Jeff Haas Comment on CAR Error handling: (thread at)

Jeff's first post: https://mailarchive.ietf.org/arch/msg/idr/qorZr8Y7Cpn4cv-BkH3LhBL391I/

DJ's response: https://mailarchive.ietf.org/arch/msg/idr/MddptzPlTu1FxAgaLAY-9liADgY/

Jeff's second post: https://mailarchive.ietf.org/arch/msg/idr/qorZr8Y7Cpn4cv-BkH3LhBL391I/

BGP-CAR authors,

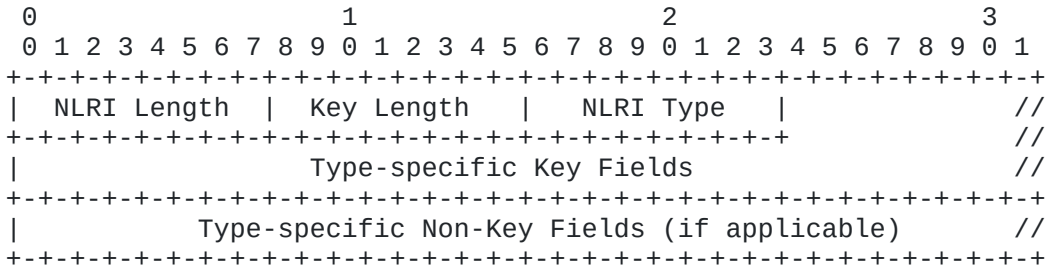
Prompted somewhat by Sue's comment in the Informational Questions:

4. RFC7606 focused on error handling in which the MP-NLRI focuses on destination keys (RD and Prefix) plus non-key material (Labels, SIDS). Attributes (generally) apply to all NLRI. For example, MED applies to all NLRIs in the packet.

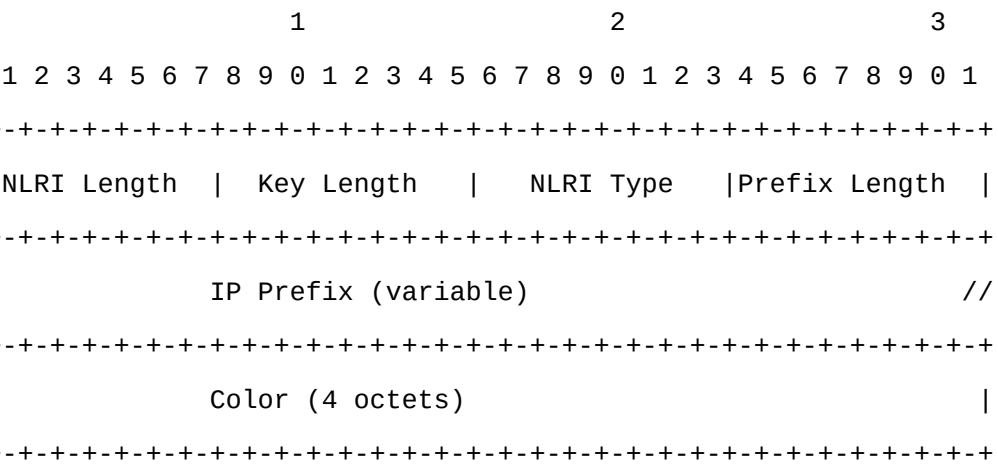
I was looking through the encodings and the related error handling text in the -05 version of the -car document. In section 2.9.1, we have the generic header:

The generic format for the BGP CAR SAFI NLRI is shown below:

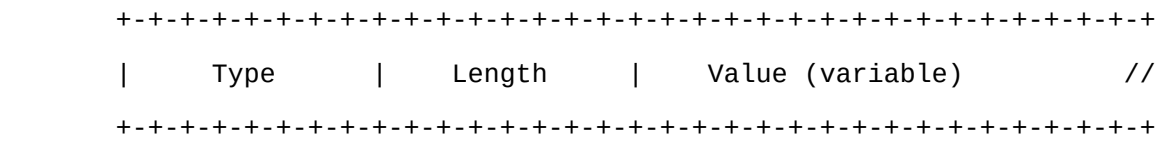
The generic format for the BGP CAR SAFI NLRI is shown below:



