

Bartosz KOWAL¹
Paweł DYMORA²
Mirosław MAZUREK³

NS2 – JAKO ŚRODOWISKO SYMULACYJNE DO BADAŃ NAD BEZPRZEWODOWYMI SIECIAMI SENSOROWYMI

Wraz z rozwojem bezprzewodowych systemów sensorowych rośnie zainteresowanie Internetem Rzeczy (ang. *Internet of Things*). Celem artykułu jest przybliżenie możliwości jakie dostarcza środowisko symulacyjne NS2 do badań nad bezprzewodowymi sieciami sensorowymi. W artykule opisano symulator sieci NS2, scharakteryzowano główne funkcje symulatora, przedstawiono i opisano przykładowy proces budowy skryptu sieci WSN oraz analizę wyników symulacji testowanej sieci.

Słowa kluczowe: sensory, WSN, NS2, bezprzewodowa sieć sensorowa.

1. Wstęp

Postęp technologiczny doprowadził do stworzenia koncepcji Internetu Rzeczy. W myśl tej koncepcji każde z urządzeń powinno być podłączone do sieci komputerowej i przez nią sterowane. Bezprzewodowa sieć sensorowa (WSN) jest infrastrukturą sieciową składającą się z czujników. Ich zadaniem jest przetwarzanie danych i wysyłanie użytkownikowi zbadanych parametrów i zareagowanie na określone przez użytkownika zdarzenie. Sieć taką tworzy się w przypadku, gdy zależy użytkownikowi na mobilności i łatwości rozlokowania czujników, niskim zużyciu zasobów energii węzłów sieciowych, długim i nieprzerwanym czasie pracy oraz niezawodności sieci. Takie właśnie sieci wprowadza się już w prawie każdej dziedzinie życia, np. sieci WSN są już używane przez architektów budowlanych podczas projektowania inteligentnych budynków. Ciągły

¹ Autor do korespondencji: Bartosz Kowal, Politechnika Rzeszowska, Zakład Systemów Złożonych, bartosz.kowal@prz.edu.pl

² Paweł Dymora, Politechnika Rzeszowska, Zakład Systemów Złożonych, pawel.dymora@prz.edu.pl

³ Mirosław Mazurek, Politechnika Rzeszowska, Zakład Systemów Złożonych, miroslaw.mazurek@prz.edu.pl

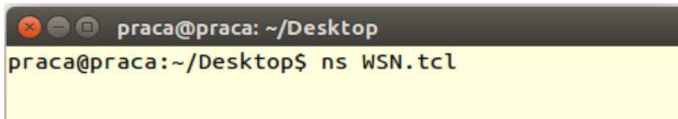
rozwój tej technologii i mnóstwo przeprowadzanych badań sprawia, że taka sieć jest co raz częściej wykorzystywana [1].

Do badań nad takimi sieciami służą symulatory badawcze, mające na celu symulację pracę rzeczywistej sieci. Dzięki nim można w łatwy sposób sprawdzić, czy badana sieć spełnia żądane wymagania, nie posiada błędów strukturalnych, spełnia wymogi QoS itp. Nie trzeba więc inwestować ogromnych pieniędzy w urządzenia fizyczne przed ich przetestowaniem. Wystarczy tylko stworzyć odpowiednio skonfigurowaną symulację sieci. Symulatory sieci, gdy poprawnie się je skonfiguruje, mogą pomóc w odzwierciedleniu rzeczywistej sieci, lecz nie zawsze można się na nich opierać. Symulacja dostarcza uogólnione informacje o parametrach sieci. Takie wykorzystanie środowisk testowych sieci może pozwolić na zlokalizowanie „wąskich gardeł” sieci, ocenie poprawności zastosowanych protokołów komunikacyjnych i routingu np. OSPF, BGP, AODV, LEACH itp. Obecnie na rynku można znaleźć wiele różnych symulatorów sieci - komercyjnych czy też bezpłatnych opartych na licencji Open Source.

2. Network Simulator v2

Jednym z symulatorów do badania WSN jest Network Simulator v2, powszechnie znany jako NS2. NS2 jest to obiektowo zorientowany symulator sieci napisany w języku C++. Pozwala on tworzyć sieci przewodowe oraz bezprzewodowe w oparciu o różne protokoły takie jak: TCP, UDP, FTP, FTP, DSR czy też protokoły routingu OSPR, RIP, AODV, DSDV, LEACH itp. Jest on odpowiedni do symulacji bezprzewodowych sieci sensorowych [2-3].

