port-ranges

Next Steps

- Complete the comparaison