EXPERIMENTAL WIRE-SPEED PACKET FORMAT

M. Gallo, D. Perino, Z. Ben-Houidi (Bell Labs, Alcatel-Lucent), L. Muscariello (Orange Labs)

ICNRG meeting, London, March 3rd, 2014

Introduction

System design and protocol stack
- Basic CCN operations
- Enhanced in-network CCN mechanisms
- End-host socket and protocol stack

➡ efficient wire speed packet format

Fixed size header plus TLVs

Extend current format for
- Flexibility
  - Application specific data/commands
  - Network commands
- More advanced security
- Interoperability among different ICN architectures
- ...

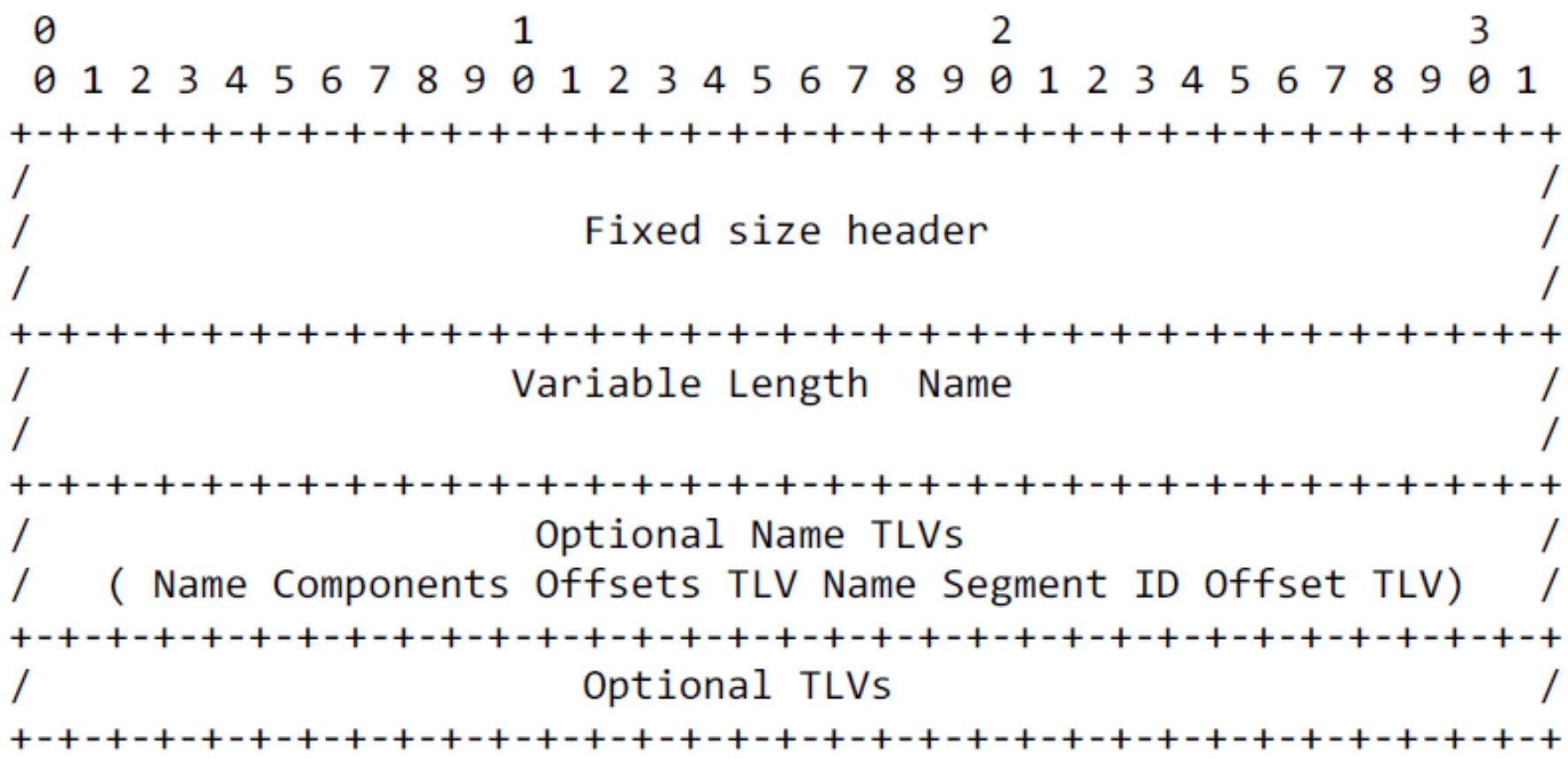
Planning to propose a draft
- Open to comments/contributions