Program ten nie posiada trybu graficznego, co pozwala na jego uruchomienie nawet na komputerach o słabszych parametrach. Symulator ten jest polecany wyłącznie tym, którzy dobrze rozumieją języki programowania C++, OTcl, gdyż modelowanie sieci jest bardzo czasochłonne i bardzo złożone. Brak trybu graficznego oraz brak specjalistycznych narzędzi do zbierania danych wymaga od użytkowników dogłębnego poznania budowy skryptów Tcl w NS2 oraz ręczne tworzenie skryptów AWK do pobrania interesujących użytkownika informacji z bardzo obszernych plików z danymi. Pomimo upływu czasu i ogromnej ilości bibliotek czy poradników, NS2 jest powoli wypierany przez nowszą wersję środowiska NS3 z trybem graficznym, która oparta jest języku C++ oraz języku Python [2-4].



```
praca@praca: ~/Desktop
praca@praca:~/Desktop$ ns WSN.tcl
```

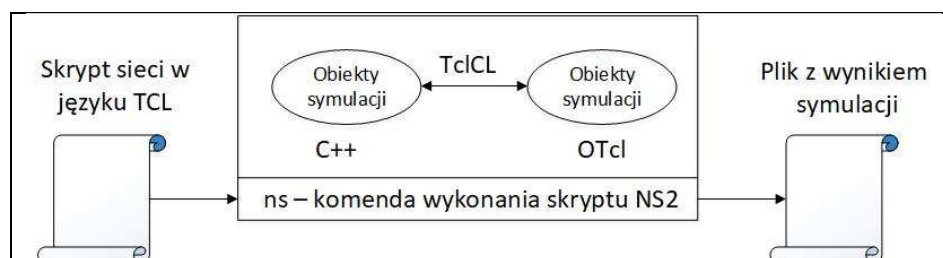
Rys. 1. Network Simulator v2 - NS2

Fig. 1. Network Simulator v2 - NS2

Do uruchomienia środowiska symulacyjnego NS2 potrzebny jest system Linux lub Microsoft Windows z oprogramowaniem Cygwin. Network Simulator v2 opiera się na kilku narzędziach oraz programach, których odpowiednie wersje powinno się pobrać przed zainstalowaniem NS2 np. Tcl 8.5, OTcl 1.14, tak aby całe środowisko mogło się bez problemu zainstalować i uruchomić. Jednym z najczęstszych problemów podczas kompilacji skryptu NS jest nieodpowiednie dobranie wersji języka Tcl do używanej wersji symulatora sieci, brak odpowiednich bibliotek, brak zmiennych środowiskowych oraz błędy składni, gdyż skrypt jest mieszaniną języków C++ oraz Tcl [2-3].

2.1. Architektura NS2

Na rys. 2 przedstawiono architekturę środowiska symulacyjnego NS2. Skrypt jest uruchamiany za pomocą polecenia *ns*, z którym zaleca się zastosowanie przekierowania standardowego wyjścia do innego pliku np. pliku wynikowego, tak aby można było później odczytać dane [5].



Rys. 2. Architektura NS2 [5]

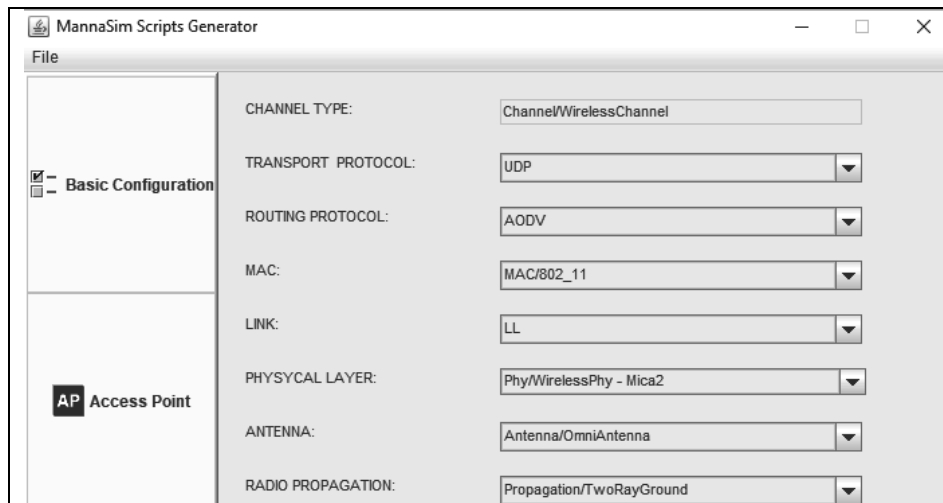
Fig. 2. Architecture of NS2 [5]

