# Options for certificate installation and retrieval

G. Kramer

# Current solution in D2.2

- Two new OAMPDU types are defined for managing certificates
- Action codes are defined to
  - 0x00 - Install NAC (or chain)
  - 0x01 - Retrieve DAC
  - 0x02 - Retrieve NAC (or chain)

**Table 13-10—eOAMPDUs and assignment of Opcode values**

| Opcode | eOAMPDUs | Defined in |
|---|---|---|
| 0x01 | *eOAM_Get_Request* | 13.4.6.2 |
| 0x02 | *eOAM_Get_Response* | 13.4.6.3 |
| 0x03 | *eOAM_Set_Request* | 13.4.6.4 |
| 0x04 | *eOAM_Set_Response* | 13.4.6.5 |
| 0x09 | *eOAM_Software* | 13.4.6.6 |
| 0x0A | *eOAM_Certificate_Request* | 13.4.6.7 |
| 0x0B | *eOAM_Certificate_Response* | 13.4.6.7 |

## Install NAC (request)

| Size (B) | Field name | Value and notes |
|---|---|---|
| 21 | eOAMPDU header | See Table 13-2 |
| 1 | Opcode | 0x0A |
| 1 | Action Code | 0x00 |
| 2 | Certificate Length | The length of the Certificate field.<br>The value of 0x00 indicates that this is a request to remove the existing NAC certificate |
| ≤ 1489 | Certificate Data | NAC certificate data.<br>This field is not present if the `CertificateLength` is 0x00. |
| ≤ 35 | Pad | |
| 4 | FCS | |

## Install NAC (response)

| Size (B) | Field name | Value and notes |
|---|---|---|
| 21 | eOAMPDU header | See Table 13-2 |
| 1 | Opcode | 0x0B |
| 1 | Action Code | 0x00 |
| 1 | Action Status | Value encoding the status of a taken/attempted action, as defined in Table 13-24 |
| 1 | Certificate Status | Value encoding the status of the installed certificate, as defined in Table 13-25 |
| 35 | Pad | 0x00-…-00 |
| 4 | FCS | |

## Retrieve NAC/DAC (request)

| Size (B) | Field name | Value and notes |
|---|---|---|
| 21 | eOAMPDU header | See Table 13-2 |
| 1 | Opcode | 0x0A |
| 1 | Action Code | `0x01`: retrieve the DAC<br>`0x02`: retrieve the NAC. |
| 37 | Pad | 0x00-…-00 |
| 4 | FCS | |

## Retrieve NAC/DAC (response)

| Size (B) | Field name | Value and notes |
|---|---|---|
| 21 | eOAMPDU header | See Table 13-2 |
| 1 | Opcode | 0x0B |
| 1 | Action Code | `0x01`: DAC certificate<br>`0x02`: NAC certificate |
| 2 | Certificate Length | The length of the Certificate field.<br>The value of 0x00 indicates that the requested certificate (NAC or DAC) is not present or cannot be retrieved. |
| ≤ 1489 | Certificate Data | DAC or NAC certificate data.<br>This field is not present if the `CertificateLength` is 0x00. |
| ≤ 35 | Pad | |
| 4 | FCS | See 13.4.2 |

If NAC comes with intermediate certificates, the entire certificate chain must fit within this field

# Questions

- During the August meeting, we discussed the constraint on the NAC certificate chain to not exceed 1489 bytes. If we are to remove the above constraint, should we
  - **A)** Allow the certificate chain to be larger than 1489 bytes, while each individual certificate (DAC, NAC, intermediates) is under 1489 bytes?

  or
  - **B)** Allow each individual certificate to be larger than 1489 bytes?

- The current **eOAM_Certificate_Request** and **eOAM_Certificate_Response** eOAMPDUs are not TLV based. They use predefined fixed fields. Is it better
  - **1)** To use these specialized OAMPDU formats ?

  or
  - **2)** To use common **Set_Request/Set_Response** to install certificates and **Get_Request/Get_Response** to retrieve certificates?

**Overall, 4 possibilities: A1, B1, A2, B2**

# Method A1

- Redefine the **ActionCode** to allow each certificate in a certificate chain to be carried in a separate OAMPDU.

| ActionCode[7-0] | |
|---|---|
| **Bit[7] - Action** | **Bits[6-0] – Certificate Code** |
| 0 - Install | **0x00** – NAC |
| 1 - Retrieve | **0x01…0x04** – Intermediate Certs. |
| | **0x05…0x7E** – Reserved |
| | **0x7F** – DAC |

**Definition in D2.2**

| Action | ActionCode value |
|---|---|
| Install NAC | 0x00 |
| Retrieve DAC | 0x01 |
| Retrieve NAC | 0x02 |

