CoAP Message Transport with Packet Wash

Authors: Lijun Dong, Richard Li

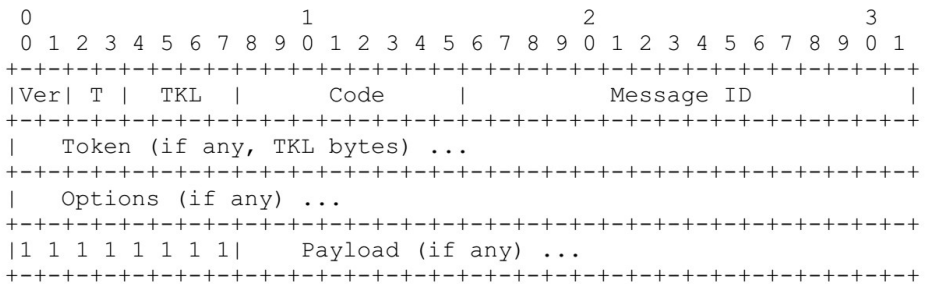
Presenter: Lijun Dong
Principle Architect,
Network Technology Lab
Futurewei Technologies Inc.

Contact: lijun.dong@futurewei.com

Basics of CoAP

- CoAP defines four types of messages:
  Confirmable, Non-confirmable, Acknowledgement, Reset.
- The CoAP messaging model is based on the exchange of messages over UDP between endpoints.
- CoAP uses a short fixed-length binary header (4 bytes) that may be followed by compact binary options and a payload.
- Reliability is provided by marking a message as Confirmable (CON). A Confirmable message is retransmitted using a default timeout and exponential back-off between retransmissions, until the recipient sends an Acknowledgement message (ACK) with the same Message ID.
- A message that does not require reliable transmission can be sent as a Non-confirmable message (NON). These are not acknowledged, but still have a Message ID for duplicate detection.

CoAP Message Format



- The message format starts with a fixed-size 4-byte header. This is followed by a variable-length Token value, which can be between 0 and 8 bytes long.
- Following the Token value comes a sequence of zero or more CoAP Options in Type-Length-Value (TLV) format, optionally followed by a payload that takes up the rest of the datagram.
  - (The field size sets a theoretical limit of **65,535 bytes**(**8 byte** header + **65,527 bytes** of data) for a UDP datagram. However, the actual limit for the data length, which is imposed by the underlying IPv4 protocol, is 65,507 bytes (65,535 − **8-byte** UDP header − **20-byte** IP header).)
  - 6LoWPAN L2 packets are limited to 127 bytes including various overheads); this may motivate implementations to be frugal in their packet sizes and to move to **block-wise** transfers.
- Type (T): 2-bit unsigned integer. Indicates if this message is of type Confirmable (0), Non-confirmable (1), Acknowledgement (2), or Reset (3).
- Token Length (TKL): 4-bit unsigned integer. Indicates the length of the variable-length Token field (0-8 bytes).
- Code: 8-bit unsigned integer, split into a 3-bit class (most significant bits) and a 5-bit detail (least significant bits)
- Message ID: 16-bit unsigned integer in network byte order. Used to detect message duplication and to match messages of type Acknowledgement/Reset to messages of type Confirmable/Nonconfirmable.

FUTUREWEI Technologies

Options

No.	C	U	N	R	Name	Format	Length	Default
1	x			x	If-Match	opaque	0-8	(none)
3	x	x	-		Uri-Host	string	1-255	(see below)
4				x	ETag	opaque	1-8	(none)
5	x				If-None-Match	empty	0	(none)
7	x	x	-		Uri-Port	uint	0-2	(see below)
8				x	Location-Path	string	0-255	(none)
11	x	x	-	x	Uri-Path	string	0-255	(none)
12					Content-Format	uint	0-2	(none)
14		x	-		Max-Age	uint	0-4	60
15	x	x	-	x	Uri-Query	string	0-255	(none)
17	x				Accept	uint	0-2	(none)
20				x	Location-Query	string	0-255	(none)
35	x	x	-		Proxy-Uri	string	1-1034	(none)
39	x	x	-		Proxy-Scheme	string	1-255	(none)
60			x		Sizel	uint	0-4	(none)

