Enabling Elastic Content Payload in Content-Centric Networking

Aytac Azgin, Ravi Ravindran, Asit Chakraborti, and Guo-Qiang Wang

Huawei Research Center

Motivation

- **Packet format** and **how to packetize the content** is a critical challenge for CCNs with direct impact on packet processing
  - With CCN we observe higher processing overheads (e.g., parsing, hashing, validation, lookup, update, storage)
    - We have dynamically formatted names, name-based stateful forwarding, content-based security, and in-network-caching

- Packet format needs to support wide diversity of usage scenarios and network technologies[1]

[1] **ICN Packet Format Design Requirements**

Why Elastic Payload is Important?

- Performance losses experienced by CCN become a **critical limiting factor over high capacity links**
  - Larger sized content requires more chunks to be processed at content routers
  - Current solutions are not sufficiently effective
    - Carrying *inefficient encoding structures*, **or**
    - Offering *multiple T/L formats* increasing the complexity of parsing, **or**
    - Presenting *simpler* but *fixed formats* that are *inflexible* in terms of adapting to variable content responses
- **We need a flexible packet format** to insert/remove elements when necessary with minimal number of required fields, while supporting efficient packet processing[2]
  - We need the ability to also support smaller content sizes to address IoT space requirements

[2] **ICN Packet Format Design Requirements**

Why IoT Support is Important?

- **New IoT requirements at the edge cause IP to fail or have scalability/efficiency issues** [3]
- Current **state-of-the-art** architectures are **highly fragmented** and **lacks sufficient interoperability at the device, service, or application level**
  - **Lack of a unified architecture creates major concerns in regards to flexibility, scalability, mobility, and reliability**
- **Lack of a common protocol abstraction leads to inefficient communication and high cost**
- **ICN encompasses IoT requirements and is capable of providing an efficient unified architecture**

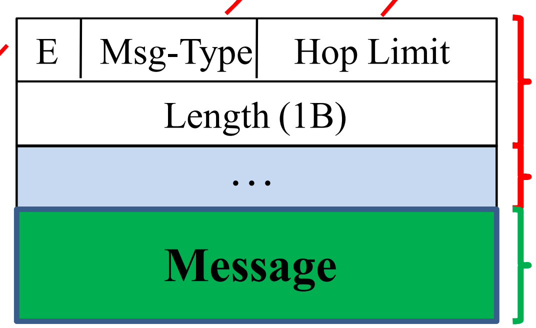
[3] **ICN Based IoT Architecture- Requirements and Challenges**

Proposed Elastic Packet Format

Proposed packet format consists of 2B long fixed header and variable length extended header

- **Message type** supporting up to **8 types** (3-bits)
- **1 generic type** reserved to support other types, for which the information is included in the TLVs as part of the extended header

- **Hop limit** supporting 15-hops (4-bits)



2B Fixed Header with 1B reserved for packet length

Extended Header (if needed)

ICN Message

- **Extended header** included or not (1-bit)?
  - **0**: Not included
  - **1**: Included

Proposed Scenario-specific Elastic Packet Formats

IoT Friendly Header

E=0	Msg-Type	Hop Limit
Length (Max:255B)		
ICN Message		

Hop-limit in an IoT setting supports mesh topologies (multi-hop transmissions triggered by asynchronous wireless transmission links)

E=1	Msg-Type	Hop Limit
Length= 0x00		
Extended Header TLV		
Optional Length TLV		
. . .		
ICN Message		