Jak już wspomniano, NS2 korzysta z dwóch języków programowania C++ i OTcl. Funkcje języka C++ definiują mechanizmy wewnętrznych symulowanych obiektów, zaś OTcl służy do konfiguracji symulacji poprzez interpretację skryptu z dostępnymi bibliotekami. Po zakończeniu symulacji, NS2 generuje wynik w postaci tekstowego pliku wynikowego. Do analizy wyników zalecane jest utworzenie skryptów AWK, pozwalających np. na odczyt opóźnień dla poszczególnego węzła w sieci [5].

2.2. MANNASIM FRAMEWORK - moduł bezprzewodowych sieci sensorowych do NS2

Mannasim Framework jest modułem zawierającym wymagane biblioteki do zasymulowania bezprzewodowych sieci sensorowych. Framework wprowadza do podstawowej wersji NS2 nowe moduły oraz biblioteki służące do projektowania i analizy bezprzewodowych sieci sensorowych. Jedną z głównych

zalet MannaSim jest dołączony generator skryptów sieci WSN. Generator ten przedstawiono na rys. 3. Napisano go w języku Java. Pozwala na elastyczne i szczegółowe tworzenie skryptów sieci WSN [6].



Rys. 3. Generator skryptów WSN [6]

Fig. 3. WSN scripts generator [6]

2.3. Etapy budowy skryptu WSN w środowisku NS2

Na samym początku tworzenia skryptu WSN należy zdefiniować rozmiar sieci, jakie protokoły routingu zostaną wdrożone, jaka technologia komunikacji zostanie wykorzystana i co ma być w niej badane. Następnie można podzielić tworzenie skryptu na kilka etapów [6]:

1. **Podstawowa konfiguracja sieci** – ustala się w niej protokół warstwy transportowej, protokół routingu, długość kolejki interfejsu, rozmiar badanej sieci oraz czas symulacji.
2. **Budowa węzła głównego** – umiejscowienie węzła, zasięg anteny, energia baterii węzła, liczba węzłów, czas i wielkość generowanych danych.
3. **Budowa węzłów komunikacyjnych** – umiejscowienie węzłów, zasięg anteny, energia baterii węzła, liczbę węzłów, czas i wielkość generowanych danych, moment wysłania danych (okresowo, na żądanie, na warunek).
4. **Klastry** – w przypadku hierarchicznych bezprzewodowych sieci sensorych, ustala się, które z węzłów tworzą klastry, tak aby zoptymalizować działanie sieci i zmniejszyć zużycie energii.

2.3.1. Podstawowa konfiguracja sieci

Konfigurację sieci zaczyna się od wyznaczenia rozmiaru scenariusza:

```
set val(x) 100.0
set val(y) 100.0
```

Następnym krokiem jest ustalenie zużycia energii dla poszczególnych operacji:

```
set mica(sensing_power) 0.018;# 18 mW = 0.018 W
set mica(processing_power) 0.027 ;# mW = 0.027 W
set mica(instructions_per_second) 8000000 ;# 8MHZ
```

Trzecim etapem jest ustawienie podstawowych parametrów węzła takich jak częstotliwość pracy nadajników, typ propagacji fal radiowych, szerokość pasma oraz zużycie energii na poszczególne operacje:

```
proc setup_mica2 { antenna range } {
Phy/WirelessPhy set Pt_ 0.281838
Phy/WirelessPhy set freq_ 2.4e09
Phy/WirelessPhy set L_ 1.0
Phy/WirelessPhy set lambda_ 0.125
Phy/WirelessPhy set RXThresh_ [TwoRay 0.281838 [$antenna set
Gt_] [$antenna set Gr_] 0.8 0.8 1.0 $range 0.125]
Phy/WirelessPhy set bandwidth_ 28.8*10e3; #28.8 kbps
Node/MobileNode/SensorNode set sensingPower_ 0.015
Node/MobileNode/SensorNode set processingPower 0.024
Node/MobileNode/SensorNode set instructionsPerSecond_ 8000000}
```

