

# Dokumentacja

Bartłomiej Sweeklej, 300517

Dominik Baczyński, 300475

Mariusz Nieciecki, 300584

## CoAP Server

### Zadanie:

Zrealizować system (serwer) udostępniający opisane zasoby za pomocą protokołu CoAP. Serwer powinien działać na platformie EBSimUnoEth, używanej w ćwiczeniu laboratoryjnym realizowanym zdalnie. Dla wspieranych zasobów należy zaprojektować URI, oraz – tam, gdzie nie jest to oczywiste lub doprecyzowane ich stan i reprezentację. Serwer ma współpracować z podanym klientem CoAP, w zakresie wynikającym z podanych możliwości stworzonej przez Zespół implementacji protokołu CoAP. Serwer powinien umieć generować różne kody odpowiedzi, stosownie do sytuacji. W przypadku błędu, serwer powinien zwracać payload diagnostyczny.

### Udostępniane zasoby:

#### 1. Zasób opisujący pozostałe zasoby. Ścieżka: */.well-known/core*

*GET:* Pobranie reprezentacji zasobu w formacie *CoRE Link Format*

#### 2. Graf skierowany. Ścieżka: */Graph*

*GET:* Pobranie wszystkich krawędzi w grafie. Krawędź jest zadawana jako para numerów wierzchołków, Np. (3,4) oznacza krawędź od wierzchołka 3 do wierzchołka 4.

*PUT:* Dodanie nowej krawędzi grafu. W payloadzie wiadomości musi być zdefiniowana jedna i tylko jedna krawędź w formacie (dwie cyfry od 0 do 9 oddzielone bądź nie, innym znakiem), np.: {1,2; (1,2); 12; 1 2; 1;2; 1-2; 1x2; itp.}.

*GET:* Pobranie zbioru krawędzi dla kwadratu grafu. **!Nie zaimplementowane!**

#### 3. Trzy metryki opisujące wymianę datagramów między klientem a serwerem:

##### a. **Received Bytes** - ilość otrzymanych bajtów po stronie serwera. Ścieżka: */ReceivedB*

*GET:* Pobranie reprezentacji metryki Received Bytes - ilość otrzymanych bajtów (liczba)

##### b. **Send Bytes** - ilość wysłanych bajtów po stronie serwera. Ścieżka */SendB*

*GET:* Pobranie reprezentacji metryki Send Bytes - ilość wysłanych bajtów (liczba)

##### c. **Total Bytes** - ilość wszystkich wymienionych bajtów w komunikacji między serwerem a klientem. Ścieżka */TotalB*

*GET:* Pobranie reprezentacji metryki Total Bytes - ilość wszystkich wymienionych bajtów (liczba)

*Uwaga:* Metryka Total Bytes jest metryką o długim dostępie, tj. Na żądanie CON odpowiada wysyłając odpowiedź ACK i po chwili odsyła wartość.

## Funkcjonalności serwera:

1. **Obsługa wiadomości NON (GET/PUT)**
  - 1.1. Serwer odpowiada wiadomością NON z odpowiednim kodem na żądanie GET/PUT
2. **Obsługa opcji Content-Format**
  - 2.1. Wystawia bądź przyjmuje do wiadomości informację o formacie payloadu. Jeśli nie jest obsługiwany pomija tę opcję.
3. **Obsługa opcji Uri-Path**
  - 3.1. W odpowiedzi na żądanie GET zwraca wartość zasobu reprezentowanego przez Uri-Path, bądź wiadomość z kodem błędu w przypadku braku zasobu pod taką ścieżką.
4. **Obsługa opcji Accept**
  - 4.1. Przyjmuje do wiadomości informację o akceptowanym formacie danych. Gdy wartość nie jest obsługiwana, wysyła wartość w formacie "text/plain".
5. **Obsługa Token'a**
  - 5.1. Zwraca wartość taką samą jak w żądaniu w celu identyfikacji odpowiedzi po stronie klienta.
6. **Obsługa MID**
  - 6.1. Zwraca różne wartości. W przypadku wiadomości CON MID jest takie samo jak w żądaniu w celu identyfikacji odpowiedzi po stronie klienta.
7. **Obsługa żądań CON(GET/PUT)**
  - 7.1. Odpowiada wiadomością ACK z odpowiednim kodem w formacie "piggybacked". W przypadku żądania CON dla metryki o "długim czasie dostępu" odsyła pustą wiadomość ACK, po czym wiadomość NON niosącą zwracaną wartość.
8. **Obsługa CoAP PING**
  - 8.1. Odsyła pustą wiadomość ACK
9. **Obsługa opcji Block2 i Size2 - Brak**
  - 9.1. *Brak implementacji*

