POLITECHNIKA WARSZAWSKA

DYSCYPLINA NAUKOWA INFORMATYKA TECHNICZNA I TELEKOMUNIKACJA / DZIEDZINA NAUK INŻYNIERYJNO-TECHNICZNYCH

Rozprawa doktorska

mgr inż. Adam Kozłowski

Adaptacyjne zarządzanie komunikacją w przemysłowych, bezprzewodowych sieciach Internetu Rzeczy (IoT)

> Promotor prof. dr hab. inż. Janusz Sosnowski

Warszawa, 2022

Streszczenie

Bezprzewodowe sieci przemysłowe (industrial wireless networks, IWN) zyskują na znaczeniu w związku z rozwojem Przemysłu 4.0. Autonomiczna wymiana informacji bezpośrednio pomiędzy węzłami sieci i odejście od hierarchicznej struktury procesów automatyzacji wymagają efektywnych metod adaptacyjnej optymalizacji konfiguracji IWN pozwalających na wypełnienie stawianych wymagań jakościowych (QoS).

W rozprawie sformułowano problem optymalizacyjny znalezienia między warstwowej konfiguracji IWN w oparciu o zgłaszane kontrakty, wymagania jakościowe QoS (tj. maksymalny czas transmisji, niezawodność, żywotność, przepływność) oraz raporty dotyczące połączeń z sąsiednimi węzłami. Opracowano i poddano analizie zestaw autorskich modeli, metod oraz algorytmów pozwalających na adaptacyjny dobór parametrów konfiguracyjnych w sieciach IWN z węzłami o ograniczonych zasobach (układowych i energetycznych). Charakter opracowania jest uniwersalny ale przyjęte, w badanych implementacjach, definicje (kontraktów, parametrów QoS, priorytetów, rodzajów ruchu) są zgodne ze standardem ISA100.11a. Otrzymane rezultaty mają wymiar praktyczny co zostało potwierdzone w ramach projektu badawczo-rozwojowego RPMA.01.02.00-14-9551/17-00.

Rozwiązanie postawionego problemu optymalizacyjnego wymagało opracowania autorskiej metody wykorzystującej heurystyczny algorytm GA-CAS oraz algorytmy planowania szczelin czasowych dla ruchu cyklicznego CAS-PERIOD oraz sekwencyjnego CAS-SEQ. Zaproponowany algorytm GA-CAS wykorzystuje algorytm genetyczny wraz z autorską metodą oceny. Algorytmy CAS-PERIOD i CAS-SEQ wykorzystują opracowany mechanizm planowania przydziału szczelin czasowych unikający kolizji (CAS) bez konieczności wykonywania planu w długim okresie (wynikającym z wielokrotności okresów kontraktów).

Założenie dotyczące ograniczonych zasobów węzłów IWN wymagało autorskiego podejście w wielu aspektach. Opracowano uniwersalny model energetyczny sieci IWN bazując na pomiarach typowego układu i uwzględniono go w opracowanej metodzie optymalizacyjnej w kontekście czasu obowiązywania kontraktu. Opracowano metodę oraz algorytmy ograniczające wpływ dryftu zegara czasu rzeczywistego (składowa jitter) w przypadku gwałtownych zmian temperatury. W ramach badań przeprowadzono szereg eksperymentów fizycznych oraz symulacyjnych. Do tego opracowano specjalne środowiska sprzętowe oraz programowe i bogaty zestaw testów, które pozwoliły ocenić opracowane rozwiązania w porównaniu z innymi.

Słowa kluczowe: bezprzewodowe sieci przemysłowe, wymagania jakościowe, algorytmy optymalizacyjne, konfiguracja sieci, problemy energetyczne, badania symulacyjne

Abstract

Industrial wireless networks (IWN) are gaining importance due to the rapid development of Industry 4.0. The autonomous exchange of information directly between network nodes and non-use of the hierarchical structure of automation processes require effective adaptive methods to optimise the IWN configuration and meet quality of service (QoS) requirements.

The thesis formulates the problem of finding the optimal cross-layer IWN configuration based on the requested contracts, QoS requirements (e.g. latency, reliability, lifetime, throughput), and reports on links with one-hop neighbours. To this aim, a set of proprietary models, methods, and algorithms were developed and analysed for adaptive selection of configuration parameters in IWN networks with limited resources (hardware and energy). The nature of the study is universal, but the definitions adopted in the tested implementations (contracts, QoS requirements, priorities, types of traffic) are compliant with the ISA100.11a standard. The obtained results have practical applications, which were confirmed in the R&D project RPMA.01.02.00-14-9551/17-00.

Solving the presented optimisation problem required the development of a proprietary method that uses the GA-CAS heuristic algorithm and time slot scheduling algorithms for the periodic CAS-PERIOD and non-periodic CAS-SEQ traffic. The proposed GA-CAS algorithm uses a genetic algorithm with a proprietary evaluation method. The CAS-PERIOD and CAS-SEQ algorithms use the developed mechanism of time slot allocation that avoids collisions (CAS) without the need to evaluate the plan in the long run (resulting from the least common multiple of contract periods).

The limited resources of IWN nodes require a proprietary approach in many aspects. The general energy model of the IWN network was developed based on the measurement of a typical node, and the model was incorporated into the developed optimisation method to meet the QoS requirements for contract lifetime. Both a method and algorithms limiting the impact of real-time clock drift (jitter component) in case of rapid temperature changes were developed. A series of physical and simulation experiments were also conducted. In addition, special hardware and software environments along with an extensive set of tests were created in order to evaluate the developed solutions in comparison with others.

Keywords: industrial wireless networks, quality of service, optimisation algorithms, network configuration, energy related problems, simulations

Spis treści

1	W	Wprowadzenie			
	1.1	Gen	eza i przedmiot rozprawy	7	
	1.2	Cele	e i teza rozprawy	. 10	
	1.3	Ukł	Układ rozprawy		
2	P	Problemy energetyczne i synchronizacii w sieciach IWN			
	2.1	Ana	liza profili energetycznych	. 13	
	2.	.1.1	Stany energetyczne wezła	. 16	
	2.	.1.2	Odbieranie i nadawanie danych	. 18	
	2.	.1.3	Łaczenie komend radiowych	. 19	
	2.2	Syn	chronizacja wezłów sieci IWN	.21	
	2.	.2.1	Badanie efektów przesunięcia czasu i temperatury	. 22	
	2.	.2.2	Adaptacyjna kompensacja przesunięcia czasu	. 25	
	2.	.2.3	Ewaluacja modelu	. 26	
3	А	daptac	yjny dobór parametrów konfiguracji sieci IWN	. 32	
	3.1	Kor	elacja parametrów IWN z wymaganiami QoS	. 33	
	3.	.1.1	Wymagania czasowe	. 33	
	3.	.1.2	Wymagania niezawodnościowe	. 35	
	3.2	Kon	figuracja sieci IWN	. 41	
	3.	.2.1	Definicja kontraktu	. 41	
	3.2.2		Komunikacja cykliczna i komunikacja sekwencyjna	. 43	
	3.2.3		Definicja sieci	. 44	
	3.2.4		Definicja problemu	. 48	
4	K	Koncepcja badań eksperymentalnych		. 53	
	4.1	Śroc	lowisko testowania	. 53	
	4.2	Scei	nariusze testowania	. 58	
	4.	.2.1	Generowanie losowej topologii sieci	. 59	
	4.	.2.2	Generowanie kontraktów	. 59	
	4.	.2.3	Generowanie ruchu w sieci	. 61	
5	А	lgorytı	my planowania rozmieszczenia szczelin czasowych w TDMA	. 63	
	5.1	Spe	cyfika algorytmów	. 63	
	5.	.1.1	Algorytm planowania szczelin AMUS	. 64	
	5.	.1.2	Algorytm planowania szczelin z metrykami LLF lub EDF	. 66	
	5.1.3		Przegląd innych rozwiązań	. 67	

	5.2	Badania eksperymentalne algorytmów				
5.2		2.1 Wyniki porównania algorytmów planowania szczelin				
	5.2	2 Analiza pracy sieci				
	5.3	Algorytm CAS-PERIOD				
	5.3	1 Definicja algorytmu CAS-PERIOD i przyjęte założenia				
5.3.2 5.3.3		2 Ewaluacja wyników planowania algorytmu CAS-PERIOD				
		3 Symulacja pracy sieci dla CAS-PERIOD i ewaluacja wyników				
	5.4	Algorytm CAS-SEQ				
	5.4	1 Definicja algorytmu CAS-SEQ i przyjęte założenia				
	5.4	2 Ewaluacja wyników planowania i symulacja pracy sieci				
6	Op	ymalizacja procesu konfiguracji sieci IWN				
	6.1	Analiza możliwości wykorzystania algorytmów genetycznych				
	6.2	Definicja algorytmu GA-CAS i przyjęte założenia				
	6.2	1 Reprezentacja problemu w algorytmie				
	6.2	2 Operatory genetyczne				
	6.2	3 Funkcja przystosowania				
	6.2	4 Szacowanie wartości funkcji przystosowania na podstawie genotypu				
	6.2	5 Implementacja algorytmu i notacja zapisu wersji algorytmu				
	6.3	Badania eksperymentalne algorytmu GA-CAS				
	6.3	1 Analiza metody in-vitro				
7	Рос	sumowanie				
	7.1	Osiągnięcia pracy i wnioski				
	7.2	Kierunki dalszych badań				
8	Lite	eratura				
A. Dodatki 1						
	A.1.	Struktura zbioru danych dla środowiska badań eksperymentalnych				
	A.2.	Plan aktywności szczelin czasowych TDMA				
	A.3.	Komplementarna dokumentacja wyników ekspervmentalnych				
	A.4.	Lista wariantów algorytmu GA-CAS				
В	. Roz	zszerzona dokumentacja przeprowadzonych badań (CD-ROM)				

1 Wprowadzenie

Obserwowany trend automatyzacji i zwiększenia wymiany danych w technologiach produkcyjnych, zwany czwartą rewolucją przemysłową (Przemysł 4.0) [1, 2] integruje szereg dziedzin w obszarze technologii informacyjnych i komunikacyjnych (ICT): cyber-physical systems (CBS) [3], Internet rzeczy (IoT) [4], cloud computing, edge computing [5, 6]. Przemysł 4.0 dąży do zwiększenia produktywności [1, 3, 7, 8] poprzez integrację urządzeń zakładu przemysłowego (ang. smart factory). W tym procesie istotną rolę odgrywają bezprzewodowe sieci przemysłowe (IWN) oraz efektywne zarzadzanie przesyłaniem informacji. Ten ostatni problem stał się przedmiotem moich badań.

1.1 Geneza i przedmiot rozprawy

Kluczowym elementem funkcjonowania "inteligentnej" fabryki (ang. smart factory), czyli autonomicznych i rozproszonych systemów automatyki przemysłowej jest niezawodna wymiana i przetwarzanie informacji. Komunikacja i różnego rodzaju standardy komunikacyjne odgrywają tutaj kluczową rolę [1, 9]. Osiągnięcie celów wymaga odejścia od hierarchicznej struktury procesów automatyzacji w kierunku integracji dwóch modeli komunikacji [10]: wykorzystanie sieci urządzeń o większej autonomiczności komunikujących się bezpośrednio ze sobą (ang. machine-to-machine, M2M) [10, 11] oraz integracja z systemami przetwarzania w chmurze. Interoperacyjność w ramach tych dwóch modeli komunikacji nazwanych odpowiednio komunikacja horyzontalna i wertykalna daje szanse na dalsze zwiększanie produktywności [12]. Koncepcja Przemysłu 4.0 została dogłębnie omówiona w literaturze, natomiast brak jest konkretnych standardów opisujących jej implementację na dużą skalę w praktyce. Wdrażanie technologii bezprzewodowej jest wciąż na wczesnym etapie szczególnie w bardziej krytycznych przypadkach, np. kontroli i ochrony w pętli zamkniętej, gdzie awaria komunikacji może spowodować poważne wypadki [13]. Jako kluczowy problem w rozwoju technologii bezprzewodowej wskazywany jest brak rozwiązań pozwalających na praktyczne wypełnienie wymagań QoS stawianych rozwiązaniom przemysłowym [7]. Do głównych wymagań QoS należy zaliczyć: maksymalny czas transmisji między punktami docelowymi, niezawodność, żywotność, przepływność, jitter.

Efektywne rozwiązanie tego problemu wymaga opracowania metod pozwalających na niezawodne i niezwłoczne przygotowanie konfiguracji sieci w taki sposób aby wszystkie warstwy stosu komunikacyjnego były optymalizowane w sposób zintegrowany [14–16]. Poprzez optymalizację i konfigurację między-warstwową (ang. cross-layer), dążymy do sytuacji w której protokoły i algorytmy warstw stosu komunikacyjnego są do siebie dopasowane i wymieniają wzajemnie informacje, aby usprawnić mikrozarządzanie [17]. Połączenie problemów optymalizacji występujących na różnych warstwach w odniesieniu do

7

wymagań QoS pozwala na ich bardziej efektywne rozwiązania niż rozwiązywanie tych problemów sekwencyjnie, warstwa po warstwie. Ograniczenia sprzętowe (moc obliczeniowa, pamięć operacyjna) oraz energetyczne (praca autonomiczna) stanowią istotny problem w praktycznych zastosowaniach IWN. Dodatkowo trudne warunki środowiskowe oraz ich zmienność muszą zostać uwzględnione w procesie optymalizacji konfiguracji sieci. Problemy optymalizacji, takie jak planowanie szczelin czasowych występujące w warstwie MAC lub routing w warstwie sieciowej zapewniający jednoczesne wypełnienie wielu wymagań QoS, są trudne do rozwiązania nawet osobno (problemy z klasy NP-zupełne dla których rozwiązanie nie jest możliwe w czasie wielomianowym) [17, 18]. Dodatkowo, ich wzajemne uwikłanie zwiększa złożoność problemu ze względu na to, że nie można ocenić jakości znalezionych tras bez wykonania planowania szczelin czasowych i odwrotnie. Heurystyczne metody optymalizacji są wymagane, aby szybko znaleźć prawidłowe i zoptymalizowane rozwiązania w tych ogromnych przestrzeniach potencjalnych rozwiązań, respektując jednocześnie wszystkie wymagania QoS oraz ograniczone zasoby elementów sieci.

Rygorystyczne wymagania QoS dotyczące wysokiej niezawodności, determinizmu czasowego oraz oczekiwanego niskiego zużycia energii powodują, że rozwiązania wykorzystujące komunikację synchroniczną [19] oraz scentralizowane zarządzanie [20], są bardziej efektywne w sieciach IWN. Zarządzanie siecią IWN wymaga przygotowania konfiguracji oraz jej adaptacji do zmieniających się warunków oraz wymagań zgłaszanych przez poszczególne procesy aplikacyjne (kontrakty QoS). Przygotowanie optymalnej konfiguracji IWN jest procesem złożonym wymagającym zarówno zasobów sprzętowych jak i energetycznych, dlatego szybkie i efektywne metody przygotowania oraz dostarczenia konfiguracji są elementem warunkującym uzyskanie wymaganej adaptacyjności. Dodatkowo, w procesie optymalizacji konfiguracji możemy stosować metody z pamięcią, bazujące na korelacji wymagań jakościowych z już wytworzonymi konfiguracjami. Zarówno metody oparte o uczenie maszynowe [21] jak również wykorzystujące dane uzyskane w procesie symulacji [22] pozwalają na efektywną korelację jedynie kilku wybranych parametrów konfiguracji oraz nie uwzględniają ograniczeń zasobów sieci IWN. Zastosowania przemysłowe wymagają ujęcia w procesie adaptacji parametrów, takich jak plan aktywności węzłów (szczelin czasowych) oraz trasy, kluczowych do zapewnienia wykorzystywanych wymagań QoS. Stosowanie komunikacji synchronicznej wymaga efektywnych metod synchronizacji węzłów oraz zapewnienia stabilności lokalnych wzorców czasu również przy zmieniających się warunkach otoczenia w szczególności gwałtownych zmianach temperatury [23]. Duży poziom autonomiczności węzłów IWN oraz wynikające z tego ograniczone zasoby energetyczne powodują konieczność uwzględnienia bilansu energetycznego sieci [24] w procesie

optymalizacji i adaptacji konfiguracji (modele energetyczne, identyfikacja krytycznych operacji).

Przedmiotem rozprawy są efektywne metody oraz algorytmy adaptacyjnego zarządzania konfiguracją bezprzewodowej sieci przemysłowej (IWN) [25], składającą się z urządzeń o ograniczonych zasobach. Mimo, iż istnieje kilka standardów dedykowanych dla IWN i definiujących warstwy stosu komunikacyjnego, w tym warstwę dostępu do medium (MAC) oraz warstwę fizyczną (PHY) (WirelessHART [26], ISA100.11a [27], TSCH [28]), nie ma kompleksowych rozwiązań pozwalających na między-warstwową konfigurację sieci przy spełnieniu szerokiego spektrum wymagań QoS formułowanych przez poszczególne aplikacje biorące udział w komunikacji wewnątrz IWN.

W pracy analizowana jest typowa sieć IWN, zarządzana centralnie poprzez dedykowany węzeł pełniący rolę menadżera sieci (SM). IWN dysponuje ograniczonymi zasobami zarówno w odniesieniu do SM i węzłów (mała moc obliczeniowa, mała pamięć operacyjna) jak również w odniesieniu do energii niezbędnej do ich funkcjonowania (zasilanie autonomiczne) oraz zasobów radiowych, gdzie wiele urządzeń korzysta ze wspólnej przestrzeni radiowej. Poprzez urządzenie o ograniczonych zasobach, w analizowanym problemie, należy rozumieć przypadek w którym zasoby urządzenia są wyraźnie ograniczone w stosunku do celów jakie przed nim stoją. Ograniczenie jest na tyle wyraźne, że wymaga stosowania dedykowanych technik do realizacji celów, takich które nie byłyby konieczne do stosowania gdyby ograniczenie nie występowało.

Każdy węzeł sieci (węzeł wnioskujący będący węzłem źródłowym) może wnioskować do SM o zawarcie kontraktu na komunikację z wybranym węzłem docelowym. Węzeł wnioskujący podaje w kontrakcie szczegóły dotyczące planowanej komunikacji w szczególności dotyczące stawianych wymagań QoS. SM w oparciu o wnioskowane i aktywne kontrakty oraz w oparciu o raporty dotyczące jakości połączeń węzłów z sąsiadami podejmuje decyzję o zawarciu lub odrzuceniu kontraktu. Zadaniem SM jest taki dobór parametrów sieci aby jak największa liczba kontraktów mogła być zawarta, z uwzględnieniem ich priorytetu. Postawiony problem wymaga opracowania efektywnych algorytmów pozwalających na przygotowanie pełnej konfiguracji sieci (tablica routingu i harmonogram transmisji) jak również jej optymalizację przy spełnieniu postawionych wymagań QoS oraz uwzględniając szybko zmieniające się warunki. Szybkie zmiany temperatury otoczenia, mobilność węzłów oraz trudne i zmienne warunki przemysłowe wymagają szybkiej reakcji i adaptacji parametrów sieci do nowych warunków. Opracowane algorytmy muszą uwzględniać specyfikę sieci wynikającą zarówno z ograniczonych zasobów (układowych i energetycznych) jak również z szybko zmieniających się trudnych warunków przemysłowych. Implementacja opracowanych algorytmów zakłada pracę IWN w oparciu o

rzeczywiste urządzenia (opracowane w ramach projektu badawczego którego jestem kierownikiem) oraz warunki pracy.

1.2 Cele i teza rozprawy

Przeprowadzone badania literaturowe oraz praktyczne doświadczenie autora stanowią motywację do powstania pracy co pozwoliło na sformułowanie tezy oraz zakresu prowadzonych badań.

Teza rozprawy:

Zaproponowany adaptacyjny dobór parametrów konfiguracyjnych w sieciach IWN z węzłami o ograniczonych zasobach, pozwala na kompleksowe wypełnienie wielowymiarowych wymagań jakościowych QoS oraz optymalizację zawieranych kontraktów komunikacyjnych.

Celem pracy było opracowanie i analiza modeli, metod oraz algorytmów pozwalających na efektywne zarządzanie siecią IWN składającą się z urządzeń o ograniczonych zasobach. Cel ten osiągnięto poprzez szereg szczegółowych badań dotyczących specyfiki transmisji w sieciach IWN na różnych poziomach ich projektowania. Badania te obejmowały rozpoznanie i identyfikację krytycznych aspektów w projektowaniu sieci IWN. Między innymi uwzględniały one analizę profilu energetycznego węzłów sieci oraz możliwości jego optymalizacji, analizę wymagań czasowych i niezawodnościowych oraz problemów zarządzania/planowania obsługi kontraktów przy wielowymiarowych wymaganiach jakościowych QoS. Wynikiem tych analiz było sformułowanie uogólnionego problemu konfigurowania i zarządzania siecią dla szerokiego spektrum parametrów. Problemy energetyczne były przedmiotem badań, które opisałem szczegółowo w publikacjach [19, 23, 24, 29], stąd też w pracy ograniczono się do przedstawienia najważniejszych wyników. Specyfikacja opracowanych algorytmów dla tego problemu jest dostępna w trybie open access [23] i pominięto ją w pracy. W ramach trzech powiązanych ze sobą zagadnień rozpatrywanych w pracy przeprowadzono analizę literatury, która jest raportowana w poszczególnych rozdziałach poświęconych tym zagadnieniom.

Istotnym elementem pracy było opracowanie systemu symulacji działania sieci pozwalającego na badanie jej własności dla bogatego zestawu testów. Wielowymiarowy zestaw testów dobrze odzwierciedla specyfikę sieci IWN. Opracowane środowisko testowe zostało wykorzystane do analizy znanych rozwiązań z literatury, identyfikacji ich niedostatków oraz wskazanie możliwych usprawnień. Wymagało to zamodelowania szeregu wariantów sieci oraz przeprowadzenia eksperymentów umożliwiających eksplorację zaproponowanych parametrów dla znanych wariantów konfiguracji sieci (w tym autorskie modyfikacje). Te badania pozwoliły na opracowanie oryginalnych algorytmów deterministycznych planowania szczelin czasowych dla kontraktów komunikacji cyklicznej i sekwencyjnej (CAS-PERIOD, CAS-SEQ). Ewaluacja tych algorytmów w środowisku testowym potwierdziła ich efektywność w zakresie znaczącego

zmniejszenia czasu konfigurowania (co jest istotne w adaptacji do zmiennych warunków pracy węzłów i kanałów komunikacyjnych), lepszych parametrów transmisyjnych oraz bardziej racjonalnego wykorzystania zasobów komunikacyjnych w obsłudze kontraktów poprzez wprowadzoną koncepcję ich grupowania.

