

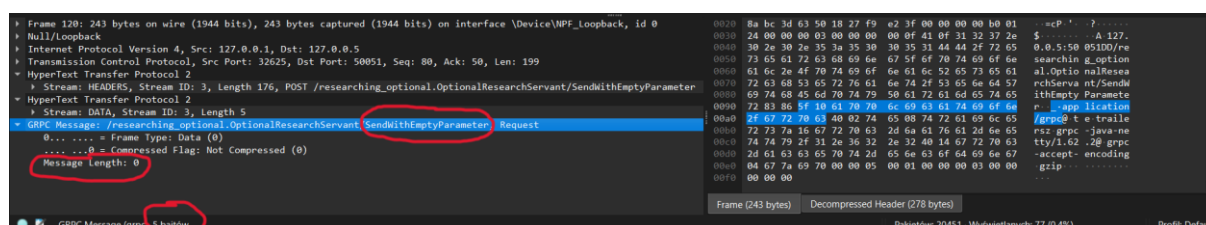
# Zadanie I.4 - Opcjonalne pola struktur danych i argumenty wywołania *middleware*

## 1 gRPC

gRPC korzysta z Protobufa jako formatu wymiany danych, dlatego analiza gRPC mocno pokrywa się z analizą sposobu działania samego Protobufa.

Zaznaczę jeszcze na wstępie, że korzystam ze składni Protobuf3.

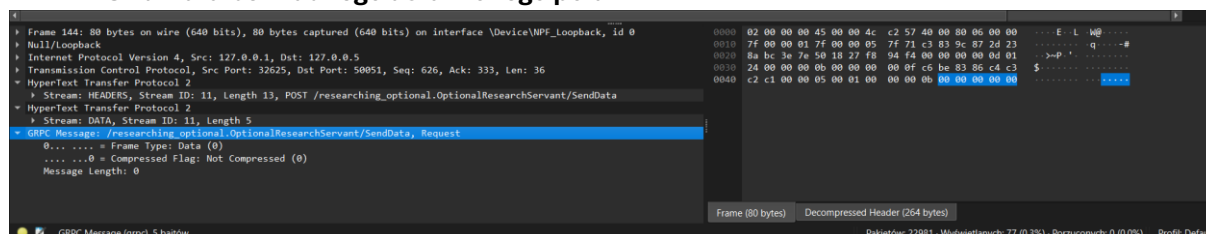
Podczas analizy gRPC najbardziej będą nas interesowały pewne części zrzutów ekranu. Zawarte są tam m.in. nazwa wywołanej usługi, zawartość wiadomości gRPC oraz zawartość struktur w tej wiadomości (HTTP2#DataStream#payload):



### 1.1 OPCJONALNOŚĆ PÓL STRUKTUR DANYCH

W protobufie3 domyślnie wszystkie pola są opcjonalne. Twórcy doszli do wniosku, że pola „required” przynoszą więcej szkody niż pożytku, więc uniemożliwili specyfikowanie pól jako obowiązkowe. Tym samym wszystkie pola automatycznie stały się opcjonalne. (jednak słówko kluczowe „optional” pozostało, ale o tym później).

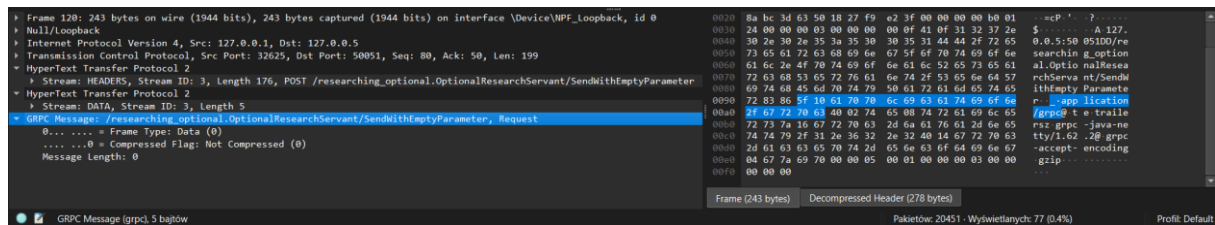
#### 1.1.1 Struktura bez żadnego ustawionego pola



Taka struktura zajmuje 0B, ale gRPC message wciąż zajmuje 5B (= 1B dla Frame Type oraz Compressed Flag + 4B dla Message Length).

