ICN Hop-By-Hop Fragmentation Update: Begin-End Fragmentation (BEF)

Marc Mosko
Palo Alto Research Center
marc.mosko@parc.com

Christian Tschudin
University of Basel
christian.tschudin@unibas.ch

IRTF ICNRG interim meeting, Prague, CZ, July 19, 2015

Introduction

- Submitted IETF document
  - https://datatracker.ietf.org/doc/draft-mosko-icnrg-beginendfragment/
- CCNx implementation status
- Draft updated for NDNLPv2
- CCN-lite implementation

IETF document

ICNRG
Internet-Draft
Intended status: Experimental
Expires: January 3, 2016

M. Mosko
PARC
C. Tschudin
University of Basel
July 2, 2015

ICN "Begin-End" Hop by Hop Fragmentation
draft-mosko-icnrg-beginendfragment-00

Abstract

This document describes a simple hop-by-hop fragmentation scheme for ICN and mappings to the CCNx 1.0 and NDN packet formats, called "begin-end fragmentation". This scheme may be used at Layer 3 when ICN packets are used natively over a Layer 2 media which does not reorder packets.

https://www.ietf.org/id/draft-mosko-icnrg-beginendfragment-00.txt

CCNx Implementation Status

- Basic protocol implemented
  - Current CCNx forwarder implements the "basic" format where fragmentation state encoded in the Fixed Header.
  - Extended format forthcoming.
- Used automatically for Ethernet links.
- UDP links still using IP fragmentation.

Begin-End Fragmentation for NDN

- Caveat:
  - Not NDN approved, not part of NDN code distribution
  - but aligned with ongoing NDN Link Protocol v2 discussion

```
NDNpacket       ::= Interest | Data | NDNLP
NDNLP           ::= NDNLP-TYPE TLV-LENGTH
                      NDNLPhdrFields*
                      NDNLPfragment?
NDNLPhdrFields  ::= BeginEndField | (other NDNLP header fields)
BeginEndField   ::= BEGIN-END-FIELD-TYPE TLV-LENGTH
                      BYTE BYTE+
NDNfragment     ::= NDN-FRAGMENT-TYPE TLV-LENGTH
                      BYTE+
```

CCN-lite Implementation Status

- Supports BEF functionality, both for
  - CCNx encoding (basic format)
  - NDN encoding

  using the same fragmentation code
- Tested on RFduino hardware
  - Bluetooth Low Energy, 20 Bytes MTU

- Insights:
  - hard to use with fragmentation factor above x10 over BTLE
  - CCNx 1.0 fixed header hurts (in IoT), too many frags/chunk
  - a case for IoT-specific encoding? (going away from TLV)

Next Steps

- CCNx implementation
  - Implement Extended format
- CCN-lite implementation
  - open question: how to (de-)activate BEF on a link? → importance of link negotiation protocol
  - add support for "official" NDN indexed fragmentation, too
  - Interops
- Protocol
  - Define BE protocol that supports out-of-order packet reception for use in UDP environments