Algorytmy planowania stały się podstawą do opracowania kompleksowego algorytmu genetycznego (GA-CAS) optymalizującego zarządzanie konfiguracją sieci z uwzględnieniem dwu typów kontraktów (cykliczne i sekwencyjne) i szerokiego spektrum wymagań QoS, co istotnie wyróżnia zaproponowane rozwiązanie od opisanych w literaturze. Przedstawiony algorytm został przebadany w środowisku testowym dla wielu parametrów sieci i węzłów uwzględniających własności typowych rozwiązań technicznych zgodnych ze standardem komunikacji bezprzewodowej ISA100.11a (norma PN-EN 62734:2015). Badania te potwierdziły jego wysoką efektywność oraz korzyści w stosunku do innych rozwiązań z literatury.

Przeprowadzone badania obejmują aspekty teoretyczne, eksperymentalne oraz praktyczne. Opracowane w ramach dysertacji metody i algorytmy zostały zaimplementowane oraz wdrożone w środowisku rzeczywistym z wykorzystaniem urządzeń wytworzonych w kierowanym przez autora projekcie badawczo-rozwojowym (dofinansowany w ramach projektu RPMA.01.02.00-14-9551/17-00 w wyniku konkursu, 1,5 mln PLN). Jego celem było opracowanie stosu komunikacyjnego do zastosowań przemysłowych (TronixIWN) oraz wytworzenie niezbędnych prototypów urządzeń tworzących IWN (GATEWAY_4B, SENSOR_NODE_1C).

1.3 Układ rozprawy

Rozprawa złożona jest z siedmiu rozdziałów, bibliografii, dodatków oraz rozszerzonej dokumentacji na nośniku CD-ROM. Rozdział pierwszy poświęcono nakreśleniu genezy pracy oraz sformułowaniu tezy i celu rozprawy. Przegląd literatury na poziomie podstawowym przedstawiono w rozdziale 1.1 oraz rozszerzony i dopasowany do specyfiki poszczególnych problemów w rozdziałach 5.1, 6.1 oraz artykułach [19, 23, 24].

W rozdziale 2 przedstawiono, w sposób syntetyczny i uogólniony, już opublikowane wyniki badań dotyczące problemów energetycznych [24] oraz synchronizacji czasu [23] w sieciach IWN. Analiza profili energetycznych (rozdział 2.1) poświęcona jest identyfikacji krytycznych operacji w sieci IWN z wykorzystaniem typowych węzłów oraz prezentacji modelu energetycznego sieci IWN. Przedstawiony w rozdziale 2.2 problem synchronizacji węzłów jest analizowany w odniesieniu do węzłów o ograniczonych zasobach i narażonych na gwałtowne zmiany temperatury otoczenia. W rozdziale 2.2 przedstawiono opracowaną metodę

(algorytmy) wykorzystującą predykcję temperatury z użyciem regresji liniowej. W rozdziale 2.2.3 przedstawiono wyniki badań eksperymentalnych i symulacyjnych.

W rozdziałach 3-6 przedstawiono problem adaptacyjnego doboru parametrów optymalnej konfiguracji wypełniającej wymagania QoS dla sieci IWN o ograniczonych zasobach (sprzętowych i energetycznych). Wykorzystano wyniki badań z rozdziału 2 (model energetyczny, minimalizacja jitter). W rozdziale 3 sformułowano problem optymalizacji konfiguracji sieci IWN oraz przedstawiono analizę wymagań QoS (rozdział 3.1) czasowych i niezawodnościowych.

Rozdział 4 prezentuje autorskie środowisko badań eksperymentalnych dedykowane określeniu wpływu wielu parametrów na proces tworzenia konfiguracji IWN. Badanie podzielono na dwie fazy: tworzenie konfiguracji na podstawie opracowanych wektorów testowych, symulację pracy sieci w oparciu o utworzone konfiguracje. W rozdziale 4.1 przedstawiono opis środowiska a definicję wykorzystywanych wektorów testowych oraz metodykę badań w rozdziale 4.2.

W rozdziale 5 przedstawiono problematykę przygotowywania planu aktywności węzłów (szczelin czasowych) w sieciach TDMA/FDMA. W rozdziale 5.2 zaprezentowano wyniki badań eksperymentalnych dla typowych algorytmów oraz ich modyfikacji. Opracowane autorskie algorytmy zaprezentowano w rozdziałach 5.3 (CAS-PERIOD) oraz 5.4 (CAS-SEQ) dla odpowiednio ruchu cyklicznego i sekwencyjnego. Algorytmy poddano wnikliwej analizie w oparciu o wyniki uzyskane podczas ewaluacji z wykorzystaniem środowiska testowego.

W rozdziale 6 przedstawiono całościowe rozwiązanie zdefiniowanego problemu optymalizacyjnego, wykorzystujące opracowany autorski algorytm GA-CAS (rozdział 6.2) oraz algorytmy CAS-PERIOD i CAS-SEQ. W rozdziale 6.3 zaprezentowano wyniki badań eksperymentalnych w których porównano różne warianty GA-CAS wraz z analizą wpływu parametrów na efektywność optymalizacji.

Podsumowanie wyników pracy oraz możliwości dalszych badań przedstawiono w rozdziale 7. Komplementarne wyniki eksperymentów (ilustracje graficzne) umieszczono w dodatkach. Rozszerzona dokumentacja przeprowadzonych badań (głównie dane liczbowe) jest przedstawiona na załączonej płycie CD ROM.

2 Problemy energetyczne i synchronizacji w sieciach IWN

Analizowana w pracy sieć IWN dysponuje ograniczonymi zasobami zarówno sprzętowymi (mała moc obliczeniowa, mała pamięć operacyjna) jak również energetycznymi (zasilanie autonomiczne). Ograniczone zasoby determinują dobór właściwych metod i algorytmów pozwalających na zarządzanie siecią IWN. W rozdziale prezentuję szczegółową analizę problemów wynikających z istnienia ograniczeń zasobów.

Opracowany model energetyczny sieci IWN prezentuję w rozdziale 2.1. Model bazuje na pomiarach typowego węzła IWN i uwzględnia identyfikację krytycznych operacji na węzłach sieci. Do charakteryzacji wykorzystano fizyczne węzły wytworzone w ramach realizacji projektu B+R. Opracowany model energetyczny został uwzględniony w sformułowanym problemie optymalizacyjnym (rozdział 3.2) oraz w opracowanej metodzie optymalizacyjnej (rozdział 6) i stanowi bazę do wypełnienia jednego z wymagań QoS dotyczącego czasu obowiązywania kontraktu.

Występujące ograniczenia oraz potrzeba implementacji opracowanych rozwiązań narzuciły konieczność opracowania autorskiego modelu synchronizacji węzłów sieci przy narażeniu na gwałtowne zmiany temperatury. W rozdziale 2.2 zbadałem wpływ temperatury otoczenia (badania eksperymentalne) na zdolność węzła do wypełnienia wymagań QoS. Poczynione obserwacje w połączeniu z wykonaną charakteryzacją węzłów sieci pozwoliły na opracowanie autorskiej metody i algorytmów (rozdział 2.2.2) pozwalających na aktywne ograniczenie wpływu dryftu zegara czasu rzeczywistego w przypadku gwałtownych zmian temperatury otoczenia dla węzłów o ograniczonych zasobach w szczególności ograniczonym bilansie energetycznym. W rozdziale 2.2.3 prezentuję badania eksperymentalne i symulacyjne potwierdzające efektywność zaproponowanych rozwiązań i ich korzyści w porównaniu z innymi publikowanymi rozwiązaniami.

2.1 Analiza profili energetycznych

Efektywne opracowanie algorytmów i protokołów dla IWN wymaga znajomości właściwości fizycznych urządzeń w konkretnych praktycznych scenariuszach. Ważnym zagadnieniem jest analiza profili zużycia energii w odniesieniu do aktywności węzła (związanych z przetwarzaniem danych, pamięcią i komunikacją radiową) oraz możliwych trybów pracy. Ten problem jest pomijany w literaturze. Korzystanie z zasilanych bateryjnie i miniaturowych węzłów jest niezbędne do wykorzystania wszystkich zalet technologii IWN. W wielu branżach tylko całkowite wyeliminowanie okablowania może zachęcić instalatorów i właścicieli zakładów do inwestowania i zmiany ich obecnych technologii przewodowych w kierunku technologii IWN. Dlatego projektując IWN składające się z węzłów o ograniczonych zasobach i zasilanych bateryjnie, oczekujemy bezobsługowej pracy przez co najmniej 5 lat. Urządzenie

zasilane z pojedynczej baterii AA np. LS 14500 SAFT, posiada ograniczony budżet energetyczny, gdzie nominalna energia jest równa 9,36Wh. Biorąc pod uwagę, że węzły pracują na wolnym powietrzu przy temperaturach od -20 °C do 55 °C a pobierany prąd waha się od 10mA do 20mA, możemy przyjąć, że energia wynosi 50% pojemności nominalnej. Odpowiada to średniemu zużyciu energii 100 μ W przy założeniu pięcioletniej żywotności. Budżet energetyczny jest ważnym czynnikiem warunkującym dobór i zastosowanie zarówno komponentów, jak i metod dla IWN. Obok zużycia energii, zależności czasowe przy realizacji konkretnych sekwencji operacji są istotnym elementem, który należy brać pod uwagę projektując i optymalizując IWN. Wymagania czasowe dotyczące konkretnych operacji wykonywanych przez warstwę fizyczną stosu sieciowego dla przykładowego standardu ISA100.11a umieściłem w tabeli B.1.

W rozdziale przedstawiam oryginalną metodologię charakteryzacji profili zużycia energii, która opiera się na określeniu podstawowych parametrów i ich prezentacji w postaci diagramu stanów dla węzła sieci [24]. Reprezentatywne parametry określam dla typowego węzła IWN. Parametry te wraz z modelem są pomocne przy tworzeniu operacyjnych modeli zużycia energii dla praktycznych aplikacji docelowych. W przeprowadzonych eksperymentach koncentruję się na problematyce komunikacji radiowej z uwzględnieniem mechanizmu "duty-cycling" [30, 31], który jest powszechnie wykorzystywany do zapewnienia małego całkowitego zużycia energii. Analizuję wpływ parametrów transmisji oraz czas potrzebny na przejście w stan uśpienia i wybudzenie (dla różnych wersji oprogramowania). Pomiary wykonuję dla typowego węzła IWN pracującego pod kontrolą układu CC1352R1 produkcji firmy Texas Instruments (TI). Pomiary wykonałem z użyciem modułu *EnergyTrace* stanowiacego element sprzetowego debugger'a XDS110 produkcji TI. Charakteryzację wykonaną dla układu CC1310F128 będącego poprzednikiem CC1352R1 zaprezentowałem w artykule [24]. W artykule określenie zużycia energii realizowane jest w odmienny sposób poprzez pomiar zmian napięcia bezpośrednio na rezystorze precyzyjnym (10Ω) z wykorzystaniem oscyloskopu z 10 bitową rozdzielczością w osi Y (RTB2004 Rohde&Schwarz). Moduł EnergyTrace dokonuje pomiaru również na rezystorze precyzyjnym, ale o mniejszej wartości nominalnej (1Ω) oraz dodatkowo wykorzystuje zespół precyzyjnych i niskoszumnych wzmacniaczy operacyjnych do wstępnego wzmocnienia sygnału bezpośrednio w układzie pomiarowym, co uwidacznia się w postaci krótszych stałych czasowych RC na wykresach. Moduł EnergyTrace dysponuje mniejszą rozdzielczością czasu w stosunku do akwizycji z wykorzystaniem oscyloskopu cyfrowego.

Na podstawie przeprowadzonej analizy wytypowałem istotne z punktu widzenia opracowywanego algorytmu parametry oraz scenariusze pracy urządzeń. Zużycie energii i czas trwania poszczególnych czynności określiłem z wykorzystaniem opracowanego dedykowanego programu, który realizuje konkretny scenariusz pracy urządzeń i pozwala na

zbieranie danych (zależność mocy chwilowej i czasu) potrzebnych do późniejszej analizy. Opracowany program wykorzystuje komponenty programistyczne (SDK) dostarczone przez TI i dedykowane dla testowanej rodziny urządzeń (SimpleLink CC13x2 26x2 SDK – 5.10.00.48). Programy wykorzystują mechanizm "Power policy" który jest elementem SDK. Celem mechanizmu jest podjęcie decyzji dotyczącej oszczędzania energii, gdy procesor jest w stanie bezczynności. Procesor jest uważany za bezczynny, gdy wykonywana jest pętla bezczynności systemu operacyjnego, gdy wszystkie wątki aplikacji są blokowane w oczekiwaniu na operacje we/wy lub blokowane w oczekiwaniu na inne zdarzenia aplikacyjne. Wszystkie pomiary zostały wykonane przy użyciu agresywnej wersji polityki "*Power_standbyPolicy()*" zawartej w kolekcji sterowników TI. Strategia ta wymusza na urządzeniu przejście do trybu STANDBY, gdy tylko jest to możliwe. Przejście do trybu STANDBY nie jest możliwe, jeśli jednostka radiowa pracuje lub gdy czas do następnego zaplanowanego zadania jest zbyt krótki (mniejszy od predefiniowanego czasu wynoszącego 1000µs), w takich przypadkach urządzenie automatycznie przełącza się w tryb IDLE.

W opracowanym oprogramowaniu zastosowałem tryb interakcji z modułem radiowym o nazwie "proprietary". Tryb daje dużą elastyczność w tworzeniu własnej struktury ramek i mechanizmu łączenia poleceń. W tabeli B.2 wymieniłem użyte polecenia bloku RF wraz z ich krótkim opisem.

Opracowane oprogramowanie korzysta z systemu operacyjnego czasu rzeczywistego (RTOS) dostarczonego przez TI o nazwie TI-RTOS. Jądro TI-RTOS wykorzystuje koncepcję mikrojądra, dlatego zapewnia minimalną funkcjonalność systemu operacyjnego, którą można dalej rozszerzać za pomocą modułów i sterowników, które zapewniają abstrakcyjny, funkcjonalny dostęp do urządzeń peryferyjnych. TI-RTOS jest częścią oficjalnego zestawu programistycznego (SDK) dostarczanego przez TI. Używam następujących sterowników dla CC26x2/CC13x2, aby rozszerzyć podstawową funkcjonalność i uzyskać dostęp do wymaganych urządzeń peryferyjnych:

- <ti/drivers/rf/RF.h> Sterownik modułu radiowego.
- <ti/drivers/power/PowerCC26XX.h> Sterownik zarządzania zasilaniem
- <ti/drivers/PIN.h> Sterownik portów dla CC26X2/CC13X2

Ponadto w jednym scenariuszu wykorzystuję aplikację typu "bare-metal" w celu porównania czasów przejścia przy wybudzaniu urządzenia. Aplikacja "bare-metal" nie wykorzystuje systemu czasu rzeczywistego a jedynie szkielet o nazwie NoRTOS dostarczony przez TI (NoRTOS.h) i bazuje na tej samej kolekcji sterowników. Takie zestawienie pozwala na porównanie narzutu systemu czasu rzeczywistego na ten konkretny scenariusz.

W kolejnych podrozdziałach prezentuję pomiary osobno dla każdego scenariusza pracy.

2.1.1 Stany energetyczne węzła

Węzły IWN budzą się okresowo ze stanu głębokiego uśpienia, aby mogły nawiązać ze sobą komunikację. Procedurę wybudzenia wyzwala przerwanie z zegara RTC, który działa również w stanie głębokiego uśpienia. Układy z rodziny CC13x2 posiadają zegar czasu rzeczywistego taktowany dodatkowym, oscylatorem niskiej częstotliwości (LF). Oscylator LF może wykorzystywać wewnętrzny obwód RC lub zewnętrzny kryształ dla większej dokładności.



Rysunek 2.1 Zużycie energii przy przejściu pomiędzy trybami STANDBY→ACTIVE→STANDBY dla TI-RTOS (lewy) oraz "bare-metal" (prawy)

W przypadku układów które wykorzystuje w badaniach trybem pracy który zapewnia najniższy pobór energii i który jednocześnie pozwala na wybudzenia z wykorzystaniem zegara RTC jest STANDBY. W praktycznych aplikacjach węzły IWN przebywają w stanie STANDBY przez większość czasu i jedynie cyklicznie przechodzą w tryb ACTIVE w celu wykonania niezbędnych zadań. Dodatkowo tryb IDLE (wyłączone CPU) można wykorzystać do zmniejszenia zużycia energii, przykładowo podczas słuchania nadchodzącej transmisji. Zużycie energii podczas procedury wybudzenia STANDBY \rightarrow ACTIVE i natychmiastowego przejścia z powrotem do trybu głębokiego uśpienia pokazano na rysunku 2.1. Pomiary wykonano dla wariantu pod kontrolą TI-RTOS oraz jako aplikacja "bare-metal". Oba warianty używają tego samego zestawu sterowników TI, opisanego w poprzednim rozdziale.

Analizując pomiary można wyróżnić kilka faz budzenia się i usypiania węzła. Faza 0 kończy się po powrocie z instrukcji oczekiwania na przerwanie (WFI). Długość tej fazy zależy wyłącznie od fizycznych właściwości układu. Fazy 1, 2 i 3 tworzą sekwencję uruchamiania oprogramowania, po której program wraca do głównej procedury, natomiast fazy 4 i 5 tworzą sekwencję przejścia w stan uśpienia, która ostatecznie kończy się instrukcją WFI. Poszczególne fazy są identyfikowane za pomocą mechanizmu powiadomień dostępnego w sterowniku PowerCC26XX.h. Mechanizm pozwala na rejestrację funkcji, która zostanie wywołana w

przypadku zaistnienia określonego zdarzenia w module zarządzania zasilaniem. Powiadomienia są rejestrowane za pomocą funkcji API "Power registerNotify()". Zakończenie fazy 1 i 2 jest określone odpowiednio przez zdarzenia PowerCC26XX AWAKE STANDBY i PowerCC26XX AWAKE STANDBY LATE. Drugie zdarzenie jest wysyłane później podczas budzenia, po ponownym włączeniu przerwań. Podczas sekwencji usypiania zdarzenie PowerCC26XX ENTERING STANDBY kończy fazę 4. Pełną sekwencję uruchamiania można znaleźć w nocie aplikacyjnej TI "CC26x0, CC13x0 SimpleLinkTM Wireless MCU Power Management Software Development" [32]. Czasy trwania poszczególnych faz oraz uzyskane zużycie energii umieściłem w tabeli B.3.

Porównując pomiary wykonane dla poprzedniego modelu CC13x0 [24] z bieżącymi pomiarami dla CC13x2, należy zwrócić uwagę na skrócenie czasów i zmniejszenie zużycia energii w fazach bezpośrednio związanych z programową realizację sekwencji. W dużej mierze wynika to z zastosowania szybszej architektury MCU w nowszej wersji układu, ale również wyraźnie widać optymalizację kodu szczególnie w wersji pod kontrolą TI-RTOS (około 3 lata między pomiarami). Narzut TI-RTOS w porównaniu do aplikacji typu "bare-metal" podczas procedury STANDBY \rightarrow ACTIVE \rightarrow STANDBY jest stosunkowo mały i równy 8% pod względem czasu i 6% pod względem energii. Rezultat jest lepszy od uzyskanego przy poprzednich pomiarach i wynoszącego odpowiednio 10% i 17%.

Warto, aby w tym miejscu przyjrzeć się bliżej fazie 0, która stanowi około 50% zarówno czasu jak i energii potrzebnej do wykonania scenariusza. Dominującym elementem tej fazy jest pierwszy impuls, związany z koniecznością uzupełnienia energii zgromadzonej w zewnętrznym kondensatorze VDDR. Energia zgromadzona w kondensatorze wykorzystywana jest głównie do podtrzymania bloku pamięci RAM i zapewnienia zasilania dla modułów z domeny "always-on" do których zaliczamy RTC. Impuls ten jest niezależny od samej sekwencji wybudzenia i jest ściśle związany z pracą układu w trybie STANDBY. Pobór prądu w trybie STANDBY stanowią dwie składowe: cykliczny impulsu prądu ładowania, służącego do ładowania kondensatora VDDR oraz pobór prądu pomiędzy ładowaniami. Ta druga składowa jest o wiele poniżej wartości, które mogłem zmierzyć i wynosi około 70nA (według informacji od TI). Średni pobór prądu w trybie STANDBY wraz z ładowaniem kondensatora VDDR, wynosi około 1µA, dla konfiguracji z zewnętrznym kryształem. Sterowanie impulsem ładowania jest w pełni adaptacyjne i wykorzystuje w rodzinie CC13x2 wbudowany komparator, co daje optymalny interwał ładowania. W starszej rodzinie CC13x0 czas ładowania dobierany był programowo bazując na wymaganym czasie przebywania w trybie STANDBY.

W wykonanych pomiarach sekwencja dla scenariusza była wykonywana co około 500ms, co pokrywa się z impulsem ładowania, dlatego wybudzenie zawierało dominujący impuls ładujący. Jeżeli sekwencja wybudzania realizowana jest częściej (np. co drugą szczelinę czasową), doładowanie kondensatora nie jest konieczne za każdym razem lub jego wartość jest ograniczona.

2.1.2 Odbieranie i nadawanie danych

Podstawowym zadaniem węzła sieci IWN jest odbieranie i nadawanie ramek danych. Ponieważ komunikacja radiowa realizowana jest w szczelinach czasowych układy radiowe podobnie jak całe urządzenie będą włączone i wyłączone cyklicznie. Odbieranie i nadawanie danych po wybudzeniu urządzenia z trybu STANDBY wymaga wykonania sekwencji włączania modułu radiowego (RF). Domyślnie sterownik RF mierzy czas potrzebny na realizację sekwencji włączenia zasilania i wykorzystuje go jako oszacowanie dla następnego cyklu. W układach z serii CC13x0 uruchomienie radia i wybudzenie ze STANDBY trwało około 1,6 ms (faza0 + faza1), podczas gdy w serii CC13x2 czas ten został znacząco skrócony i wynosi około 0,8 ms. Dodatkowo sterownik RF wstawia margines czasu, oznaczony jako faza 3, aby pokryć wszelkie niepewności podczas wykonywania automatycznych pomiarów. Jest to konieczne zwłaszcza w aplikacjach z dużym obciążeniem przerwaniami sprzętowymi, które mogą opóźnić wykonanie wewnętrznej maszyny stanu sterownika RF. Automatyczny pomiar czasu potrzebnego na włączenie bloku radiowego można wyłączyć, określając niestandardowy czas włączenia. Zużycie energii podczas odbierania i nadawania ramki o całkowitej długości 32B i przepływności 400kBodów/s, pokazano na rysunku 2.2. W tym konkretnym scenariuszu użyłem przepływności większej niż podstawowa w standardzie 802.15.4, czyli 250kBodów/s, aby wizualnie skrócić czas trwania fazy transmisji i uwypuklić szczegóły pozostałych faz wyodrębnionych w scenariuszu.