2.3.2. Budowa węzła głównego

Budowę węzła głównego zaczyna się od ustalenia zasięgu anteny:

```
set_default_settings
set val(range) 50
setup_mica2 $val(antenna) $val(range)
```

Kolejnym krokiem jest ustalenie położenia węzła w sieci:

```
set local(x) 50.0
set local(y) 50.0
set local(z) 0.0
```

Następnie ustawiana jest pojemność baterii węzła:

```
set local(start) ""
set local(stop) ""
set local(energy) 100.0
```

Ostatnim krokiem jest utworzenie węzła:

```
set val(apApp) Application/AccessPointApp
create_access_point $local(energy) $local(x) $local(y) $local(z)
$local(start) $local(stop)
```

2.3.3. Budowa węzłów komunikacyjnych

Budowę węzła komunikacyjnego zaczyna się od ustalenia zasięgu anteny:

```
set_default_settings
set val(range) 50
setup_mica2 $val(antenna) $val(range)
```

Następnie określany jest czas i typ przesyłania danych np. programowo, okresowo, na żądanie:

```
set val(disseminating_type) 1;
set val(disseminating_interval) 20.0
set val(processing) Processing/AggregateProcessing
set val(data_generator) "[create_data_generator {TemperatureDataGenerator} {5.0} {programmed} {25.0} {5.0} {30.0}]"
```

Trzecim etapem jest ustawienie pojemności baterii węzła:

```
set local(start) ""
set local(stop) ""
set local(energy) 10.0
```

Kolejnym krokiem jest zdefiniowanie położenia węzła w sieci:

```
set local(x) 10.0
set local(y) 10.0
set local(z) 0.0
set local(father_addr) $val(father_addr)
```

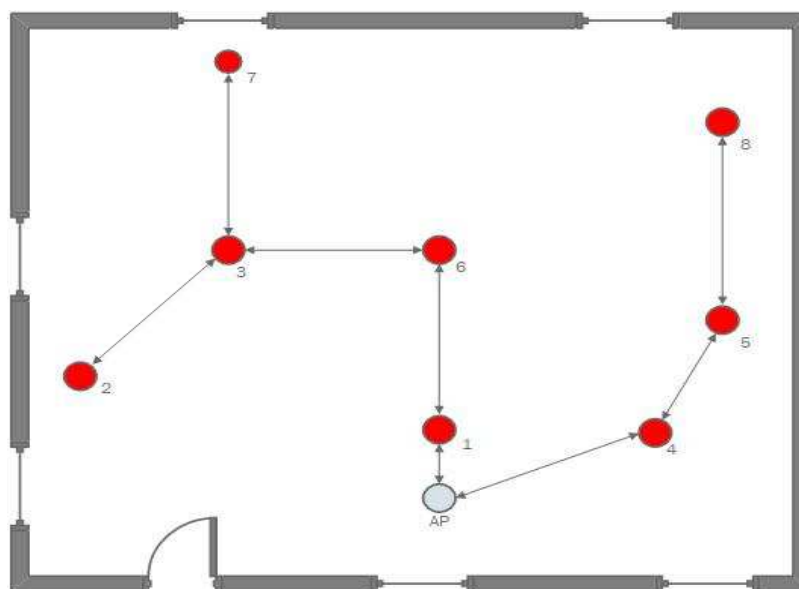
Ostatnim krokiem jest utworzenie węzła:

```
set val(commonApp) Application/SensorBaseApp/CommonNodeAp
create_common_node $local(energy) $local(x) $local(y) $local(z)
$local(father_addr) $local(start) $local(stop)
```

3. Analiza rezultatów symulacji WSN w środowisku NS2

Plik wynikowy z danymi symulacji bezprzewodowej sieci sensorowej dostarcza wielu informacji. Sam NS2 z dołączonym modułem Mannasim służącym do badania WSN nie posiada narzędzi graficznych do przeglądania zebranych danych. Zalecane jest stworzenie własnym skryptów do przetworzenia danych.

