

Universal Object Delivery (UOD) [®] Generalized Object Encoding (GOE)

IETF 83 MARCH 25 – 30, 2012 Paris, France

Kevin Fall, Mike Luby & Thomas Stockhammer



Comments on GOE

- GOE provides some interesting ways to combine protection over different objects (or portions of objects)
- > We believe it should be considered for adoption
- > We would be willing to contribute to GOE
 - Qualcomm does have IPR in this general area
 - ➢ If it is unavoidable, IPR statement would probably be similar to that of RFC 6330
 - We would be willing to work with the group to avoid IPR if this is considered important



UOD proposal

➢ Overall UOD

- Basic
 - Simpler and more natural FEC Payload ID for object delivery
 - One parameter instead of two
- Extension
 - Capability to put symbols for unrelated objects in the same packet
 - Improvement in reliability of delivering small objects
 - > We provide some further generalization in this presentation
- Currently specified for RaptorQ (RFC6330)
- Qualcomm has IPR on UOD and will make an IPR disclosure
 - IPR statement would probably be similar to that of RFC 6330
- International Computer Science Institute has IPR related to UOD
 - Probably will make IPR royalty-free



GOE and UOD

- GOE and UOD address complementary important use cases for joint protection of multiple objects
 - ➢ GOE somewhat akin to streaming bundling in FECFRAME (RFC 6363)
 - UOD somewhat akin to sub-blocking specified in RaptorQ (RFC 6330)
- Use cases for GOE are important
 - We support working group adoption of GOE
 - Should consider generalizing to a *framework* specification so that GOE can be leveraged by all FEC Encoding Schemes

Use cases for UOD are important

- We are seeking support for working group adoption of UOD
- We would consider generalizing to a framework specification so that UOD can be leveraged by other FEC Encoding Schemes
- GOE and UOD can be used together in some use cases



Basic UOD for RaptorQ

FEC OTI is the same

FEC OTI for Basic UOD RaptorQ is exactly the same as for RaptorQ in RFC 6330

FEC Payload ID is simplified

- FEC Payload ID for RaptorQ in RFC 6330 is (SBN, ESI)
- FEC Payload ID for Basic UOD RaptorQ is UOSI
 - No separate limitations on the number of source blocks and on the number of source symbols per source block
 - > Transmitting symbols in UOSI sequence is network friendly

0	1	2	3
0123456789	0123456789	0123456789	01

Universal Object Symbol Identifier (UOSI) (32 bits)



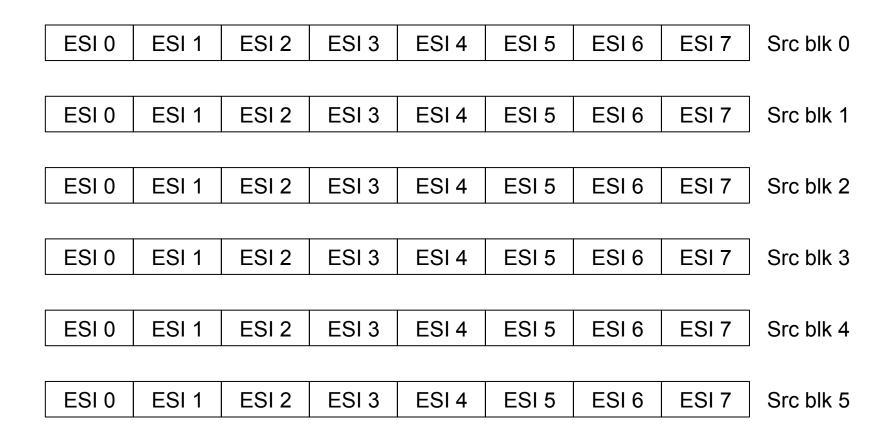
UOSI $\leftarrow \rightarrow$ (SBN, ESI) mapping

- From UOSI value C to (SBN,ESI) for an object with Z source blocks:
 - ESI = floor(C/Z)
 - SBN = C (ESI)*Z
- From (SBN,ESI) values (A,B) for an object with Z source blocks to UOSI value C:
 - C = SBN + (ESI)*Z
- Example for Z=5:

UOSI (C)	SBN	ESI
0	0	0
1	1	0
2	2	0
3	3	0
4	4	0
5	0	1
6	2	2

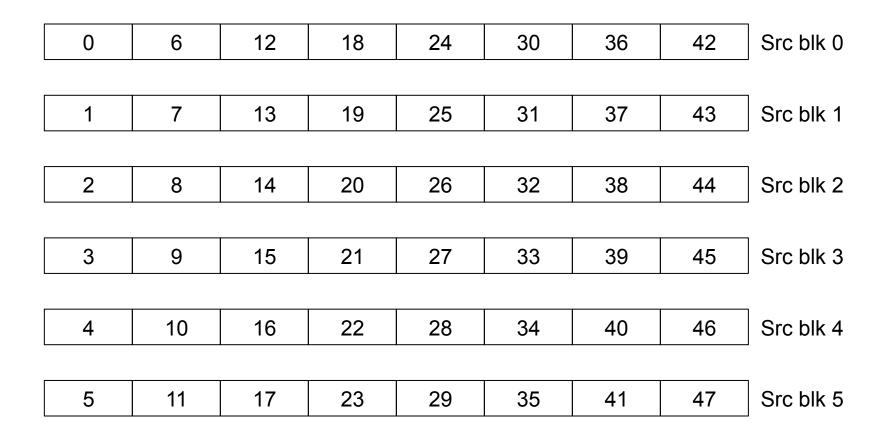


Example object with FEC Payload ID (SBN, ESI)





Example object with FEC Payload ID (UOSI)



Sending symbols in UOSI order ensures network-friendly interleaving



Extended UOD

Capability to put multiple symbols for unrelated objects in the same packet



Extended UOD

- New table Packet Structure Table (PST)
 - Each entry is identified by a unique Packet Format Identifier (PFI)
 - Each entry in the PST contains:
 - The number and sizes of FEC Payload IDs in a packet
 - Specifies which object is associated with each symbol in a packet
 - Uses TOIs to make this association
 - Specifies which FEC Payload ID to use for each symbol in a packet
 - PFI is carried in each packet to identify the format and data that is carried in the packet
- Example
 - First PST entry specifies packet format of:
 - Two FEC Payload IDs of two bytes each UOSI format
 - One symbol for the first object with TOI that uses the first FEC Payload ID
 - · One symbol for the second object with TOI that uses the second FEC Payload ID



Sketch (exact formats TBD)