**New definition**

| Action | ActionCode value |
|---|---|
| Install NAC | 0x00 |
| Install IC1 | 0x01 |
| Install IC2 | 0x02 |
| Install IC3 | 0x03 |
| Install IC4 | 0x04 |
| reserved | 0x05…0x7E |
| Retrieve NAC | 0x80 |
| Retrieve IC1 | 0x81 |
| Retrieve IC2 | 0x82 |
| Retrieve IC3 | 0x83 |
| Retrieve IC4 | 0x84 |
| reserved | 0x85…0xFE |
| Retrieve DAC | 0xFF |

**Note:** ActionCode 0x7F (Install DAC) is disallowed.

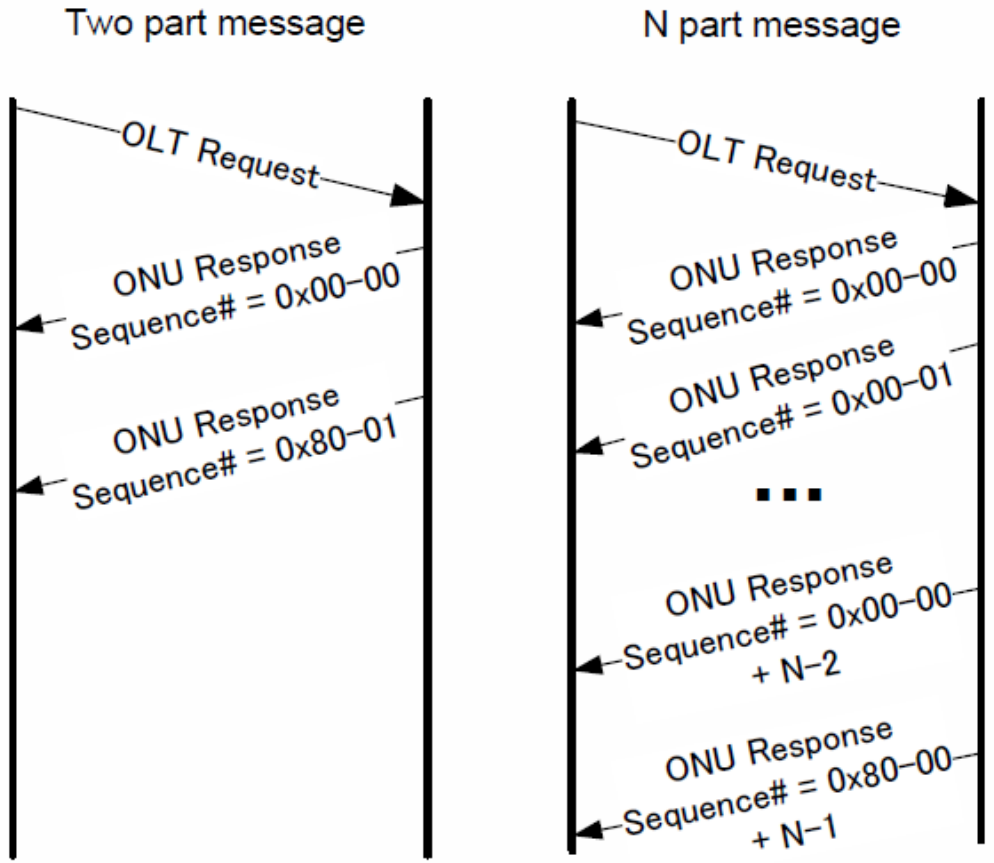
The above table includes 4 code points for intermediate certificates (ICn).
**Is this enough?**

- Each certificate in the chain is installed and confirmed individually
  - In case of installation failure, the OLT re-installs only the failed certificate
- Each individual certificate is not to exceed 1489 bytes
- This method requires small draft changes.

# Method B1 (1/3)

- If we allow certificate size > 1489 bytes, then such certificate must be carried in multiple OAMPDUs.
- The existing multi-PDU mechanism is TLV-based and is defined only for ONU's **Get_Response** and **Set_Response** OAMPDUs:

  "To indicate that additional eOAMPDUs comprising a complete response from the ONU are forthcoming, the ONU shall add an instance of the ***Sequence* TLV** (0xDB/0x00-01) to the response eOAMPDU to denote the response sequence."




"To send a multiple part response requiring N eOAMPDUs, the ONU does the following:

- For the first eOAMPDU in the response sequence, set the value in the Sequence# field to 0x00.
- For the last eOAMPDU in the response sequence, set bit 15 in the Sequence# field to 1.
- For all eOAMPDUs in the response sequence, increment the value of the Sequence# field."

