Core -13 to -15 updates


• New option area structure
  – OCS field required and word-aligned
  – Zero padding for alignment
  – Followed by sequence of option TLVs

• Integrate UNSAFE kinds in main option range
  – 194-254 now UNSAFE
  – Simplifies parsing UNSAFE options

• FRAG requirements updated
  – At least 2 fragments for a total of at least 3,000 bytes (2x IPv6 EMTU_R)

• REQ / RES now required
  – To enable ubiquitous PLPMTUD via UDP-Opt
Other updates

- Clarify short/long option formats
  - All options except EOL, NOP MAY use either
  - SHOULD use most compact

- How to make required OCS field “optional” – same as UDP CS (zero)
  - MAY be zero, e.g., when UDP CS=0 {slide #2 IETF 110, 112}
  - MAY be zero prior to fragmentation

- Two experiment kinds
  - Optional
  - Required if present (“unsafe”)

- Expanded set of recurring options
  - NOP and now EXP and UEXP
  - Allows multiple experiments

- ACS -> APC
  - More clearly “Alternate Payload Checksum”
Additional -15 to -16 updates

• Specific limits on options
  – Regarding DOS in security considerations
  – Based on RFC8504

• Correct, updated TOC

• Terminology consistency pass
  – Added terminology section
  – Consistent use of ‘surplus area’, rather than also ‘option area’; use of ‘user data’, rather than also ‘UDP payload’, etc.
  – Changed MSS/MRSS to MDS/MRDS, reflecting datagram terminology of UDP
Known pending updates

• More overall consistency/clarity checks, especially failure handling
  – UDP user data is **always** passed to the app*
  – ”hard” option errors **are no exception**
    • OCS does not succeed
    • Parsing / structure error (wrong len, UNSAFE outside FRAG)
    • Any UNSAFE option does not succeed (zero-len user data)
  – “soft” option errors also pass UDP option list to the app
    • Any SAFE option does not succeed
  – *Receiver CAN be configured to do “more”
    • E.g., prevent receiving packets at all if certain options fail

• API summary – description of required aspects
  – Outgoing packet and fragment options, per-socket default or per-packet
  – Incoming option failure handling: silently discard, count and discard, or pass
  – Incoming option omission handling: silent discard, count and discard, or pass
  – Incoming option info MUST be available for each packet (even if raw)