Invalid TLV Handling in IS-IS

draft-ginsberg-lsr-isis-invalid-tlv-02

Les Ginsberg, Cisco
Paul Wells , Cisco
Tony Li, Arista
Tony Przygienda, Juniper
Shraddha Hegde, Juniper

Motivations

- **Explicit statement for handling TLVs which are disallowed in a given PDU type not easily available**
- **Some Interoperability issues seen in handling TLVs which are unrecognized/incorrectly formatted**
  - **LSPs rejected because of unsupported TLVs/sub-TLVs**
  - **LSPs rejected because of malformed TLVs**
- **Purge Handling now has multiple modes – interoperability issues seen here as well**
  - **Non-compatible imposition of TLV allowance rules**
- **Interoperability issues compromise network operation (inconsistent LSPDB)**

Changes Since IETF 104

- **Clarified the text in a number of places**
- **Thanx to Bruno Decraene for his review**
- **Last Call Started June 12, 2019 – significant support expressed – no objections voiced**

Next Steps

(assuming WG Adoption has completed ⩵)

Ready for Last Call

Backup Slides (Presented in Prague)

POI TLV Registry Issue

Value	Name	IIH	LSP	SNP	Purge	Reference
13	POI	N	Y	N	Y	[RFC6232]

Section 3 of RFC 6232:

“The POI TLV SHOULD be found in all purges and MUST NOT be found in LSPs with a non-zero Remaining Lifetime.”

13	POI	N	N	N	Y	[RFC6232]

Control of non-backwards compatible extensions

Specification	Requirements
RFC 5304/5310 (Crypto auth)	Body of LSP (TLVs) MUST be removed on transmission. Purges which have TLVs other than authentication MUST be ignored on receipt.
RFC 6233 POI TLV	Additional TLVs allowed in purges (POI, hostname, MI IID, Fingerprint) Not backwards compatible w RFC 5304/5310

A need to control when POI TLVs can be sent.

“It is recommended that implementations provide controls for the enablement of behaviors that are not backward compatible.”

Backup Slides (Presented in Bangkok)

Handling Received TLVs

ISO 10589 Section 9.3

"Any codes in a received PDU that are not recognised shall be ignored.“

New TLVs are unrecognized by older implementations => older implementations do not know allowed status for new TLVs

Unsupported == Disallowed

(This applies to sub-TLVs as well.)

Category	Action
Supported	Process
Supported – incorrectly formatted	Ignore TLV
Unsupported	Ignore
Disallowed	**Ignore**

LSP Acceptance (non purge)

The unit of propagation for the Update process is an LSP (not a TLV).

LSP Acceptance tests:

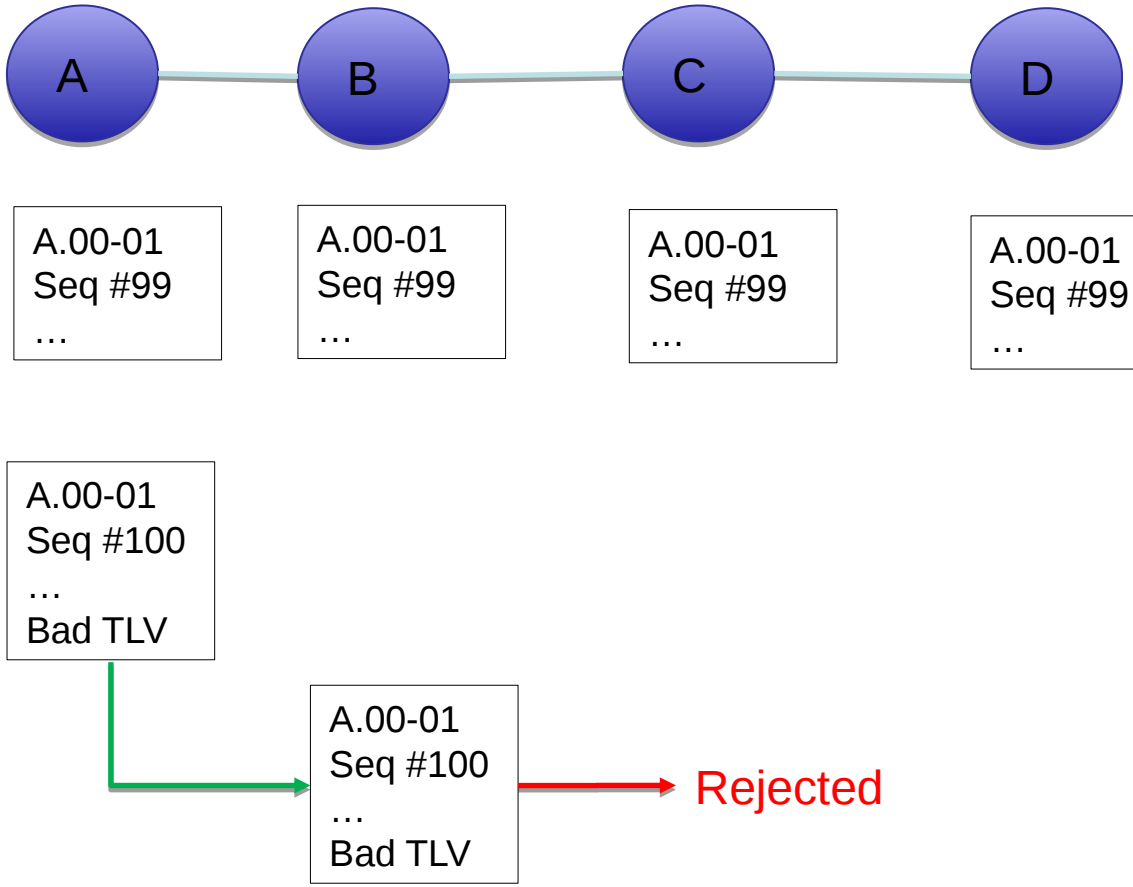
Checksum valid

Authentication valid (if present and in use)

LSP is “newer” or the “same” (based on sequence #)

TLV content is NOT relevant!!

Interoperability Issues



“Bad TLV” => Unsupported, disallowed, malformed

Purged LSP Acceptance

Specification	Requirements
ISO 10589	Body of LSP (TLVs) should be removed on transmission – but is ignored on receipt (no checksum) Only plain text authentication supported
RFC 5304/5310 (Crypto auth)	Body of LSP (TLVs) MUST be removed on transmission. Purges which have TLVs other than authentication MUST be ignored on receipt.(Not backwards compatible)
RFC 6233 POI TLV	Additional TLVs allowed in purges (POI, hostname, MI IID, Fingerprint) Not backwards compatible w either of the above modes

POI Implementation Issues

POI extensions are NOT backwards compatible w strict RFC 5304/RFC 5310 compliance. Therefore POI enablement in the presence of crypto authentication is dependent on the entire area supporting the extension.

Without crypto authentication POI can be accepted under base 10589 rules.

With crypto authentication TLVs fall into following categories:

TLV Category	Actions
Supported	Reject Purge if TLV is disallowed in purges
Not supported	Ignore (implementation does not know if TLV is allowed or not) This is key to allow new TLVs to be defined and allowed in purges.

Interoperability Issues Purges