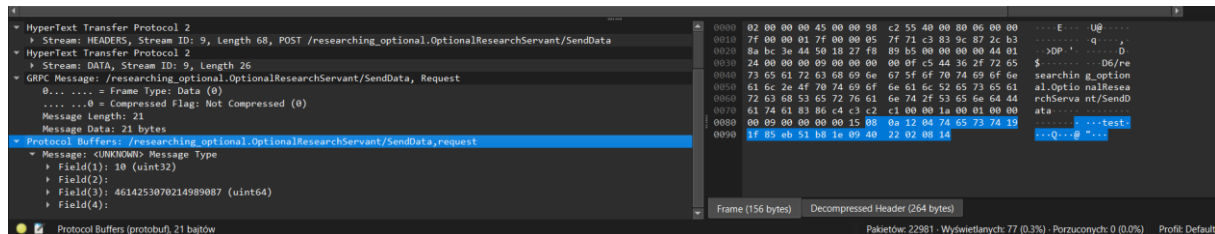
#### 1.1.2 Struktura w ogóle bez żadnego możliwego pola

Mowa o strukturze „message Empty {}”.



Sytuacja analogiczna jak w przypadku struktury z polami, ale bez żadnego ustawionego pola.

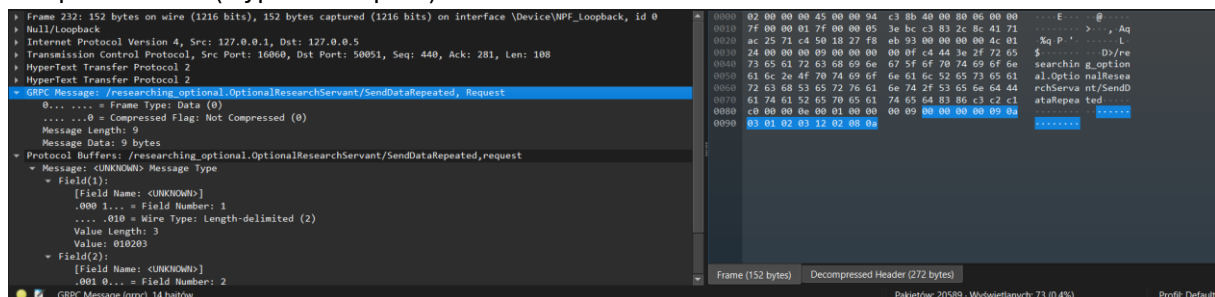
Dla porównania to samo wywołanie, ale z w petni uzupełnioną strukturą. Zajętość jest oczywiście większa:



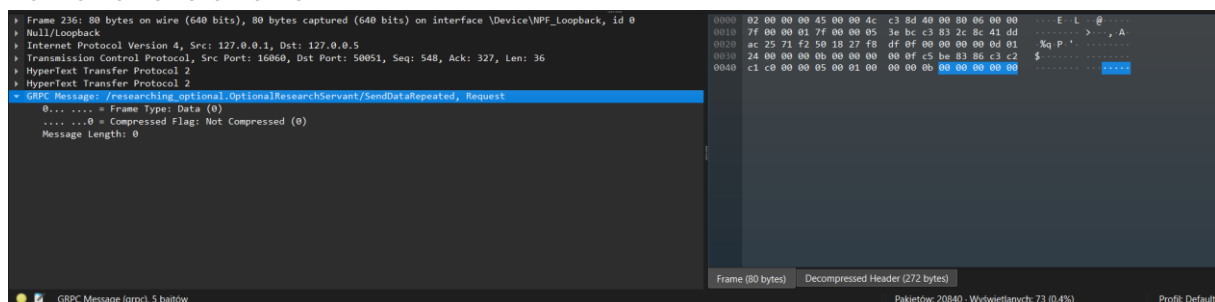
### 1.1.3 Pola „repeated”

Pola typu „repeated” także można, w pewnym sposób, traktować jako pola opcjonalne (a w sumie jako powtórzenie danego pola od zero- do wielokrotnie).

Dla porównania (wypełnione pola):



Powtórzenie zerokrotne:

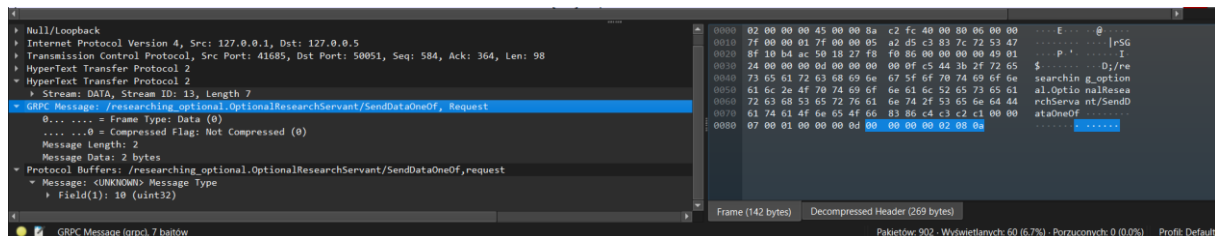


Jak widać, powtórzenie zerokrotne jest równoważne niedefiniowaniu żadnego pola.