C=Critical, U=Unsafe, N=NoCacheKey, R=Repeatable

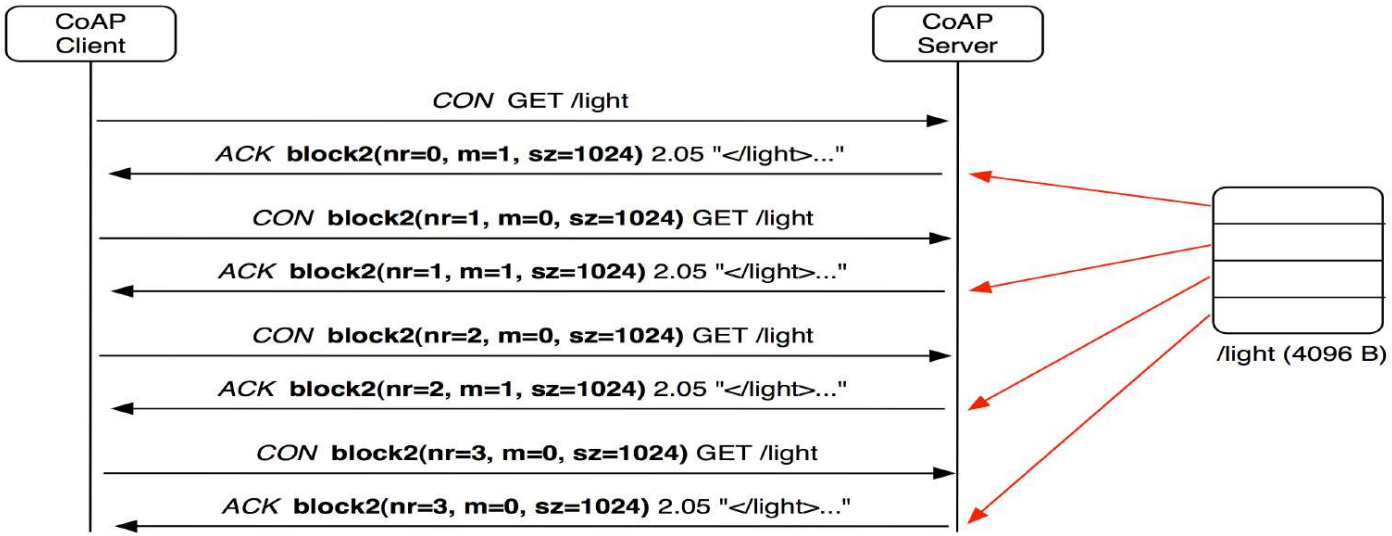
Media type	Encoding	ID
text/plain; charset=utf-8	-	0
application/link-format	-	40
application/xml	-	41
application/octet-stream	-	42
application/exi	-	47
application/json	-	50

- The Uri-Host Option specifies the Internet host of the resource being requested,
- The Uri-Port Option specifies the transport-layer port number of the resource,
- Each Uri-Path Option specifies one segment of the absolute path to the resource
- Each Uri-Query Option specifies one argument parameterizing the resource.
- The Proxy-Uri Option is used to make a request to a forward-proxy. The forward-proxy is requested to forward the request or service it from a valid cache and return the response. The option value is an absolute-URI.
- The Content-Format Option indicates the representation format of the message payload. The representation format is given as a numeric Content-Format identifier that is defined in the "CoAP ContentFormats" registry.

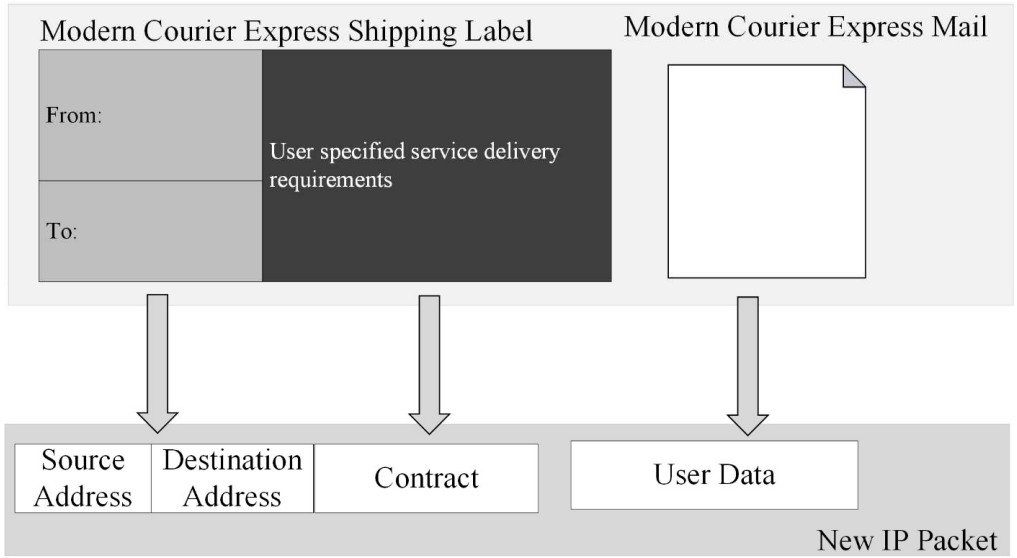
Options(Continue)

- The CoAP Accept option can be used to indicate which Content-Format is acceptable to the client.
- The Max-Age Option indicates the maximum time a response may be cached before it is considered not fresh.
- An entity-tag(ETag) is intended for use as a resource-local identifier for differentiating between representations of the same resource that vary over time. The Etag option can be included in the request or response.
- The Location-Path and Location-Query Options together indicate a relative URI that consists either of an absolute path, a query string, or both.
- Conditional request options enable a client to ask the server to perform the request only if certain conditions specified by the option are fulfilled.
  - The If-Match Option MAY be used to make a request conditional on the current existence or value of an ETag for one or more representations of the target resource.
  - The If-None-Match Option MAY be used to make a request conditional on the nonexistence of the target resource.
  - The Size1 option provides size information about the resource representation in a request.