Dzięki odpowiedniemu wykorzystaniu symulatora sieci NS2, można zasymulować np. pracę bezprzewodowej sieci sensorowej mającej na celu badanie poziomu zapylenia w hali produkcyjnej (70x100 m). Strukturę takiej sieci przedstawiono na rys. 4, zaś część zebranych danych przedstawiono w tabeli 2. Poniżej zostaną przedstawione wybrane wyniki symulacji, mające na celu zbadanie właściwości sieci WSN przy użyciu algorytmu routingu AODV i zasięgu nadajnika 15 m. Zakładając ciągłą pracę 8 czujników działających na częstotliwości 2,4 GHz oraz znając maksymalną pojemność baterii wynoszącą 10 J można zacząć badać zebrane dane symulacyjne.



Rys. 4. Topologia badanej bezprzewodowej sieci sensorowej

Fig. 4. Wireless sensor network topology

W tabeli 1 pokazano wybrane typy wiadomości z pliku wynikowego wraz z krótkim komentarzem. Pełna specyfikacja, funkcje oraz pozostałe informacje można odczytać z dokumentacji Mannasim Framework [7].

Tabela 1. Wybrane parametry pliku wynikowego

Table 1. Selected parameters of the result file

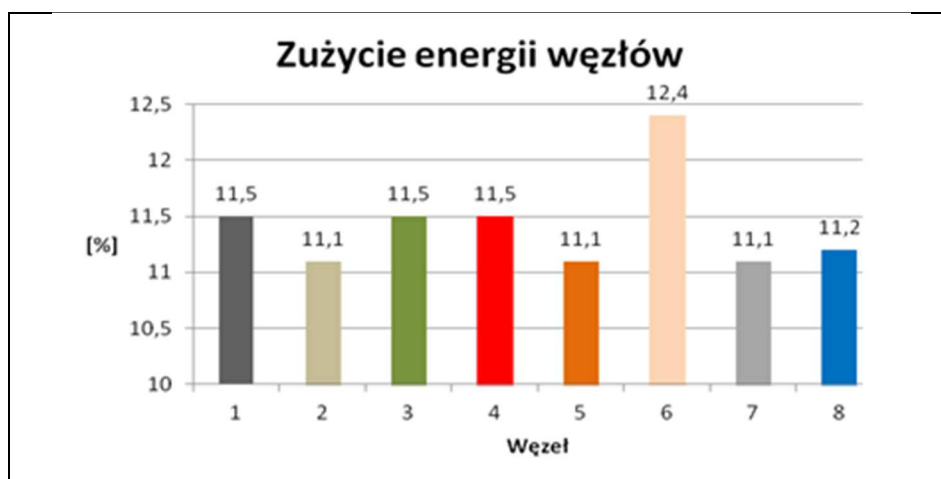
Typ zdarzenia	s – wysłany pakiet
	r – otrzymany pakiet
Identyfikator przeskoku pakietów (hop)	Hs – identyfikator węzła
	Hd – identyfikator kolejnego węzła
Położenie węzła w sieci	Nx – położenie węzła na osi x
	Ny – położenie węzła na osi y
	Nz – położenie węzła na osi z
Energia węzła	Nd – pozostała energia węzła
Czas	t – czas zdarzenia