Poszczególne fazy są identyfikowane za pomocą przerwań rdzenia RF. Rdzeń RF ma cztery linie przerwań do MCU, które są mapowane na zdarzenia systemowe. Koniec fazy 2 został określony na podstawie zdarzenia "*RF_EventPowerUp*". Zdarzenie "*RF_EventCmdDone*" pozwoliło określić koniec fazy 5. Czas trwania i odpowiednie zużycie energii w każdej fazie przedstawiam w tabeli B.4. Dodatkowo podano krótki opis każdej fazy, a całkowity czas i energię przedstawiono w ostatniej kolumnie. Faza 4' jest specyficzna dla trybu odbioru, w tej fazie właściwa transmisja z nadajnika jeszcze się nie rozpoczęła, a węzeł jedynie nasłuchuje kanału transmisyjnego oczekując na właściwą umówioną transmisję. W praktycznych zastosowaniach czas trwania fazy 4' jest różny i zależy od samego protokołu MAC, a także od dokładności zegarów RTC w węzłach. Obecność tej fazy (ang. idle-listening) jest zjawiskiem nieporządnym i w praktyce dążymy do jego minimalizacji, mając na uwadze kompromis

pomiędzy zużyciem energii w tej fazie a energii potrzebnej na zapewnienie synchronizacji urządzeń co pozwala na ograniczenie jej długości.



Rysunek 2.2 Zużycie energii podczas odbierania (lewy) oraz nadawania (prawy) ramki o całkowitej długości 32B (400kBodów/s, 2-GFSK, 868MHz)

W oparciu o wykonane pomiary, wprowadzam następujące stałe, do dalszego wykorzystania:

- Całkowita energia potrzebna na uzyskanie gotowości bloku RF węzła E_{ONtotal} = E₀ + E₁ + E₂ ≈ 14,4 μJ oraz całkowity czas τ_{ONtotal} = τ₀ + τ₁ + τ₂ ≈ 1,1 ms
- Całkowita energia potrzebna na powrót do trybu STANDBY E_{OFFtotal} = E₆ + E₇ ≈ 3,2 μJ oraz całkowity czas τ = τ₆ + τ₇ ≈ 0,2 ms
- Zużycie energii podczas odbierania E_{RX} = f(l_F, R) = l_F/R × E_{4,RX}/τ₄ = l_F/R × P_{RX} ≈ l_F/R × 23 mW oraz nadawania E_{TX12.5dBm} = f(l_F, R) = l_F/R × E_{4,TX}/τ₄ = l_F/R × P_{TX12.5dBm} ≈ l_F/R × 59 mW, gdzie l_F jest całkowitą długością ramki w bitach a R jest przepływnością wyrażoną w bitach na sekundę
- Zużycie energii w trakcie "idle-listening" $E_{IL} = f(t_{IL}) = t_{IL} \times \frac{E_{4'}}{\tau_{4'}} = t_{IL} \times P_{IL} \approx t_{IL} \times 23 \ mW$, gdzie t_{IL} jest czasem trwania fazy "idle-listening"
- Zużycie energii podczas inicjalizacji komendy radiowej E<sub>INIT_{CMD} = E₃ ≈ 1,7 μJ oraz czas τ_{INIT_{CMD}} ≈ 0,1 ms
 </sub>
- Zużycie energii podczas kończenia komendy radiowej $E_{END_{CMD}} = E_5 \approx 2 \,\mu J$ oraz czas $\tau_{END_{CMD}} \approx 0.1 ms$

2.1.3 Łączenie komend radiowych

Komendy radiowe są wykonywane w sekwencjach powiązanych ze sobą pojedynczych komend. Najlepszymi przykładami sekwencji są: wysyłanie wielu ramek, procedura "Listen

Before Talk" (LBT) lub oczekiwanie na potwierdzenie po transmisji ramki (ACK). Pomiary poboru energii dla przejścia w różnych sekwencjach przedstawiono na rysunku 2.3 (TX \rightarrow TX, TX \rightarrow RX) i rysunku 2.4 (RX \rightarrow TX, RX \rightarrow RX). MCU może planować polecenia RF, używając wskaźnika następnej operacji w dowolnym poleceniu operacji radiowej. Ten wskaźnik może wskazywać na następną komendę do wykonania w łańcuchu, dlatego złożone sekwencje mogą być wykonywane bez udziału MCU. Czas trwania i odpowiednie zużycie energii dla każdej fazy przedstawiono w tabeli B.5. Numeracja faz odpowiada podanej w tabeli B.4. Fazy drugiej komendy w łańcuchu są dodatkowo oznaczone apostrofem.



Rysunek 2.3 Zużycie energii podczas sekwencji operacji radiowych: TX->TX (lewy), TX->RX (prawy)



Rysunek 2.4 Zużycie energii podczas sekwencji operacji radiowych: RX->TX (lewy), RX->RX (prawy)

Czas pomiędzy poszczególnymi komendami, mierzony jako odstęp pomiędzy zakończeniem pierwszej komendy w łańcuchu i rozpoczęciem wykonywania drugiej, nie jest pomijalny i

wynosi 0,13 ms dla sekwencji RX->RX oraz 0,16 ms dla sekwencji TX->TX. Zużycie energii to odpowiednio 2,3 μJ oraz 2,8 μJ. Porównując wyniki z poprzednią wersją układu, czyli CC13x0, również w tej sekwencji widoczna jest znacząca redukcja zarówno czasu jak i energii potrzebnej do przejścia. Dla układu z rodziny CC13x0 czasy mieszczą się w przedziale 0,14 ms (RX->RX) oraz 0,25ms (TX->TX) a zużycie energii odpowiednio w przedziale 2,4 μJ do 5,2 μJ.

2.2 Synchronizacja węzłów sieci IWN

Synchronizacja czasu jest krytyczną kwestią w działaniu przemysłowych sieci bezprzewodowych (IWN), ponieważ zapewnia wspólne ramy czasowe dla różnych węzłów. Węzły sieci pracują pod dużymi ograniczeniami związanymi z energią, pamięcią i możliwościami obliczeniowymi, ponadto są narażone na zmiany otoczenia (np. temperatury). Węzły zazwyczaj wykonują zadania rozproszone i zespołowe w sposób zsynchronizowany. Służą do wykrywania sygnałów, przetwarzania, generowania odpowiednich danych, a także mogą realizować różne funkcje kontrolne. Wszystkie te działania polegają na współdzieleniu wspólnej skali czasu w węźle sieci co jest podstawą efektywnej komunikacji.

Węzły IWN pracujące w środowisku przemysłowym lub montowane na maszynach mobilnych narażone są na częste i nieprzewidywalne zmiany temperatury, które wpływają na dokładność lokalnego zegara powodując problemy z synchronizacją (przesunięcie czasowe). Kwestie te należy wziąć pod uwagę podczas optymalizacji sieci, aby uniknąć większego zużycia energii i zatorów komunikacyjnych. Problem przesunięcia czasowego można minimalizować za pomocą kompensacji zależnej od temperatury. W rozdziale prezentuję efektywną metodę wraz z uniwersalnym algorytmem pozwalającą na rozwiązanie tego problemu. Algorytm uwzględnia powolne i gwałtowne wahania temperatury. Opracowany algorytm łączy fizyczny model oscylatora węzła z oryginalną metodą przewidywania opartą na regresji liniowej. Proces ten jest realizowany przez oryginalny schemat przewidywania zmian zegara czasu rzeczywistego (RTC) w połączeniu z wymianą komunikatów synchronizacji. Algorytm pozwala na uzyskanie dokładnej synchronizacji poprzez zapewnienie wysokiej stabilności czasowej w węzłach sieci, tolerując jednocześnie szeroki zakres zmian temperatury. Proponowane rozwiązanie dynamicznie dostosowuje okresy resynchronizacji i jest kompatybilne z różnymi architekturami IWN oraz protokołami synchronizacji.

Opracowany uniwersalny algorytm z powodzeniem zaimplementowano w typowym węźle IWN (SENSOR_NODE_1C) wyposażonym w mikrokontroler o ograniczonych zasobach z rodziny CC13x2 firmy TI. Prezentowane podejście wymagało przeprowadzenia eksperymentalnych badań właściwości oscylatorów wykorzystywanych w IWN oraz możliwości samych mikrokontrolerów. Wykonano szereg badań w środowisku rzeczywistym pozwalających na określenie wpływu temperatury na pracę węzłów IWN. Wykonane

implementacje i przeprowadzone testy wykazały wysoką skuteczność prezentowanej metody w środowisku przemysłowym w porównaniu z innymi podejściami. Osiągnięto wysoką stabilność czasową przy zachowaniu niskiego poboru energii, co zostało udowodnione w przeprowadzonych eksperymentach (fizycznych i symulacyjnych) z węzłami sieci o ograniczonych zasobach.

Prezentowane w rozdziale wyniki badań opublikowałem w [23], natomiast tutaj przytaczam je w formie skrótowej w celu udokumentowania prowadzonych prac i otrzymanych wyników. Przeprowadzona w artykule [23] (rozdział 2) analiza literatury potwierdza duże znaczenie synchronizacji węzłów IWN w różnych aspektach ich pracy [19, 33–38]. Jednocześnie literatura wykazuje braki w zakresie metod i algorytmów pozwalających na zapewnienie stabilnej pracy zegara czasu rzeczywistego (RTC) węzła IWN o ograniczonych zasobach, przy szybko zmieniających się warunkach otoczenia [33, 35, 37, 39–47].

2.2.1 Badanie efektów przesunięcia czasu i temperatury

Opracowanie efektywnego algorytmu pozwalającego na zapewnienie wymaganego w IWN poziomu synchronizacji węzłów, tolerującego zmienne warunki otoczenia, wymaga praktycznej weryfikacji wpływu temperatury na prace węzłów sieci w szczególności przesunięcie czasu RTC. W tym celu został przygotowany zestaw eksperymentów obejmujących reprezentatywne węzły IWN oraz scenariusze testowe. Specyfikacja stanowiska została precyzyjnie opisana w [23]. Eksperymenty obejmują dwie wersje węzłów (wcześniejsze wersje SENSOR NODE 1C) oparte na mikrokontrolerze CC13X2 i dwóch rodzajach czujników temperatury o różnym poziomie dokładności pomiaru. Do pomiaru temperatury z bardzo niską dokładnością (±2,5 °C) używam wbudowanego (w CC13x2) monitora stanu baterii (BATMON). Natomiast do pomiaru temperatury o większej dokładności (± 0,2°C) dodatkowego obwodu Z zewnętrznym termistorem używam nieliniowym (NCP15XH103F03RC) i wbudowanego 12-bitowego przetwornika analogowo-cyfrowego (ADC). Węzły zawierają specjalnie opracowane autorskie moduły oprogramowania. Scenariusze testowe odnoszą się do powolnych i szybkich zmian temperatury, W celu sprawdzenia wpływu pojemności cieplnej węzła (pomijanego w literaturze), eksperymenty prowadzę w dwóch wariantach, gdzie węzeł pracuje bez obudowy i w obudowie.

Eksperymenty polegają na znalezieniu rozkładu przesunięcia czasowego przy odpowiadającej mu temperaturze otoczenia dla dwóch faz ekspozycji węzła. W pierwszej fazie (0 min do 900 min) węzeł pracuje na zewnątrz, gdzie temperatura otoczenia stopniowo spada w naturalny sposób (cykl dzień-noc). W drugiej fazie (>900 min) węzeł przenoszony jest do ogrzewanego pomieszczenia na około 40 min, następnie ponownie na zewnątrz na około 40 min i ponownie do temperatury pokojowej. Na rysunku 2.5 prezentuję pomiary temperatury (lewa sekcja) i

przesunięcia czasu (prawa sekcja). Temperaturę mierzę co sekundę, korzystając z dwóch dostępnych czujników. W górnej części (lewa sekcja) przedstawiam pomiar za pomocą zintegrowanego czujnika temperatury o niskiej dokładności (BATMON). Dolna część prezentuje wyniki dokładniejszego czujnika temperatury, składającego się z taniego termistora NTC i 12-bitowego wewnętrznego przetwornika ADC. Przesunięcie czasowe (prawa sekcja) mierzę przy stałym okresie resynchronizacji, RP = 30 s. W górnej części przedstawiam przesunięcie czasowe między węzłem lokalnym a źródłem zegara podczas każdej resynchronizacji. Wartość dodatnia oznacza, że zegar lokalnego węzła wyprzedza zegar źródłowy, więc wartość przyrostu RTC (rejestr RTC_SUBSECINC) jest zmniejszana. Podczas każdej synchronizacji czas lokalny jest regulowany, a RTC_SUBSECINC jest korygowane o wartość wynikającą z rzeczywistego przesunięcia czasu podzielonego przez liczbę cykli RTC od ostatniej synchronizacji.



Rysunek 2.5 Temperatura (lewa) mierzona poprzez: BATMON (góra) i NTC+ADC (dół) oraz mierzone przesunięcie czasu (prawa-góra) i wartość komponsacji w rejestrze RTC_SUBSECINC (prawa-dół)

Бада	Przesunięcie czasu [µs]				
Faza	mean	std	min	max	
<900 min	-0.62	1.48	-5.00	4.00	
>900 min	0.22	6.72	-19.00	71.00	

Tabela 2.1 Parametry rozkładu przesunięcia czasu dla obu faz

W tabeli 2.1 przedstawiam wartości średnie, odchylenia standardowego, minimum oraz maksimum rozkładu przesunięcia czasowego dla każdej fazy. W pierwszej fazie, gdy węzeł pracuje w warunkach zewnętrznych a zmiana temperatury wynika wyłącznie z naturalnego i powolnego przejścia między dniem a nocą, wartość przesunięcia czasowego mieści się w zakresie od -5µs do 4µs i możliwe jest dalsze wydłużenie okresu RP. Gdy węzeł przechodzi

między temperaturą pokojową a warunkami zewnętrznymi, przesunięcie czasowe wzrasta do niedopuszczalnej wartości nawet przy stosunkowo krótkim RP wynoszącym 30s i w takiej sytuacji okres RP należy skrócić aby zapewnić wymagany poziom synchronizacji przyjęty w przykładowej sieci standardu ISA100.11a jako 32µs.



Rysunek 2.6 Temperatura mierzona (A_1, B_1) poprzez: BATMON (góra) i NTC+ADC (dół) oraz odpowiadające jej funkcje dopasowane numerycznie (A_2, B_2) do równania (3), artykuł [23]

Analizując wyniki eksperymentu można zauważyć, że w przypadku gwałtownych zmian temperatury węzeł może być traktowany jako zbiornik ciepła (pojemność cieplna). Dlatego zgodnie z prawem stygnięcia Newtona, szybkość zmiany temperatury jest wprost proporcjonalna do różnicy temperatur między ciałem a otoczeniem (równanie (3), artykuł [23]). Na rysunku 2.6 przedstawiam funkcję z równania (3) [23] numerycznie dopasowaną do zmierzonych wartości temperatury. Symbole A_2 i B_2 używam do oznaczenia dopasowanej funkcji odpowiednio dla węzła bez obudowy (A) i węzła z obudową (B), natomiast A_1 i B_1 odnoszą się do wartości mierzonej temperatury. Prezentuję oddzielnie fazę chłodzenia (lewa) i ogrzewania (prawa). W celu dopasowania do funkcji nieliniowej, używam algorytmu trustregion reflective least squares z wykorzystaniem pakietu SciPy/Python. Należy zauważyć, że funkcje numeryczne dopasowują się do zmierzonych danych z dużą dokładnością. Dokładność dopasowania jest lepsza dla węzła o większej pojemność cieplnaj (przypadek B). W praktycznych zastosowaniach, spodziewam się że pojemność cieplna obudowy będzie jeszcze większa co wydłuży stałą czasową. Należy również zauważyć, że pojemność cieplną węzła można dodatkowo zwiększyć poprzez zastosowanie kropli żywicy termoprzewodzącej

zlokalizowanej w obszarze gdzie znajduje się obwód rezonatora kwarcowego wraz z termistorem na płytce PCB.

2.2.2 Adaptacyjna kompensacja przesunięcia czasu

Opracowany autorski model pracuje w dwóch trybach, pomiędzy którymi przełącza się automatycznie na podstawie detekcji trendu. Tryb *LINEAR* jest używany, gdy charakter zmian temperatury można określić za pomocą liniowej funkcji czasu. Tryb *EXP* jest używany, gdy temperatura zmienia się wykładniczo zgodnie z równaniem (3), artykuł [23]. W celu określenia średniego przesunięcia czasu pomiędzy chwilą t_1 i t_2 , obliczam średnią wartość zmiany częstotliwości w odniesieniu do czasu (równanie (1)) dla obwodu z rezonatorem kwarcowym (ang. tuning fork crystal) dla którego odchylenie częstotliwości opisane jest wzorem (1), artykuł [23].

$$\frac{\overline{\Delta f}}{f_0} = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} \gamma \cdot (T(t) - T_{tr})^2 dt \tag{1}$$

Przyjmując, że zmiana temperatury w funkcji czasu ma charakter liniowy $T(t) = \alpha t + \beta$, równanie można rozwiązać do postaci (5) artykuł [23]. Natomiast jeżeli przyjmiemy, że zmiana temperatury odpowiada zależności wynikającej z prawa stygnięcia Newtona (równanie (3), artykuł [23]), równanie można rozwiązać do postaci (6) artykuł [23]. W modelu szacuję przewidywane wartości dryftu zegara na podstawie równań (5) i (6) odpowiednio dla trybów: *LINEAR* i *EXP*, dla stałych interwałów, dlatego mogę częściowo stablicować obliczone wartości. Opracowany algorytm został podzielony na trzy niezależne części:

- Algorytm 1, artykuł [23] pomiar temperatury i detekcja trendu;
- Algorytm 2, artykuł [23] predykcja temperatury, szacowanie dryftu zegara oraz kompensacja RTC w odniesieniu do temperatury;
- Algorytm 3, artykuł [23] korekta zegara i kompensacja RTC w odniesieniu do synchronizacji czasu;

Każda z procedur wykonywana jest cyklicznie z okresem odpowiednio P_1 , P_2 , P_3 , gdzie $P_3 > P_2$; $P_2 = n \cdot P_1$; $n \in \mathbb{N}$. P_3 jest równe okresowi resynchronizacji (*RP*), w testach *RP* = 30*s*. W praktycznych zastosowaniach dążymy do maksymalnego wydłużenia *RP*, które wiąże się z poniesieniem dużego wydatku energetycznego związanego z komunikacją, okres ten wacha się w przedziale 5s do 120s. Do predykcji temperatury, używam regresji liniowej dla obu trybów *LINEAR* i *EXP*. W trybie *EXP* wykorzystuję linearyzację danych w celu przekształcenia równania (3) artykuł [23] do postaci funkcji liniowej $y = \alpha x + \beta$.

$$x = e^{-kt}; \ \alpha = T_0 - T_{env}; \ \beta = T_{env}$$
(2)

Szczegółowy opis poszczególnych algorytmów wraz z ich pseudokodem został opublikowany w [23].

2.2.3 Ewaluacja modelu

W celu oceny skuteczności i dokładności opracowanego modelu kompensacyjnego wykonałem eksperymenty praktyczne i symulacyjne. Pierwsze bazują na opracowanych w ramach projektu RPMA.01.02.00-14-9551/17-00 węzłach (wczesne prototypowe wersje SENSOR_NODE_1C, rozdział 4.1, [23]) oraz pomiarach w terenie. Eksperymenty symulacyjne koncentrują się na analizie wpływu wybranych parametrów pracy węzła i porównaniu z modelem [39].

2.2.3.1 Badania eksperymentalne

W opracowanym modelu kompensacji czasu istotnym elementem jest właściwa detekcja trendu zmian temperatury (LINEAR / EXP). Do oceny poprawności wykrywania trendu użyłem kilku rzeczywistych zestawów danych. Zdefiniowałem trzy metryki, aby ocenić dokładność wykrywania trendów. Współczynnik good/all (wyrażony w procentach) definiowany jako liczba pożądanych detekcji trendu EXP do liczby detekcji trendu EXP wykrytych przy użyciu algorytmu 1 według [23]. Współczynnik wynoszący 100% oznacza brak fałszywych oraz pominietych detekcji. Współczynniki exp/total i lin/total (wyrażone w procentach) określają odpowiednio całkowity czas trybów EXP i LINEAR. Reprezentatywne wyniki detekcji trendu dla obu rozważanych czujników temperatury opublikowałem w tabeli 3 [23]. Stosując NTC + ADC, który jest znacznie dokładniejszy niż BATMON, uzyskałem poprawną detekcję z oknami 4 razy krótszymi. Długość okna dla SHORT SMA SIZE określa, jak szybko algorytm wykryje zmianę temperatury. Ponieważ przesunięcie czasu jest największe przy pierwszej synchronizacji czasu występującej po zmianie trendu, dobre dopasowanie początkowego przebiegu krzywej temperatury jest szczególnie ważne dla prawidłowej kompensacji. W zastosowaniach praktycznych parametry te muszą być dostosowane do zadanej dokładności pomiaru temperatury. Większa dokładność pomiaru temperatury pozwala na krótsze rozmiary okien i dlatego potrzebny jest mniejszy bufor cykliczny do przechowywania danych. W testach najlepsze wyniki uzyskano dla SHORT SMA SIZE, LONG SMA SIZE, SIGMA SIZE i K odpowiednio: 4, 8, 32, 2 dla NTC+ADC i 16, 64, 256, 2 dla BATMON. Co więcej, większe okna zbierające (ang. rolling windows) wymagają więcej pamięci RAM, co jest szczególnie ważne w domenie kontrolera czujników (rozdział 4.1, [23]) ze względu na ograniczoną ilość pamięci. Niższe wartości zapewniają najszybsze wykrywanie, ale wiaże się z większą liczbą fałszywych wykryć. Zauważyłem, że w testach stosunek LONG SMA SIZE do SHORT SMA SIZE, a także SIGMA SIZE do LONG SMA SIZE wynosi 4 dla najlepszych rezultatów.