Block Transfer



New IP Packet and its Contract Component



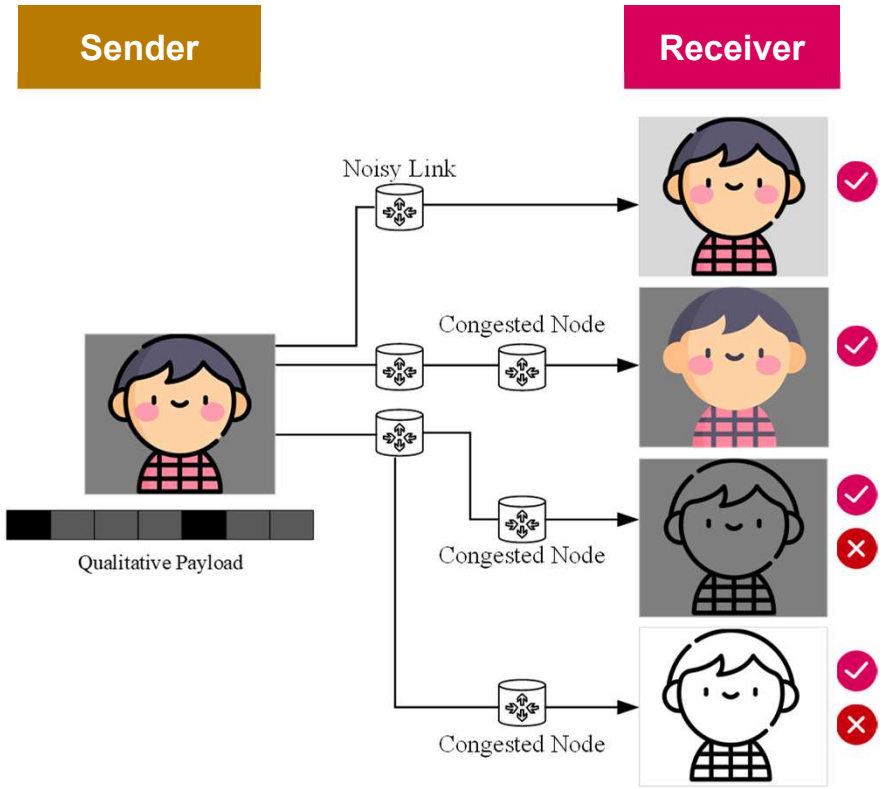
<Contract> := <Contract Clause> | <Contract Clause> AND <Contract>

<Contract Clause> := <Contract Item> | <Contract Item> OR <Contract Clause>

<Contract Item> := <Event, Condition, Action> | <Metadata> | <Event, Condition, Action> <Metadata>

<Event, Condition, Action> := <Action> | <Condition> <Action> | <Event> <Condition> <Action>

Qualitative Communication: A Structure of Bits and Bytes



Ref: A Framework for Qualitative Communications using Big Packet Protocol, ACM Sigcomm 2019 NEAT Workshop, Beijing, August 19, 2019. Available at: https://dl.acm.org/citation.cfm?id=3342201

Semantics

What is received = What is *maximally meant*

In payload, bits and bytes are not equally significant. Instead, they are different in their entropies

Less significant bits and bytes may be dropped

Partial or degraded, yet useful, packets may be repaired and recovered before being rendered