The type 1 format in section 2.9.2 builds on that:



Followed by optional TLVs encoded as below:



It creates the definition:

: o Key Length: variable. It indicates the total length comprised of
: the Prefix Length field, IP Prefix field, and the Color field, as
: described below. For IPv4 (AFI=1), the minimum length is 5 and
: maximum length is 9. For IPv6 (AFI=2), the minimum length is 5
: and maximum length is 21.

Section 2.10 covers the error handling procedures, which I quote:

: When the error determined allows for the router to skip the malformed
: NLRI(s) and continue processing of the rest of the update message,
: then it MUST handle such malformed NLRIs as 'Treat-as-withdraw'. In
: other cases, where the error in the NLRI encoding results in the
: inability to process the BGP update message, then the router SHOULD
: handle such malformed NLRIs as 'AFI/SAFI disable' when other AFI/SAFI
: besides BGP-CAR are being advertised over the same session.
: Alternately, the router MUST perform 'session reset' when the session
: is only being used for BGP-CAR.

The general idea makes sense here: Use the key field and as long as you trust its contents, that's enough to do RFC 7606 procedures.

: Following errors result in 'AFI/SAFI disable' or 'session reset':
: o Minimum NLRI length check error.

: o NLRI length conflict with key length.

: o Key length encoding errors (such as minimum, maximum and conflict

: with prefix length).

For Type 1, these also make sense.

: There can be cases where the NLRI length value is in conflict with

: the enclosed non-key TLVs, which themselves carry length values.

: Either the length of a TLV would cause the NLRI length to be exceeded

: when parsing the TLV, or fewer than 2 bytes remain when beginning to

: parse the TLV.

For non-key, overclaim and too-short can cause the TLV set to be malformed. However, this is in the optional stuff, so aside from avoiding issues that TLV protocols introduce of parsing past the content, or interpreting content outside of the containing TLV for a truncated sub-TLV, it's still clear.

I think some additional text covering length checks may still be appropriate. There's particular motivation between the normative RFC 2119 keywords for whether the MUST in question for a specific description of procedure implies malformed packets or not. It might be worth tagging each element of procedure where violations may imply the reset or not.

> **[DJ (CAR author)]:** We will take a second look.
>
> **[Jeff]: Thanks!**

(Note that I'll flag items as needing session reset for brevity. The text already discusses afi/safi shutdown.)

Here's some notes working through the encoding:

Generically:

- The minimum NLRI Length is 2 for unknown types. A key length and an NLRI Type are required fields. If shorter, the NLRI are malformed and we need a session reset.

> **[DJ]:** We could explicitly state the minimum
>
> **[DJ]:** For the General error handling comment, it's not entirely clear if it needs to be repeated explicitly everywhere, or is [it] sufficient to capture in the Error handling section. But we will discuss.
>
> > [**Jeff2:]** I understand it may need a bit of trial and error in the document to find a format that works best. The main challenge I was noting is that our RFC2119 keywords vary in this circumstances between "fatal" and "non-fatal" behaviors with regard to the session. I

[Jeff2:] It is the usual problem of overloaded exception handling. 😊

Why state this when the next requirement implies it? Clarity and a desire to help people avoid underflow of unsigned integers. :-)

- Key Length MUST be at least two less than NLRI Length. (Already stated.) If not, the NLRI are malformed and we need a session reset.

- The NLRI Type may contain unknown fields. The intent is to permit a route reflector to transparently carry unknown types. As will be noted further down, once we hit a device that understands the contents, the downstream BGP Speaker may hit a failure condition that requires session reset. This will be illustrated using the documented Type 1's as an example. I think this may violate the spirit of RFC 7606 where our old Path Attribute form of "optional, transitive nonsense" gets a new life in this kind of NLRI.