## Przyjęte rozwiązania programistyczne:

Dodanie własnych bibliotek:

a) coap.h - zawiera definicje nagłówka oraz samej wiadomości + przydatne funkcje do obsługi żądań

klasa **CoapHeader** - reprezentuje nagłówek wiadomości CoAP

klasa **CoapMessage** - reprezentuje całą wiadomość CoAP (wraz z nagłówkiem i innymi opcjami)

funkcja **SetContentFormat** - funkcja wpisująca opcję Content-Format do wiadomości

funkcja **SetPayload** - funkcja wpisująca tablice bajtów o podanej długości jako payload wiadomości CoAP

funkcja **Send** - funkcja wysyłająca całą wiadomość CoAP za pomocą obiektu OirEthernetUDP

funkcja **GetPacketLen** - funkcja zwracająca aktualny rozmiar wiadomości CoAP (wraz z nagłówkiem, opcjami i zawartością)

b) resources.h

klasa **Graph** - reprezentuje obiekt grafu skierowanego za pomocą macierzy sąsiedztwa

funkcja **AddEdge** - funkcja dodająca połączenie między węzłami

funkcja **GetGraph** - funkcja prezentująca graf jako zbiór połączeń

funkcja **Received** - funkcja aktualizująca metryki "bytesReceived" i "bytesTotal" o ilość bajtów podanych w parametrze

funkcja **Send** - funkcja aktualizująca metryki "bytesSend" i "bytesTotal" o ilość bajtów podanych w parametrze

funkcja **GetResource** - funkcja dostępu do wartości metryk. Przyjmuje Uri-Path jako parametr i odpowiada odpowiednią reprezentacją zasobu

Działanie głównej pętli programu:

1. Nasłuchiwanie na pakiety UDP
2. Odczyt nagłówka i tokenu wiadomości
3. Parsowanie i obsługa opcji – zapis obsługiwanych opcji do lokalnych zmiennych
4. Odczyt payloadu wiadomości
5. Reakcja na otrzymaną wiadomość

### **Instrukcja obsługi:**

1. Skompilować plik server.ino.
2. Odnaleźć plik wynikowy server.ino.hex.
3. Przenieść plik wynikowy do folderu z emulatorem EBSimUnoEth.
4. W run\_sim.bat zmienić ścieżkę do pliku wynikowego oraz IP na adres w podłączonej sieci.
5. Uruchomić EBSimUnoEth za pomocą run\_me.bat
6. Uruchomić przeglądarkę Firefox z zainstalowaną wtyczką COPPER
7. Wpisać adres: *coap://<IP:Port>/*
8. Sprawdzić działanie programu

## Scenariusz testów:

### I. NON:

1. Discover w celu ujawnienia zasobów
2. GET na każdym udostępnianym zasobie:
  - a. Graph
  - b. ReceivedB
  - c. SendB
  - d. TotalB
3. PUT – dodanie nowej krawędzi do grafu (można wykonać kilkakrotnie)
4. GET – Graph, w celu zobaczenia zaktualizowanego grafu

### II. CON:

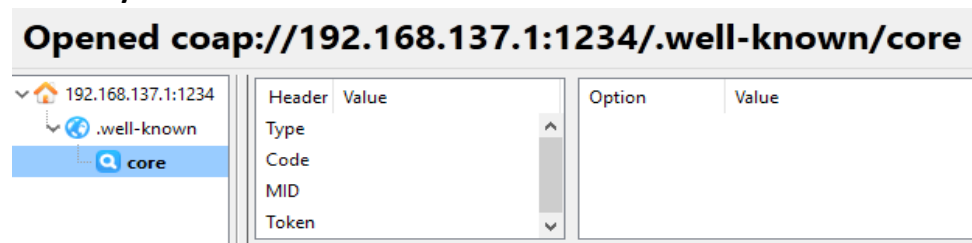
1. GET na zasobie o długim czasie dostępu:
  - a. TotalB
2. GET na innym zasobie
3. CoAP Ping (CON-EMPTY)