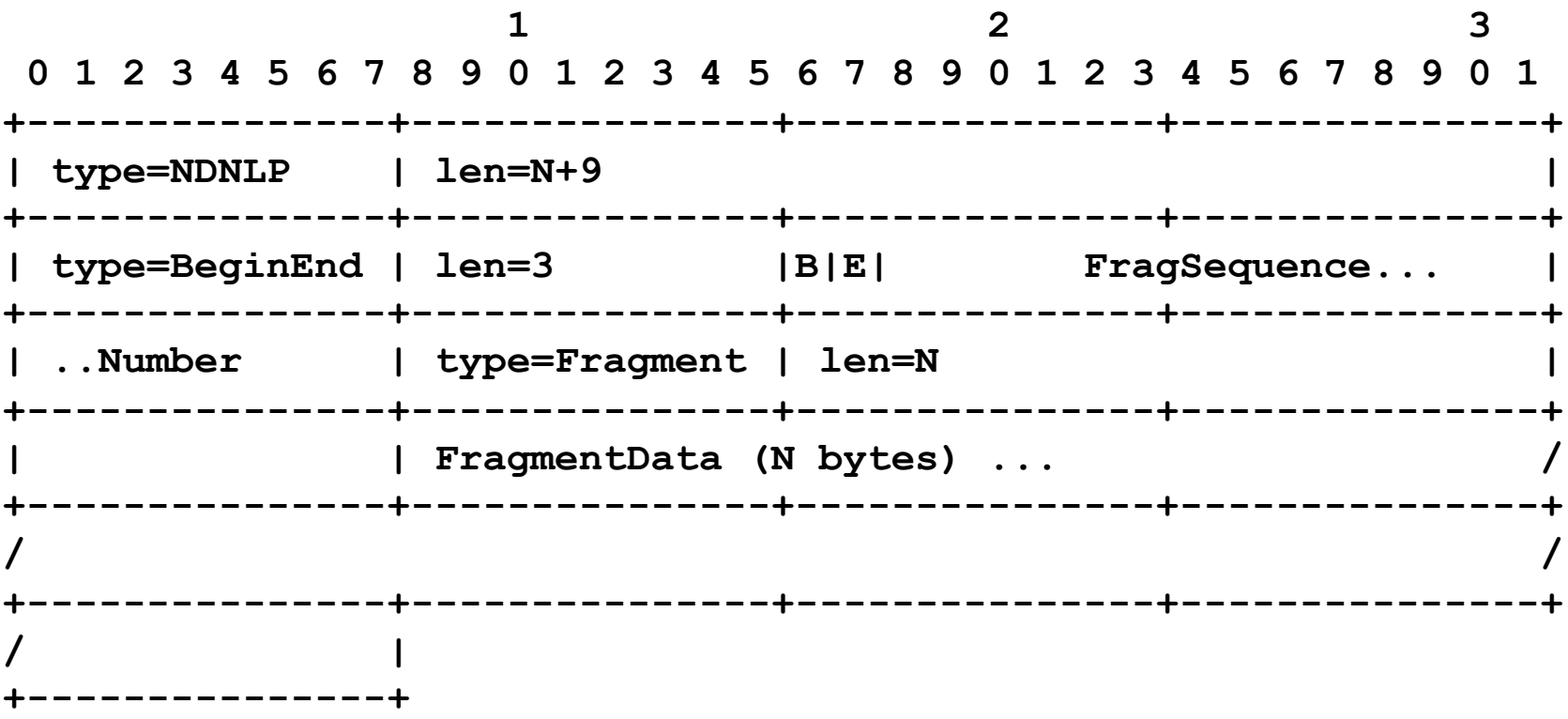
Appendix: NDN Format Specified

NDNLPv2 packets have a start type NDNLP-TYPE which distinguishes them from the classic Interest and Data packets. Inside the NDNLPv2 TLV structure, a sequence of NDNLPv2 header fields preceed the payload (fragment data) which is introduced by the type value NND-FRAGMENT-TYPE.

```
NDNpacket       ::= Interest | Data | NDNLP
NDNLP           ::= NDNLP-TYPE TLV-LENGTH
                      NDNLPhdrFields*
                      NDNLPfragment?
NDNLPhdrFields  ::= BeginEndField | (other NDNLP header fields)
BeginEndField   ::= BEGIN-END-FIELD-TYPE TLV-LENGTH
                      BYTE BYTE+
NDNfragment     ::= NDN-FRAGMENT-TYPE TLV-LENGTH
                      BYTE+
```

Appendix: NDN Example

We present an example of the basic fragment encoding for a payload of size larger than 253 Bytes and less than 64KB.



- B: Begin flag.
- E: End flag.
- FragSequenceNumber: a 22-bit sequence number to identify the fragment.