# Method B1 (2/3)

- Allow each certificate or certificate chain to use multiple OAMPDUs
- Add "**Sequence**" field to **Install NAC request** and **Retrieve DAC/NAC response** OAMPDUs

**Install NAC (request)**

| Size (B) | Field name | Value and notes |
|---|---|---|
| 21 | eOAMPDU header | See Table 13-2 |
| 1 | Opcode | 0x0A |
| 1 | Action Code | 0x00 |
| 2 | Sequence | |
| 2 | Certificate Length | The length of the Certificate field.<br>The value of 0x00 indicates that this is a request to remove the existing NAC certificate |
| ≤ 1487 | Certificate Data | NAC certificate data. This field is not present if the `CertificateLength` is 0x00. |
| ≤ 35 | Pad | |
| 4 | FCS | |

**Install NAC (response)**

| Size (B) | Field name | Value and notes |
|---|---|---|
| 21 | eOAMPDU header | See Table 13-2 |
| 1 | Opcode | 0x0B |
| 1 | Action Code | 0x00 |
| 1 | Action Status | Value encoding the status of a taken/attempted action, as defined in Table 13-24 |
| 1 | Certi… Status | Value encoding the status of …e installed certificate, as …in Table 13-25 |
| 35 | Pad | 0x00-…-… |
| 4 | FCS | |

**Retrieve NAC/DAC (request)**

| Size (B) | Field name | Value and notes |
|---|---|---|
| 21 | eOAMPDU header | See Table 13-2 |
| 1 | Opcode | 0x0A |
| 1 | Action Code | `0x01`: retrieve the DAC<br>`0x02`: retrieve the NAC. |
| 37 | Pad | 0x00-…-00 |
| 4 | FCS | |

**Retrieve NAC/DAC (response)**

| Size (B) | Field name | Value and notes |
|---|---|---|
| 21 | eOAMPDU header | See Table 13-2 |
| 1 | Opcode | 0x0B |
| 1 | Action Code | `0x01`: DAC certificate<br>`0x02`: NAC certificate |
| 2 | Sequence | |
| 2 | Certificate Length | The length of the Certificate field.<br>The value of 0x00 indicates that the requested certificate (NAC or DAC) is not present or cannot be retrieved. |
| ≤ 1487 | Certificate Data | DAC or NAC certificate data. This field is not present if the `CertificateLength` is 0x00. |
| ≤ 35 | Pad | |
| 4 | FCS | See 13.4.2 |

- This method also requires small and simple draft changes.

- In a single OAMPDU, Sequence = 0x80-00
- In multi-PDU message, Sequence increments from 0 to N.
- In the last OAMPDU of a multi-PDU message, Sequence = 0x80-00 + N

# Method B1 (3/3)

- How large the "**Sequence**" field needs to be?

- In **Sequence TLV**, the **SequenceNumber** field is 15 bits wide
  - Max. attribute size = $2^{15} \times 128$ (B) $\approx$ 4.2 (MB)

- If Sequence field is also 15 bits wide, then
  - Max. certificate (chain) size = $2^{15} \times 1487$ (B) $\approx$ 48.7 (MB)
- If Sequence field is reduced to 7 bits, then
  - Max. certificate (chain) size = $2^{7} \times 1488$ (B) $\approx$ 190.5 (KB)

- Do we expect the total size of a certificate chain to ever approach 190 KB?
- What is a practical size of secure non-volatile storage (trust store) in the ONU?

# Method B1+ (= A1 + B1)

- Add "**Sequence**" field as in method B1,
  and
  redefine **ActionCode** field as in method A1.

- Each certificate in the chain is installed and confirmed individually
  - In case of installation failure, the OLT re-installs only the failed certificate
- Each individual certificate may exceed 1488 bytes
- Each certificate exceeding 1488 bytes will use multiple OAMPDUs

**New ActionCode definition**

| Action | ActionCode value |
|---|---|
| Install NAC | 0x00 |
| Install IC1 | 0x01 |
| Install IC2 | 0x02 |
| Install IC3 | 0x03 |
| Install IC4 | 0x04 |
| reserved | 0x05…0x7E |
| Retrieve NAC | 0x80 |
| Retrieve IC1 | 0x81 |
| Retrieve IC2 | 0x82 |
| Retrieve IC3 | 0x83 |
| Retrieve IC4 | 0x84 |
| reserved | 0x85…0xFE |
| Retrieve DAC | 0xFF |

- This method is less efficient than B1, i.e., it will require more OAMPDUs in case of certificate chain.
- Requires special logic in the ONU to construct certificate chain from individual certificates received from the OLT.

# Method A2

- Each certificate does not exceed 1436 (!) bytes, plus using common **Set_Request/ Set_Response** and **Get_Request/Get_Response**
- The certificates are treated as any other OAM attributes
  - Certificates are installed using Set_Request/Set_Response
  - Certificates are retrieved using Get_Request/Get_Response

- Variable Container TLV payload size is limited to 128 bytes. TLVs can carry large certificates by breaking them into 128-byte chunks and adding the terminating TLV at the end.