### III. BAD REQUEST:

1. Nieobsługiwana opcja, np. DELETE
2. Nieistniejący zasób
3. Dodanie złej krawędzi

## Zestawienie przeprowadzonych testów:

Przed wykonaniem testów:

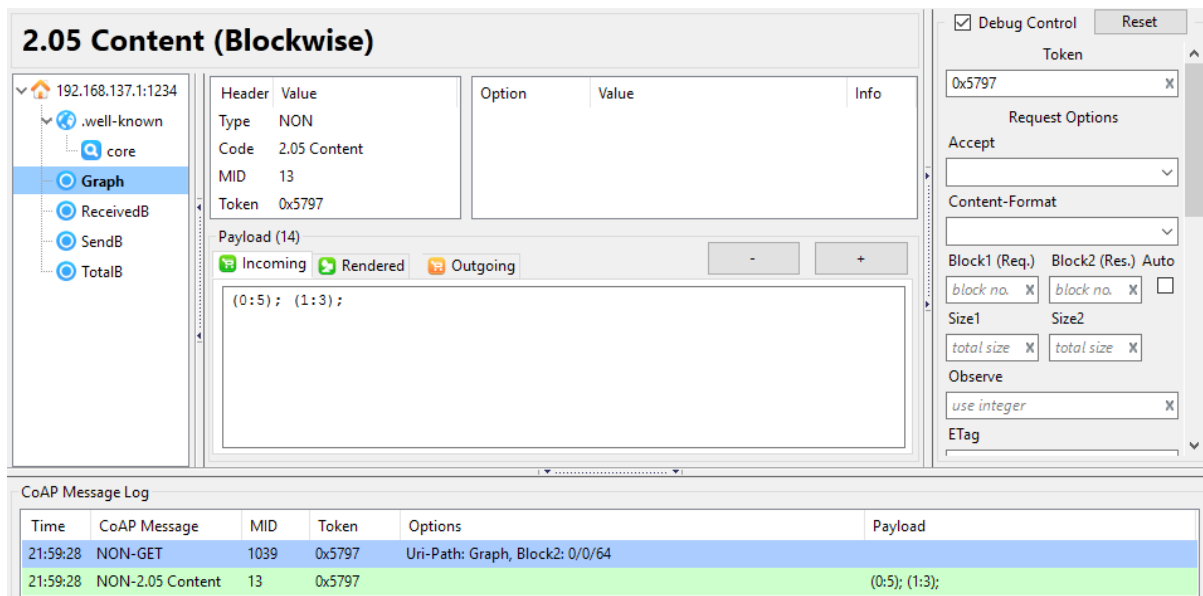


## Testy:

### I. 1. Wykonanie żądania Discover:

Time	CoAP Message	MID	Token	Options	Payload
21:54:14	NON-GET	35086	empty	Uri-Path: .well-known/core	
21:54:14	NON-2.05 Content	11	empty	Content-Format: 40	</ReceivedB>;</SendB>;</TotalB>;</Graph>
21:54:31	NON-GET	35087	empty	Uri-Path: .well-known/core	
21:54:31	NON-2.05 Content	12	empty	Content-Format: 40	</ReceivedB>;</SendB>;</TotalB>;</Graph>

## I. 2. a. Wykonanie żądania **NON GET** na Graph:



**2.05 Content (Blockwise)**

192.168.137.1:1234

- .well-known
  - core
    - Graph**
    - ReceivedB
    - SendB
    - TotalB

Header Value | Option Value | Info

Type: NON

Code: 2.05 Content

MID: 13

Token: 0x5797

Payload (14)

Incoming | Rendered | Outgoing

(0:5); (1:3);

Debug Control | Reset

Token: 0x5797

Request Options

Accept: [dropdown]

Content-Format: [dropdown]

Block1 (Req.) | Block2 (Res.) | Auto

block no. | block no. | [checkbox]

Size1 | Size2

total size | total size

Observe: use integer

ETag: [dropdown]