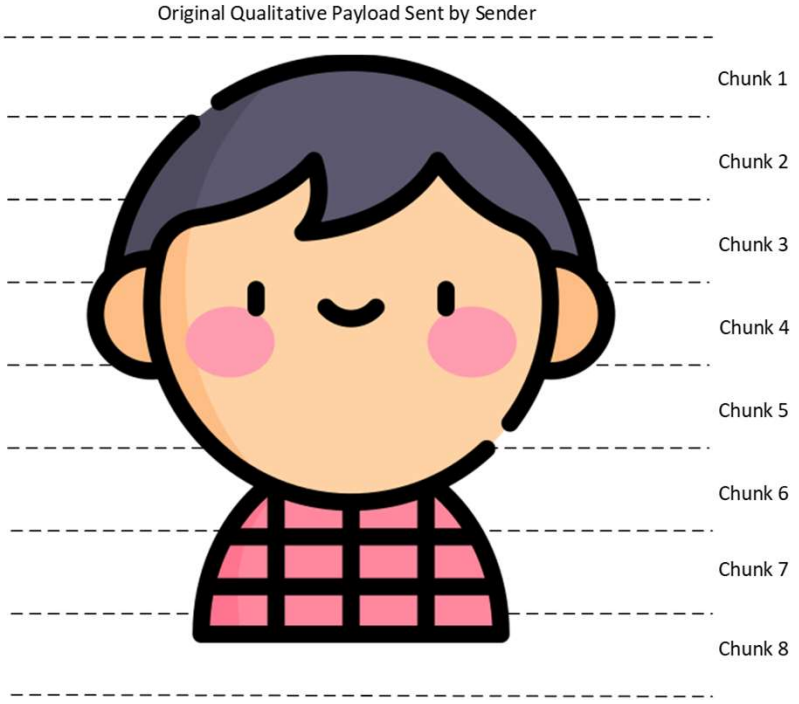
Good for

- Large volume of image-like data
- Holographic type communications
- Media with digital senses
- Disaster Environment

FUTUREWEI Technologies

Qualitative Communications: an Example for Illustration Only

User Data Payload: Divided into 8 chunks.



Contract:

Event: Congested **OR** Radio Unstable
Action: Packet Wash
Meta-Data: Chunk 1, Chunk 2, chunk 6, Chunk 7, Chunk 8

On congested node: after cut

Packet Wash is preferred to drop-and-retransmit

FUTUREWEI Technologies

Benefits Provided by Packet Wash

- Retransmission may not be needed if the receiving node has the capability or intelligence to deduce the entire information from what is left in the packet after packet washing of the original packet by the network nodes.

- The network resources usage can be tremendously reduced and the network resources can be better prioritized for other packet delivery.

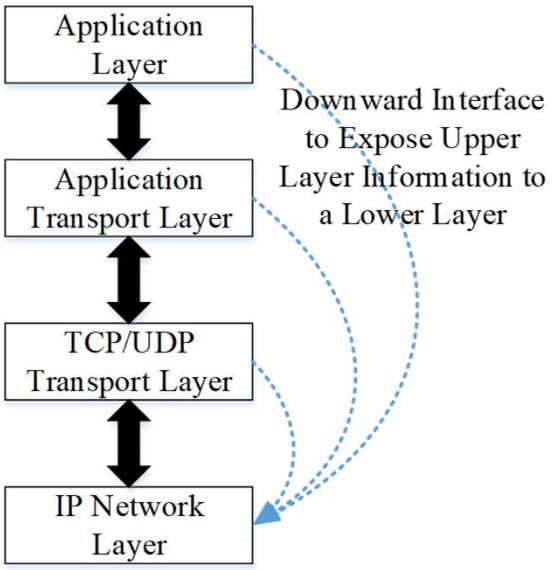
- The latency of transmitting the packet can be significantly reduced due to no retransmission, smaller packet size after packet washing.

Summary of the Paper

- Proposal of New IP Metadata with very limited overhead in order to enable in-network packet wash on CoAP messages (mainly messages with payload).
- Proposed actions on encrypted payload.
- Proposed actions on block-wise transfer payload, which is unique to CoAP protocol.
- Proposed actions on unencrypted payload.
- Proposal of CoAP header and options compression.

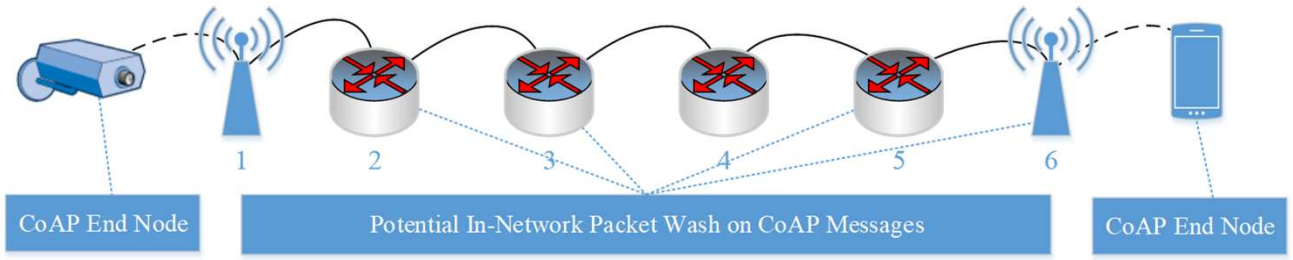
Cross-Layer Design for Internet Protocol Stack

- In principle, one layer is not be able to access the fields in the header of another layer, either it is upper layer or lower layer. The methodology of layered protocol design possesses some advantages from the protocol transparency perspective. For example, protocols in one layer can be designed, improved, or even substituted without imposing any influence on other protocol layers.
- However, it is likely that the information from one layer may be useful to another layer. As a result, the performance optimization between different protocol layers becomes impossible under such methodology of layered protocol design, which can significantly degrade the network performance. This unavoidably leads to the cross-layer design.
- The concept of cross layer design is about sharing of information among different protocol layers for adaptation purposes and to increase the inter-layer interactions.