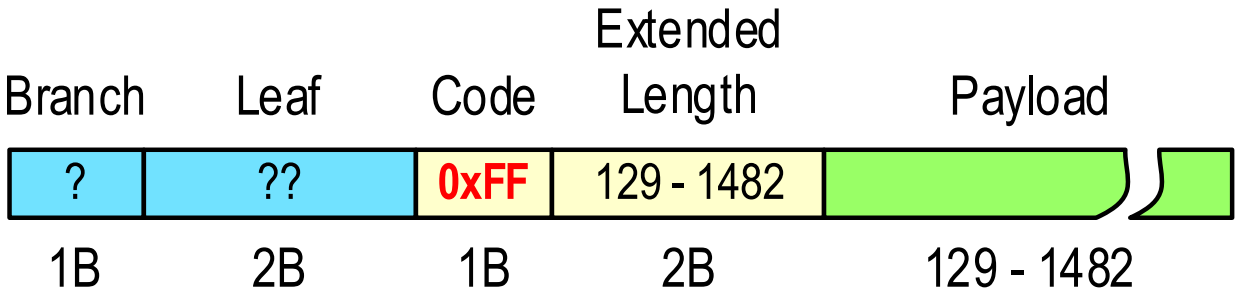
- Accounting for the TLV overhead (4B/132B), one OAMPDU can accommodate a 1436-byte certificate spread over 12 TLVs (the last TLV is shortened)

**13.4.3.3 TLVs carrying large values**

The maximum length of data that can fit into a single Variable Container is equal to 128 octets. Some attribute values may be larger than the 128 octets, requiring a series of TLVs to transfer them between the source OAM client and the target OAM client, using a repeated branch/leaf tuple for the attribute in question. Such a series of TLVs is terminated with a TLV with the same branch/leaf tuple, and a length of zero, to indicate the end of multi-TLV value.

# Method A2+ (large TLV)

- The efficiency can be somewhat improved if we define a new “large” TLV format, such that the entire OAMPDU payload can be just one TLV.




Add new code point

| Code | Name |
|---|---|
| 0x80 | No Error |
| 0x81 | Too Long |
| 0x86 | Bad Parameters |
| 0x87 | No Resources |
| 0x88 | System Busy |
| 0xA0 | Undetermined Error |
| 0xA1 | Unsupported Attribute/Action |
| 0xA2 | May Be Corrupted |
| 0xA3 | Hardware Failure |
| 0xA4 | Overflow |
| 0xA5 | Invalid Context Object |
| 0xFF | Extended Length TLV |

- Maximum TLV size that fits into one OAMPDU (assuming 4-byte object context TLV) is 1488 bytes (payload = 1482 bytes)

- The “large” TLV may be useful for other large attributes as well, but existing systems are implemented without it. It may be confusing to allow different TLV formats for the same attributes. Is it worth adding a new option now?

# Method B2

- Each certificate may exceed 1436 bytes, plus using common **Set_Request/ Set_Response** and **Get_Request/Get_Response** OAMPDUs

- **<u>Certificate retrieval:</u>**
  - If method A2 or A2+ is implemented, then the existing eOAM spec already supports retrieving large certificates using multi-PDU messages with the help of **Sequence TLV**.
- **<u>Certificate Installation:</u>**
  - In current OAM, the **Sequence TLV** is only allowed in **Get_Response** and **Set_Response** OAMPDUs.
  - To install large certificates, the **Sequence TLV** must be allowed for **Set_Request** OAMPDU as well.
  - The ONU shall wait for the complete **Set_Request** sequence before sending **Set_Response**.

# Summary of required changes (against D2.2)

| | **A)**<br>**Each certificate must fit in single OAMPDU (Single-PDU messages)** | **B)**<br>**Certificates have no size constraints (multi-PDU messages)** |
|---|---|---|
| **1)**<br>**Use certificate-specific OAMPDUs (as in D2.2)** | 1. Redefine the **ActionCode** field to identify DAC, NAC, and intermediate certificates | 1. Add **Sequence** field to **Install NAC request** and **Retrieve DAC/NAC response** OAMPDUs<br>2. **"Plus" option:** Redefine the **ActionCode** field as in A1 |
| **2)**<br>**Revert to common Set_Request/ Set_Response and Get_Request/ Get_Response** | 1. Delete subclause ***13.4.6.7 eOAM_Certificate_Request and eOAM_Certificate_Response eOAMPDUs***<br>2. Define new OAM attributes/TLVs for DAC, NAC, and intermediate certificates<br>3. **"Plus" option**: define a new "large" TLV format, such that the entire OAMPDU payload can be just one TLV. | 1. As A2 or A2+, and …<br>2. Re-write subclause ***13.4.5 Multipart eOAMPDU response sequence*** to allow the **Sequence** TLV in **Set_Request,** in addition to **Get_Response** |