  [DJ]: Please see response at the end.

For Type 1:

- Key Length gets additional semantic checks. This is our first example that validation may change depending on whether the type is known vs. unknown. Failure of those minimums require session resets.

  This likely motivates the error handling text being specific about general behavior (small set of things) vs. type-specific. The text is mostly structured this way, but clarifying that we've moved into type-specific validation would be good for readability.

  [DJ]: We will check.

- Prefix Length specifies the usual limits for IPv4/IPv6. It'd be good to make sure the definition section notes that violating length is worth a session reset.

- IP Prefix: Consider borrowing the text from the main BGP RFC. This would also cover what to do about "trailing bits".

  I suspect the usual caveats in RFC 4271, §6.3 apply about semantically incorrect addresses being ignored?

  [DJ]: We believe so. We will check if we can borrow the text.

  [Jeff-2]: Thanks. I think the somewhat novel consideration, going with the novel encoding, is that with the intent of targeting generic reflectors is that situations like the "trailing bits" accommodation may be worth abandoning.

  [**Jeff-2]** The path that got us to our behaviors in RFC 4271 were based on a certain level of forgiveness (Postel's Maxim) with regard to trailing bits based on long history. But even with long history, when implementations do have accidental trailing bits, we still see bugs in the field with regard to processing them.

  [Jeff-2]: In formats like BGP-LS, BGP Flowspec, and structured NLRI as seen in many of the BESS protocols, we have additional issues that "don't care" bits become problematic because it's not clear when an implementation should or should not ignore

> them for key considerations. Most implementations fall back to simple memcmp() semantics without a canonicalization step. Thus, two theoretically comparable NLRI may not compare identically depending don't-care semantics.
>
> **[Jeff-2]:** As a new NLRI, you do have the option to make a choice in the matter. The challenge is when that choice conflicts with established practice (existing library code!), even if for good reasons.

Not specific to the key fields:

- The optional TLVs are largely targeted toward forwarding behaviors. It's not clear what should be done if more than one TLV is carried at the same time.

> **[DJ]:** This is a matter of local policy to decide which one or more forwarding types We can make it explicit.

The above comments on the key field validation is, I think, the essence of whether the NLRI keys can be safely used or not. Making sure that the violations and their impact are close together in the text would be helpful.

Where we see a more general issue is the transitional stage of going from an unknown type to a known type could result in downstream session resets.

The goal was to provide transparent carriage of unknown types, but unless I'm missing something, we didn't quite succeed at this when errors are present and raise issues similar to those we were trying to mitigate in RFC 7606.

> **[DJ]:** Since the intent itself is to allow new route types to transit through an RR transparently without requiring an upgrade, the RR cannot detect an error in that NLRI. It is expected though that the client of the RR (for example, an ABR) which is supposed to consume this route for installation will be upgraded, and hence can do the error handling. The route will therefore not propagate any further. This is also true if the ABR is not upgraded and does not recognize the new route type. It will not install and hence not propagate any further.
>
> > **[Jeff-2]** Agreed. However, that means that like the situations that motivated RFC 7606, the session reset happens "far away" from the route origination.
>
> **[DJ]:** One possible option is to enable the operator to make the decision on whether they want to allow this propagation through an RR; or enforce the validation at the RR, in which case the RR will need to be upgraded. We could describe this consideration and option in the Deployment section. Does that look useful to you?
>
> > **[Jeff-**2] Possibly so. Minimally, the Error Handling section could use the text. One additional way to consider addresssing this is to capability negotiate whether a given NLRI type for the afi/safi is understood. Implementations may thus have the operational ability to choose whether they accept unknown NLRI types for that afi/safi or not.
> >
> > **[Jeff-**2]: You may find my somewhat recent flowspec capability bits draft a possible example for encoding such a thing.