Rysunek 2.7 Temperatura i przesunięcie czasu w węzłach podczas grzania (lewa), chłodzenia (prawa): Przesunięcie czasu i kompensacja czasu (wiersz 1); Temperatura mierzona BATMON i predykcja LINEAR, EXP (wiersz 2); Temperatura mierzona NTC+ADC i predykcja LINEAR, EXP (wiersz 3); Przesunięcie czasu po kompensacji używając BATMON (wiersz 4)

Na rysunku 2.7 przedstawiam eksperymentalne wyniki dotyczące pomiarów temperatury i przesunięcia czasowego odpowiednio dla scenariuszy grzania i chłodzenia (czas grzania w zakresie od 1054 min do 1063 min i czas chłodzenia w zakresie od 1100 min do 1108 min). Korzystam z parametrów algorytmu detekcji trendu wymienionych powyżej i RP=30 s. LINEAR SIZE i EXP SIZE to odpowiednio 32, 32 dla NTC+ADC i 256, 128 dla BATMON. W wierszu 1 prezentuję przesunięcie czasowe bez kompensacji (kolor niebieski) oraz uzyskane wartości kompensacji temperatury za pomocą opracowanego algorytmu (kolor czerwony). W wierszach 2 i 3 przedstawiam temperaturę zmierzoną (kolor niebieski) wraz z temperaturą przewidywaną (kolor czerwony) przy zastosowaniu odpowiednio czujników BATMON i NTC + ADC. Pomarańczowa linia obrazuje wykryty trend, 0 dla trybu LINEAR i 1 dla trybu EXP. Dodatkowa zielona linia pokazuje hipotetyczną przewidywaną temperaturę gdyby algorytm pracował wyłącznie w trybie LINEAR. Należy zauważyć, że tryb LINEAR nie może być stosowany, gdy temperatura otoczenia zmienia się gwałtownie, ponieważ prowadzi to do znacznych różnic między temperaturą faktycznie zmierzoną a przewidywaną. Ponieważ przesunięcie czasowe jest funkcją kwadratową temperatury (równanie (1) [23]), możemy przyjąć, że aproksymacja liniowa samego przesunięcia czasowego [33] da jeszcze gorsze rezultaty, gdy założymy gwałtowne zmiany temperatury. W wierszu 4 prezentuję przesunięcie czasowe uzyskane po kompensacji w oparciu o BATMON. Parametry rozkładu przesunięcia czasowego uzyskane przy użyciu kompensacji BATMON przedstawiono w tabeli 4 [23]. Widzimy, że nawet gdy węzeł używa bardzo niedokładnego czujnika temperatury BATMON, proponowana metoda kompensacji przesunięcia czasowego daje akceptowalne rezultaty. Uzyskane maksymalne przesunięcie czasowe nie przekracza 7µs, i jest o wiele mniejsze od uzyskanych 71µs podczas pracy bez opracowanego algorytmu kompensacji. Dodatkowo w publikacji [23] na rysunku 9 przedstawiam powiększone wykresy przewidywanej temperatury w trybie *EXP* oraz *LINEAR*. Możemy zauważyć, jak ważne jest prawidłowe wykrycie trendu EXP w pierwszym odcinku krzywej, tam gdzie przesunięcie czasowe jest największe.

Głównym założeniem opracowanego algorytmu jest zapewnienie wymaganego poziomu synchronizacji węzłów przy jednoczesnym zapewnieniu zużycia energii wynikającego z funkcjonowania algorytmu na bardzo niskim poziomie (porównywalnym lub niższym w stosunku do TCXO). W tabeli 5 [23] podsumowałem obliczone zużycie energii związane z synchronizacją czasu dla węzła pracującego bez kompensacji temperaturowej i z kompensacją temperaturową wykorzystującą BATMON lub NTC+ADC. Zużycie energii przedstawiam osobno dla każdego z algorytmów oraz wskazuję poszczególne źródła zużycia energii. W przypadku gdy węzeł pracuje bez kompensacji temperatury wykorzystuje wyłącznie algorytm 3 [23], ponieważ wykonuje wyłącznie synchronizację czasu poprzez wymianę pary komunikatów ze źródłem odniesienia czasu. W bilansie algorytmu 3 nie uwzględniłem wydatku energetycznego który wynika z samej komunikacji. Wydatek ten jest zdecydowanie największy ale taki sam dla wszystkich opisanych w tabeli wariantów. Algorytm 1 [23] działa na rdzeniu SC, podczas gdy algorytmy 2 i 3 [23] wykorzystują rdzeń M4F. W przypadku SC łączną liczbę cykli określiłem, zliczając liczbę instrukcji asemblera, zakładając, że każda instrukcja wymaga 2 cykli do wykonania. W przypadku rdzenia M4F określiłem liczbę cykli za pomocą debugera. Zużycie energii potrzebnej do wykonania kodu określiłem na podstawie liczby cykli oraz energii potrzebnej do wykonania pojedynczego cyklu maszyny dla odpowiedniego rdzenia (specyfikacja układu CC13x2). Ponieważ węzły mają ograniczony budżet energetyczny, w przypadku bezczynności muszą przechodzić do trybów o obniżonym zużyciu energii. W zestawieniu uwzględniam energię potrzebną do wybudzenia węzła oraz przejścia w tryb głębokiego uśpienia. W rozdziale 2.1.1 oraz [24] przeprowadziłem szczegółowe pomiary i analizę rdzenia M4F układów CC13x2, skupiając się na różnych scenariuszach przejścia między trybami pracy. Dla algorytmów 2 i 3 [23] nie uwzględniam energii wynikającej z przechodzenia między poszczególnymi trybami pracy rdzenia, zakładając, że główny rdzeń będzie budził się co najmniej raz na sekundę, aby wykonać inne zadania.

W celu dalszej redukcji zużycia energii, możliwe jest przeniesienie algorytmu 2 [23] do domeny kontrolera czujników (rdzeń SC). Ta implementacja jest trudniejsza ze względu na brak pełnej

jednostki arytmetycznej i dostęp tylko do 16-bitowych rejestrów. SC ma dodatkową jednostkę akceleratora matematycznego (MAC) do 16-bitowego mnożenia i do 32-bitowego dodawania. Co więcej, w praktycznych zastosowaniach, gdy używamy SC do innych zadań okresowych (np. pomiarów), energia potrzebna do wybudzenia SC może być pominięta w całkowitym budżecie energetycznym algorytmu 1. W takim przypadku algorytm 1 zużyje jedynie 0,5 μW nawet w konfiguracji NTC+ADC. Dodatkowo nieliniowy element NTC można zastąpić prawie-liniowym, np. TMP61 (Texas Instruments), który niewielkim kosztem zapewni liniowość szerokim zakresie temperatur (zastosowano W W finalnej wersji SENSOR NODE 1C). Algorytm 1 może zostać wyposażony w dodatkowy mechanizm dostosowujący okres P1 do aktualnej temperatury. Interwał można skrócić, gdy temperatura węzła jest dalej od T_{tr} i wydłużyć w przeciwnym przypadku. Taki mechanizm pozwali na zwiększenie dokładności pomiaru oraz szybkości wykrywania trendu przy jednoczesnym ograniczeniu zużycia energii.

2.2.3.2 Symulacja numeryczna

Opracowałem autorski symulator i przeprowadziłem serię symulacji numerycznych pozwalających na porównanie opracowanego algorytmu z prezentowanym w [39]. Symulator składa się z modeli analitycznych odpowiednich dla rozważanych algorytmów oraz modułu wizualizacji do prezentacji wyników w postaci numerycznej i graficznej. Eksperymenty symulacyjne dotyczą dwóch przypadków zmiany temperatury: (a) 0 - 25 °C, (b) -35 - 25 °C przy stałej cieplnej układu τ równej 100s i 200s. Przesunięcie czasowe zostało obliczone dla okresów resynchronizacji (RP) 30, 60, 120, 300 i 600 s. Ponadto wyniki obejmują 5 okresów pomiaru temperatury (0,5, 1,0, 2,0 i 5,0 s) oraz 2 poziomy niepewności pomiaru temperatury (T_{err} : ± 0.1 °C; ± 0.2 °C). Symulacja dla każdego zestawu parametrów została uruchamiana 500 razy w celu uzyskania pełnego rozkładu wyników. Predefiniowane stałe dotyczące wykrywania trendu odpowiadają najlepszym wynikom uzyskanym w rozdziale 6.1 [23] dla zestawu ADC+NDC (*SHORT_SMA_SIZE*, *LONG_SMA_SIZE*, *SIGMA_SIZE* i *K*: 4, 8, 32, 2). *LINEAR_SIZE* i *EXP_SIZE* to odpowiednio 32, 32.

W celu lepszego odwzorowania warunków rzeczywistych w symulatorze uwzględniłem trzy klasy błędów:

 <u>Niepewność pomiaru temperatury.</u> Proces kalibracji i kompensacji wykorzystują te same obwody pomiarowe, dlatego dokładność pomiaru można pominąć. Najważniejszą składową niepewności pomiaru jest precyzja. Dla zestawu ADC+NTC, precyzja pomiaru zdeterminowana jest poprzez rozdzielczość/nieliniowość ADC i nieliniowość NTC. W badanym przypadku zakładam że dokładność rzeczywistego pomiaru to ±0,2 °C. Dokładność można dodatkowo zwiększyć poprzez zastosowanie prawie-liniowego termistora lub uśrednianie wielu sąsiednich odczytów (dodatkowy wydatek energetyczny). Aby zasymulować niepewność pomiaru temperatury, do symulowanej temperatury dodawany jest losowy błąd o rozkładzie równomiernym (T_{err}).

- <u>Precyzja ustawienia szybkości RTC</u>. Wartość przyrostu RTC jest przechowywana w rejestrze RTC_SUBSECINC. Rozdzielczość rejestru RTC_SUBSECINC pozwala na 119,21ns niedokładności na 1s. Tą wartość uznałem za zbyt wysoką dla długich (>1 min) RP, dlatego dalszą kompensację nachylenia RTC przeprowadziłem w oprogramowaniu. Dodatkowe 4 bity zwiększają precyzję do 7,45ns na 1s. Aby zasymulować ten błąd, losowa wartość o rozkładzie równomiernym (generowana raz na RP) jest proporcjonalnie dodawana do obliczonego przesunięcia czasowego dla każdego postąpienia symulacji.
- <u>Precyzja znakowania czasem</u>. Synchronizacja czasu oparta o dwustronną wymianę komunikatów wykorzystuje znakowanie czasem w celu obliczenia wartości korekty. W badanym zestawie znakowanie czasem wykorzystuje dedykowany zegar modułu radiowego (RAT). RAT to 32-bitowy licznik działający z częstotliwością 4 MHz. Aby zasymulować ten błąd, do obliczonego przesunięcia czasowego dodaje losową wartość o rozkładzie równomiernym z przedziału odpowiadającego rozdzielczości RAT.



Rysunek 2.8 Przesunięcie czasu w węźle IWN podczas szybkiej zmiany temperatury otoczenia: 0°C do 25°C (lewy), -35°C do 25°C (prawy) dla opracowanego algorytmu i [39]

Symulator pracuje z zadanym krokiem symulacji (przyjęto 10ms). Dla każdej kolejnej chwili czasu oblicza wartość różnicy czasu pomiędzy badanym węzłem a czasem odniesienia. Na rysunku 2.8 prezentuję wartość przesunięcia czasu w funkcji czasu dla dwóch reprezentatywnych zestawów parametrów (wartości parametrów umieściłem nad wykresami). Bazując na obliczonej wartości chwilowej symulator gromadzi również wartość przesunięcia czasu jakie obserwowane jest w węźle w chwili wykonywania synchronizacji czasu (RP). Rozkład wartości chwilowych przesunięcia czasu oraz rozkład wartości przesunięcia czasu w chwili synchronizacji zostały zaprezentowane w tabelach 6 i 7 w publikacji [23] i oznaczone odpowiednio "Inst." oraz "Sync.". Wyniki obu porównywanych algorytmów zestawiłem dla

dwóch reprezentatywnych wartości RP. Wartość RP=30s jest interpretowana jako "krótka" dla węzłów o ograniczonych zasobach energetycznych podczas gdy RP=600s jest wartością "długą", która pozwala na korzystny bilans energetyczny ale jest nieosiągalna w układach bez stabilizacji zegara RTC lub dodatkowej kompensacji.



Rysunek 2.9 Porównanie wyników symulacji szybkiej zmiany temperatury otoczenia: 0°C *do* 25°C (1, 2 kol.), -35°C *do* 25°C (3, 4 kol.) dla opracowanego algorytmu i [39]

Na rysunku 2.9 porównałem uzyskane wyniki algorytmu [39] (1, 3 kolumna) i opracowanego (2, 4 kolumna). Przedstawione wyniki pokazują poprawę dla każdego zestawu parametrów. Maksymalne przesunięcie czasowe jest o 50% niższe dla 0 *do* 25 °C, $\tau = 200s$, P1 = 0.5s i ponad 5 razy mniejsze dla -35 *do* 25 °C $\tau = 100s$, P1 = 5 s. Poprawa znacznie wzrasta wraz ze wzrostem zakresu temperatur oraz wydłużeniem okresu pomiaru temperatury.

3 Adaptacyjny dobór parametrów konfiguracji sieci IWN

W rozdziale prezentuję problem doboru optymalnych parametrów sieci IWN. Model prezentowanej sieci zakłada zarządzanie poprzez rolę dedykowanego menadżera sieci (SM). Węzły tworzące sieć mogą komunikować się wzajemnie ze sobą oraz przekazywać komunikację od innych węzłów. Wszystkie elementy sieci mogą wykorzystywać wielodostęp do medium z podziałem w dziedzinie czasu (ang. Time-division multiple access, TDMA) i częstotliwości (ang. Frequency-division multiple access, FDMA). Model sieci zakłada komunikację horyzontalna (bezpośrednio pomiedzy urządzeniami wykonawczymi zaangażowanymi w procesy automatyzacji produkcji) [10] i jest odpowiedzia na potrzeby Przemysłu 4.0 w szczególności komunikacji M2M (ang. machine to machine). Zarówno SM jak i węzły sieci są urządzeniami o ograniczonych zasobach. Ograniczenia dotyczą dostępnej pamięci operacyjnej, mocy obliczeniowej oraz dostępnej energii zasilającej urządzenia. Każdy węzeł sieci (węzeł wnioskujący będący węzłem źródłowym) może wnioskować do SM o zawarcie kontraktu na komunikację z wybranym węzłem docelowym. Węzeł wnioskujący podaje w kontrakcie szczegóły dotyczące planowanej komunikacji w szczególności dotyczące stawianych wymagań QoS. SM w oparciu o wnioskowane i aktywne kontrakty oraz w oparciu o raporty dotyczące jakości połączeń węzłów z sasiadami podejmuje decyzję o zawarciu lub odrzuceniu kontraktu. Zadaniem SM jest taki dobór parametrów sieci aby jak największa liczba kontraktów, z uwzględnieniem ich priorytetu, mogła być zawarta. Postawiony problem wymaga opracowania efektywnych algorytmów pozwalających na przygotowanie pełnej konfiguracji sieci (tablica routingu i harmonogram transmisji) jak również jej optymalizacje przy spełnieniu postawionych wymagań QoS.

W rozdziale 3.2 opisuję przyjęte założenia w szczególności dotyczące definicji kontraktu oraz modelu sieci. Pełną notację użytych symboli oraz definicję pojęć prezentuję zbiorczo w tabeli 3.5. Formalny zapis postawionego problemu optymalizacyjnego prezentuję w rozdziale 3.2.4. W rozdziale 6.2 prezentuję nowatorski algorytm GA-CAS który jest dedykowany całościowemu rozwiązaniu postawionego problemu. GA-CAS wykorzystuje algorytm genetyczny i integruje dwa algorytmy CAS-PERIOD (5.3) i CAS-SEQ (5.4) dedykowane do wykonania planu szczelin czasowych w TDMA. Porównuję opracowane algorytmy planowania CAS-PERIOD i CAS-SEQ względem kilku dobrze znanych i opisanych w literaturze algorytmów EDF/LLF/AMUS (rozdział 5). Opracowane algorytmy poddaję badaniu z wykorzystaniem autorskiego środowiska testowego (rozdział 4) przy użyciu opracowanych własnych wektorów testowych (rozdział 4.2) pozwalających na analizę wpływu szeregu parametrów na uzyskiwane rezultaty.

3.1 Korelacja parametrów IWN z wymaganiami QoS

Quality of Service (QoS) odnosi się do "the collective effect of service performance which determine the degree of satisfaction of a user of the service". Przy czym użytkownik definiowany jest szeroko jako jednostka korzystająca z sieci. W przypadku IWN taką jednostką będzie np. czujnik lub element wykonawczy, którego użycie zakłada i wymaga dostępności usług sieciowych na określonym ściśle zdefiniowanym poziomie. Zupełnie inne wymagania jakościowe wobec sieci ma czujnik obecności metanu w kopalni węgla a zupełnie inne wymagania ma czujnik wilgotności w gospodarstwie rolnym wykorzystywany do sterowania pracą systemu nawadniania. W naturalny sposób, mimo że rola obu czujników jest kluczowa do funkcjonowania systemu to ich priorytetyzacja musi być inna w przypadku funkcjonowania w obrębie jednej sieci.

Wymagania QoS dla bezprzewodowych sieci automatyki przemysłowej (IWN) różnią się od zwykłych sieci komputerowych. Surowe środowisko pracy w przemyśle pogorsza wydajność komunikacji bezprzewodowej z powodu utraty widoczności radiowej, zacieniania ścieżki transmisji, propagacji wielotorowej i innych zakłóceń [48]. Co więcej, ruch składa się głównie z krótkich pakietów zawierających pomiary czujników lub polecenia siłowników, które muszą być dostarczone na czas, zamiast dużych strumieni multimedialnych lub ruchu interaktywnego, który przeważa w sieciach komputerowych. Klasyczne sieci komputerowe dysponują również dużą nadmiarowością zasobów w stosunku do średnich potrzeb użytkowników. W sieciach IWN dysponujemy ograniczonymi zasobami zarówno w odniesieniu do samych urządzeń (mała moc obliczeniowa, mała pamięć operacyjna) jak również w odniesieniu do energii niezbędnej do ich funkcjonowania oraz zasobów radiowych, gdzie wiele urządzeń korzysta ze wspólnej przestrzeni radiowej.

Symbole wykorzystywane w opisach w niniejszym rozdziale zostały zebrane w tabeli 3.5 przy okazji rozdziału 3.2.3 dotyczącego definicji modelu sieci.

3.1.1 Wymagania czasowe

Dokonując klasyfikacji instalacji przemysłowych ze względu na potrzebę pracy w czasie rzeczywistym (ang. real-time) [49], większość z nich reprezentuje Klasę 1 – soft real-time. Jedynie pewien ściśle określony podzbiór zastosowań wymaga bardziej restrykcyjnego podejścia do problemu czasu rzeczywistego. Aplikacje wymagające pracy w pętli zamkniętej reprezentują klasę 2 – hard real-time, gdzie narzucone są ścisłe wymagania dotyczące opóźnienia end-to-end. Wąską grupę aplikacji stanowią te z Klasy 3 – isochronous real-time, gdzie dodatkowo do bardziej wygórowanych niż w klasie 2 wymagań dotyczących opóźnienie end-to-end, wymagany jest ściśle zdefiniowany jitter.

	End-to-end delay	Jitter	Przykładowe aplikacje
Brak	-	-	Administracja, zarządzanie, diagnostyka
Miękki	>10ms	-	Automatyzacja procesów, akwizycja danych
Twardy	<1ms; 10ms>	-	Sterowanie w pętli zamkniętej
Izochroniczny	<250us; 1ms)	<1µs	Sterowanie położeniem i ruchem pracujące w pętli zamkniętej (ang. Motion control applications)

Tabela 3.1 Wymagania czasu rzeczywistego

Zgodnie z definicją kontraktów i sformułowanych w rozdziale 3.2.1 wymagań QoS, całkowity czas transmisji (ang. end-to-end delay) będę analizował wyłącznie w odniesieniu do ruchu cyklicznego. Załóżmy, że węzeł s generuje cyklicznie pakiety danych z okresem T [s] przeznaczone dla węzła docelowego d. Ponieważ w przypadku TDMA czas przyjmuje wartości wynikające z długości szczeliny czasowej, łatwiej posługiwać się liczbą szczelin czasowych w odniesieniu do zdarzeń związanych z transmisją danych. Dyskretna wartość okresu T [ts], przyjmuje postać T [ts] = $\begin{bmatrix} T[s] \\ |ts| \end{bmatrix}$, gdzie |ts| jest długością szczeliny czasowej. Konwencja zaokrąglenia w dół wynika wyłącznie z łatwości implementacji. Zdefiniujmy pojęcie "szczelina czasowa uwolnienia" (ang. release timeslot) $ts_release$ jako najwcześniejsza szczelina czasowa w której pakiet gotowy jest do wysłania. Czas, w którym n-ty pakiet gotowy jest do wysłania wynosi $T \times n$. Wprowadźmy pojęcie fazy uwolnienia phase, która stanowi przesunięcie czasu uwolnienia względem początku okresu T. Przyjmijmy że faza definiowana jest jako wartość procentowa względem długości okresu. Uwzględniając okres i fazę numer szczeliny czasowej uwolnienia j-tego pakietu w ramach ruchu generowanego dla kontraktu k wyrażamy jako:

$$ts_release_n^k = n \times T + \left\lfloor \frac{phase \times T}{100} \right\rfloor$$
(3)

Jeżeli pakiet został uwolniony w węźle źródłowym w szczelinie czasowej l oraz został w całości dostarczony do węzła docelowego w szczelinie czasowej j to całkowite maksymalne opóźnienie end-to-end dla tej trasy, wyrażone w liczbie szczelin, wynosi: j - l + 1.

Zdefiniujmy pojęcie "ostateczna szczelina czasowa" (ang. deadline timeslot) jako najpóźniejsza szczelina czasowa w której pakiet ma być dostarczony i gotowy do wykorzystania w węźle docelowym. Przyjmijmy, że strumień ruchu generowanego przez węzeł źródłowy jest uwolniony w szczelinie czasowej $ts_release_j^k$ oraz że strumień posiada zdefiniowany maksymalny czas, w którym ma być dostarczony pakiet (ang. end-to-end delay)

 Δ , wtedy numer ostatecznej szczeliny czasowej dla n-tego pakietu w ramach ruchu dla kontraktu k wyrażamy jako:

$$ts_deadline_n^k = ts_release_n^k + \left\lfloor \frac{\Delta}{|ts|} \right\rfloor - 1 \tag{4}$$

Opisaną sytuację przedstawiam na rysunku B.1. Należy zwrócić uwagę, że szczelinom czasowym dedykowanym do nadawania towarzyszy szczelina czasowa zarezerwowana u wszystkich bezpośrednich sąsiadów, którzy nie są stroną transmisji. Warunek ten dotyczy pracy na pojedynczym kanale i jest konieczny w celu uniknięcia kolizji w transmisji.