Our Needs

Current Design

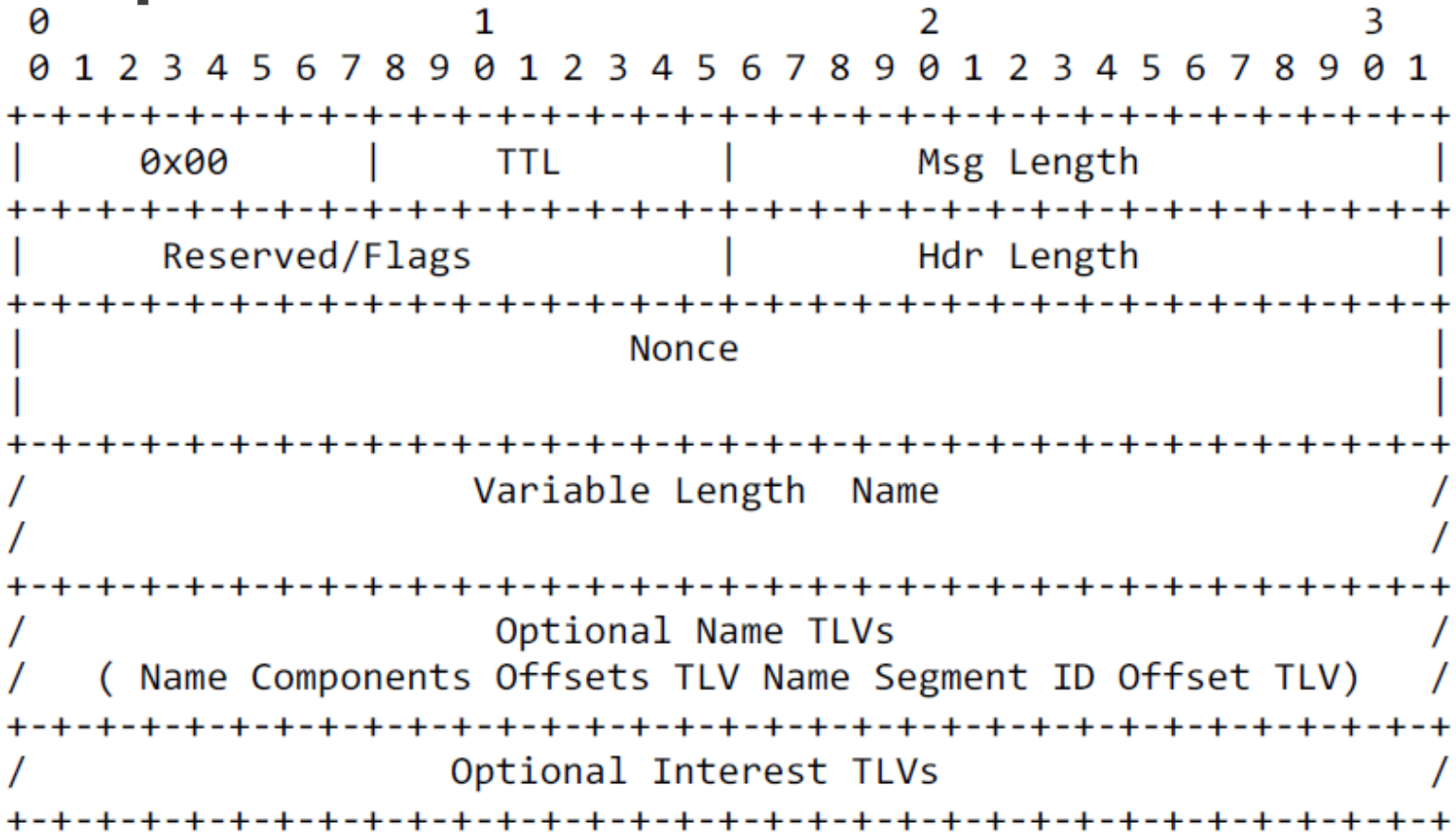
Future evolution

Generic packet format



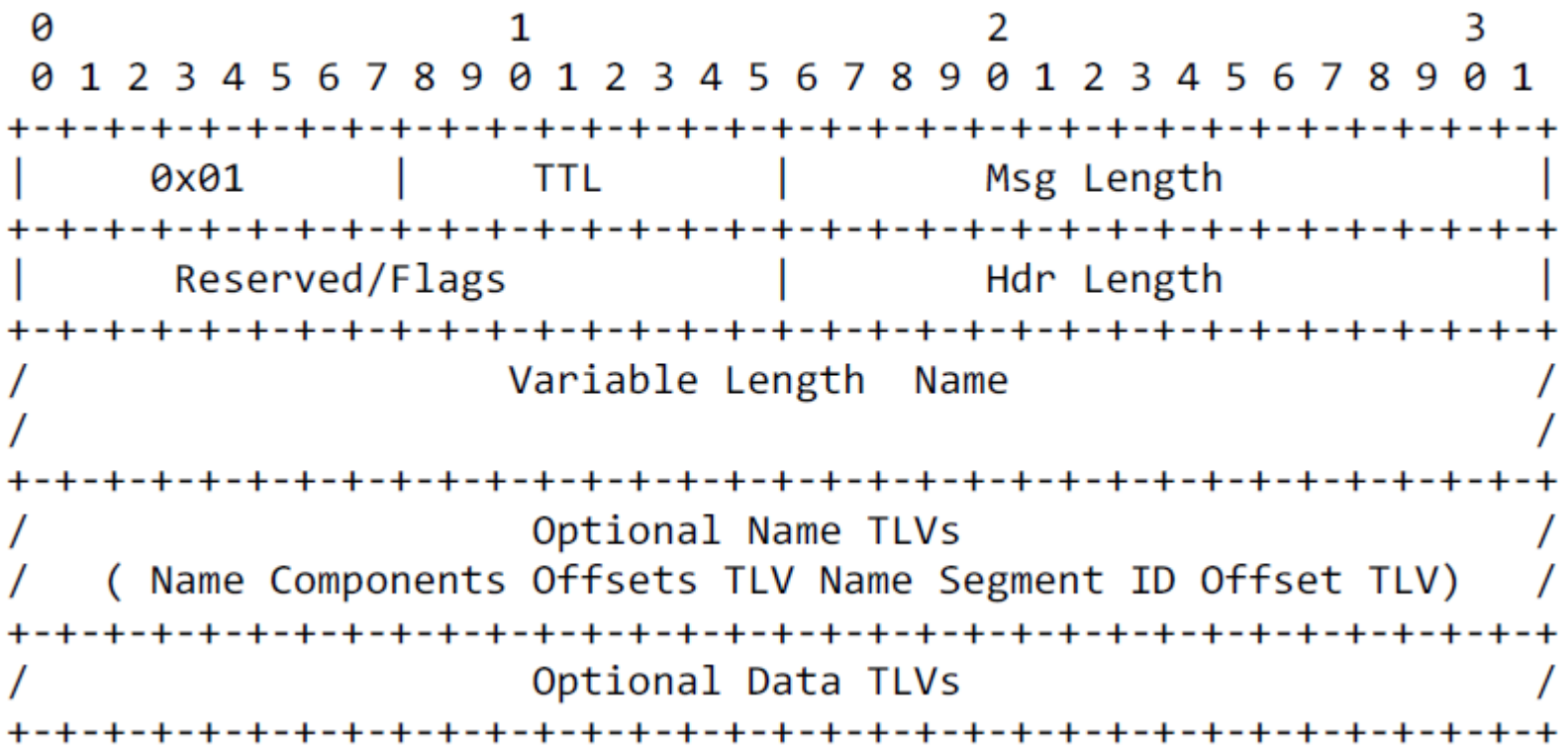
Common fixed size header for fast access to critical fields (e.g., name).
Two packet types defined so far: Interest and Data.