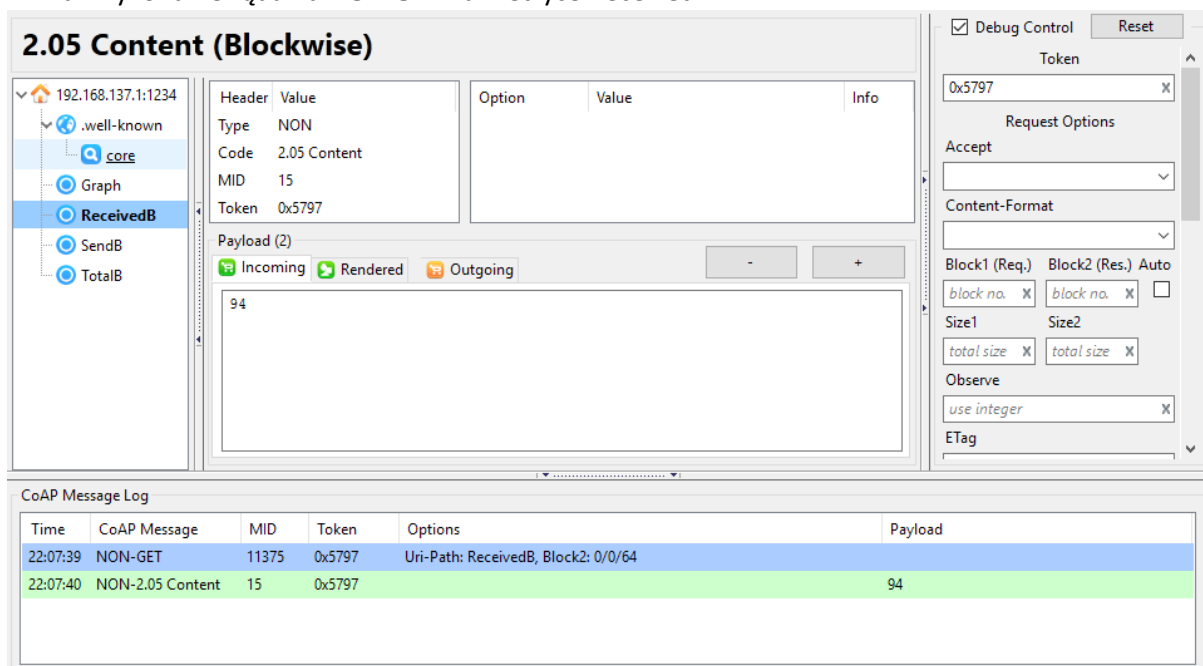
CoAP Message Log

Time	CoAP Message	MID	Token	Options	Payload
21:59:28	NON-GET	1039	0x5797	Uri-Path: Graph, Block2: 0/0/64	
21:59:28	NON-2.05 Content	13	0x5797		(0:5); (1:3);

Póki nie zostało wykonane żądanie PUT na zasobie Graph i nie dodaliśmy nowych węzłów, wyświetlają się jedynie przykładowe, dodane bezpośrednio w kodzie:

```
27 | AddEdge(0, n-1);
28 | AddEdge(1, 3);
```

## I. 2. b. Wykonanie żądania **NON GET** na metryce ReceivedB:



**2.05 Content (Blockwise)**

192.168.137.1:1234

- .well-known
  - core
    - Graph
    - ReceivedB**
    - SendB
    - TotalB

Header Value | Option Value | Info

Type: NON

Code: 2.05 Content

MID: 15

Token: 0x5797

Payload (2)

Incoming | Rendered | Outgoing

94

Debug Control | Reset

Token: 0x5797

Request Options

Accept: [dropdown]

Content-Format: [dropdown]

Block1 (Req.) | Block2 (Res.) | Auto

block no. | block no. | [checkbox]

Size1 | Size2

total size | total size

Observe: use integer

ETag: [dropdown]

CoAP Message Log

Time	CoAP Message	MID	Token	Options	Payload
22:07:39	NON-GET	11375	0x5797	Uri-Path: ReceivedB, Block2: 0/0/64	
22:07:40	NON-2.05 Content	15	0x5797		94

### I. 2. c. Wykonanie żądania **NON GET** na metryce SendB:

**2.05 Content (Blockwise)**

192.168.137.1:1234

- .well-known
- core
- Graph
- ReceivedB
- SendB**
- TotalB

Header	Value	Option	Value	Info
Type	NON			
Code	2.05 Content			
MID	16			
Token	0x5797			