W tym miejscu musimy uporządkować dwa pojęcia, które są zamiennie wykorzystywane w literaturze i prowadzą do wieloznaczności w interpretacji. Pakiet sieciowy stanowi paczkę danych która przemieszcza się w sieci. W modelu ISO/OSI nazwa *pakiet* (ang. packet) zarezerwowana jest do określenia paczki danych na warstwie sieciowej (warstwa 3). Na warstwie dostępu do medium (warstwa 2) paczkę danych nazywamy *ramką* (ang. frame). Ramka jest podstawowym blokiem danych który przekazywany jest na warstwie fizycznej pomiędzy nadajnikiem i odbiornikiem. W przypadku szczelin czasowych zakładamy, że pojedyncza ramka może być przesłana w pojedynczej szczelinie. Ponieważ jeden pakiet może się składać z wielu ramek, jeden pakiet do wysłania może wymagać kilku szczelin czasowych.

Zdefiniujmy miarę będącą stopniem swobody w planowaniu (ang. least laxity) jako różnicą pomiędzy czasem pozostającym do ostatecznego wykonania zadania a czasem potrzebnym na jego wykonanie. Przekładając to na transmisję ramki $\tau_k^{\{i,j\}}$ wzdłuż trasy $r_k^{\{s,d\}}$ możemy zapisać stopień swobody transmisji ramki [50] (ang. laxity of frame) jako różnicę pomiędzy ilością szczelin pozostających do *ts_deadline* a ilością szczelin potrzebnych do transmisji wzdłuż pozostającej do pokonania trasy. Dla dowolnej szczeliny *ts* możemy określić stopień swobody dla transmisji ramki w ramach kontraktu *k* jako:

$$laxity_{ts}^{k} = \left(ts_deadline_{j}^{k} - ts + 1\right) - h_{ts}^{k}$$

$$\tag{5}$$

gdzie h_{ts}^k jest ilością pozostających transmisji ramki wzdłuż trasy dla kontraktu k.

3.1.2 Wymagania niezawodnościowe

W przypadku transmisji bezprzewodowej głównym czynnikiem wpływającym na poziom niezawodność jest kanał radiowy. Duża liczba różnych zjawisk fizycznych takich jak: interferencje, zacienienie, niejednorodność tłumienia, odbicia powodują, że określenie pożądanego poziomu niezawodności przez aplikację jest kluczowe do zapewnienia świadczenia usług na zakładanym poziomie.

Zakładając, że pojedyncza ramka danych jest przesyłana, z wykorzystaniem pojedynczego przeskoku, pomiędzy węzłami: nadawcą *i* a odbiorcą *j*, to prawdopodobieństwo odebrania poprawnej ramki [19] przez odbiorcę definiujemy jako:

$$p_{suc_frame}^{\{l,j\}} = p_{tr} \times (1 - BER)^{l_F}$$
(6)

gdzie p_{tr} jest prawdopodobieństwem rozpoczęcia transmisji (dla dostępu do medium z rywalizacją) [51], l_F jest całkowitą długością ramki w bitach a BER jest bitowym współczynnikiem błędu (ang. bit error rate). Praktyczną miarą prawdopodobieństwa odebrania poprawnej ramki/pakietu jest stosunek liczby poprawnie otrzymanych do liczby wszystkich nazwany współczynnikiem poprawnie otrzymanych ramek/pakietów (ang. packet delivery ratio) (PDR). Każdy węzeł działający w sieci bez dodatkowego nakładu może zliczać liczbę wszystkich odebranych ramek od konkretnych sąsiadów wraz z liczbą ramek, które rozpoznał jako niepoprawne. Mimo że to podejście nie daje możliwości zastosowania innego prawdopodobieństwa dla ramek o różnej długości, w praktyce daje najlepsze rezultaty i jest najczęściej stosowane.

W przypadku transmisji pakietu składającego się ze zbioru F ramek, prawdopodobieństwo otrzymania poprawnego pakietu wymaga poprawnego odebrania |F| ramek, w takim przypadku prawdopodobieństwo dla odebrania pakietu definiujemy jako:

$$p_{suc} = \prod_{f}^{|F|} p_{suc_frame,f}^{\{i,j\}}$$
(7)

Zakładając, że znamy średnie prawdopodobieństwo dostarczenia ramek (bez uwzględniania np. różnicy wynikającej z ich długości), to wyrażenie uprościmy do postaci:

$$p_{suc} = \left(p_{suc_frame}^{\{i,j\}}\right)^{|F|} \tag{8}$$

W rozważanej konfiguracji sieci, zakładam, że ruch pomiędzy węzłem źródłowym a węzłem docelowym może odbywać się za pośrednictwem innych węzłów (ang. multi-hop). W takim przypadku liczba transmisji jest dodatkowo uzależniona od ilości przeskoków które znajdują się na trasie do węzła docelowego oraz od prawdopodobieństwa dostarczenia ramki dla linku pomiędzy nadajnikiem *i* a odbiornikiem *j*. Prawdopodobieństwo dostarczenia pakietu definiujemy jako:

$$p_{suc} = \prod_{l_j^i \in r_k^{\{s,d\}}} \left(\prod_f^{|F|} p_{suc_frame,f}^{\{i,j\}} \right)$$
(9)
3.1.2.1 Korelacja BER/PER z fizycznymi parametrami toru radiowego

Parametry takie jak *BER* i *PER* (1 - PDR) doskonale opisują poziom niezawodność transmisji. Natomiast aktualnie zakładamy, że jedynym sposobem na ich uzyskanie jest gromadzenie statystyk dotyczących błędów przy przesyłaniu ramek pomiędzy dwoma węzłami biorącymi udział w transmisji. Takiej oceny możemy niejako dokonać *post factum*. W celu określenia poziomu PER należy zgromadzić informacje z wielu transmisji i dopiero na tej podstawie obliczyć wartość średnią. Wartości tak uzyskane zwłaszcza dla transmisji o małych przepływnościach oddają obraz kanału uśredniony za długi okres wynoszący od kilku do kilkuset sekund. Im czas ten jest dłuższy tym więcej zjawisk o charakterze chwilowym mogło mieć na niego wpływ. W tym miejscu rodzą się następujące pytania:

- Czy istnieje możliwość określenia poziomu BER/PER z parametrów samego połączenie radiowego i położenia węzłów względem siebie?
- Czy istnieje możliwość efektywnej korelacji PER/BER, których natura jest długookresowa, z innymi parametrem które są odpowiednie do określenia chwilowego stanu kanału.
- Jakie dane możemy gromadzić w celu określenia niezawodności połączenia pomiędzy dwoma węzłami komunikującymi się bezpośrednio.

Radiowe układy nadawczo-odbiorcze (ang. transceiver) dysponują możliwością pomiaru dwóch wielkości, które mogą posłużyć do oceny jakości kanału radiowego i w konsekwencji możliwości oszacowania niezawodności łącza. Pomiar mocy odbieranego sygnału wykowany jest podczas odbioru i udostępniany w postaci Received Signal Strength Indicator (RSSI) dla każdej odebranej ramki danych. Dokładność pomiaru RSSI nie jest duża i wacha się w przedziale od $\pm 3dB$ do $\pm 5dB$, w zależności od konkretnego układu radiowego oraz od użytej modulacji. Zakres pomiaru jest szeroki i zazwyczaj zaczyna się od poziomu czułości odbiornika i wynosi 90-100dB. RSSI jest wskaźnikiem mocy odbieranego sygnału i bezpośrednio nie wskazuje na jakość tego sygnału. W przypadku układu CC1352R wartość RSSI zwracana jest jako 8-bitowa liczba wyrażona w dBm i przesunięta o stałą wartość, którą należy uwzględnić przy odczycie z rejestru. Drugą wielkością wskazującą na jakość kanału jest współczynnik korelacji (ang. correlation value) (CORR) stanowiący miarę dopasowania sygnału odbieranego do sygnału oczekiwanego podczas odbioru symboli specjalnej sekwencji "Start Frame Data" (znajdującej się za preambułą sygnału radiowego). W przypadku układu CC1352R wartość CORR zapisywana jest jako 6-bitowa liczba. Powołując się na rozmowę z inżynierami TI zakładam, że miara ta przyjmuje wartości z przedziału <14; 62> gdzie 14 oznacza najniższą jakość a 62 najwyższą.

Ze względu na to, że RSSI i CORR są mierzone inaczej przez różnych producentów układów oraz że nie wskazują bezpośrednio na jakość sygnału, w protokołach do komunikacji radiowej wprowadzono dodatkowy wskaźnik *Link Quality Indicator* (LQI). Wskaźnik ten ma różną nazwę w różnych standardach komunikacji i jest definiowany odmiennie. W przypadku warstwy MAC standardu IEEE 802.15.4 [28] mamy LQI z przedziału <0x00; 0xFF>, gdzie wartości minimalna i maksymalna powinny odpowiadać sygnałom zgodnym z najniższą i najwyższą jakością wykrywaną przez odbiornik, a wartości pomiędzy nimi powinny być równomiernie rozłożone. W przypadku ISA100.11.a [27] definiujemy Received Signal Quality Indicator (RSQI), gdzie dla analogicznego przedziału wyodrębniono cztery podzakresy: 1..63 oznacza słaby sygnał, 64..127 dobry sygnał, 128..191 dobry sygnał, a 192..255 doskonały sygnał. Porównanie wartości pomiędzy różnymi węzłami ma być gwarantowane w przypadku przynależności do konkretnego podzakresu natomiast w obrębie podzakresu jedynie dla tego samego węzła.

W celu obliczenia LQI na podstawie zmierzonego RSSI, Texas Instruments w swoim oprogramowaniu stosuje następującą formułę:

$$LQI = (LQI_{MAX} * (RSSI - RSSI_{MIN})) / (RSSI_{MAX} - RSSI_{MIN})$$
(10)

gdzie LQI_MAX jest maksymalną wartością wskaźnika LQI (0xFF), RSSI_MIN i RSSI_MAX odpowiadają poziomowi RSSI dla którego LQI ma przyjmować odpowiednio najniższą i najwyższą wartość. RSSI_MIN określane jest jako poziom 10dBm powyżej czułości odbiornika (czułość dla CC1352R przy O-QPSK, 250kbps - wynosi -100dBm). RSSI_MAX przyjmowane jest jako najwyższa mierzona wartość RSSI (dla CC1352R - -100dBm + 95dB=-5dBm)

W artykule [52] przedstawiono pomiary wykonane w warsztacie przemysłowym z wykorzystaniem węzłów pracujących w standardzie IEEE 802.15.4 [28] (O-QPSK with DSSS, 250kbps). Precyzyjnie opisano warunki panujące w warsztacie wraz z położeniem wszystkich ważnych przeszkód terenowych. Przestrzeń na których badano połączenia to 16m długości, 10.5m szerokości i 5m wysokości. Węzły pracują przy mocy nadajnika -5dBm a ramka testowa ma 94 bajty. Wyniki pomiarów wskazują, że dla 90% wszystkich połączeń pomiędzy węzłami, PER jest na poziomie 0.005 lub niższym. Dokonano podziału wszystkich połączeń na cztery grupy ze względu na dostępność połączenia linii wzroku (LOS) pomiędzy węzłami i ilości ciężkich obiektów przy lub w torze radiowym: LOS-light, LOS-heavy, NLOS-light i NLOS-heavy. Pomiary tłumienia trasy (ang. path loss) na podstawie średniego RSSI są odpowiednio: -72dBm, -75dBm, -77dbm, -82dBm. W trzech pierwszych grupach średnie PER jest w przedziale 0.002 do 0.003, jedynie dla grupy NLOS-heavy rośnie do poziomu 0.009. Pomiary wskazują na całkowitą bezużyteczność stosowanego wskaźnika LQI, który praktycznie we

wszystkich pomiarach wynosi powyżej 100 (gdzie 110 jest najwyższą wartością). W pomiarach dokonano uproszczenia i przyjęto LQI = CORR, gdzie CORR jest współczynnikiem korelacji sygnału radiowego dla 8 pierwszych bajtów ramki danych (występujących po SFD). Z pomiarów jednoznacznie wynika, że współczynnik CORR daje dużo gorszą niż RSSI korelację z PER. Zaprezentowano również rezultaty dotyczące niesymetryczności połączeń radiowych w obu kierunkach. Poziom asymetrii definiuje jako różnicę pomiędzy średnim PER w jedną i drugą stronę (uplink vs downlink). W przeprowadzonym eksperymencie 90% połączeń ma asymetrię na poziomie 0.004 lub poniżej. Pomiary prowadzą do wniosku, że asymetria jest zjawiskiem, które towarzyszy połączeniom słabym. W przypadku połączeń silnych asymetria jest zjawiskiem całkowicie pomijalnym.

Podobne rezultaty zostały zaprezentowane w artykule [53], gdzie przeprowadzono pomiary dla czterech różnych lokalizacji działalności przemysłowej, dwie z nich sklasyfikowano jako "przetwarzanie drewna" i dwie jako "przetwarzanie metalu". W prezentowanym eksperymencie umieszczono antenę nadawczą na wysokości 6m (znacznie wyżej niż w innych porównywalnych doświadczeniach) dlatego porównywalne tłumienie trasy obserwujemy przy odległościach średnio o 20m większych. Autorzy potwierdzili doświadczalnie dużą korelację zaników sygnału z modelowaniem przy użyciu rozkładu Rice'a. Uzyskane wartości współczynnika K rozkładu to: 16dB dla w pełni zautomatyzowanej produkcji (znikoma liczba ruchomych obiektów), 5dB dla produkcji manualnej.

W artykule [54] zaprezentowano wyniki szacowania niezawodności połączenia poprzez określenie PER z wykorzystaniem emulatora kanału radiowego. Wyniki PER odpowiadają ramce o rozmiarze PSDU=127B wraz z ACK. Przedstawino zależność PER w funkcji RSSI przy różnych modelach kanału: idealny AWGN, Rice'a (dla $K \in \{2,4,8,16\}$ oraz Rayleigha. Zakładając dla połączenie pomiędzy dwoma węzłami PER=10e-4 dla modelu Rice'a (K=4) średnie RSSI musi być na poziomie -70dBm lub więcej.

W artykule [55] zaprezentowano wyniki pomiarów PER i RSSI dla dwóch fabryk zlokalizowanych w Szwecji: walcownii i fabryce papieru. W przeciwieństwie do już omawianych publikacji, dane zebrano z dłuższego okresu czasu wynoszącego około 20h. Na podstawie wyników wyciągnięto wniosek że model oparty na rozkładzie Rice'a przeszacowuje potrzebny RSSI do uzyskania zakładanego BER. Porównano wartości mierzone z wartościami wygenerowanymi dla wielu modeli kanału radiowego w tym Rice'a i Rayleigha. W praypadku modelu Rice'a zastosowano współczynnik K=3.55, co jest wartością bardzo niską. Na podstawie wyciągniętych wniosków stworzono nowy model stanowiący połączenie modelu Nakagami i logarytmicznego.

W artykule [56] porównano korelację PDR (1-PER) z dwiema wielkościami wskazującymi na jakość połączenia (LQI). Wykorzystano dwie dostępne w odbiorniku wielkości: RSSI i CORR. Zauważono, że zależność PDR od RSSI jest bardzo nieliniowa. Dla szerokiego zakreu wartości RSSI współczynniki PDR utrzymuje się na poziomie 1. Dla wartości w przedziale 10-20dB powyżej czułości odbiornika następuje gwałtowny spadek współczynnika. Wskazano, że zależność ta jest lepsza w przypadku uzycie CORR jako LQI. W tym przypadku zależność jest o wiele bardziej równomierna. Natomiast w dalszym ciągu istnieje pewien zakres w którym ciężko sklasyfikować połączenie jako dobre lub złe gdy LQI jest powyżej określonej wielkosci (>165).

Moc sygnału radiowego który dociera do odbiornika jest uzależniona od poziomu tłumienia trasy łączącej nadajnik z odbiornikiem. Do modelowania tłumienia trasy w funkcji odległości między węzłami wykorzystywany jest model logarytmiczny [57] (ang. log-distance path-loss), jako

$$PL(d) = PL(d_0) + 10 \times \alpha \times \log_{10}(d/d_0)$$
⁽¹¹⁾

gdzie: d jest odległością między węzłami a $PL(d_0)$ jest tłumieniem trasy na odcinku odniesienia obliczonym z wykorzystaniem modelu Friis'a dla otwartej przestrzeni (ang. Friis free-space path loss). Wartość d_0 dla przestrzeni zamkniętych przyjmuje się z przedziału <1;10>m, natomiast w otwartej przestrzeni do około 100m.

W literaturze możemy spotkać bardzo wiele oszacowań wartości współczynników $PL(d_0)$ oraz α uzyskanych na podstawie przeprowadzonych praktycznych pomiarów [53, 57, 58]. Do modelowania przyjmowana jest pojedyncza prosta (ang. one-slope model) lub podwójna (ang. two-slope model), gdzie końcowy fragment modelowany jest prostą o większym nachyleniu [57]. W artykule [59] wykonano dodatkowo pomiary dla różnych konfiguracji wysokości anten. Dla konfiguracji anten 0,25m nadawcza i 0,25 odbiorcza zarówno współczynnik α jak również odchylenie standardowe rozkładu rosną o około 40% w stosunku do konfiguracji gdzie obie anteny są na wysokości 1,75m.

W tabeli B.6 zebrałem, na podstawie literatury, charakterystyczne wartości odpowiadające interesującym dla IWN warunkom przemysłowym. Wartości prezentuję dla przypadku, gdy istnieje widoczność bezpośrednia (LOS) oraz dwóch przy braku widoczności (NLOS) i różnej liczbie przeszkód na trasie sygnału. Na rysunku B.2 prezentuję tłumienie trasy (lewy) w funkcji odległości dla różnych wartości parametrów z tabeli B.6 oraz zależność PER w funkcji średniego RSSI na podstawie [54] (prawy).

Podsumowując przeprowadzoną w niniejszym rozdziale analizę, należy odnotować następujące wnioski:

- Modelowanie kanału za pomocą AWGN dla warunków sieci przemysłowych ma znikomą użyteczność
- W środowisku przemysłowym mamy propagację sygnału wielokrotnie odbitego (ang. heavy multipath propagation)
- Charakterystyka kanału radiowego w interesujących nas warunkach przemysłowych może być modelowana za pomocą rozkładu Rice'a lub w skrajnym przypadku rozkładu Rayleigha.
- Współczynnik K dla modelu Rice'a należy dobierać jako K=4 dla powierzchni przemysłowych z wieloma ruchomymi obiektami w torze radiowym i K=8 lub 16 dla powierzchni przemysłowych o dużym poziomie automatyzacji, gdzie istnieją względnie duże korytarze między wyspami bloków urządzeń i gdzie mało jest obiektów ruchomych tymczasowo przesłaniających tor sygnału.

Podsumowując, w celu określenia jakości połączenia pomiędzy dwoma węzłami dysponujemy następującymi wielkościami:

- RSSI oraz LQI
- Liczba powtórzeń w procedurze CCA
- Liczba wysłanych bajtów w tym liczba wysłanych bajtów, dla których nie odebrano ACK lub odebrano NACK
- Liczba odebranych bajtów w tym liczba bajtów, dla których nie wysłano ACK lub wysłano NACK

3.2 Konfiguracja sieci IWN

3.2.1 Definicja kontraktu

Kontrakt			
Cykliczny (periodic)	Aperiodyczny (aperiodic)	Opis	
Priority		Priorytet kontraktu: 3 – kontrola sieci, krytyczne zarządzanie 2 – komunikacja cykliczna w czasie rzeczywistym 1 – komunikacja sekwencyjna w czasie rzeczywistym 0 – pozostały ruch	
Destination_Address		IPv6 adres oraz numer portu UDP	
Payload_Size		Rozmiar pakietu [bajty] który węzeł będzie wysyłać w ramach kontraktu	
Reliability		Prawdopodobieństwo dostarczenia pakietu (PDR)	
Expiration		Czas obowiązywania kontraktu	
Period		Okres z jakim wysyłany jest pakiet	

Phase Deadline		Określa miejsce w obrębie okresu, w którym dane powinny być gotowe do wysłania
		Maksymalne opóźnienie end-to-end
	Committed_Burst	Długookresowa średnia liczba pakietów w ciągu sekundy
	Excess_Burst	Krótkookresowa średnia liczba pakietów w ciągu sekundy

Tabela 3.2 Pola kontraktu

Węzeł źródłowy w celu komunikacji z wybranym węzłem docelowym musi wystąpić do SM o zawarcie kontraktu. Kontrakt opisuje podstawowe parametry dotyczące ruchu między dwoma węzłami (źródłowy \rightarrow docelowy) w szczególności dotyczące wymaganych parametrów QoS. Kontrakt określa również czas trwania oraz typ ruchu jaki będzie obsługiwany. Kontrakt stanowi umowę pomiędzy menadżerem sieci (SM) a urządzeniem w sieci, która obejmuje przydział zasobów sieciowych przez SM w celu zaspokojenia określonej potrzeby komunikacyjnej tego urządzenia. Urządzenie które wnioskuje o kontrakt jest źródłem generowanego ruchu, a urządzenie, z którym chce się komunikować, jest węzłem docelowym. Zawarta umowa powinna ustanawiać i obsługiwać ścieżkę komunikacyjną między urządzeniami w sieci, aby zaspokoić potrzeby komunikacyjne procesu aplikacji. Proces aplikacyjny, który wymaga komunikacji z procesem aplikacyjnym w innym urządzeniu, wymaga zawarcia umowy.

W tabeli 3.2 prezentuję pola definiujące kontrakt według standardu ISA100.11a [27]. Użyte angielskie nazwy pól odpowiadają zdefiniowanym w standardzie i będę się nimi posługiwał według wzorca $c(p|a)_k^{\{s,d\}}$. $\{f\}$, szczegóły notacji w tabeli 3.5. Priorytet kontraktu oraz typ ruchu są podstawowymi parametrów określającymi jakie metody i algorytmy muszą być użyte w celu alokacji zasobów tak aby możliwe było wypełnienie wymagań QoS. W tabeli 3.3 przedstawiam charakterystykę ruchu dla każdego z priorytetów kontraktów (Tabela 3.2). Określam podstawowe wielkości takie jak: rodzaj kontraktu, sposób rywalizacji w szczelinie czasowej, wielkość pakietu oraz przepływność, podając oczekiwane wartości.

