

Current issues with existing RBNF notation for PCEP messages and extensions

draft-cmfg-pce-pcep-grammar-01

PCE WG, IETF 87 Berlin

Very early version of a work in progress.
This is just starting the discussion and we expect a lot
of refinement

Ramon Casellas, CTTC

Cyril Margaria, Coriant

Adrian Farrel, Old Dog

Oscar González de Dios, Telefónica I+D

Dhruv Dhody, Huawei

Xian Zhang, Huawei

Contributors:

Robert Varga

Overview

- **Introduction and current issues**
 - PCEP has been defined in [RFC5440] and later extended.
 - PCEP RFCs describe specific extensions and focus on their constructs.
- **When implementing a set of extensions**
 - Lack of global view of related extensions – ordering issues?
 - Inconsistent naming
 - Lack of semantics and formal structure
- **Goal**
 - Identify document inconsistencies, provide a reference, complete and formal RBNF for PCEP messages, include object ordering and precedence rules.
 - Ease the development of automated parses & error handling. Avoid interpreting just “text”
- **Note**
 - Do not modify the content of defined PCEP objects and TLVs.
 - Not normative, the normative definition is included in the existing specs (not precluding integration with a future revision of such documents).

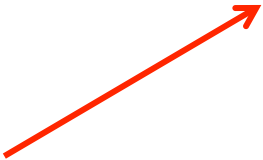
Object ordering

- PCEP uses RBNF, and “an implementation MUST form the PCEP messages using the object ordering specified.” -- [RFC5440], section 6
- RBNF : “ordering (...) in an assignment is **explicit**, (...) specifications MAY (...) state (...) RECOMMENDED..” -- [RFC5511], section 2.3.3

```
<request> ::= <RP>
    <END-POINTS>
    [<LSPA>]
    [<BANDWIDTH>]
    [<metric-list>]
    [<RRO> [<BANDWIDTH>]]
    [<IRO>]
    [<LOAD-BALANCING>]

<request> ::= <RP>
    <end-point-rro-pair-list>
    [<OF>]
    [<LSPA>]
    [<BANDWIDTH>]
    [<metric-list>]
    [<IRO>]
    [<LOAD-BALANCING>]
```

[RFC6006] Note that we preserve compatibility with the [RFC5440] definition of <request> [not really...?]



- **Unspecified**, e.g., [RFC5521] only states “the XRO is OPTIONAL and MAY be carried within PCReq and PCRep”. (**before or after which object?** and SVEC?)
- **Confusing...** e.g. “if a metric is to be applied to a set of synchronized requests, the METRIC object MUST follow the SVEC object “

```
<svec-list> ::= <SVEC> [<OF>] [<metric-list>]
```

Lack of “expressiveness”, “semantics”, “structure”

- If `<response> ::= <RP> [<NO-PATH>] [<attribute-list>] [<path-list>]` then is `<RP><NO-PATH><ERO>` ok?
- Re-arrange to avoid such cases:

```
<response> ::= <RP> ( < path-list > | <NO-PATH> [<attributes>] )
<path-list> ::= <path> [<path-list>]
```

OR even this? – Intermediate constructs? (reuse them in other contexts, etc.) or excessive?

```
<response> ::= <RP> (<success> | <failure>)
<success> ::= <path-list>
<failure> ::= <NO-PATH> [<attributes>]
<path-list> ::= <path> [<path-list>]
```

- Similarly

```
<PCErr Message> ::= <Common Header>
                  ( <error-obj-list> [<Open>] ) | <error>
                  [<error-list>]
<error-obj-list> ::= <PCEP-ERROR> [<error-obj-list>]
<error> ::= [<request-id-list>]
          <error-obj-list>
```

```
<PCErr Message> ::= <Common Header>
                  ( <solicited-error> | <unsolicited-error> )
```

```
<solicited-error> ::= <request-id-list> <pcep-error-list>
<unsolicited-error> ::= <handshake-error> | <pcep-error-list>
<handshake-error> ::= <pcep-error-list> <OPEN>
<request-id-list> ::= <RP> [<request-id-list>]
<pcep-error-list> ::= <PCEP-ERROR> [<pcep-error-list>]
```

Not straightforward... Difference between error and error-obj? Why not OPEN btw?

Minor aspects

- **Naming conventions:** If, given $\langle A \rangle$, $\langle a\text{-list} \rangle ::= \langle A \rangle [\langle a\text{-list} \rangle]$ then $\langle \text{svec-list} \rangle ::= ?$

```
 $\langle \text{svec-list} \rangle ::= \langle \text{SVEC} \rangle$   
                                   $[\langle \text{OF} \rangle] \dots$   
                                   $[\langle \text{svec-list} \rangle]$ 
```

- **Confusing (i.e., Correct from ordering, but the order depends on the message / construct, it makes things a bit more complicated for implementations)**

```
 $\langle \text{svec-list} \rangle ::= \langle \text{SVEC} \rangle$   
                                   $[\langle \text{OF} \rangle]$   
                                   $[\langle \text{metric-list} \rangle]$   
  
 $\langle \text{request} \rangle ::= \langle \text{RP} \rangle$   
                                  (snip)  
                                   $[\langle \text{metric-list} \rangle]$   
                                   $[\langle \text{OF} \rangle]$   
  
 $\langle \text{attribute-list} \rangle ::= [\langle \text{OF} \rangle]$   
                                   $[\langle \text{LSPA} \rangle]$   
                                   $[\langle \text{BANDWIDTH} \rangle]$   
                                   $[\langle \text{metric-list} \rangle]$ 
```

RBNF could be extended for convenience

- Lack of convenient notation,

- e.g. [RFC5886]

- ```
<metric-pce> ::= <PCE-ID> [<PROC-TIME>] [<OVERLOAD>]
```

- > is the intent that at least one is required? Does it in fact mean:

- ```
<metric-pce> ::= <PCE-ID> (<PROC-TIME>|<OVERLOAD>| <PROC-TIME><OVERLOAD>)
```

- Extending RBNF could be useful :

- New convention : “A or B or both, but at least one...”

- ```
<a> || is <a> | | <a> -- “Exclusive OR”
```

- Non-empty sets

- ```
<set> ::= { <a> | <b> | <c> } - repetition not allowed
```

- ```
<set> ::= { <a> <c> } - repetition allowed,
```

- can also be expressed 

```
<set> ::= { <a>... | ... | <c>... }
```

- Capture compound conditional cases, where value of a dictates what follows

- ```
<Q> ::= <a>
```

- ```
 | <c>
```

- If object a field x has value v then object b, else object c.

- ```
(<a with x=v> <b>) | (<a with x!=v> <c>) [Ex. RP flags]
```

Notes:

- Some of the new proposals are overlapping (e.g. Exclusive OR & non empty set w. rep)
- Some rules can be written as per [RFC5511] although way more verbosely (<a> || || <c> || <d>).
- Authors may consider a new I.-D. for RFC5511bis if appropriate.

Conclusions

- Request WG feedback on
 - Are the current specs (specially when combined) + common sense + reading interpretation + “conservative in tx, liberal in rx” + some “errata” to be reported + some luck, good enough?
 - Is it worth the effort? Wasting our time? Obsolete when finished?
- If it is worth the effort, then
 - Should we adhere strictly to RBNF, extend it?
 - Do we need expressive grammars (e.g. intermediate constructs) or not?
 - Do we also address “minor” things (e.g. naming conventions, etc.)?
- Note
 - Effort just barely started, triggered in ML after I.-D. review,
 - We need lots of “eyes” ...
 - All comments are welcome, specially from implementers.