CoAP Message Type Illustrated in the Paper

- Due to the simplicity and RESTFUL nature of CoAP protocols, the most common CoAP messages that are sent between two end nodes are the combinations of: [Confirmable, Non-Confirmable, Acknowledgement, Reset]+[GET, PUT, POST, DELETE]+[Request, Response].
- Since the only difference between Confirmable and Non-Confirmable messages is that whether the message is acknowledged by the recipient upon arrival. Thus, we don't distinguish them in the invention. Among those messages, the GET response, POST request, PUT request may contain data in the payload, which the invention will focus on.
- Given the GET response, POST request, PUT request messages have similar syntax, depending on whether it is initiated by the client or server, in the invention, we use the GET response message as the example to illustrate how a CoAP message may be washed by the in-network network nodes. Other types of CoAP messages can be treated similarly.



FUTUREWEI Technologies

In-Network Packet Wash on the GET Response Message

- In order to facilitate the in-network packet wash on the GET response message, we propose that in the corresponding GET request message, the requesting end node's application layer will optionally pass the following parameters to the BPP metadata field:
  - Type of requested data that corresponds to the application (e.g., surveillance video, temperature)
  - Latency budget for returned data
  - Required quality of returned data (e.g., for video data, view angle, resolution etc.)
  - Tolerance degree on the distortion of the data quality

- When packet wash capable network node receives a GET response packet, it may make decisions on revising the packet, depending on the current network condition. The actions it may take include:
  - Compress the CoAP header
  - Cache the payload data and remove the payload from the packet
  - Modify/compress the payload data if it is not encrypted

New IP Metadata Design To Enable In-Network Packet Wash on CoAP GET Response Message

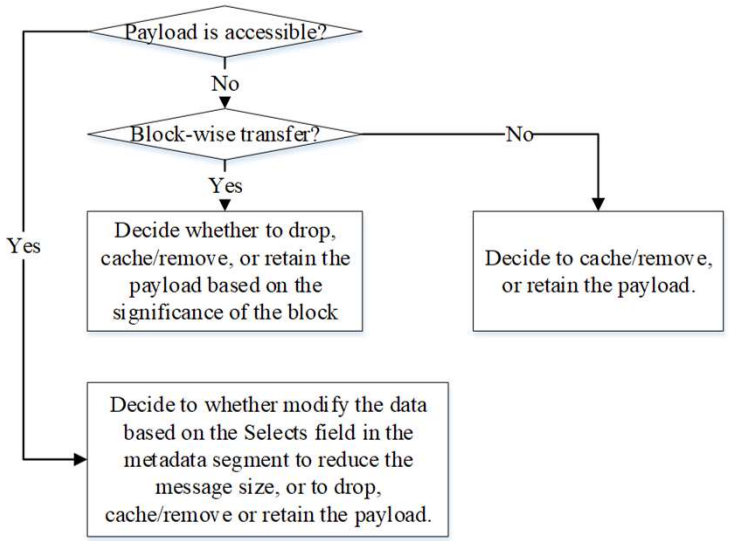
- *Status* (3 bits): It indicates whether the packet has been washed by the previous network nodes which have forwarded the packet. The very first bit indicates whether the packet is original or not . The second bit indicates whether the packet's CoAP header is compressed or not . The third bit indicates whether the packet's payload is modified or not (including the actions such as the payload is completely dropped, or the unencrypted payload is revised) (101). Any combination of the 2 types of revisions to the original packet could exist by setting the corresponding bit to 1 and results in the first bit to be 1.
- *CoAPWash* (2 bits)*:* The first bit indicates whether that the *TKL*, *MessagID* and *Tokens* fields are removed. The second bit indicates whether the *content-format*, *max-age*, and *ETag* options are removed and cached.
- *Multi* (1 bit): It indicates whether there are simultaneous requests between the requesting end node and responding end node. If the bit is set, the *Tokens* field must be included in the CoAP header and cannot be removed. If the bit is not set, a network node may remove the *TKL* and *Tokens* field in the CoAP header. On the other hand, if there is only one request between the two end nodes, the *Message ID* field can also be removed since the requesting end node can match the response to the request by the responding end node's address.
- *TagCache* (1-8 bytes): The local identifier of the payload and other information cached by a network node, which can be used by the requesting node to retrieve the payload later when the network condition becomes satisfactory.
- *Significance* (1-8 bits): If the message is one block in the block-wise transfer, this field can be used to indicate the significance of the block in recovering or interpreting the original data. A network node can use this field to decide whether the payload in the message may be dropped.

New IP Metadata Design (continue)

- *Selects* (length varies): This field is used to include any possible requirements from the requesting node, as well as properties related to the actual data being returned by the responding node.
  - The *Type-of-Data* lets the network nodes understand the type of data included in the payload, such that it may use the corresponding algorithm, tool to process the data.
  - The *Tolerance-Degree* option following the previous option is to indicate the requesting end node could have some level of acceptance if the data is not exactly matching the requirement it set up. This Tolerance-Degree could give the flexibility to the network node to compress/revise the payload to fit the current network condition, given that the payload is not encrypted or the network node is authorized to access the payload.