Interest packet format



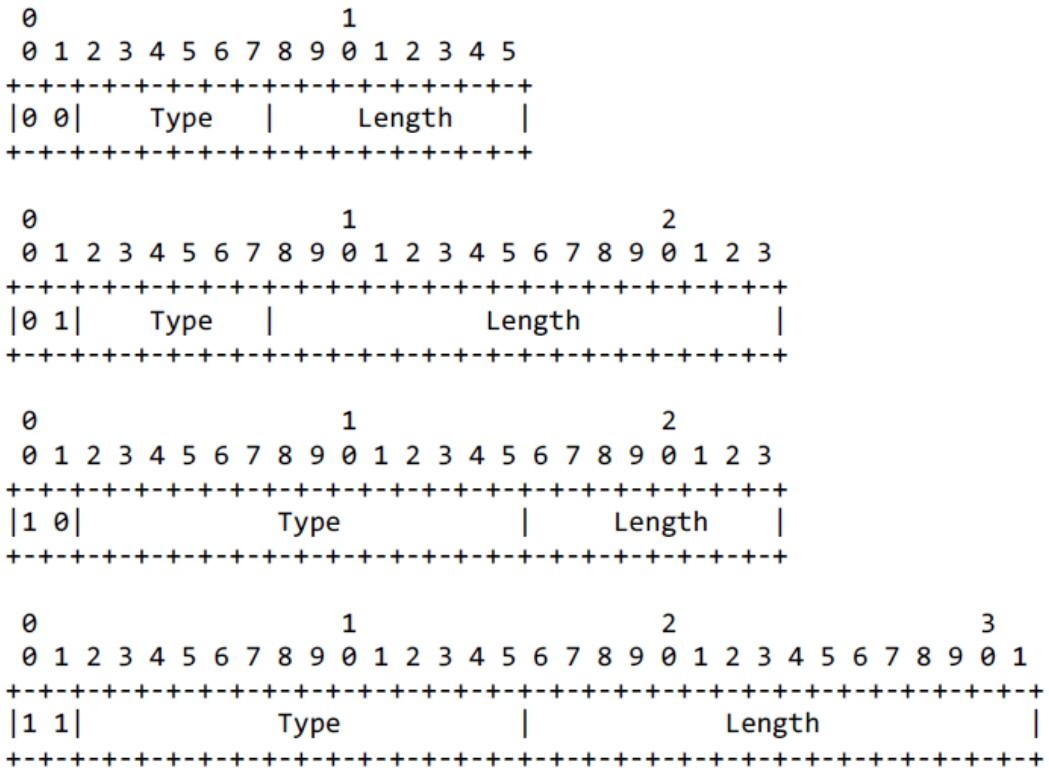
- TTL and Nonce are mandatory fields
- A TLV for the name is not required
- Versioning is currently not supported, but we could include it in the future
- Nonce can be optional in future versions with the use of a flag (this is currently under discussion)

Data packet format



- Data Packet header is the same of Interest except for the nonce field
- Fragmentation/reassembly initial thoughts:
  - They are required but currently not defined
  - Reassembly should be required for Interest (e.g. name lookup)
  - Data can be forwarded without reassembly but Data reassembly is required for caching