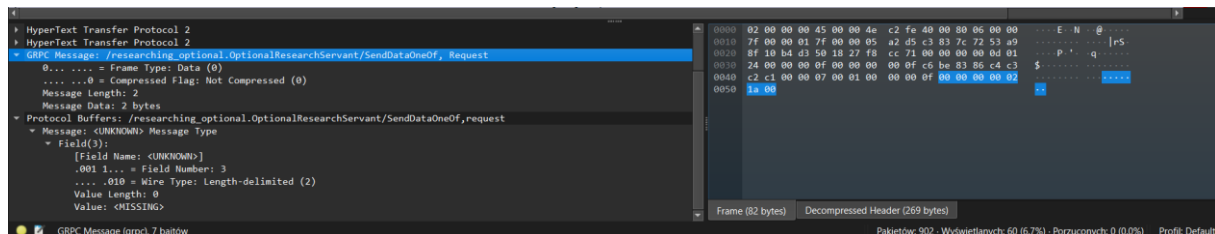
### 1.1.4 One of

Słowo kluczowe “one of” daje możliwość zdefiniowania co najwyżej jednego pola spośród wymienionych. Co istotne, użycie „one of” umożliwia zbadanie, czy dane pole zostało ustawione explicite przez klienta, czy znajduje się tam domyślna wartość.

Ustawiony int:

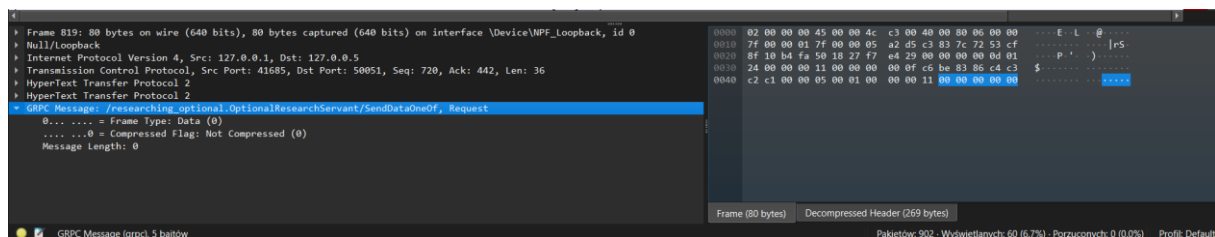


Ustawiony Empty:



**Ważne!** Empty zajmuje w tym wypadku miejsce!

Nic nie ustawione:



Wyniki metod `hasSomething()` dla trzech przedstawionych powyżej przykładów

```
==> all
sendDataOneofWithInt
response.hasInt32(): true
response.hasData(): false
response.hasEmpty(): false

sendDataOneofWithEmptyValue
response.hasInt32(): false
response.hasData(): false
response.hasEmpty(): true

sendDataOneofWithUnsetValue
response.hasInt32(): false
response.hasData(): false
response.hasEmpty(): false
```

Jak widać, dzięki użyciu „one of” jesteśmy w stanie rozróżnić te trzy sytuacje. Nawet sytuacje, kiedy ustawiamy strukturę Empty oraz kiedy nie ustawiamy nic.

### 1.1.5 “Optional” jawne vs “optional” domyślne

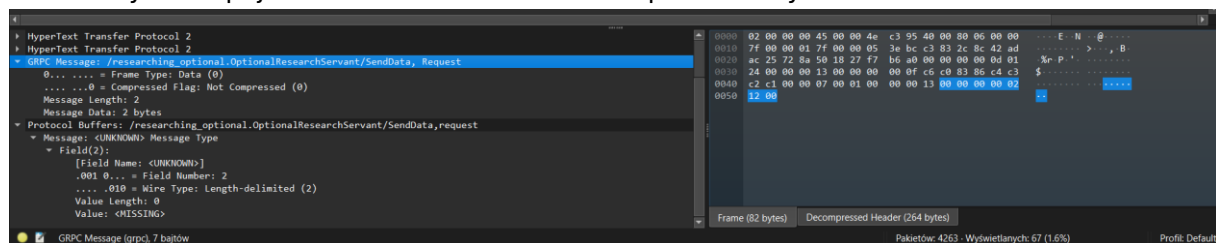
Pomimo tego, że Protobuf3 wyeliminował słowo „required” a wszystkie pola są opcjonalne, to wciąż istnieje słowo kluczowe „optional”. Dlaczego? Ponieważ jest różnica między domyślną opcjonalnością a jawną opcjonalnością. Chociaż w sumie okazuje się, że to zależy od konkretnego języka, bo w wielu jest inaczej.