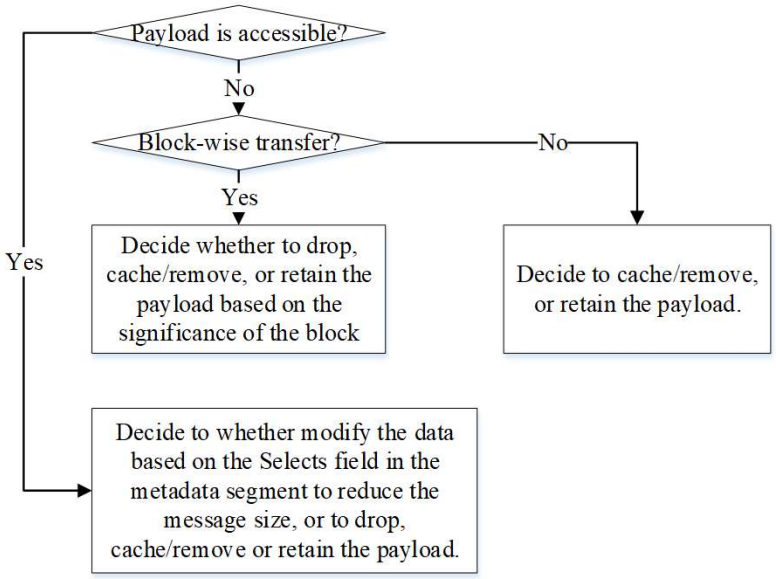
No.	Name	Format	Length
1	Type-of-Data	string	0-1
2	Latency-Budget	unit	0-1
3	Tolerance-Degree	unit	0-1
4	View-Angle	unit	0-2
5	Tolerance-Degree	unit	0-1
6	Resolution	unit	1
7	Tolerance-Degree	unit	0-1

Action on Encrypted Payload



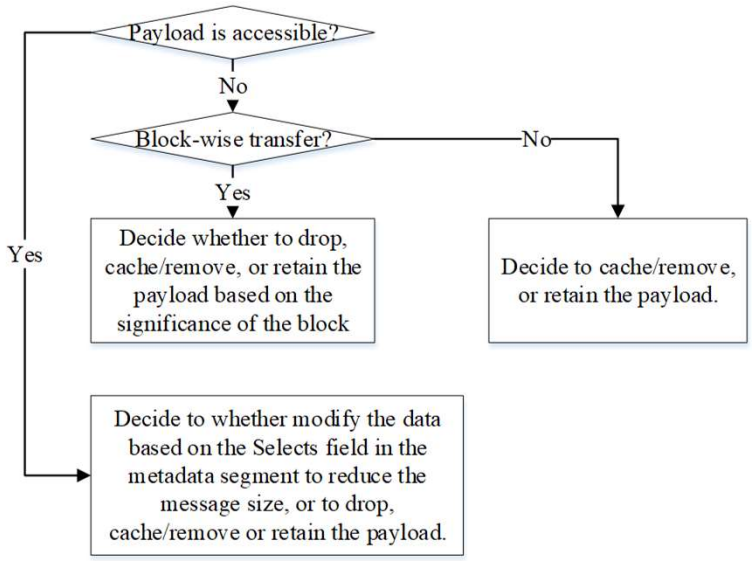
- If the payload is encrypted, the data itself is invisible to the network node and cannot be processed in any method.
- However, the network node can decide to whether to cache the payload and other associating information in the CoAP header/remove it from the original message.
- If the network node caches the payload and other associating information in the CoAP header (e.g., Content-Format, Max-Age, ETag) and removes the payload from the original message, the network node will set the very first bit in the *Status* field in the metadata segment to 1 to indicate such action.
- On the other hand, the network node can set the *TagCache* field in the metadata segment to include its local identifier of the cached content for the requesting node to retrieve the data when the network condition becomes better.
- If the *TagCache* is not setup after the network node drops the payload from the original message, it indicates the network node is charge of sending the data contained in the payload to the requesting node after it sees the network condition becomes better. The requesting node only needs to expect and wait for the data to be delivered later.

Action on Block-Wise Transfer Payload



- If the message is one block in the block-wise transfer, we propose that the responding node may organize the blocks such that each of the blocks may contain different parts of the requested data with different importance.
- In the *Significance* field of the metadata segment in the message, it will indicate the relative importance of the current block compared to the other blocks. In the Block2 option, the *SZX* field shows the number of blocks in total. We define that the *Significance* could take the value of [1, *SZX*]. The larger the relative *Significance* value is, the more important the block is to recover the information contained in the data.
- The network node could decide whether to drop the block given that the network condition is not satisfactory to transport the block to the next hop. If the block is relatively important, the network node needs to try its best to retain the payload and send to the requesting node. Otherwise, the payload could be dropped, removed/cached for later retrieval or delivery.