TLV formats

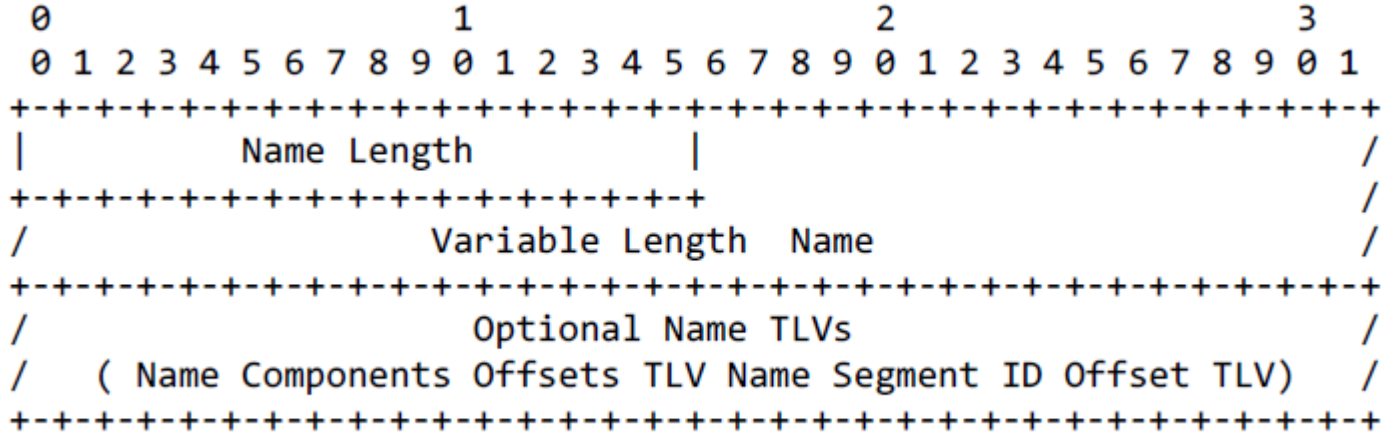


4 types of TLVs

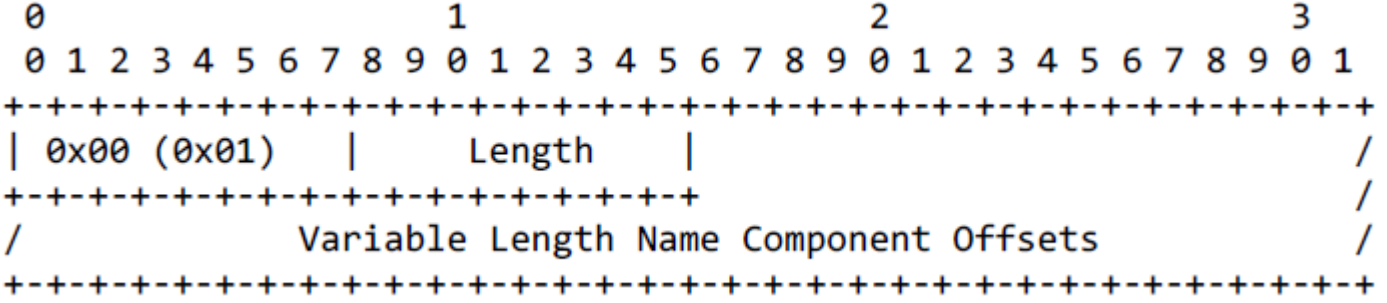
- Two length formats are required to support long Value fields (e.g. name) and to limit the overhead;
- Last two TLVs could be useful if many type values will be defined (e.g. application layer TLVs) .

Name encoding

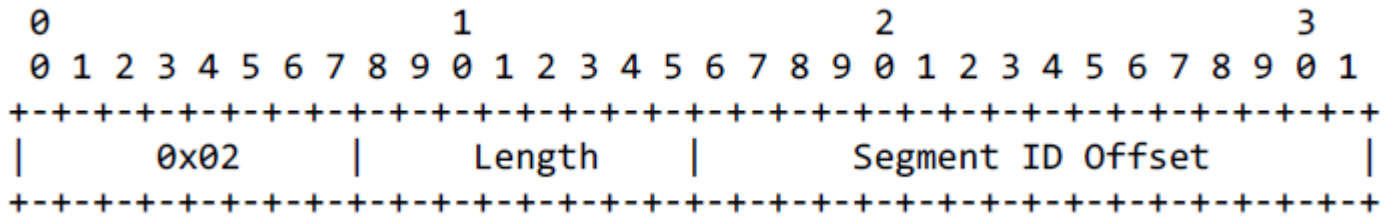
Basic fields:



Component offset TLV is defined for fast parsing (in constant time).
It is also possible to specify 2 Bytes component offsets with a different Type:



Segment ID offset:



TLV types

First Byte	Field Name	Field Description
0x00 0x40	Compact Name Components' Offset	Offsets (expressed using 1 Byte) of the components in the Name field.
0x01 0x41	Extended Name Components' Offset	Offsets (expressed using 2 Bytes) of the components in the Name field.
0x02 0x42	Name Segment ID's Offset	Offset of the segment ID in the Name field.
0x03 0x43	Interest Lifetime	Validity time for an Interest expressed in milliseconds.
0x04 0x44	Data Lifetime	Validity time for a Data Payload expressed in seconds.
0x05 0x45	Data Path ID	Identifier of the path followed by the Data packet.
0x06 0x46	Signature	Author's Signature for the Data payload
0x07 0x47	Key	The Content Object(s) that should be requested in order to retrieve the key used to sign the Data packet.

Packet format evolution and discussion

Extend the packet format to support:

- Fragmentation and reassembly
- Enhanced security
- ICN architecture interoperability
- Application specific data/commands
- Network management/processing commands
- ...

Other relevant questions:

- Applications-ICN interface definition
- ...