Priorytet	3	2	1	0
Kontrakt	$cp_k^{\{s,d\}} \lor ca_k^{\{s,d\}}$	$cp_k^{\{s,d\}}$	са	{s,d} k
Szczelina	dedyk	towana	dedykowana /	współdzielona
Pakiet	50-500B	5-50B	100-	1000B
Aplikacja	Krytyczne zarządzanie siecią, rekonfiguracja	Sterowanie w pętli zamkniętej, automatyzacja procesów	Aplikacje strumieniowe, transmisja dźwięku/obrazu	Komunikacja klient/serwer
Przepływność	0.5-2kbit/s	100kbit/s	1-2Mbit/s	-

Tabela 3.3 Charakterystyka ruchu z podziałem na priorytety kontraktu

Standard ISA100.11a definiuje niezawodność w kontrakcie w sposób skwantyfikowany {low, medium, high} i nie przypisuje konkretnych wartości oraz nie definiuje miar. Formaty typów wykorzystywanych do przesyłania danych kontraktu, posiadają 7 bitów zarezerwowanych na zakodowanie niezawodności. W opracowaniu będę posługiwał się następującym zapisem niezawodności w ramach dostępnej 7-bitowej przestrzeni. Do opisu poziomu niezawodności PDR = 1 - PER w ramach kontraktu przechowuję PER, który zapisuje z wykorzystaniem mantysy i wykładnika. Bity 1..4 opisują mantysę, natomiast bity 5..7 opisują ujemny wykładnik, zbiór możliwych wartości PER definiuję zatem jako { $x \cdot 10^{-y} | x \in \{0,...,15\}, y \in \{0,...,7\}$ }. Zapisanie PER zamiast PDR, który jest wprost miarą niezawodności, podyktowane jest wyłącznie łatwością i kompaktowością zapisu wartości liczbowych występujących w praktycznych aplikacjach (PDR=99,999% vs PER=1e-3%).

Standard ISA100.11a definiuje dla pola "deadline" jednostkę 10ms, takie ograniczenie wydaje się nieuzasadnione. Na potrzeby niniejszego opracowania przyjmuję, że jednostką dla pola "deadline" opisującego kontrakt jest długość szczeliny czasowej. Domyślnie przyjmowana długość szczeliny to ~10ms (10485 $\cdot 2^{-20}s$) ale może być zmieniona w celu badania wpływu długości szczeliny na wypełnienie wymagań QoS.

Standard ISA100.11a definiuje wartość w polu "period" jako niezerową liczbę całkowitą. Dla wartości ujemnych okres jest równy $-\frac{1}{x}[s]$ a dla dodatnich x[s]. W ten sposób przy zapisie 16-bitowym uzyskuje się szeroki zakres w praktyce od 10ms do 32767 s. Dolny zakres wynika z długości pojedynczej szczeliny czasowej.

3.2.2 Komunikacja cykliczna i komunikacja sekwencyjna

Rozróżniam dwa typy ruchu dla których w odmienny sposób definiowane są wymagania QoS. Ruch cykliczny jest szczególnym rodzajem ruchu dedykowanym do wykonywania komunikacji okresowej i jest odpowiedzią na istnienie potrzeby komunikacji w czasie rzeczywistym szczególnie w układach przemysłowych z zamkniętą pętlą. Ruch cykliczny jest definiowany poprzez określenie: okresu z jakim uwalniany jest ruch, fazy okresu w której następuje uwolnienie oraz maksymalnego wymaganego czasu na dostarczenie pakietu do węzła docelowego. Drugim rodzajem ruchu jest aperiodyczny (ang. aperiodic), który w niniejszym opracowaniu nazywam ruchem sekwencyjnym. Ten rodzaj ruchu jest definiowany poprzez określenie średniej wymaganej przepływności oraz przepływności chwilowej. W rozdziale 4.2.3 opisałem wykorzystywany sposób modelowania ruchu cyklicznego i sekwencyjnego. Ruch sekwencyjny zamodelowałem w oparciu o przesunięty rozkład Bernoulliego na podstawie parametrów contract.committed burst oraz contract.excess_burst.

3.2.3 Definicja sieci

Rozpatrywana sieć IWN jest typu kratownicy (ang. mesh) i została zamodelowana jako graf skierowany G = (V, E), gdzie V stanowi zbiór węzłów sieci (v_i) a E jest zbiorem krawędzi skierowanych (e) pomiędzy węzłami. Krawędź $e = (v_i, v_i)$ istnieje wtedy i tylko wtedy, gdy pomiędzy węzłami v_i, v_j istnieje połączenie radiowe l_j^i zdolne przesyłać dane z wykorzystaniem warstwy fizycznej (PHY) tej samej dla wszystkich węzłów sieci. $N_V = |V|$ jest liczbą wszystkich węzłów w sieci. Każdy węzeł sieci ($v_i \in V$) może być źródłem transmisji danych (wtedy $v_i \in S$) oraz stanowić punkt docelowy transmisji danych (wtedy $v_i \in D$) tak, że $S \subset V$; $D \subset V$; $S \cup D = V$; $S \cap D \ge 0$. Dane są przesyłane od źródła ($v_i \in S$) do punktu docelowego ($v_i \in D$) za pośrednictwem innych węzłów ($v_i \in R$) (ang. multihop). Każdy węzeł $(v_i \in V)$ może wysyłać i odbierać ramki danych. Wysłanie i odbiór pojedynczej ramki danych realizowane pomiędzy dwoma węzłami v_i, v_j nazywamy transmisją $\tau_k^{\{i,j\}}$, gdzie węzeł v_i nazywamy nadawcą a węzeł v_i odbiorcą transmisji. Maksymalna długość pojedynczej ramki danych jest zdefiniowana dla całej sieci (np. 127B). Generowany ruch jest typu unicast, gdzie strumień danych od konkretnego węzła źródłowego jest przeznaczony dla jednego węzła docelowego. Unicast jest najczęściej stosowany w sieciach przemysłowych, natomiast prezentowana w opracowaniu metodologie można łatwo rozciagnąć na ruch typu multicast, gdzie strumień danych od pojedynczego węzła źródłowego jest przeznaczony dla wielu węzłów docelowych. Ruch typu multicast jest stosowany rzadko w sieciach przemysłowych, głównie ze względu na możliwość jego zastąpienia kilkoma strumieniami unicast.

Ruch w sieci realizowany jest wyłącznie w oparciu o przydzielane kontrakty $c_k^{\{s,d\}} \in C$, gdzie $c_k^{\{s,d\}}$ to kontrakt o numerze k przydzielony dla węzła źródłowego *s* do komunikacji z węzłem docelowym d. Rozróżniamy dwa rodzaje kontraktów:

- Kontrakt dla komunikacji cyklicznej $cp_k^{\{s,d\}}$
- Kontrakt dla komunikacji sekwencyjnej (aperiodycznej) $ca_k^{\{s,d\}}$

Zakładamy, że węzeł respektuje ustalenia kontraktu, tym samym parametry kontraktu są parametrami definiującymi ruch w sieci. Pojedyncza transmisja ($\tau_k^{\{i,j\}}$) może być realizowana wyłącznie w związku z jednym kontraktem k. Parametry opisujące oba rodzaje kontraktów zostały zebrane w tabeli 3.2. Kontrakt definiuje wymagania jakościowe QoS jako QP_k = ($\rho_k, \Delta_k, \lambda_k, \xi_k$) dla komunikacji cyklicznej i QA_k = (ρ_k, λ_k, ξ_k) dla komunikacji sekwencyjnej, gdzie:

- ρ_k wymagana niezawodność (ang. reliability),
- Δ_k wymagane maksymalne opóźnienie przesłania pakietu w ramach kontraktu pomiędzy węzłem źródłowym a węzłem docelowym (ang. end-to-end delay),

- λ_k wymagana minimalna średnia przepływność dla trasy pomiędzy węzłem źródłowym a węzłem docelowym (ang. throughput)
- ξ_k wymagany minimalny czas, w którym obowiązują zdefiniowane parametry jakościowe (QoS)

W tabeli 3.4 zdefiniowałem zależności pomiędzy wymaganiami QoS a parametrami definiowanymi w kontrakcie (tabela 3.2).

Symbol	Kontrakt	Opis [Jednostka]
ρ_k	$c(p a)_k^{\{s,d\}}$. Reliability	Minimalne prawdopodobieństwo dostarczenia pakietu do węzła docelowego. [%]
Δ_k	$cp_k^{\{s,d\}}$. Deadline	Oczekiwana wartość czasu stanowiącego różnicę pomiędzy czasem odebrania pakietu na węźle docelowym i czasem w którym pakiet powinien być gotowy do wysłania na węźle źródłowym. [s]
λ_k	$\frac{cp_{k}^{\{s,d\}}. \text{Payload}_\text{Size}}{cp_{k}^{\{s,d\}}. \text{Period}}$ $ca_{k}^{\{s,d\}}. \text{Payload}_\text{Size}$ $\times ca_{k}^{\{s,d\}}. \text{Committed}_\text{Burst}$	Minimalna średnia wartość przepływności pomiędzy węzłem źródłowym i docelowym liczona inaczej dla cyklicznego i sekwencyjnego. [bajt/s]
ξ_k	$c(p a)_k^{\{s,d\}}$. Expiration	Minimalny czas do kiedy obowiązuje ustalony kontrakt i wymagane parametry QoS. [s]

Tabela 3.4 Zależność pomiędzy wymaganiami QoS a parametrami kontraktu dla ruchu cyklicznego QP_k i sekwencyjnego QA_k

Dla sieci IWN należy założyć, że połączenie radiowe pomiędzy węzłami v_i i v_j musi być modelowane za pomocą krawędzi skierowanej, ponieważ:

- Każdy węzeł $v_i \in V$ może dysponować inną efektywną mocą wyjściową nadajnika
- Każdy węzeł $v_i \in V$ może dysponować inną czułością odbiornika
- Przeszkody terenowe oraz źródła zakłóceń mogą powodować, że tłumienie trasy w obu kierunkach może być inne (asymetria połączenia, rozdział 3.1.2.1)

Transmisja może odbywać się na jednym kanale ch ze zbioru wszystkich kanałów CH dostępnych dla danej definicji warstwy fizycznej (PHY).

Ruch w sieci dystrybuowany jest w sposób pozwalający unikać kolizji. Definiujemy następujące reguły w celu unikania kolizji w sieci:

- 1. Każdy węzeł $v_i \in V$ może wykonywać tylko jedną operację radiową w tym samym czasie, tj. nie może odbierać, gdy nadaje i nadawać, gdy obiera dane;
- Każdy węzeł v_j ∈ V który odbiera na kanale ch nie może być zakłócany w tym samym czasie przez swojego bezpośredniego sąsiada, który nadaje na kanale *ch* i nie jest stroną połączenia lⁱ_i;

Złamanie zdefiniowanych reguł nazywamy odpowiednio kolizją pierwszego i drugiego stopnia.

Wszystkie węzły sieci obsługują automatyczne żądanie powtórzenia (ang. automatic repeat request) (ARQ), w którym odbiorca wykorzystuje potwierdzenie (ang. acknowledgement) aby powiadomić nadawcę w określonym czasie (ang. timeout) o poprawnym odebraniu przesłanej ramki. Potwierdzenie ACK jest małą ramką kontrolną wysyłaną w tej samej szczelinie czasowej co ramka z danymi. Zakładamy, że jeżeli pomiędzy węzłami v_i, v_j istnieje połączenie l_j^i stanowiące krawędź grafu połączeń *G*, to mimo braku połączenia odwrotnego l_i^j w tym grafie węzeł v_j ma zdolność poprawnego przesłania ACK. Takie założenie możemy poczynić, ponieważ długość ACK jest o wiele mniejsza od długości ramki danych co zgodnie ze wzorem (6) prowadzi do uzyskania porównywalnego prawdopodobieństwa otrzymania ramki nawet przy dużo gorszych parametrach kanału zwrotnego.

Zakładamy, że sieć jest zarządzana w sposób scentralizowany poprzez jednostkę System Managera (SM). Zadaniem SM jest gromadzenie informacji o sieci oraz tworzenie konfiguracji dla każdego z węzłów sieci na podstawie zgłoszeń dotyczących zawarcia kontraktu. W wyniku realizacji procedury "odkrywania topologii sieci" (ang. neighbor discovery), każdy z węzłów zbiera informacje o swoich bezpośrednich sąsiadach. Te informacje przekazywane są do SM w sposób cykliczny lub na żądanie SM tak że SM posiada aktualny obraz topologii sieci. Częstość gromadzenia danych jest definiowana przez SM i odpowiada charakterowi węzłów sieci, np. mobilności.

W tabeli 3.5 zebrałem wykorzystywaną notację. Pierwsza kolumna zawiera symbole a druga
zawiera opis wraz z notacją matematyczną definiującą pojęcie i prezentującą zależności.

Symbol	Definicja		
G	Graf połączeń węzłów sieci, skierowany, $G = (V, E)$		
$v_i; V; N_V$	Węzeł v_i o numerze <i>i</i> w zbiorze węzłów sieci V o liczebności N_V , $V = \{v_i \mid i = 1,, N_V\}; N_V = V $		
S; D	Zbiór węzłów odpowiednio źródłowych i docelowych transmisji, $S \subset V$; $D \subset V$; $S \cup D = V$; $S \cap D \ge 0$;		
10; R; B	Zbiór węzłów pełniących rolę w sieci odpowiednio: węzeł końcowy bez zdolności przekazywania ramek; węzeł posiadający zdolność przekazywania ramek do innych węzłów; węzeł brzegowy do sieci szkieletowej (ang. backbone router); $IO \subset V$; $R \subset V$; $B \subset V$; $IO \cup R \cup B = V$; $IO \cap R = 0$; $B \cap (IO \cup R) \ge 0$		
$l_j^i; E$	Połączenie l_j^i z węzła o numerze i do węzła o numerze j w zbiorze połączeń $E, l_j^i = \{v_i, v_j\}; E = \{l_j^i \mid v_i, v_j \in V \land v_i \neq v_j\}$		
$r_k^{\{s,d\}}; RT$	Trasa dla kontraktu o numerze <i>k</i> z węzła źródłowego o numerze <i>s</i> do węzła docelowego o numerze <i>d</i> w zbiorze tras routingu <i>RT</i> (tablica routingu), $r_k^{\{s,d\}} = \{l_i^s,, l_d^j \mid v_s \in S, v_d \in D, v_i, v_j \in R\}; RT = \{r_k^{\{s,d\}} \mid c_k^{\{s,d\}} \in C\};$		
NE _i	Zbiór bezpośrednich sąsiadów węzła v_i , $NE_i = \{v_j \mid l_j^i \in E\}$		

NL_j^i	Zbiór połączeń węzłów sieci, narażonych na konflikt 2'ego stopnia w przypadku realizacji transmisji ramki pomiędzy nadawcą <i>i</i> a odbiorcą <i>j</i> , $NL_j^i = \left\{ l_{j'}^{i'} \mid v_{i'} \in (NE_i \cup NE_j) \right\}$		
CAP _i	N-ka z parametrami opisującymi możliwości węzła v_i .		
C; CP; CA	Zbiór kontraktów C będący sumą zbiorów kontraktów komunikacji cyklicznej CP i sekwencyjnej CA, $C = CP \cup CA$; $CP \cap CA = 0$;		
$c_{k}^{\{s,d\}}; cp_{k}^{\{s,d\}}; \ ca_{k}^{\{s,d\}}$	Kontrakt <i>k</i> dla węzła źródłowego o numerze <i>s</i> do komunikacji z węzłem docelowym o numerze <i>d</i> , $c_k^{\{s,d\}} \in C$; dla komunikacji cyklicznej $cp_k^{\{s,d\}} \in CP$ oraz sekwencyjnej $ca_k^{\{s,d\}} \in CA$		
$wc_k^{\{s,d\}}$	Waga kontraktu k, zależna od priorytetu kontraktu. Waga kontraktu o najniższym priorytecie to 1, $wc_k^{\{s,d\}} = \left(\sum_{x \in \{x \in C \mid x. priority = c_k^{\{s,d\}}. priority - 1\}} wc_x^{\{s,d\}}\right) + 1$		
$c(p a)_k^{\{s,d\}}.\{f\}$	Wartość pola o nazwie f definiującego kontrakt $c(p a)_k^{\{s,d\}}$, zgodnie z definicją pól w Tabela 3.2		
$QP_k; QA_k$	Wymagania jakościowe QoS dla ruchu cyklicznego, $QP_k = (\rho_k, \Delta_k, \lambda_k, \xi_k)$ oraz ruchu sekwencyjnego, $QA_k = (\rho_k, \lambda_k, \xi_k)$ w ramach kontraktu k		
$ ho_k$	Wymagana niezawodność (ang. reliability) dla kontraktu k		
\varDelta_k	Wymagane maksymalne opóźnienie przesłania pakietu w ramach kontraktu k pomiędzy węzłem źródłowym a węzłem docelowym (ang. end-to-end delay)		
λ_k	Wymagana minimalna średnia przepływność dla trasy pomiędzy węzłem źródłowym a węzłem docelowym w ramach kontraktu k (ang. throughput)		
ξ_k	Wymagany minimalny czas, w którym obowiązują zdefiniowane parametry jakościowe (QoS) w ramach kontraktu k		
$ au_k^{\{i,j\}}$	Transmisja ramki danych pomiędzy nadawcą <i>i</i> a odbiorcą <i>j</i> , realizowana w związku z kontraktem <i>k;</i>		
BUF _i ; BUF	Bufor węzła v_i z transmisjami ramek oczekującymi na wysłanie, $BUF_i = \{\tau_k^{\{i,j\}} \mid l_j^i \in r_k^{\{s,d\}}\}; BUF = \{BUF_i \mid v_i \in V\}$		
<i>ts</i> ; <i>ts</i>	Numer szczeliny czasowej i jej długość [s]		
ch; CH; N _{CH}	Numer kanału radiowego ch w zbiorze dostępnych kanałów CH o liczebności $N_{CH}, ch \in CH; N_{CH} = CH $		
$CH_i; N^i_{CH}$	Zbiór kanałów, na których węzeł v_i może nadawać, $N_{CH}^i = CH_i $		

SCH _i ; SCH ^{CP} ; SCH ^{CA} ; SCH	Harmonogram transmisji $\tau_k^{\{i,j\}}$ węzła v_i . Zbiór elementów $SCH_i = \{el \mid el = (p, c_k^{\{s,d\}}, l_j^i, ts, ch); v_i, v_j \in V; c_k^{\{s,d\}} \in C; v_s \in S; v_d \in D\},$ gdzie p jest okresem powtarzania elementu harmonogramu. Znając SCH_i można skonstruować odpowiadający mu harmonogram odbiorów SCH_j^{RX} dla węzła v_j ; $SCH_i^{CP} = \{el \mid el \in SCH_i \land c_k^{\{s,d\}} \in CP\}$; $SCH_i^{CA} = \{el \mid el \in SCH_i \land c_k^{\{s,d\}} \in CP\}$; $SCH_i^{CA} = \{el \mid el \in SCH_i \land c_k^{\{s,d\}} \in CA\}$; $SCH = \{SCH_i \mid v_i \in V\}$
E _{SCHi}	Energia jaką zużyje węzeł v_i przy realizacji harmonogramu SCH_i
$E^i_{ON_{total}}; \ E^i_{OFF_{total}}; \ E^i_{TX}; E^i_{RX}$	Poziomy zużycia energii dla węzła υ_i dla poszczególnych faz pracy zdefiniowanych w rozdziale 2

Tabela 3.5 Wykorzystywana notacja

3.2.4 Definicja problemu

Każdy węzeł sieci ($v_i \in V$) może być źródłem transmisji danych i formułować wymagania jakościowe QoS jako $q \in QP_k \cup QA_k$ w ramach kontraktu $c_k^{\{s,d\}}$, który zamierza zawrzeć z menadżerem sieci (SM). SM w procesie formowania sieci i na bieżąco w postaci cyklicznych raportów uzyskuje od węzłów informacje dotyczące ich parametrów oraz topologii sieci. Zakładamy, że uzyskane informacje pozwalają SM na zbudowanie grafu połączeń G = (V, E) oraz informacji o parametrach każdego z węzłów CAP_i . SM dla wszystkich wymagań QoS jako $QP_k \cup QA_k$ szuka poprawnej i optymalnej konfiguracji sieci *conf* = $\{RT, SCH\}$, gdzie *RT* jest zbiorem wszystkich tras routingu a *SCH* jest zbiorem wszystkich harmonogramów transmisji dla wszystkich węzłów sieci.