Tabela 2. Wybrane dane symulacji sieci

Table 2. Selected network simulation data

s	-t 0.158	-Hs 3	-Hd -	-Ni 3	-Nx 25.00	-Ny 25.00	-Ne 9.999987
r	-t 0.159	-Hs 3	-Hd -	-Ni 3	-Nx 25.00	-Ny 25.00	-Ne 9.999987
s	-t 0.160	-Hs 3	-Hd -	-Ni 3	-Nx 25.00	-Ny 25.00	-Ne 9.999837
r	-t 0.159	-Hs 2	-Hd -	-Ni 2	-Nx 15.00	-Ny 15.00	-Ne 9.999980
r	-t 0.159	-Hs 6	-Hd -	-Ni 6	-Nx 25.00	-Ny 40.00	-Ne 9.999980
r	-t 0.159	-Hs 7	-Hd -	-Ni 7	-Nx 40.00	-Ny 25.00	-Ne 9.999980
r	-t 0.159	-Hs 2	-Hd -	-Ni 2	-Nx 15.00	-Ny 15.00	-Ne 9.999980
r	-t 0.159	-Hs 6	-Hd -	-Ni 6	-Nx 25.00	-Ny 40.00	-Ne 9.999980
r	-t 0.159	-Hs 7	-Hd -	-Ni 7	-Nx 40.00	-Ny 25.00	-Ne 9.999980
s	-t 0.159	-Hs 2	-Hd -	-Ni 2	-Nx 15.00	-Ny 15.00	-Ne 9.999980
s	-t 0.159	-Hs 2	-Hd -	-Ni 2	-Nx 15.00	-Ny 15.00	-Ne 9.999980
s	-t 0.160	-Hs 6	-Hd -	-Ni 6	-Nx 25.00	-Ny 40.00	-Ne 9.999959
r	-t 0.160	-Hs 3	-Hd -	-Ni 3	-Nx 25.00	-Ny 25.00	-Ne 9.999787
r	-t 0.160	-Hs 3	-Hd -	-Ni 3	-Nx 25.00	-Ny 25.00	-Ne 9.999787
s	-t 0.161	-Hs 6	-Hd -	-Ni 6	-Nx 25.00	-Ny 40.00	-Ne 9.999959
r	-t 0.161	-Hs 3	-Hd -	-Ni 3	-Nx 25.00	-Ny 25.00	-Ne 9.999766
r	-t 0.161	-Hs 1	-Hd -	-Ni 1	-Nx 10.00	-Ny 40.00	-Ne 9.999939
r	-t 0.161	-Hs 0	-Hd -	-Ni 0	-Nx 5.00	-Ny 40.00	-Ne 99.998728
r	-t 0.161	-Hs 3	-Hd -	-Ni 3	-Nx 25.00	-Ny 25.00	-Ne 9.999766
r	-t 0.161	-Hs 1	-Hd -	-Ni 1	-Nx 10.00	-Ny 40.00	-Ne 9.999939
r	-t 0.161	-Hs 0	-Hd -	-Ni 0	-Nx 5.00	-Ny 40.00	-Ne 99.998728

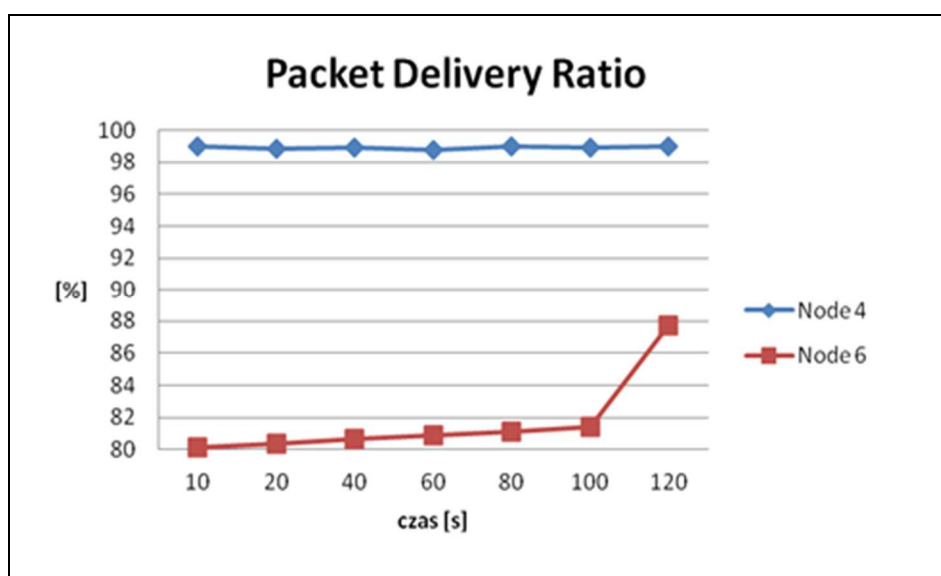
Zebrane dane pozwalają dokonać oceny poprawności działania sieci, czasu bezawaryjnej pracy, współczynnika dostarczania pakietów, zużycia energii węzłów itp. Wyniki w postaci graficznej przedstawiono na rys. 5 i 6:



Rys. 5. Przykładowy wykres pozostałej energii węzłów po 30 sekundach symulacji

Fig. 5. Sample graph of remaining energy in nodes after 30 seconds of simulation

Jak pokazano na rys. 5 największe zużycie energii bo aż 12,4% ma węzeł nr 6. Jego większe zużycie energii jest spowodowane przekazywaniem dużej ilości pakietów do i z węzła nr 3. Dzięki takiemu diagramowi można oszacować maksymalny czas pracy sieci, który w tym przypadku wynosi 241 sekund, po tym czasie sieć przestanie być spójna, a pakiety od węzłów nr: 2, 3 i 7 nie zostaną dostarczone do węzła AP.