Extended Header Envelope

High Capacity Transport Friendly Header
to support payload sizes >> 64 KB

$T=T_6$ (2B)	Length = 6B

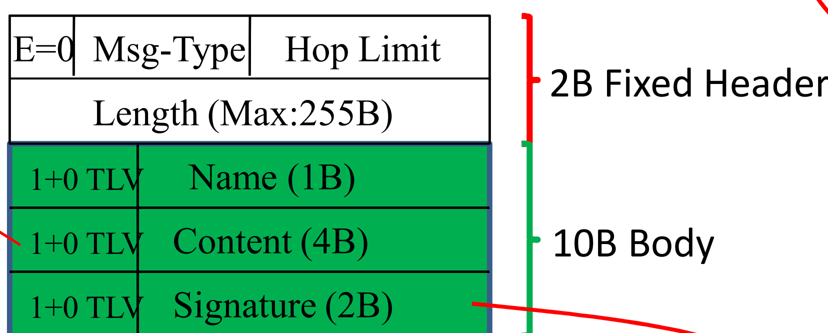
Message Length header

Example Use Cases for the Scenario-specific Elastic Packet Formats

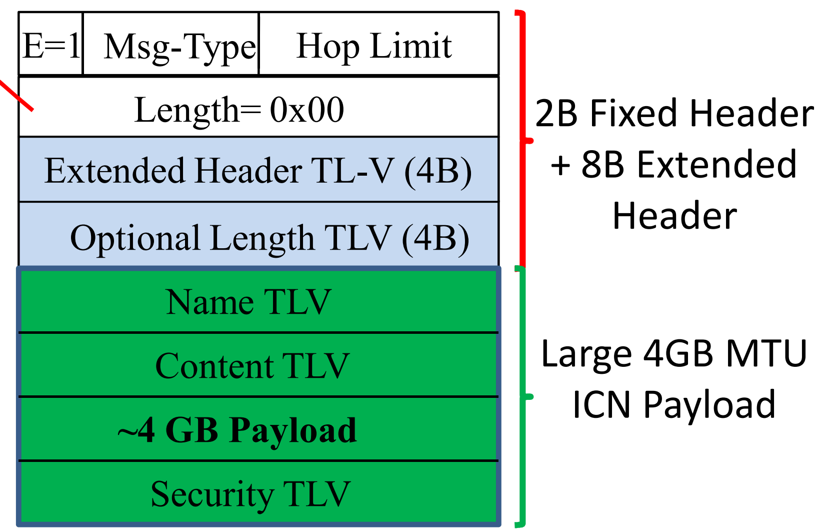
- Support header compression for constrained network
- Expand to 2+2 TLV at the gateway

- Non-zero if message length < 256
- If > 255, use the optional length TLV

12-Bytes IoT Packet



4 GB ICN Content Object



SigFox Cellular Technology[4]: Payload 12-Bytes

LoRaWAN [5]: Payload 2-255 Bytes

Bluetooth Low Energy (BLE): Data payload of 27 Bytes (Advertisement payload of 6-37 Bytes)

Can support < 128 bit output (e.g., short 32-bit or 64-bit implementations) for the **Galois/Counter Mode** (authenticated encryption with associated data) and **GMAC** (generating message authentication code on non-encrypted data) [6]

[4] http://www.sigfox.com/en/#!/

[5] Recommended SX1272 Settings for EU868 LoRaWAN Network Operation, **AN1200.23**

[6] Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, **NIST Special Publication 800-38D**

Updated 2+2 T/L Format for the Payload

- 2-Bytes **Type** field; 2-Bytes **Length** field is represented as two sub components
  - **2 bit Flag** representing the **granularity of the component (headers, payload) length**
  - **14 bit Length information**

new type	flags	block count
$T=T_8$ (2B)	2b	Length = 14 bits

2+2 T/L format supporting different granularities or block sizes, *e.g.*, 00:B, 01:KB, 10:MB, 11:GB

- Proposed format supports payload sizes of up to $\mathbf{l_\lambda \boldsymbol{x} 2^{14}}$ **Bytes**
  - l_λ represents unit size at the selected granularity level (λ)
  - If **max granularity level** represents GB units, **maximum payload** equals 2^{44} Bytes
  - Using 6 Bytes (**length type T_6**) to represent message length allows for message sizes up to 2^{48}

Minimum Required Network Support (1/3)

- **Interest probing** is used to support **path setup** for elastic content delivery

- **Two major objectives to maximize routing efficiency**
  - **Identify supported end-to-end path MTUs** *to avoid fragmentation*
  - **Select paths that meet the minimum path MTU requirements** as dictated by the consumer side application

- We introduce **interface-capability flag** and **interface marking** to support the above requirements

Minimum Required Network Support (2/3)

- **Interface-capability flag (IC-flag)**
  - ***k*-bit** flag integrated within the hop-by-hop header of (**discovery type**) Interest packets
  - IC-flag *may* indicate the following states for an interface
    - **high-capacity** (*e.g.*,>1GB)
    - **medium capacity** (*e.g.*,1MB< <1GB)
    - **low-capacity** (*e.g.*,1KB< <1MB)
    - **very low capacity** (*e.g.*,<1KB)

- **Intermediate routers can cache multiple content responses** based on supported interface capabilities
  - **PIT entries needs to include information on the IC-flag**
  - Potentially affecting state aggregation at the content routers

Minimum Required Network Support (3/3)

- **Interface marking** is used at the content routers for two main purposes
  - **Update IC-flag** within Interest packet header
  - **Select an interface with matching capability** (as requested by the consumer side application
  - **Forward Interest if matching interface found, or drop if no match is found**
    - **Active** (**NACK-driven**) or **passive** (**timeout-driven**) **signaling** is used if selected path MTU cannot be supported
    - For **multi-path probing**, **received NACKs** are only **forwarded**, if **no matching outgoing interface** is found at the host

Final Comments

- Optimistic about the future, our architecture will be helpful in the long-term
- **Fairness constraints can be managed**
  - **Latency impact** of *transmitting 150MB on 1Tbps link rate* is **equivalent to** *transmitting 1500B on 10Mbps link rate*
- **Network overhead can be reduced** by **integrating link quality metrics** into decision process to **determine optimal packet sizes**
- **Processing overhead can be reduced** as **less chunks are processed per content** (proposed architecture limits #chunks to 4)
- **Storage requirements need to be addressed** as **larger buffers are needed** to accommodate larger packet sizes
  - **Caching policies** need **to adapt to diverse content sizes** to **ensure fair access to content**