Problem znalezienia poprawnej konfiguracji sieci, która optymalnie wypełnia przyjęte wymagania jakościowe (QoS) definiujemy jako:

$$\max_{conf} o_q(conf)$$

pod warunkiem,

$$k_{0}: \quad \forall v_{i} \in V \left(\neg \exists el \in SCH_{i} (\exists el \in SCH_{i} (el. ts = el'. ts \land el \neq el')) \right)$$

$$k_{1}: \quad \forall el \in SCH_{i}^{CP} \left(\neg \exists el' \in SCH_{i}^{CP} \begin{pmatrix} el. ts = el'. ts \land el. ch = el'. ch \\ \land el. l_{j}^{i} \in NL_{j'}^{i'} \land el \neq el' \end{pmatrix} \right)$$

$$k_{2}: \qquad \forall v_{i} \in V (E_{SCH_{i}} \leq CAP_{i}. Energy_left)$$

$$(12)$$

gdzie

$$o_q(conf) = \sum_{\substack{c_k^{\{s,d\}} \in C}} \begin{cases} wc_k^{\{s,d\}} & \text{if soft}\left(c_k^{\{s,d\}}, conf\right) = 1\\ 0 & \text{otherwise} \end{cases}$$
(13)

$$soft\left(c_{k}^{\{s,d\}},conf\right) = \begin{cases} softPeriodic\left(c_{k}^{\{s,d\}},conf\right) & if \ c_{k}^{\{s,d\}} \in CP\\ softAperiodic\left(c_{k}^{\{s,d\}},conf\right) & if \ c_{k}^{\{s,d\}} \in CA \end{cases}$$
(14)

$$softPeriodic\left(cp_{k}^{\{s,d\}},conf\right) = \begin{cases} r_{k}^{\{s,d\}} \in RT \land :mk_{0} \\ 1 & if \ \rho_{k}' \ge \rho_{k} \land :mk_{1} \\ \Delta_{k}' \le \Delta_{k} & :mk_{2} \\ 0 & otherwise \end{cases}$$
(15)

$$softAperiodic\left(ca_{k}^{\{s,d\}},conf\right) = \begin{cases} 1 & if \ r_{k}^{\{s,d\}} \in RT \land :mk_{0} \\ \rho_{k}' \ge \rho_{k} & :mk_{1} \\ 0 & otherwise \end{cases}$$
(16)

Apostrofem oznaczyłem parametry uzyskane przez sieć w wyniku zastosowania przygotowanej konfiguracji, bez apostrofu parametry wymagane przez kontrakt. Funkcja celu (12) poszukuje optymalnej między-warstwowej (ang. cross-layer) konfiguracji sieci $conf = \{RT, SCH\}$ poprzez maksymalizację wartości funkcji $o_q(conf)$. Wartość tej funkcji należy interpretować jako liczbę możliwych do zawarcia kontraktów, przy czym waga kontraktu jest różna i zależna od priorytetu kontraktu (13). Waga została skonstruowana w taki sposób, że waga kontraktu o priorytecie $c_k^{\{s,d\}}$. priority jest o 1 większa od wagi wszystkich możliwych do zawarcia kontraktów o priorytecie $c_k^{\{s,d\}}$. priority – 1, gdzie $c_k^{\{s,d\}}$. priority – 1 oznacza priorytet o 1 stopień niższy od priorytetu $c_k^{\{s,d\}}$. priority na skali porównawczej. Dla funkcji celu (12) zdefiniowano trzy twarde warunki (ang. hard constraints): k_0, k_1, k_2 . Warunki k_0 oraz k_1 definiują brak kolizji odpowiednio 1'ego i 2'go stopnia dla harmonogramu SCH generowanego dla wszystkich węzłów sieci. Warunek k_0 odnosi się do wszystkich elementów planu natomiast warunek k_1 odnosi się wyłącznie do kontraktów cyklicznych, ponieważ dla nich ruch odbywa się w dedykowanych szczelinach czasowych (ang. contention free), w przeciwieństwie do ruchu sekwencyjnego, gdzie zakładam rywalizację między węzłami w dostępie do szczeliny (ang. contention based). Warunek k_0 jest spełniony, jeżeli w każdym harmonogramie SCH_i dla węzła $v_i \in V$ istnieje tylko jedna aktywność w tej samej szczelinie czasowej. Warunek k_1 jest spełniony, jeżeli we wszystkich harmonogramach SCH_i dla węzłów $v_i \in V$ istnieje tylko jedna aktywność w tej samej szczelinie czasowej i na tym samym kanale realizowana przez nadawcę lub jego bezpośrednich sąsiadów, taka, że $c_k^{\{s,d\}} \in CP$. Warunek k_2 określa zdolność sieci do poniesienia wydatku energetycznego niezbędnego do wykonania planu *SCH*. Każdy węzeł sieci deklaruje ilość energii jaką dysponuje CAP_i . Energy_left a SM określa wydatek energetyczny dla każdego węzła sieci (E_{SCH_i}) . Warunek uznajemy za spełniony, jeżeli każdy węzeł jest zdolny do poniesienia obliczonego wydatku energetycznego. Wydatek energetyczny poniesiony przez węzeł $v_i \in V$ na realizację planu SCH_i (17) definiujemy jako sumę energii zużytej na transmisję ramek i sumę energii zużytej na odbiór ramek w czasie wynikającym z parametru QoS dotyczącego czasu obowiązywania kontraktu ($\xi_{el.c_k}^{\{s,d\}}$) z uwzględnieniem okresu powtarzania elementu planu (*el. p*). Poziomy zużycia energii dla poszczególnych faz pracy węzła odpowiadają zdefiniowanym w rozdziale 2 [24].

$$E_{SCH_i} = E_{SCH_i}^{TX} + E_{SCH_i}^{RX}$$
(17)

$$E_{SCH_i}^{TX} = \sum_{el \in SCH_i} \left[\left(E_{ON_{total}}^i + E_{OFF_{total}}^i + \frac{PPDU^{el}}{R} \cdot E_{TX}^i + \frac{PPDU_{ACK}^{el}}{R} \cdot E_{RX}^i \right) \cdot \frac{\xi_{el.c_k^{\{s,d\}}}}{el.p} \right]$$
(18)

$$E_{SCH_i}^{RX} = \sum_{el \in SCH_i} \left[\left(E_{ON_{total}}^i + E_{OFF_{total}}^i + \frac{PPDU^{el}}{R} \cdot E_{RX}^i + \frac{PPDU_{ACK}^{el}}{R} \cdot E_{TX}^i \right) \cdot \frac{\xi_{el.c_k^{\{s,d\}}}}{el.p} \right]$$
(19)

gdzie $PPDU^{el}$ jest całkowitą długością ramki na warstwie fizycznej wyrażoną w bitach (ang. Physical layer protocol data unit), $PPDU_{ACK}^{el}$ jest długością ramki z potwierdzeniem otrzymania ramki danych a *R* jest przepływnością wyrażoną w bitach na sekundę. Długość PPDU wynika wprost z zawartej w kontrakcie (*el.* $c_k^{\{s,d\}}$) "Payload_Size" (tabela 3.2) powiększonej o długości nagłówków na poszczególnych warstwach.

Funkcja celu (12) zlicza kontrakty zgodnie z ich wagami, jedynie pod warunkiem wypełnienia przez kontrakt miękkich warunków (14) (ang. soft constraints). W zależności od typu kontraktu do wypełnienia są warunki: mk_0, mk_1, mk_2 dla kontraktów cyklicznych (15) i mk_0, mk_1 dla kontraktów sekwencyjnych (16). Warunek mk_0 odnosi się do poprawności wygenerowanej trasy routingu. Warunek uznajemy za wypełniony jeżeli w tablicy routingu *RT* została wygenerowana i umieszczona trasa $r_k^{\{s,d\}}$ pozwalająca na dotarcie z węzła źródłowego *s* do węzła docelowego *d*. Warunki mk_1, mk_2 odnoszą się do zdolności realizacji zadanych parametrów QoS kontraktu *k*. Warunek mk_1 odnosi się do wymaganej niezawodności i jest spełniony jeżeli niezawodność ρ'_k dostarczenia ruchu dla kontraktu *k* jest nie mniejsza niż niezawodność ρ_k zadana w wymaganiach QoS kontraktu. Warunek mk_2 odnosi się do wymaganego maksymalnego czasu transmisji pakietu i jest spełniony, jeżeli czas transmisji Δ'_k pakietu dla kontraktu *k* jest nie większy niż czas Δ_k zadany w wymaganiach QoS kontraktu. Należy zauważyć, że warunki mk_0, mk_1 mogą być weryfikowane w oparciu o znajomość *RT* (znajomość *SCH* nie jest konieczna w momencie ich weryfikacji), natomiast warunek mk_2 wymaga znajomości obu elementów zbioru konfiguracji sieci conf = {RT, *SCH*}. W ramach warunku mk_2 można wyodrębnić warunek konieczny, ale nie wystarczający, który do weryfikacji wymaga jedynie znajomości RT. Dla każdej trasy $r_k^{\{s,d\}}$ istnieje minimalna liczba szczelin czasowych length(h, f) w których kontrakt można zaplanować. Minimalna długość odpowiada sytuacji, w której dla każdej transmisji nie występuje konieczność jej przeniesienia do kolejnej szczeliny ze względu na konflikt 1'ego lub 2'ego stopnia (Rysunek A.2 b). Tak zdefiniowana długość jest funkcją dwóch parametrów, gdzie *h* jest liczbą przeskoków na trasie a *f* jest ilością ramek.

$$length(h,f) = \begin{cases} h \cdot f & h \le 2\\ h + 2 \cdot (f-1) & h > 2 \end{cases}$$
(20)

Kontrakt, dla którego trasa ma większą liczbę przeskoków od tak określonej nie spełnia warunku mk_2 , w przeciwnym wypadku weryfikacja wymaga znajomości harmonogramu SCH. W celu oszacowania złożoności, tak zdefiniowany problem możemy zdekomponować i przedstawić jako dwa problemy znane z literatury. Pierwszym problemem jest znalezienie tras dla każdego z kontraktów pod warunkiem spełnienia wymagań QoS, co zostało zapisane w postaci równań (15) oraz (16). Problem ten jest znany w literaturze jako problem znalezienia trasy pod wieloma warunkami (ang. multi-constrained path problem, MCP). Drugim problemem jest znalezienie minimalnego planu SCH na podstawie utworzonych tras dla poszczególnych kontraktów. MCP, który dotyczy znalezienia RT, został sklasyfikowany jako problem NP-zupełny a dowód przedstawiony w [60]. Istnieją trzy rodzaje metryk QoS wykorzystywane w praktyce: addytywne, multiplikatywne i wklęsłe (ang. concave), z czego warunek mk_1 jest multiplikatywny a warunki mk_2 oraz k_2 (dotyczący całej sieci) addytywne. Trzy twierdzenia przedstawione w przytoczonej publikacji dowodzą, że znalezienie ścieżki pod warunkiem dwóch lub więcej warunków addytywnych lub multiplikatywnych w każdej dowolnej konfiguracji jest problemem NP-zupełnym i należy zakładać, że znalezienie rozwiązania nie jest możliwe w czasie wielomianowym. Podobnie problem znalezienia minimalnego planu SCH został opisany w [61] i sklasyfikowany jako NP-zupełny a przedstawiony dowód zakłada redukcję problemu do znalezienia liczby chromatycznej grafu. W praktyce prezentowany problem jest bardziej złożony niż jego dekompozycja na dwa osobne problemy ze względu na to, że nie można ocenić jakości znalezionych tras bez wykonania planowania i odwrotnie.

Całościowe rozwiązanie zdefiniowanego w tym rozdziale problemu wymaga opracowania efektywnych algorytmów pozwalających zarówno na przygotowanie pełnej konfiguracji sieci

jak również jej optymalizację przy spełnieniu postawionych wymagań. W celu analizy istniejących rozwiązań oraz opracowania i ewaluacji algorytmów własnych w rozdziale 4 prezentuję opracowane autorskie środowisko testowe pozwalające na prowadzenie badań:

- w środowisku rzeczywistym z wykorzystaniem GATEWAY_4B, które jest urządzeniem dedykowanym do pełnienia roli SM w sieci IWN i opracowanym w ramach realizacji projektu badawczego RPMA.01.02.00-14-9551/17-00 którego jestem kierownikiem.
- w środowisku laboratoryjnym na stacji roboczej

4 Koncepcja badań eksperymentalnych

W rozdziale prezentuję autorskie środowisko testowania, dedykowane do eksperymentalnego zbadania wpływu różnych parametrów na proces tworzenie konfiguracji sieci. Proces badania został podzielony na dwie fazy obejmujące: tworzenie konfiguracji sieci na podstawie opracowanych wektorów testowych, symulację pracy sieci w oparciu o utworzone konfiguracje. Środowisko testowe pozwala na ewaluację zarówno samych algorytmów planowania (rozdział 5) (pozostała część konfiguracji stanowi element wektora testowego) jak również kompletnego algorytmu optymalizacyjnego (rozdział 6) dostarczającego całą konfigurację sieci. Środowisko testowe wykorzystuje zarówno rzeczywiste elementy sieci IWN takie jak GATEWAY_4B opracowany w ramach projektu B+R do pełnienia funkcji SM, jak również środowiska wielostrumieniowe pozwalające na ewaluację wielu elementów wektora jednocześnie. W rozdziale 4.1 prezentuję syntetyczny opis opracowanego środowiska, natomiast w rozdziale 4.2 przedstawiam założenia dotyczące przygotowanych wektorów testowych dla dwóch wariantów generowania topologii sieci.

4.1 Środowisko testowania

Algorytmy planowania oraz algorytmy optymalizacyjne GA-CAS zostały uruchomione w środowisku rzeczywistym z wykorzystaniem GATEWAY 4B oraz dla zestawienia na stacji roboczej Dell Precision T3600 (E5-1650 @ 3.20GHz). Każdy z algorytmów uruchomiono w oparciu o wygenerowane zestawy parametrów wejściowych stanowiące element opracowanego wektora testowego (4.2). Zbiór zestawów parametrów wejściowych utworzono jako iloczyn kartezjański zbiorów wartości każdego z parametrów prezentowanych w tabeli 4.1. Zbiór zestawów parametrów wejściowych ma 480 i 144 elementy odpowiednio dla wariantu 1 (rozdział 4.2.1) i wariantu 2 (rozdział 4.2.1.2) generowania topologii sieci. Dla każdego zestawu parametrów wejściowych wygenerowano 10 losowych zbiorów definiujących sieć węzłów. Zbiór "graph" zawiera krawędzie grafu stanowiące potencjalne połączenie pomiędzy węzłami (nadajnik, odbiornik), wygenerowane zgodnie z opisem 4.2.1 oraz 4.2.1.2. Liczba elementów w zbiorze "graph": wariant 1 - 3122400; wariant 2 - 25032000. Zbiory "contract periodic" oraz "contract aperiodic" zawierają parametry każdego z kontraktów zestawione w tabeli 4.2 oraz wygenerowane zgodnie z opisem 4.2.2. Liczba elementów w tych zbiorach to odpowiednio: wariant 1 - [124080, 124080]; wariant 2 - [88200, 88200]. Dodatkowo zbiór "contract" zawiera informacje wspólne dla obu typów kontraktów, takie jak np. numer węzła źródłowego i docelowego. Liczebność zbioru stanowi sumę liczebności obu zbiorów dedykowanych dla konkretnego typu kontraktu. Zbiór "contract path" zawiera trasy routingu dla każdego z kontraktów wygenerowane zgodnie z opisem 4.2.2, liczba elementów: wariant 1 - 1810323; wariant 2 - 1455548. Trasy 1 oraz 2 ze zbioru "contract path" wykorzystywane są podczas ewaluacji algorytmów planowania. Natomiast w przypadku ewaluacji algorytmu optymalizacyjnego GA-CAS w zbiorze tym zapisywane są rezultaty optymalizacyjne odpowiadające zawartości zbioru *RT*. Algorytmy zaimplementowano w środowisku rzeczywistym w języku C++, również środowisko testowe odpowiedzialne z uruchamianie poszczególnych testów i gromadzenie rezultatów ich wykonania zaimplementowano w języku C++. W przypadku GATEWAY_4B użyto kompilatora i linkera z zestawu "*GNU Toolchain for the A-profile Architecture 8.3*" oraz "*PROCESSOR-SDK-LINUX-AM335X_06.03.00.106*", natomiast w przypadku uruchomienia na stacji roboczej użyto LLVM 6.0. Dane stanowiące zbiór zestawów parametrów wejściowych oraz zbiory definiujące stan sieci wygenerowano z wykorzystaniem języka Python 3.8.5 (środowisko Anaconda 4.9.2). W tym samym środowisku wykonano analizę i prezentację danych wyjściowych. Dane wejściowe i wyjściowe zapisywano w bazie danych pod kontrolą SQLite.

W wyniku przebiegu pojedynczego algorytmu planowania uzyskano następujące dane wyjściowe:

- Czas wykonania planu dla badanego algorytmu [s].
- Liczbę zaplanowanych kontraktów cyklicznych.
- Liczbę wszystkich kontraktów cyklicznych.
- Liczbę zaplanowanych kontraktów sekwencyjnych.
- Liczbę wszystkich kontraktów sekwencyjnych.
- Liczba wszystkich elementów wykonanego planu.
- Zbiór planów dla wszystkich węzłów sieci, $SCH = \{SCH_i \mid v_i \in V\}$.

Wartości liczbowe przywołane powyżej zapisywane są w zbiorze "result", natomiast plan zapisany jest w zbiorach "superframe", "schedule" oraz "*contract_graph*", bardziej szczegółowo opisanych w dalszej części.

W wyniku przebiegu pojedynczego algorytmu GA-CAS uzyskano dane wyjściowe jak wymienione powyżej (dotyczące algorytmów samego planowania) oraz dodatkowo:

- Liczbę wykorzystanych pokoleń.
- Liczbę wykorzystanych pokoleń "in-vitro".
- Czas wykonywania optymalizacji dla badanego algorytmu [s].
- Czas operacji "rozmnażanie" stanowiącej składową czasu wykonania optymalizacji [s].
- Czas operacji "wzrost" stanowiącej składową czasu wykonania optymalizacji [s].
- Czas operacji "inicjalizacja" stanowiącej składową czasu wykonania optymalizacji [s].
- Czas innych operacji w ramach wykonania optymalizacji i nie ujętych w operacjach "rozmnażanie", "wzrost" oraz "inicjalizacja" [s].
- Liczbę kontraktów cyklicznych posiadających "możliwą" trasę w zbiorze RT.

- Liczbę kontraktów sekwencyjnych posiadających "możliwą" trasę w zbiorze RT.
- Zbiór tras routingu dla wszystkich kontraktów wszystkich węzłów sieci, $RT = \{r_k^{\{s,d\}} \mid c_k^{\{s,d\}} \in C\}.$

Wartości liczbowe opisujące rezultaty wykonania algorytmów optymalizacyjnych GA-CAS zostały zapisywane w zbiorze "*result_ga*". Plan *SCH* zapisany jest w zbiorach "superframe", "schedule" oraz "*contract graph*", natomiast zbiór tras *RT* w zbiorze "*contract path*".

Zbiory "superframe", "schedule", "contract graph" stanowią rezultat wykonania etapu planowania, który jest obecny zarówno w algorytmach planowania jak i w algorytmach optymalizacyjnych GA-CAS. Wygenerowany plan (harmonogram transmisji) stanowiący element konfiguracji konkretnej sieci zawiera plany dla wszystkich tworzących ją węzłów $SCH = \{SCH_{\{i\}} \mid v_i \in V\}, \text{ gdzie } i \text{ indeksem węzła } v_i. \text{ Pojedynczy element planu } SCH_{\{i\}} =$ $\{el \mid el = (p, c_k^{\{s,d\}}, l_i^i, ts, ch), v_i \in V, c_k^{\{s,d\}} \in C\}$ definiowany jest jako n-ka obejmująca: okres powtarzania, kontrakt (związek z kontraktem), połączenie (nadawca-odbiorca), numer szczeliny czasowej oraz numer kanału. Przenosząc notację matematyczną na zapis zbiórzwiązek wyodrębniłem zbiór "schedule" zawierający wszystkie elementy planów el' = $(c_k^{\{s,d\}}, l_j^i, ts, ch)$; oraz zbiór "*superframe*" zawierający okres powtarzania p związanych z nim elementów planu. Zbiór "superframe" grupuje elementy planu posiadające wspólny okres powtarzania. Zgodnie z koncepcją sieci IWN, ruch w ramach szczeliny czasowej może być dedykowany konkretnemu kontraktowi (ang. contention-free traffic) lub kilka kontraktów może współdzielić ta sama szczeline czasowa (ang. contention-based traffic). Bazując na tym założeniu dla każdego kontraktu w procesie planowaniu przypisywany jest numer "graphId". Numer ten w zależności od implementacji algorytmu jest unikatowy dla kontraktu lub dla kilku kontraktów uzyskuje ten sam numer "graphId". Wszystkie plany utworzone dla ruchu cyklicznego zakładają pracę w trybie dedykowanym (ang. contention-free traffic), tym samym zbiór "contract graph" posiada ten sam numer "graphId" dla każdego zaplanowanego kontraktu cyklicznego. W przypadku ruchu sekwencyjnego generowanie numerów "graphId" uzależnione jest od konfiguracji lub zachowania konkretnego algorytmu. W przypadku opracowanych algorytmów definiowane jest poprzez parametr CONTENTION przyjmujący jedną z wartości: DISABLED, ENABLED.

Liczebność zbioru "schedule" jest prezentowana jako jeden z wyników ewaluacji konkretnych algorytmów, np. Tabela B.9, Tabela B.13. Najdłuższy okres powtarzania spośród wszystkich elementów *"superframe"* prezentowany jest jako jeden z wyników ewaluacji algorytmów, np. Tabela B.8, Tabela B.12.

Do ewaluacji rezultatów przygotowanej konfiguracji sieci w tym w szczególności wygenerowanych planów *SCH* oraz zbioru tras *RT*, został opracowany autorski symulator sieci. Podstawowe zadania jakie zostały powierzone opracowanemu symulatorowi to:

- Weryfikacja poprawności wykonanych planów.
- Weryfikacja poprawności wykonanych tras
- Zgromadzenie statystyk dotyczących czasu dostarczania ramek i pakietów w sieci
- Zgromadzenie statystyk dotyczących rozkładu ramek w buforach węzłów sieci

Symulator pozwala na pracę w oparciu o definicję sieci dla obu wariantów generowania topologii. Symulator został zaimplementowany w języku C++. Do budowy symulatora wykorzystano ten sam zestaw narzędzi programistycznych jak w przypadku implementacji algorytmów planowania i optymalizacji. Dane wejściowe dla algorytmów wraz z wygenerowanymi planami oraz trasami dla każdego z analizowanych algorytmów stanowią dane wejściowe dla symulatora. Wszystkie dane analogicznie jak w fazie ewaluacji algorytmów zapisywane są w bazie SQLite, dzięki takiemu rozwiązaniu możliwe jest przygotowanie konfiguracji sieci conf = {RT, SCH} oraz symulacji na innych urządzenie do roli SM) a symulacji na stacji roboczej. Opracowany symulator dedykowane urządzenie do roli SM) a symulacji na stacji pracy sieci w rzeczywistym środowisku, która może być wykonana przed właściwym wdrożeniem sieci w celu weryfikacji działania w konkretnych praktycznych warunkach. Przedstawiona w artykule metoda symulacji sieci ISA100.11a wykorzystuje symulator NS3 i certyfikowany komercyjny stos komunikacyjny.

W wyniku wykonania symulacji działania sieci dla pojedynczego algorytmu w oparciu o wygenerowaną konfigurację sieci generowane są następujące zbiory danych wyjściowych:

- Zbiór "result_contract" zawierający dla każdego zaplanowanego kontraktu następujące informacje: liczbę uwolnionych/dostarczonych ramek oraz pakietów, średni/maksymalny czas dostarczania ramek praz pakietów, odchylenie standardowe rozkładu czasu dostarczania ramek oraz pakietów.
- Zbiór "result_node" zawierający dla każdego węzła sieci informację o rozkładzie czasu zajętości bufora ramek. Informacja gromadzona jest w postaci histogramu czasu pracy węzła z podziałem na 65 koszyków reprezentujących liczbę ramek w buforze, Wartość 64 (ostatni koszyk) jest zdefiniowany jako zbiorczy i zliczane w nim są czasy, gdy w buforze jest ≥ 64 ramki.
- Zbiór "result_node_agg" stanowi zagregowaną wersję informacji dotyczących rozkładu czasu zajętości buforów bez podziału na węzły.