Payload (3)

Incoming Rendered Outgoing

148

Debug Control ☒ Reset

Token: 0x5797

Request Options

Accept:

Content-Format:

Block1 (Req.) Block2 (Res.) Auto

block no.  block no.  ☐

Size1  Size2

total size  total size

Observe:

use integer

ETag:

CoAP Message Log

Time	CoAP Message	MID	Token	Options	Payload
22:08:10	NON-GET	4154	0x5797	Uri-Path: SendB, Block2: 0/0/64	
22:08:11	NON-2.05 Content	16	0x5797		148

### I. 2. d. Wykonanie żądania **NON GET** na metryce TotalB

**2.05 Content (Blockwise)**

192.168.137.1:1234

- .well-known
- core
- Graph
- ReceivedB
- SendB
- TotalB**

Header	Value	Option	Value	Info
Type	NON			
Code	2.05 Content			
MID	17			
Token	0x5797			

Payload (3)

Incoming Rendered Outgoing

281

Debug Control ☒ Reset

Token: 0x5797

Request Options

Accept:

Content-Format:

Block1 (Req.) Block2 (Res.) Auto

block no.  block no.  ☐

Size1  Size2

total size  total size

Observe:

use integer

ETag:

CoAP Message Log

Time	CoAP Message	MID	Token	Options	Payload
22:08:36	NON-GET	10570	0x5797	Uri-Path: TotalB, Block2: 0/0/64	
22:08:36	NON-2.05 Content	17	0x5797		281

### I. 3. Wykonanie żądania PUT na Graph:

The screenshot displays the CoAP client interface for a PUT request to the Graph resource. The main window is titled "2.01 Created". On the left, a tree view shows the network structure with "Graph" selected. The central pane shows the message details: Header (Type: NON, Code: 2.01 Created, MID: 14, Token: 0x5797), Option (empty), and Payload (15 bytes) containing the text "Edge was added.". The right pane shows the "Request Options" for a PUT request, including Token (0x5797), Content-Format, and Block1/Block2 settings. Below the main window is the "CoAP Message Log" table.

Time	CoAP Message	MID	Token	Options	Payload
21:59:28	NON-GET	1039	0x5797	Uri-Path: Graph, Block2: 0/0/64	
21:59:28	NON-2.05 Content	13	0x5797		(0:5); (1:3);
22:06:09	NON-PUT	1040	0x5797	Uri-Path: Graph	4 5
22:06:09	NON-2.01 Created	14	0x5797		Edge was added.

### I. 4. Ponowne wykonanie żądania GET na Graph:

The screenshot displays the CoAP client interface for a GET request to the Graph resource. The main window is titled "2.05 Content (Blockwise)". On the left, the tree view shows "Graph" selected. The central pane shows the message details: Header (Type: ACK, Code: 2.05 Content, MID: 2984, Token: 0x5797), Option (empty), and Payload (21 bytes) containing the text "(0:5); (1:3); (4:5);". The right pane shows the "Request Options" for a GET request. Below the main window is the "CoAP Message Log" table.

Time	CoAP Message	MID	Token	Options	Payload
22:12:23	CON-GET	2984 (0)	0x5797	Uri-Path: Graph, Block2: 0/0/64	
22:12:23	ACK-2.05 Content	2984	0x5797		(0:5); (1:3); (4:5);

## II. 1. Wykonanie żądania CON GET na metryce TotalB

**2.05 Content (Blockwise)**

192.168.137.1:1234

- .well-known
- core
- Graph
- ReceivedB
- SendB
- TotalB**

Header	Value	Option	Value	Info
Type	NON			
Code	2.05 Content			
MID	19			
Token	0x5797			

Payload (3)

Incoming Rendered Outgoing

339

Debug Control ☒ Reset

Token: 0x5797

Request Options

Accept:

Content-Format:

Block1 (Req.) Block2 (Res.) Auto

block no.  block no.  ☐

Size1  Size2

total size  total size

Observe:

ETag:

CoAP Message Log

Time	CoAP Message	MID	Token	Options	Payload
22:09:52	CON-GET	739 (0)	0x5797	Uri-Path: TotalB, Block2: 0/0/64	
22:09:52	ACK-EMPTY	739	empty		
22:09:55	NON-2.05 Content	19	0x5797		339