Rys. 6. Przykładowy wykres współczynnika dostarczania pakietów dla węzłów 4 i 6

Fig. 6. Sample graph of sums of delay in simulated network for nodes 4 and 6

Kolejnym bardzo ważnym parametrem jest współczynnik dostarczania pakietów (ang. *Packet Delivery Ratio*, PDR), określający ile pakietów zostało poprawnie dostarczonych w porównaniu do liczby wysłanych pakietów. Taki współczynnik pozwala stwierdzić, gdzie leży „wąskie gardło” sieci oraz sprawdzić czy sieć spełnia założenia QoS i działa prawidłowo, nie pomijając istotnych informacji. Rys. 6 przedstawia współczynnik dostarczania pakietów węzłów 4 i 6 dla badanej sieci. Dla analizowanej sieci współczynnik PDR dla węzła 4 kształtuje się w przedziale 98-100%, a dla węzła 6 jest to 80-88%. Taka sieć spełni swoje zadanie, gdyż wartość progowa PDR nie powinna być mniejsza od 80%.

4. Podsumowanie

Bezprzewodowe sieci sensorowe stały się bardzo popularną technologią, zwiększającą komfort życia człowieka. Użycie takiego systemu w systemach pomiarowych pozwala na sprawne i bezpieczne monitorowanie danych takich jak temperatura, dźwięk, ruch itp. Symulatory sieciowe są dobrym narzędziem do badań nad parametrami sieci WSN. NS2 jest jednym z bardziej powszechnych symulatorów sieci WSN, dostarczających bogatą liczbę bibliotek i poradników. Zaletą zastosowania NS2 w projektowaniu i badaniu sieci bezprzewodowej sieci sensorowej jest fakt, że nie jest konieczny zakup fizycznych urządzeń, co znacząco zmniejsza koszty związane z niewłaściwym doбором parametrów urządzeń.

Literatura

- [1] P. Dymora, M. Mazurek, P. Hadaj, *Education set for collecting and visualizing data using sensor system based on AVR microcontroller*, International Journal of Modern Engineering Research (IJMER), Vol. 4, Issue. 10 (Version 2), October 2014, pp. 38-42, ISSN: 2249-6645, 2014.
- [2] <https://www.isi.edu/nsnam/ns/>, [Dostęp 01.10.2017].
- [3] <https://www.nsnam.org/docs/tutorial/html/>, [Dostęp 01.10.2017].
- [4] http://nsnam.sourceforge.net/wiki/index.php/Main_Page, [Dostęp 01.10.2017].
- [5] Issariyakul T., Hossain E.: *Introduction to Network Simulator NS2*, Springer Publisher; 2nd ed. 2012 edition (December 2, 2011).
- [6] <http://www.mannasim.dcc.ufmg.br>, [Dostęp 01.10.2017].
- [7] <http://www.mannasim.dcc.ufmg.br/download/mannasim-classes-manual.pdf>, [Dostęp 01.10.2017].

NS2 - A SIMULATION ENVIRONMENT FOR RESEARCH ON WIRELESS SENSOR NETWORKS

S u m m a r y

Nowadays, with the growing interest in the Internet of Things, in most cases sensor network technology is used. The purpose of this article is to introduce the readers to the possibilities that provides the NS2 simulation environment for the study of wireless sensor networks. The article describes the network simulator NS2, characterized the most essential functions that it meets. In this article will be described a diagram of the WSN networks script, and how to correctly read the results obtained from the simulation.

Keywords: sensors, WSN, NS2, wireless sensor network

DOI: 10.7862/re.2017.14

Tekst złożono w redakcji: wrzesień 2017

Przyjęto do druku: październik 2017