Na ten przykład, w Javie wywołanie `DataExplicitlyOptional.newBuilder().setData(null)` skutkuje wyjątkiem `NullPointerException`. Poprawnym wywołaniem w tym wypadku jest `DataExplicitlyOptional.newBuilder().setData(Data.getDefaultInstance())`. W niektórych językach można, a czasem nie można, stosować odpowiedniki nulla (`None`, `missing`, `undefined` itp.). Więcej informacji na ten temat:

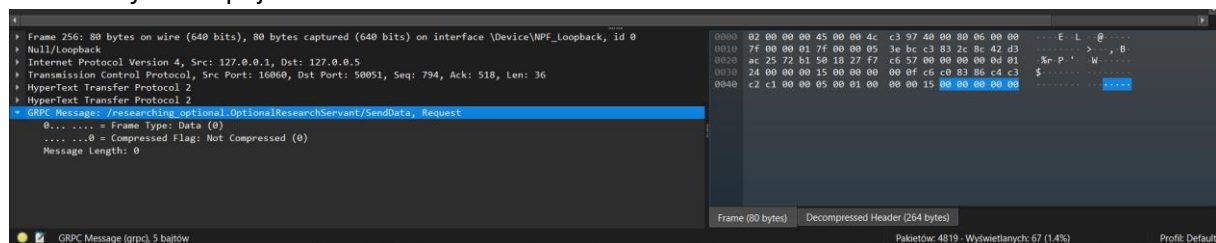
<https://gist.github.com/kshchepanovskyi/c0bf9b9f95a02742a310>

Spróbowałem wywołać dla pól niejawnie opcjonalnych oraz dla jawnie opcjonalnych (ze słowem „optional”).

Pola domyślnie opcjonalne ustawione na wartości puste/domyślne:



Pola domyślnie opcjonalne nie ustawiane na żadne wartości:

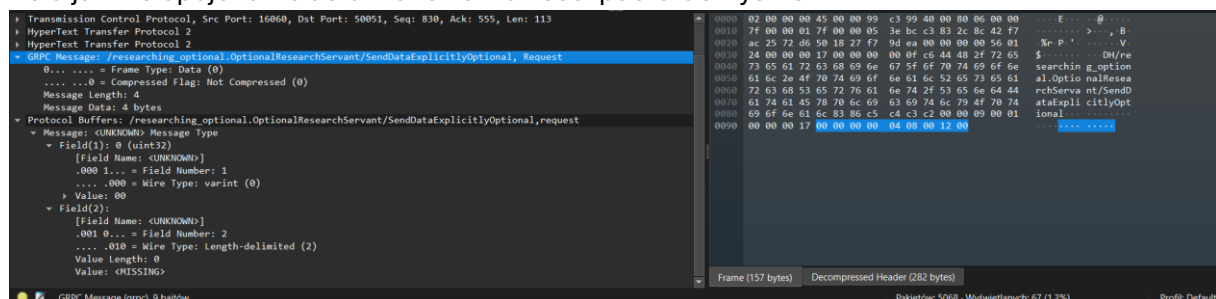


Pola domyślnie opcjonalne. Działanie metod „hasField()”:

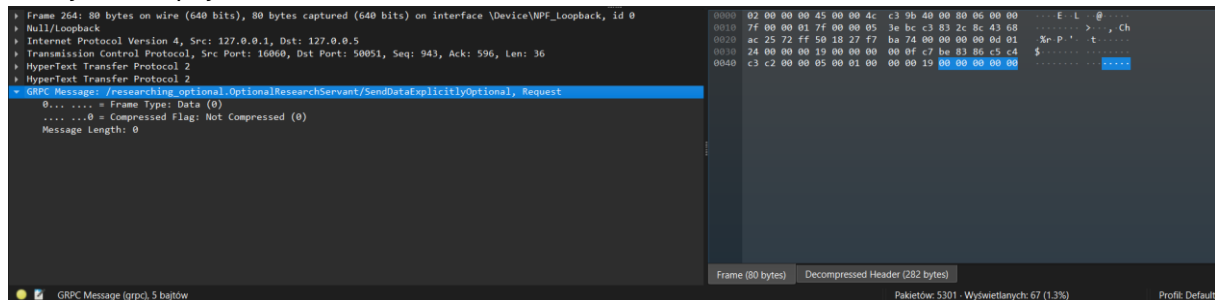
```
sendImplicitlyOptionalEmptyData
response.hasInt32(): doesn't exist for implicitly optional
response.hasData(): true

sendImplicitlyOptionalUnsetData
response.hasInt32(): doesn't exist for implicitly optional
response.hasData(): false
```

Pola jawnie opcjonalne ustawione na wartości puste/domyślne:



Pola jawnie opcjonalne nie ustawiane na żadne wartości :



Pola domyślnie opcjonalne. Działanie metod „hasField()”:

```
sendExplicitlyOptionalEmptyData
response.hasInt32(): true
response.hasData(): true

sendExplicitlyOptionalUnsetData
response.hasInt32(): false
response.hasData(): false
```

Jak widać, różnica jest znacząca.