Action on Unencrypted Payload



- If the payload is not encrypted, which means the data may be visible to the network node, or the network nodes are given the permission to access the data, the network nodes can revise the data based on the network condition, user's requirement and application layer parameters included in the metadata segment of the message.
- The *Selects* field is used to specify those proposed requirements, which could be set up by the requesting node when the CoAP request message is sent.
- The responding node prepares the data according to the requesting node's requirements specified in the *Selects* field. When the response message is sent, the actual properties of the data (e.g. *Type-of-Data*, *Resolution*, *View-Angle*) are setup accordingly. The related *Tolerance-Degree* is copied from the request message. On the other hand, the *Latency-Budget* is modified for the response message to reach the requesting node by deducting the used time from the original latency budget.
- Based on the *Tolerance-Degree*, the data contained in the payload may be revised to reduce the message size. For example, the resolution may be adapted by the local transcoder enabled at the network node, such that it could be lower than the current value, but higher than the requesting node's tolerable value.

CoAP Header Compression

- The CoAP header of a GET response packet could be possibly compressed for reduced size, under the condition that some of the fields in the header may be removed without influencing the processing or understanding of the CoAP header at the receiver side.
- In the CoAP header, the *TKL*, *MessagID* and *Tokens* field can be removed if there are no simultaneous requests between the requesting end node and responding end node, which is indicated in the *Mutli* field in the metadata by the responding end node.
- The CoAP options that could be included in a GET response message are: *Observe*, *Content-Format*, *Max-Age* and *ETag*.
  - The *Observe* Option indicates that this is a notification for the subscription. This option should not be removed.
  - The *Content-Format* Option indicates that the representation format of the message payload. The representation format is given as a numeric content format identifier that is defined in the "CoAP Content-Formats" registry. It should not be removed if the payload stays in the message.
  - The *Max-Age* Option indicates that the maximum time a response may be cached before it is considered not fresh. It should not be removed if the payload stays in the message.
  - The *ETag* Option is generated by the responding node and used as a resource-local identifier for differentiating between representations of the same resource that vary over time. The *ETag* Option in a response provides the current value of the entity-tag for the requested resource representation in the payload. The *ETag* Option can be removed, resulting in that the requesting node is not aware of the entity-tag of the received data, which is not crucial information in interpreting the representation.
- After the CoAP header is processed and compressed according to the algorithm proposed above, the first two bits in the *CoAPWash* field in metadata segment of the message is set up accordingly and the first and second bit in the *Status* field is set to “11”. For the future network nodes, they would not act on the CoAP header anymore.

CoAP Header Compression Algorithm

Algorithm 1 CoAP Header Compression
If the first two bits of Status is "11", **then**
    The CoAP header has been compressed and no further actions
    needed on the CoAP header.
else if the second bit of Status is "0", **then**
    if $multi = 0$, **then**
        remove TKL, MessageID and Tokens field if they exist.
        Set the first bit of *CoAPWash* to "1".
    end
    if payload is removed from the message and cached, **then**
        Remove content-format, max-age, and ETag options from the message
        and cache content-format, max-age and ETag along with the payload.
        Set the second bit of *CoAPWash* to "1".
    else
        Retain content-format and max-age options, remove ETag option.
    end
        Set the first and second bit in the *Status* field is set to "11".
end