## II. 2. Wykonanie żądania CON GET na metryce SendB

**2.05 Content (Blockwise)**

192.168.137.1:1234

- .well-known
- core
- Graph
- ReceivedB
- SendB**
- TotalB

Header	Value	Option	Value	Info
Type	ACK			
Code	2.05 Content			
MID	31148			
Token	empty			

Payload (3)

Incoming Rendered Outgoing

104

Debug Control ☐ Reset

Token: use hex (0x...) or string

Request Options

Accept:

Content-Format:

Block1 (Req.) Block2 (Res.) Auto

block no.  block no.  ☒

Size1  Size2

total size  total size

Observe:

ETag:

CoAP Message Log

Time	CoAP Message	MID	Token	Options	Payload
23:07:48	CON-GET	31148 (0)	empty	Uri-Path: SendB, Block2: 0/0/64	
23:07:48	ACK-2.05 Content	31148	empty		104



### II. 3. Wykonanie żądania Ping:

**Pong: Remote responds to CoAP (RTT 34 ms)**

192.168.137.1:1234

- .well-known
  - core
    - Graph
    - ReceivedB
    - SendB
    - TotalB

Header	Value	Option	Value	Info
Type	ACK			
Code	EMPTY			
MID	47159			
Token	empty			

Payload

Incoming Rendered Outgoing

CoAP Message Log

Time	CoAP Message	MID	Token	Options	Payload
22:10:24	CON-EMPTY	47159 (0)	empty		
22:10:24	ACK-EMPTY	47159	empty		

### III. 1. Wykonanie żądania DELETE:

**5.01 Not Implemented**

192.168.137.1:1234

- .well-known
  - core
    - Graph
    - ReceivedB
    - SendB
    - TotalB

Header	Value	Option	Value	Info
Type	NON			
Code	5.01 Not Implemented			
MID	22			
Token	empty			

Payload

Incoming Rendered Outgoing

CoAP Message Log

Time	CoAP Message	MID	Token	Options	Payload
23:10:42	NON-DELETE	5469	empty		
23:10:42	NON-5.01 Not Implemented	22	empty		

### III. 2. Wykonanie żądania GET na nieistniejącym zasobie:

The screenshot shows the CoAP client interface with the title "4.04 Not Found (Blockwise)". On the left, a tree view shows the network structure with "192.168.137.1:1234" expanded, showing ".well-known", "core", "Graph", "ReceivedB", "SendB", and "TotalB". The "core" node is selected. The main panel displays the message details:

Header	Value	Option	Value	Info
Type	NON			
Code	4.04 Not Found			
MID	23			
Token	empty			

The Payload section shows "Incoming", "Rendered", and "Outgoing" tabs. The "Rendered" tab is active, showing an empty payload. On the right, the "Request Options" section includes "Accept", "Content-Format", "Block1 (Req.)", "Block2 (Res.)", "Auto", "Size1", "Size2", "Observe", and "FTan".

Below the main panel is the "CoAP Message Log" table:

Time	CoAP Message	MID	Token	Options	Payload
23:12:12	NON-GET	29684	empty	Uri-Path: something, Block2: 0/0/64	
23:12:12	NON-4.04 Not Found	23	empty		

### III. 3. Dodanie złej krawędzi:

The screenshot shows the CoAP client interface with the title "4.00 Bad Request". On the left, the "Graph" node is selected. The main panel displays the message details:

Header	Value	Option	Value	Info
Type	NON			
Code	4.00 Bad Request			
MID	25			
Token	empty			

The Payload section shows "Incoming", "Rendered", and "Outgoing" tabs. The "Rendered" tab is active, showing the payload: "Not valid syntax or edge already exists." On the right, the "Request Options" section includes "Accept", "Content-Format", "Block1 (Req.)", "Block2 (Res.)", "Auto", "Size1", "Size2", "Observe", and "FTan".

Below the main panel is the "CoAP Message Log" table:

Time	CoAP Message	MID	Token	Options	Payload
23:15:03	NON-PUT	42430	empty	Uri-Path: Graph	3 6
23:15:03	NON-4.00 Bad Request	25	empty		Not valid syntax or edge already exists.