Po pierwsze, dla pól domyślnie opcjonalnych nie istnieją wygenerowane metody sprawdzające obecność pól dla typów prymitywnych. Dla pól jawnie domyślnych jest to możliwe. Wynika to z tego, że pod spodem słowo kluczowe „optional” jest zamieniane na „oneof”, a, jak przedstawiłem we wcześniejszym podpunkcie, użycie „oneof” zapewnia metody typu „hasField” także dla typów prymitywnych

Po drugie, warto pamiętać, że jeśli chcemy „usunąć” jakieś pole („odustawić” jego wartość?) i zrobimy to ustawiając wartość domyślną (np. 0 dla inta, pusty ciąg znaków dla stringa, niewypętnioną strukturę dla struktury), to:

- dla opcjonalności niejawnej wartości prymitywne „znikną z przewodu”, ale struktury już nie
- dla opcjonalności jawnej zarówno wartości prymitywne jak i struktury pozostaną „w przewodzie”.

Z tego też powodu do tego celu zaleca się używanie metod „clearField”.

Więcej na ten temat tutaj:

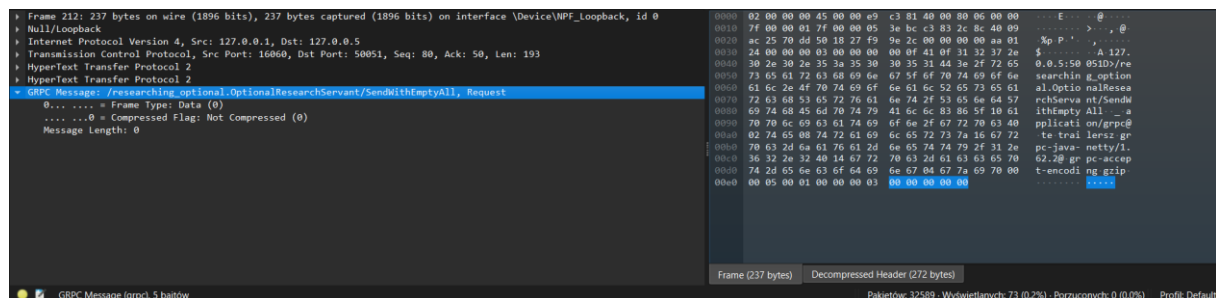
[https://github.com/protocolbuffers/protobuf/blob/v3.12.0/docs/field\\_presence.md](https://github.com/protocolbuffers/protobuf/blob/v3.12.0/docs/field_presence.md)

## 1.2 OPCJONALNOŚĆ ARGUMENTÓW WYWOŁAŃ

Definiowanie serwisów w plikach .proto nie pozwala na używanie czegoś, co miałyby oznaczać brak wartości podawanej (parametru) lub zwracanej (return), tzn. nie istnieje słówko „void”. Każde odwołanie do usługi wymaga podania argumentów oraz otrzymania odpowiedzi, nawet jeśli nas ona nie interesuje (choć możemy zignorować odpowiedź wysyłając zapytanie nieblokujące, ale dane i tak zostaną przesłane klientowi).

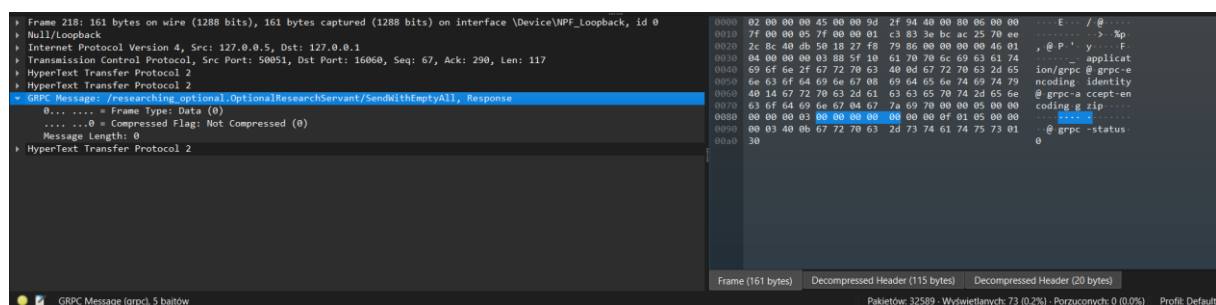
Oczywiście, w sytuacji, gdy nie musimy podać argumentów lub nie potrzebujemy wartości zwracanej, możemy skorzystać ze specjalnie przygotowanej przez Google’a struktury Empty w pliku "google/protobuf/empty.proto". Lub też, jak ja, możemy ją sami napisać, bo jest to po prostu „message Empty {}”.

Wywołanie usługi z argumentem „Empty”:



The image shows a Wireshark packet capture of a gRPC message. The left pane shows the packet list and details. The selected packet is a gRPC message (Frame 212) with details: Frame Type: Data (0), Compressed Flag: Not Compressed (0), Message Length: 0. The right pane shows the raw data in hexadecimal and ASCII. The ASCII view shows the gRPC message structure, including the magic number 0x00000000, the length 0, and the message type 0. The packet size is 237 bytes, and the decompressed header is 272 bytes.