Optional UDP Header and IP Header Compression

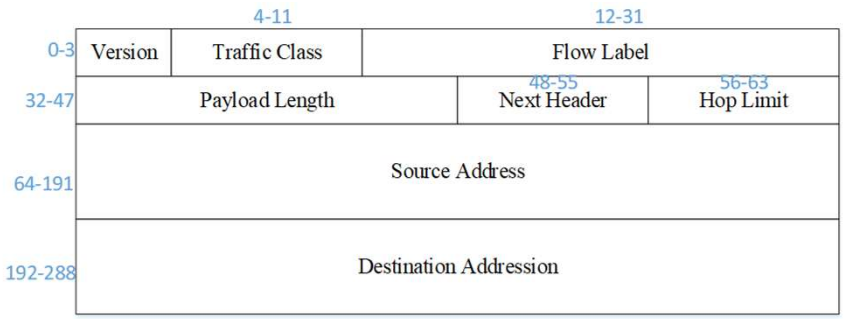
- *Status* (6 bits): It indicates whether the packet has been washed by the previous network nodes which have forwarded the packet. The very first bit indicates whether the packet is original or not (000000). The second bit indicates whether the packet's CoAP header is compressed or not (110000). The third bit indicates whether the packet's UDP header is compressed or not (101000). The fourth bit indicates whether the packet's IP header is compressed or not (100100). The fifth bit indicates whether the packet's payload is compressed or not (100010). The sixth bit indicates whether the packet's payload is modified or not (100001). Any combination of the 5 types of revisions to the original packet could exist by setting the corresponding bit to 1.
- UDPWash (4 bits): The first bit indicates whether the *source port* is compressed to 4 bits. The second bit indicates whether the *destination port* is compressed to 4 bits. The third bit indicates whether the *Length* field is removed. The fourth bit indicates whether the *Checksum* field is removed.
- IPWash (6 bits): The first bit indicates whether the *Traffic Class* field is removed. The second bit indicates whether the *Flow Label* is removed. The third field indicates that the *Payload Length* field is removed. The fourth bit indicates whether the *Next Header* is removed and cached. The fifth bit indicates whether the *Hop Limit* field is removed and cached. The six bit indicates whether the Source Address field and Destination Address fields are removed.
- IPExtCache: Each bit indicates whether the extension header is removed and cached, following the sequence as shown in TABLE III. The first bit indicates whether the Hop-by-Hop extension header is removed and cached. The second bit indicates whether the Destination extension header is removed and cached. The first Destination extension header is supposed to be processed by the first and subsequent destinations. The third bit indicates whether whether Routing extension header is removed and cached. The fourth bit indicates whether the Fragment extension header is removed and cached. The fifth bit indicates whether the Authentication extension header is removed and cached. The sixth bit indicates whether the Encapsulating Security Payload extension header is removed and cached. The seventh bit indicates whether the Destination extension header is removed and cached. This Destination extension header is supposed to be processed by Final Destination.

UDP Header Compression

- UDP (source and destination) ports may be compressed to 4 bits, if the requesting and responding nodes agree to only use 16 number of specified ports for different applications which are using CoAP as the application layer protocol. Other than the source port and destination port, there are two other fields in the UDP header:
  - Length: It indicates the length in bytes of the UDP header and the encapsulated data. The minimum value for this field is 8. This field can be removed without influencing packet interpretation.
  - Checksum: This is computed as the 16-bit one's complement of the one's complement sum of a pseudo header of information from the IP header, the UDP header, and the data, padded as needed with zero bytes at the end to make a multiple of two bytes. If the checksum is set to zero, then checksuming is disabled. If the computed checksum is zero, then this field must be set to 0xFFFF. Since a network node may revise the IP header, UDP header and the payload based on the mechanisms proposed in the invention, the Checksum field needs to be disabled if any modifications happen to the original packet, thus is removed (in this paper, removing Checksum field means disabling the checksuming).
- After a network node processes the UDP header, the *UDPWash* field in the metadata segment is setup correspondingly.

IP Header Compression

- Version (4-bits): It represents the version of Internet protocol, i.e. 0110, which can be fixed, thus can be removed.
- Traffic Class (8-bits): These 8 bits are divided into two parts. The most significant 6 bits are used for Type of Service to let the Router know what services should be provided to this packet. The least significant 2 bits are used for Explicit Congestion Notification (ECN). For a CoAP message, if this field is set up by the source, it should be not removed and the routers need to take actions according to the Type of Service and Explicit Congestion Notification to forward the message. Otherwise, this field can be removed.
- Flow Label (20-bits): This label is used to maintain the sequential flow of the packets belonging to a communication. The source labels the sequence to help the router identify that a specific packet belonging to a particular flow of information. This field helps avoid re-ordering of data packets. Since the data can be carried in one CoAP message or multiple ones with block-wise transfer, the Flow Label is not necessary for the requesting node to recover the data accurately. The Block2 option contains the order information of the blocks. Thus the Flow Label field can be removed.
- Payload Length (16-bits): This field is used to tell the routers how much information a particular packet contains in its payload. Payload is composed of Extension Headers and Upper Layer data. The field needs to be changed if the upper layer compression and payload modification happen. On the other hand, we can also consider to remove this field for IP header length reduction.



IP Header Compression

- Next Header (8-bits): This field is used to indicate whether the type of Extension Header is present.
- If the message is one block in the block-wise transfer and if it is the first block as indicated in the Block2 option in the CoAP header, then the network node caches those Extension Headers and keep them in the IPv6 header.
- If the message is one block in the block-wise transfer and if it is the later blocks other than the first block, then the network node compares the Extension Headers with the cached ones associated with the first block.
- If there is an exact match, then it will be indicated in the IPExtCache field in the metadata segment that the type of Extension Header exist in the current IPv6 header, which does not need to be transported in the network, but only needs to be extracted from the cached copy in each of the intermediate network nodes.
- Here we assume that the blocks in the block-wise transfer share the same values for the Extension Headers if they are included in the IPv6 header. This proposal also applies to the Hop Limit, Source Address and Destination Address.

IPv6 header
Hop-by-Hop Options header
Destination Options header
Routing header
Fragment header
Authentication header
Encapsulating Security Payload header
Destination Options header
Upper-layer header

Thank You!

Contact: lijun.dong@futurewei.com