Odpowiedź na powyższe zapytanie:



The image shows a Wireshark packet capture of a gRPC message. The left pane shows the packet list and details. The selected packet is a gRPC message (Frame 218) with details: Frame Type: Data (0), Compressed Flag: Not Compressed (0), Message Length: 0. The right pane shows the raw data in hexadecimal and ASCII. The ASCII view shows the gRPC message structure, including the magic number 0x00000000, the length 0, and the message type 0. The packet size is 161 bytes, and the decompressed header is 115 bytes.

Jak widać, sytuacja jest analogiczna jak podczas badania samej struktury „Empty”, tzn. sama struktura zajmuje 0B, ale żeby gRPC mogło ją przesłać, trzeba wykorzystać 5B.

## 2 ICE

### 2.1 OPCJONALNOŚĆ PÓL STRUKTUR DANYCH

W przypadku Slice’a struktury nie mogą mieć opcjonalnych pól. Jedynymi typami, które mogą mieć opcjonalne pola są:

- klasy,
- wyjątki.



## 2.1.1 „Optional” w klasach

Z uzupełnionym opcjonalnym polem:

```
Frame 88: 115 bytes on wire (920 bits), 115 bytes captured (920 bits) on interface lDevice\WPF_Loopback, id 0
NullLoopback
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.2
Transmission Control Protocol, Src Port: 34473, Dst Port: 10000, Seq: 85, Ack: 41, Len: 71
Internet Communications Engine Protocol
  Magic Number: ICMP
  Protocol Major: 1
  Protocol Minor: 0
  Encoding Major: 1
  Encoding Minor: 0
  Message Type: Request (0)
  Compression Status: Uncompressed, sender can accept a compressed reply (1)
  Message Size: 71
  Request Message Body
    Request Identifier: 2
    Object Identity Name: optionalResearcher
    Object Identity Content: (empty)
    Facet Name: (empty)
    Operation Name: sendc
    Icci:OperationMode: normal (0)
    Invocation Context: (empty)
    Input Parameters Size: 24
    Input Parameters Encoding Major: 1
    Input Parameters Encoding Minor: 1
    Encapsulated parameters: 0125093a3a4465d6f3a3a430aba000000ff
    ...
    0070 00 00 ff
```

Bez opcjonalnego pola:

```
Frame 89: 109 bytes on wire (872 bits), 109 bytes captured (872 bits) on interface lDevice\WPF_Loopback, id 0
NullLoopback
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.2
Transmission Control Protocol, Src Port: 34473, Dst Port: 10000, Seq: 159, Ack: 84, Len: 65
Internet Communications Engine Protocol
  Magic Number: ICMP
  Protocol Major: 1
  Protocol Minor: 0
  Encoding Major: 1
  Encoding Minor: 0
  Message Type: Request (0)
  Compression Status: Uncompressed, sender can accept a compressed reply (1)
  Message Size: 65
  Request Message Body
    Request Identifier: 3
    Object Identity Name: optionalResearcher
    Object Identity Content: (empty)
    Facet Name: (empty)
    Operation Name: sendc
    Icci:OperationMode: normal (0)
    Invocation Context: (empty)
    Input Parameters Size: 18
    Input Parameters Encoding Major: 1
    Input Parameters Encoding Minor: 1
    Encapsulated parameters: 0121093a3a4465d6f3a3a43
    ...
    0070 00 00 ff
```

Faktycznie, brak ustawionej wartości w opcjonalnym polu powoduje przesył mniejszej ilości bajtów. Zgadza się to z opisem w dokumentacji: <https://doc.zeroc.com/ice/3.7/best-practices/optional-values>

## 2.1.2 „Optional” w wyjątkach

Z ustawioną wartością:

```
Frame 91: 107 bytes on wire (856 bits), 107 bytes captured (856 bits) on interface lDevice\WPF_Loopback, id 0
NullLoopback
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.2
Transmission Control Protocol, Src Port: 34473, Dst Port: 10000, Seq: 226, Ack: 121, Len: 63
Internet Communications Engine Protocol
  Magic Number: ICMP
  Protocol Major: 1
  Protocol Minor: 0
  Encoding Major: 1
  Encoding Minor: 0
  Message Type: Request (0)
  Compression Status: Uncompressed, sender can accept a compressed reply (1)
  Message Size: 63
  Request Message Body
    Request Identifier: 4
    Object Identity Name: optionalResearcher
    Object Identity Content: (empty)
    Facet Name: (empty)
    Operation Name: throwException
    Icci:OperationMode: normal (0)
    Invocation Context: (empty)
    Input Parameters Size: 7
    Input Parameters Encoding Major: 1
    Input Parameters Encoding Minor: 1
    Encapsulated parameters: 01
    ...
    0070 00 00 ff
```

Bez ustawionej wartości:

```
Frame 96: 107 bytes on wire (856 bits), 107 bytes captured (856 bits) on interface lDevice\WPF_Loopback, id 0
NullLoopback
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.2
Transmission Control Protocol, Src Port: 34473, Dst Port: 10000, Seq: 287, Ack: 173, Len: 63
Internet Communications Engine Protocol
  Magic Number: ICMP
  Protocol Major: 1
  Protocol Minor: 0
  Encoding Major: 1
  Encoding Minor: 0
  Message Type: Request (0)
  Compression Status: Uncompressed, sender can accept a compressed reply (1)
  Message Size: 63
  Request Message Body
    Request Identifier: 5
    Object Identity Name: optionalResearcher
    Object Identity Content: (empty)
    Facet Name: (empty)
    Operation Name: throwException
    Icci:OperationMode: normal (0)
    Invocation Context: (empty)
    Input Parameters Size: 7
    Input Parameters Encoding Major: 1
    Input Parameters Encoding Minor: 1
    Encapsulated parameters: 00
    ...
    0070 00 00 ff
```

Co ciekawe, oba wyjątki zajmują taką samą liczbę bajtów.

## 2.1.3 Puste sekwencje

Ustawione dwie wartości:

```
Frame 100: 115 bytes on wire (920 bits), 115 bytes captured (920 bits) on interface \Device\NPF_{...}
Null/Loopback
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.2
Transmission Control Protocol, Src Port: 34473, Dst Port: 10000, Seq: 350, Ack: 219, Len: 71
Internet Communications Engine Protocol
  Magic Number: 1c0f
  Protocol Major: 1
  Protocol Minor: 0
  Encoding Major: 1
  Encoding Minor: 0
  Message Type: Request (0)
  Compression Status: Uncompressed, sender can accept a compressed reply (1)
  Message Size: 71
  Request Message Body
    Request Identifier: 6
    Object Identity Name: optionalResearcher
    Object Identity Content: (empty)
    Facet Name: (empty)
    Operation Name: sequenceMethod
    Ici::OperationNode: normal (0)
    Invocation Context: (empty)
    Input Parameters Size: 15
    Input Parameters Encoding Major: 1
    Input Parameters Encoding Minor: 1
    Encapsulated parameters: 02010000002000000
```

Ustawiona jedna wartość:

```
Frame 104: 111 bytes on wire (888 bits), 111 bytes captured (888 bits) on interface \Device\NPF_{...}
Null/Loopback
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.2
Transmission Control Protocol, Src Port: 34473, Dst Port: 10000, Seq: 421, Ack: 253, Len: 67
Internet Communications Engine Protocol
  Magic Number: 1c0f
  Protocol Major: 1
  Protocol Minor: 0
  Encoding Major: 1
  Encoding Minor: 0
  Message Type: Request (0)
  Compression Status: Uncompressed, sender can accept a compressed reply (1)
  Message Size: 67
  Request Message Body
    Request Identifier: 7
    Object Identity Name: optionalResearcher
    Object Identity Content: (empty)
    Facet Name: (empty)
    Operation Name: sequenceMethod
    Ici::OperationNode: normal (0)
    Invocation Context: (empty)
    Input Parameters Size: 11
    Input Parameters Encoding Major: 1
    Input Parameters Encoding Minor: 1
    Encapsulated parameters: 0101000000
```

Brak ustawionych wartości:

```
Frame 108: 107 bytes on wire (856 bits), 107 bytes captured (856 bits) on interface \Device\NPF_{...}
Null/Loopback
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.2
Transmission Control Protocol, Src Port: 34473, Dst Port: 10000, Seq: 488, Ack: 283, Len: 63
Internet Communications Engine Protocol
  Magic Number: 1c0f
  Protocol Major: 1
  Protocol Minor: 0
  Encoding Major: 1
  Encoding Minor: 0
  Message Type: Request (0)
  Compression Status: Uncompressed, sender can accept a compressed reply (1)
  Message Size: 63
  Request Message Body
    Request Identifier: 8
    Object Identity Name: optionalResearcher
    Object Identity Content: (empty)
    Facet Name: (empty)
    Operation Name: sequenceMethod
    Ici::OperationNode: normal (0)
    Invocation Context: (empty)
    Input Parameters Size: 7
    Input Parameters Encoding Major: 1
    Input Parameters Encoding Minor: 1
    Encapsulated parameters: 00
```

Zgodnie z oczekiwaniami, im więcej liczb w sekwencji, tym większej liczby bajtów potrzebujemy (za każdym razem o 4 więcej dla intów).

## 2.2 OPCJONALNOŚĆ ARGUMENTÓW WYWOŁAŃ

### 2.2.1 „void” jako parametr metody lub wartość zwracana

```
Frame 112: 102 bytes on wire (816 bits), 102 bytes captured (816 bits) on interface \Device\NPF_{...}
Null/Loopback
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.2
Transmission Control Protocol, Src Port: 34473, Dst Port: 10000, Seq: 551, Ack: 309, Len: 58
Internet Communications Engine Protocol
  Magic Number: 1c0f
  Protocol Major: 1
  Protocol Minor: 0
  Encoding Major: 1
  Encoding Minor: 0
  Message Type: Request (0)
  Compression Status: Uncompressed, sender can accept a compressed reply (1)
  Message Size: 58
  Request Message Body
    Request Identifier: 9
    Object Identity Name: optionalResearcher
    Object Identity Content: (empty)
    Facet Name: (empty)
    Operation Name: voidMethod
    Ici::OperationNode: normal (0)
    Invocation Context: (empty)
    Input Parameters Size: 4
    Input Parameters Encoding Major: 1
    Input Parameters Encoding Minor: 1
```



## 2.2.2 „Optional” w parametrach metody lub w zwracanej wartości

Ustawione parametry:

```
Frame 168: 115 bytes on wire (920 bits), 115 bytes captured (920 bits) on interface \Device\NPF_{...}
Null/Loopback
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
Transmission Control Protocol, Src Port: 3453, Dst Port: 10000, Seq: 689, Ack: 334, Len: 71
Internet Communications Engine Protocol
  Magic Number: Icmp
  Protocol Major: 1
  Protocol Minor: 0
  Encoding Major: 1
  Encoding Minor: 0
  Message Type: Request (0)
  Compression Status: Uncompressed, sender can accept a compressed reply (1)
  Message Size: 71
  Request Message Body
    Request Identifier: 18
    Object Identity Name: optionalResearcher
    Object Identity Content: (empty)
    Fact Name: (empty)
    Operation Name: optionalParameters
    Icsi:OperationMode: normal (0)
    Invocation Context: (empty)
    Input Parameters Size: 11
    Input Parameters Encoding Major: 1
    Input Parameters Encoding Minor: 1
    Encapsulated parameters: 0a05000000
```

Niestawione parametry:

```
Frame 168: 110 bytes on wire (880 bits), 110 bytes captured (880 bits) on interface \Device\NPF_{...}
Null/Loopback
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
Transmission Control Protocol, Src Port: 3453, Dst Port: 10000, Seq: 689, Ack: 364, Len: 66
Internet Communications Engine Protocol
  Magic Number: Icmp
  Protocol Major: 1
  Protocol Minor: 0
  Encoding Major: 1
  Encoding Minor: 0
  Message Type: Request (0)
  Compression Status: Uncompressed, sender can accept a compressed reply (1)
  Message Size: 66
  Request Message Body
    Request Identifier: 11
    Object Identity Name: optionalResearcher
    Object Identity Content: (empty)
    Fact Name: (empty)
    Operation Name: optionalParameters
    Icsi:OperationMode: normal (0)
    Invocation Context: (empty)
    Input Parameters Size: 6
    Input Parameters Encoding Major: 1
    Input Parameters Encoding Minor: 1
```

Zajętość wiadomości zmniejszyła się.

Ustawione wartości zwracane:

```
Frame 162: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{...}
Null/Loopback
Internet Protocol Version 4, Src: 127.0.0.2, Dst: 127.0.0.1
Transmission Control Protocol, Src Port: 10000, Dst Port: 3453, Seq: 334, Ack: 689, Len: 30
Internet Communications Engine Protocol
  Magic Number: Icmp
  Protocol Major: 1
  Protocol Minor: 0
  Encoding Major: 1
  Encoding Minor: 0
  Message Type: Reply (2)
  Compression Status: Uncompressed, sender can accept a compressed reply (1)
  Message Size: 30
  Reply Message Body
    Request Identifier: 18
    Reply Status: Success (0)
    Reported reply data: 0a00000001011205000000
```

Niestawione wartości zwracane:

```
Frame 166: 69 bytes on wire (552 bits), 69 bytes captured (552 bits) on interface \Device\NPF_{...}
Null/Loopback
Internet Protocol Version 4, Src: 127.0.0.2, Dst: 127.0.0.1
Transmission Control Protocol, Src Port: 10000, Dst Port: 3453, Seq: 364, Ack: 746, Len: 25
Internet Communications Engine Protocol
  Magic Number: Icmp
  Protocol Major: 1
  Protocol Minor: 0
  Encoding Major: 1
  Encoding Minor: 0
  Message Type: Reply (2)
  Compression Status: Uncompressed, sender can accept a compressed reply (1)
  Message Size: 25
  Reply Message Body
    Request Identifier: 11
    Reply Status: Success (0)
    Reported reply data: 0a0000000101
```

Tutaj, analogicznie, zajętość wiadomości zmniejszyła się