• Czas trwania symulacji – wartość utrwalana w zbiorze "results"

Podstawowe pojęcia takie jak: ramka, pakiet, uwolnienie ramki/pakietu oraz dostarczenie ramki/pakietu zostały omówione w 3.1.1. Kompletny model koncepcyjny opisujący związki pomiędzy zbiorami, zapisany w notacji UML, umieszczono w Dodatku, rysunek A.1. Zbiory pogrupowano kolorami:

- kolorem niebieskim oznaczono zbiory stanowiące dane wejściowe w procesie uruchamiania algorytmów i symulacji;
- kolor żółty to zbiory z rezultatami planowania;
- kolor zielony to dane opisujące uruchomienie algorytmów planowania
- kolor różowy to dane opisujące uruchomienie algorytmów optymalizacyjnych
- kolor turkusowy to zbiory stanowiące rezultat wykonania symulacji.

Praca symulatora polega na generowaniu ruchu (uwalnianiu pakietów) zgodnie z definicją kontraktów (4.2.3) oraz ich dystrybucji w oparciu o przygotowany plan i zgodnie z konfiguracją sieci. Symulator pracuje przy kroku równym długości pojedynczej szczeliny czasowej. Symulator wykorzystuje te same bloki funkcjonalne które stanowią element stosu komunikacyjnego rzeczywistego węzła analizowanej sieci IWN. Można wyodrębnić dwie fazy jego pracy. Faza 1 realizowana jest przez zadaną liczbę szczelin i polega na wykonywaniu pełnego cyklu zawierającego uwalnianie ruchu oraz jego dystrybucję. Faza 2 następuje bezpośrednio po skończeniu Fazy 1 i polega jedynie na dystrybucji ruchu wygenerowanego w Fazie 1 który nie został jeszcze dostarczony. Faza 2 nie zawiera elementu związanego z uwalnianiem nowego ruchu. Faza 2 kończy się w momencie dostarczenia całego ruchu lub w momencie, gdy dostarczenie ruchu nie jest możliwe, np. z powodu błędnie wygenerowanego planu. Symulator gromadzi dane statystyczne zarówno w odniesieniu do kontraktów jak również węzłów sieci. Dane dotyczące kontraktów utrwalane są w zbiorze "result contract" natomiast dotyczące węzłów w zbiorze "result node". W zbiorze "result node", dla każdego węzła sieci, symulator zlicza liczbę szczelin czasowych w których w buforze węzła znajduje się konkretna liczba ramek. Liczba ramek reprezentowana jest poprzez 65 koszyków. Koszyki ponumerowane są od 0 do 64, tak że w koszyku 1 zliczana jest liczba szczelin w których w buforze znajduje się jedna ramka, w koszyku 2 – 2 ramki, itd. W koszyku 64 zliczane są szczeliny w których w buforze konkretnego węzła jest ≥ 64 ramki. Wartość dla koszyka 0 obliczana jest dla każdego węzła po zakończeniu symulacji na podstawie całkowitej liczby szczelin czasowych, stanowiącej czas trwania symulacji oraz wartości zapisanych w pozostałych 64 koszykach. Zbiór "result node agg" tworzony jest poprzez obliczenie sumy liczby szczelin zgromadzonych w zbiorze "result node" przy pominięciu podziału na węzły.

Opracowane środowisko testowe pozwala na jednoczesne wykonywanie badanych algorytmów dla poszczególnych elementów wektora testowego z wykorzystaniem wielu równoległych strumieni przetwarzania. Środowisko jest przygotowane z myślą o pracy równoległej i istnieje zastosowaniu możliwość uruchomienia badań przy specjalistycznych środowisk wieloprocesorowych (w badaniach wykorzystano 12 równoległych strumieni przetwarzania z wykorzystaniem stacji roboczej). Zastosowanie maszyn o większej skalowalności pozwoli na skrócenie badania pojedynczego wektora testowego lub umożliwi rozszerzenie wektora o kolejne wartości parametrów lub zwiększenie liczby losowo generowanych konfiguracji sieci (ma znaczenie przy większej liczbie węzłów). W ramach prowadzonych badań wykorzystano 10 losowo generowanych konfiguracji sieci dla każdego elementu wektora testowego.

4.2 Scenariusze testowania

Parametry użyte do definicji wektorów testowych wraz z podaniem zbioru ich wartości zestawiłem w tabeli 4.1. Parametry N_V oraz D_G są użyte jedynie w wariancie 1 generowania topologii sieci podczas gdy N_{CUBIC} oraz [X, Y, Z] jedynie w wariancie 2.

N_V	Liczba wszystkich węzłów w sieci; {10,30,,90}		
$ S /N_V$	Stosunek liczby węzłów będących źródłem transmisji do liczby wszystkich węzłów w sieci; {0.6, 0.8}		
$ D /N_V$	Stosunek liczby węzłów docelowych do liczby wszystkich węzłów w sieci; {0.2, 0.4}		
D _G	Współczynnik gęstości grafu, $D_G = \frac{ E }{N_V \times (N_V - 1)}; \{0.15, 0.25\}$		
N _{CUBIC}	Średnia liczba węzłów sieci przypadająca na m ² ; {0.01, 0.02, 0.04}		
[X, Y, Z]	Rozmiar poszczególnych wymiarów przestrzeni euklidesowej w jakiej węzły są rozmieszczone; [50, 25, 4]		
N _{CH}	Liczba kanałów w sieci; {2, 8, 16}		
$N_C/ S $	Średnia liczba kontraktów przypadająca na jeden węzeł źródłowy, ≥ 1 ; {1, 2}		
$N_{CP}/N_C;$ N_{CA}/N_C	Stosunek liczby kontraktów dla ruchu odpowiednio cyklicznego i sekwencyjnego do liczby wszystkich kontraktów; $N_{CP}/N_C \in \{0.3, 0.7\}$		

Tabela 4.1 Notacja oraz wartości parametrów użytych w wektorach testowych wykorzystywanych do analizy algorytmów planowania.

Wariant 1 generowania topologii sieci jest często stosowany w literaturze ze względu na łatwość generowania różnych wariantów grafów połączeń. Prezentowane w dalszej części wyniki badań pokazują, że otrzymywane rezultaty dla wariantu 1 są mniej rzeczywiste. Wariant 2 daje rezultaty bardziej skorelowane z rezultatami otrzymywanymi w praktycznych scenariuszach pracy urządzeń IWN.

4.2.1 Generowanie losowej topologii sieci

4.2.1.1 Wariant 1 – generowanie grafu

Zakładając daną liczbę wszystkich węzłów sieci N_V oraz gęstość grafu D_G generowana jest losowa sieć, gdzie całkowita liczba krawędzi wynosi:

$$|E| = N_V \times (N_V - 1) \times D_G \tag{21}$$

Krawędzie skierowane generowane są pomiędzy losowo wybranymi węzłami nadawcą i odbiorcą. Dla każdej krawędzi losowo w rozkładzie jednostajnym generowany jest współczynnik RSSI (ang. Received Signal Strength Indicator) w przedziale < -85 dBm; -5 dBm >. Wykorzystując wyniki prezentowane w [54] i omówione w 3.1.2.1 oszacowałem PER dla każdego RSSI z użyciem modelu Rician przy współczynniku K=2.

4.2.1.2 Wariant 2 – przestrzenne rozmieszczenie węzłów

W tym wariancie węzły są losowo rozmieszczone na przestrzeni euklidesowej o wymiarach [X, Y, Z]. Zakładając daną średnią liczbę węzłów sieci N_{CUBIC} dla każdego węzła generowane jest losowe położenie w rozkładzie jednostajnym w każdym z wymiarów X, Y, Z. Dla każdej 2elementowej kombinacji ze zbioru wszystkich węzłów, przy pominięciu kombinacji dwóch tych samych węzłów, obliczana jest odległość euklidesowa oraz poziom tłumienia trasy łączącej nadajnik (pierwszy element kombinacji) z odbiornikiem (drugi element kombinacji) PL(d) na podstawie wzoru (11). Do obliczenia PL(d) wykorzystałem dane zebrane w tabeli B.6 i przyjąłem odpowiednio: $d_0 = 1m$; $PL(d_0) = 46,20 \, dBm$; $\alpha = 1,7$. Zakładając moc nadajnika taka samą we wszystich węzłach sieci i rowną 5 dBm, dla każdej kombinacji obliczyłem współczynnik RSSI przy danym PL(d). Dla wszystkich kombinacji gdzie RSSI > -80dBm utworzyłem krawędź skierowaną grafu pomiędzy nadawcą i odbiorcą. Dla każdej tworzonej krawędzi oszacowałem PER na podstawie RSSI wykorzystując wyniki prezentowane w [54] dla modelu Rician ze wspólczynnikiem K=2, rysunek B.2. Rozmieszczenie węzłów sieci wraz z zaznaczeniem dwóch tras dla wybranego kontraktu zaprezentowano na rysunku 4.1.

4.2.2 Generowanie kontraktów

Sieć generowana jest losowo do czasu aż możliwe jest utworzenie wymaganej liczby kontraktów zdefiniowanej jako średnia liczba kontraktów przypadająca na jeden węzeł źródłowy $N_C/|S|$. Generowany kontrakt uznawany jest za poprawny, jeżeli istnieje co najmniej jedna ścieżka łącząca węzeł źródłowy i docelowy. Liczba węzłów źródłowych i docelowych w sieci określona jest względem całkowitej liczby węzłów w sieci, jako: $|S|/N_V$ oraz $|D|/N_V$.



Rysunek 4.1 Przykładowe rozmieszczenie węzłów sieci wraz z trasą wybranego kontraktu dla $N_V = 100$ (lewy) oraz $N_V = 400$ (prawy)

Ko	ntrakt	Wantość	
Cykliczny	Aperiodyczny	wartosc	
Paylo	oad_Size	<50; 1200> B	
Rel	iability	Zależnie od wariantu generowania topologii	
Expiration		1 dzień do 5 lat	
Period		<0,1; 100> s (bez 0)	
Phase		<0; 99> %	
Deadline		Liczba przeskoków dla Trasa 1 * uniform<1,5;2>	
	Committed_Burst	<0.01; 10> APDU/s (bez 0)	
	Excess_Burst	Committed_Burst * 1,1	

Tabela 4.2 Wartości parametrów opisujących tworzone w wektorach testowych kontrakty

Wartości parametrów opisujących tworzone kontrakty generowane są losowo w rozkładzie jednostajnym zgodnie z zestawionym w tabeli 4.2 zakresem wartości. Nazwy pół kontraktu odpowiadają definicji w tabeli 3.2. Dla każdego kontraktu tworzone są dwa wpisy w tablicy routingu odpowiadające pojedynczej trasie pomiędzy węzłem źródłowym a docelowym. Trasa 1 odpowiada trasie z najmniejszą liczbą przeskoków, generowana jest z wykorzystaniem algorytmu Dijkstry, gdzie przyjęto dla każdej krawędzi grafu wagę 1. Trasa 2 odpowiada trasie z najwyższym całkowitym prawdopodobieństwem dostarczenia pakietu p_{suc} , wzór (9). W tym wypadku użyto również algorytmu Dijkstry z zastrzeżeniem, że mnożenie prawdopodobieństw dla poszczególnych transmisji przekształcono w dodawanie ich logarytmów. Zastosowana w algorytmie waga jest równa $p_log_{suc_frame,f}^{\{i,j\}} = -1 \times \ln\left(p_{suc_frame,f}^{\{i,j\}}\right)$.

4.2.3 Generowanie ruchu w sieci

Ruch w sieci został podzielony na dwie klasy bazując na typie kontraktu, w ramach którego jest generowany. Ruch cykliczny jest modelowany jako proces deterministyczny, gdzie czas pomiędzy dwoma sąsiednimi pakietami jest stały i określony w kontrakcie jako $cp_k^{\{s,d\}}$. *period*. Czas pierwszego uwolnienia pakietu dla kontraktu k uwzględniający fazę wyrażoną w procentach definiuję jako:

$$ts_release_0^k = \left\lfloor cp_k^{\{s,d\}}.phase \times cp_k^{\{s,d\}}.period/|ts|/100\right\rfloor$$
(22)

Kolejne uwolnienia obliczam przesuwając numer szczeliny uwolnienia o liczbę szczelin czasowych przypadających na okres kontraktu.

$$ts_release_{n+1}^{k} = ts_release_{n}^{k} + cp_{k}^{\{s,d\}}.period/|ts|$$
(23)

Ruch sekwencyjny jest modelowany w oparciu o przesunięty rozkład Bernoulliego na podstawie parametrów $ca_k^{\{s,d\}}$. *committed_burst* oraz $ca_k^{\{s,d\}}$. *excess_burst*. Rozkład Bernoulliego jest analogicznym rozkładem do rozkładu Poissona w domenie rozkładów dyskretnych. W procesie Bernoulliego prawdopodobieństwo uwolnienia pakietu dla dowolnej szczeliny czasowej wynosi *pr*. Czas uwolnienia jest opisany rozkładem geometrycznym z funkcją rozkładu prawdopodobieństwa (PMF):

$$P(X = ts) = pr \cdot (1 - pr)^{ts - 1}$$
(24)

gdzie ts = 1,2,3, ... jest numerem szczeliny uwolnienia.

Klasyczny rozkład Bernoulliego pozwala opisać ruch posługując się jedynie średnią przepływnością (liczbą pakietów/s). W modelu zakładam, że dodatkowo istnieje parametr $ca_k^{\{s,d\}}$. excess_burst określający minimalny czas pomiędzy uwolnieniami (β):

$$\beta = \frac{1}{ca_k^{\{s,d\}} \cdot excess_burst \cdot |ts|}$$
(25)

Prawdopodobieństwo uwolnienia w $ts < \beta$ pierwszych szczelinach czasowych wynosi 0, w $ts \ge \beta$ kolejnych szczelinach wynosi *pr*. Tak przesunięty rozkład Bernoulliego opisuje następująca funkcja rozkładu prawdopodobieństwa względem numeru szczeliny:

$$P(X = ts) = \begin{cases} 0 & ts < \beta \\ pr \cdot (1 - pr)^{ts - \beta} & ts \ge \beta \end{cases}$$
(26)

Średnia liczba szczelin po których następuje uwolnienie pakietu jest zatem równa

$$\bar{ts} = \beta - 1 + \frac{1}{pr} \tag{27}$$

Z powyższego możemy znaleźć parametr pr rozkładu jako:

$$pr = \frac{1}{\overline{ts} - \beta + 1} \tag{28}$$

gdzie $\overline{ts} = 1/(ca_k^{\{s,d\}}.committed_burst \times |ts|)$

5 Algorytmy planowania rozmieszczenia szczelin czasowych w TDMA

Zdefiniowany w rozdziale 3.2 problem znalezienia optymalnej konfiguracji sieci IWN wymaga zastosowania metod pozwalających na przygotowanie planu szczelin czasowych dla sieci IWN z wielodostępem TDMA/FDMA. W rozdziale 5.1 przedstawiam specyfikę algorytmów planowania na podstawie znanych z literatury typowych rozwiązań. Wykorzystuję algorytm AMUS (ruch sekwencyjny) oraz algorytmy oparte na miarach EDF/LLF (ruch cykliczny) do zbadania efektywności oraz wpływu szeregu parametrów na działanie algorytmów. W rozdziale 5.2 przedstawiam wyniki badań eksperymentalnych tych algorytmów. Analizuję możliwość ich zastosowania do rozwiązania postawionego problemu optymalizacyjnego przy przyjętych założeniach dotyczących ograniczonych zasobów elementów sieci IWN. Przedstawiam i poddaję ewaluacji modyfikacje badanych algorytmów. Otrzymane wyniki badań pozwalają mi na sformułowanie zestawu istotnych ograniczeń w ich zastosowaniu. W rozdziałach 5.3 oraz 5.4 prezentuje autorskie algorytmy wykorzystujące opracowany mechanizm unikania kolizji CAS. Algorytm CAS-PERIOD dedykowany jest do ruchu cyklicznego natomiast CAS-SEQ ruchu sekwencyjnego. Opracowane algorytmy poddaję szczegółowej analizie w oparciu o opracowane środowisko testowe (rozdział 4) a wyniki zestawiam z najlepszymi rezultatami uzyskanymi dla przebadanych wariantów algorytmów dostępnych w literaturze.

5.1 Specyfika algorytmów

Dostępne w literaturze rozwiązania można podzielić na dwie grupy: planowanie centralne i rozproszone [63, 64]. Planowanie centralne zakłada istnienie w sieci specjalnej roli managera sieci, którego zadaniem jest gromadzenie informacji o topologii sieci oraz wymaganiach dotyczących ruchu i na ich podstawie wykonywanie planów dla całej sieci oraz ich dystrybucja do każdego z węzłów. Drugą grupę algorytmów planowania stanowią algorytmy rozproszone (autonomiczne), w których decyzje dotyczące planu podejmują wezły lokalnie [65-67]. Ta grupa algorytmów ma mniejszą niezawodność ze względu na ograniczoną wiedzę o otoczeniu, sprowadzającą się jedynie do informacji o najbliższych sąsiadach. Algorytmy rozproszone pozwalają na efektywne budowanie planu szczelin przy założeniu, że ruch jest pionowy, np. w oparciu o topologie drzewa ukorzenionego, gdzie szczeliny zapełniane sa kolejno od korzenia do węzłów znajdujących się najdalej [20]. Dodatkowo w zastosowaniach IWN rygorystyczne wymagania dotyczące niezawodności i oczekiwanego determinizmu czasowego często prowadzą do stosunkowo stabilnej topologii sieci i stałych okresowych przepływów danych, co jest bardziej odpowiednie do przyjęcia scentralizowanego mechanizmu zarządzania w tym planowania [68]. Szczególnym przykładem planowania scentralizowanego są rozwiązania scentralizowane ale realizowane w oparciu o technikę "edge computing" [5, 6], gdzie zadania realizowane centralnie, przenoszone są bliżej węzłów lokalnych w celu ograniczenia narzutu wynikającego z przesyłania informacji do węzła centralnego i efektywnego skrócenia czasu na dostarczenie wymaganej konfiguracji do węzła. W naturalny sposób ruch przemysłowy w praktycznych scenariuszach wykazuje tendencję do tworzenia klastrów w ramach pojedynczych urządzeń, zespołów urządzeń, linii produkcyjnych, hal, kompleksów, itd., każdy taki lokalny klaster może być optymalizowany lokalnie i koordynowany ma poziomie całej sieci. Tworzenia planu szczelin czasowych realizowane centralnie, zarówno daje lepsze rezultaty w IWN niż planowanie rozproszone, jak również stanowi punkt wyjście do stosowania technik typu "edge computing", gdzie zadania mogą być realizowane w oparciu o mniejsze klastry węzłów sieci [69–71]. Biorąc powyższe pod uwagę, w opracowaniu koncentruję się na rozwiązaniach mogących znaleźć zastosowania w sieciach IWN z centralną funkcją managera sieci (SM), takich jak ISA100.11a lub WirelessHART.

Ze względu na to, że w definicji algorytmów w literaturze panuje dość duża dowolność i analiza algorytmów definiowanych przez różnych autorów w postaci pseudokodu jest bardzo czasochłonna, wraz z opisem każdego algorytmu prezentuję pseudokod w sposób ujednolicony, tak aby zarówno parametry wejściowe i wyjściowe były takie same oraz stosuję jednolitą notację i styl przedstawiania problemu. Takie podejście pozwoli na łatwe porównanie różnych algorytmów i ich implementację. Dostępne w literaturze algorytmy posługują się zbliżonym wzorcem działania [72]. Budują plan dla każdej kolejnej szczeliny czasowej do czasu aż cały generowany ruch jest zaplanowany.

5.1.1 Algorytm planowania szczelin AMUS

W artykule [73] zaprezentowano algorytm planowania szczelin AMUS (Adaptive Multi-hop Scheduling Method) który zapewnia małe opóźnienia i jest dedykowany dla aplikacji czasu rzeczywistego. Algorytm jest stworzony na bazie algorytmu TESA [74], który jest dedykowany jedynie dla sieci w których ruch modelowany jest za pomocą drzewa i skierowany jest w stronę pojedynczego węzła dedykowanego do akwizycji danych (ang. sink node). Oba algorytmy wykorzystują wielkość ruchu generowanego w węźle jako kryterium pierwszeństwa w procesie planowania. Algorytm AMUS wykorzystuje dwie macierze SS oraz MSS. Macierz "Scheduling Sequence" (SS) o rozmiarze 2/n, gdzie n jest liczbą węzłów stanowiących źródło ruchu, zawiera numer węzła w pierwszym wierszu i liczbę danych generowanych przez ten węzeł w jednostce czasu. Dane zawarte w macierzy SS można u nas odnaleźć w zbiorze wszystkich kontraktów z tą różnicą, że węzeł źródłowy może mieć 1 lub więcej kontraktów. Liczba danych wysyłana przez wezeł, odpowiada przepływności λ zdefiniowanej jako wymagania QoS kontraktu. Macierz "Muli-hop scheduling sequence" (MSS) tworzona jest z SS i tablicy routingu (RT). Dla każdej pozycji w SS odnajdywane są kolejne przeskoki w ścieżce routingu i w kolejnych pozycjach umieszczane są dane tego przeskoku takie jak: numer węzła źródłowego, docelowego i liczba ramek. AMUS proponuje również wykorzystanie nadmiarowych elementów planu (tentative cell), które w połączeniu z proponowanym protokołem pozwalają na dalszą redukcję opóźnienia transmisji. Ponieważ proponowana technika wymaga rozszerzenia protokołu i może być de facto użyta przez różne algorytmy dedykowane planowaniu, ten element AMUS nie będę uwzględniał w ewaluacji.

Kluczowym do efektywnego działania algorytmu jest dobór wartości parametru p, który jest długością okresu planowania. Okres planowania jest jednocześnie okresem powtarzania elementu planu $(p, c_k^{\{s,d\}}, l_j^i, ts, ch)$. Niestety, [73] nie określa w jaki sposób należy dokonać doboru parametru p. Większa wartość pozwala na objęcie planem większej ilości ruchu, natomiast pociąga za sobą zwiększenie maksymalnego czasu po jakim pakiet dotrze do węzła docelowego. Analizując algorytm, należy stwierdzić, że optymalna, z punktu widzenia "pojemności" planu, wartość p odpowiada okresowi, dla którego kontrakt o najniższej przepływności potrzebuje dokładnie jednej ramki na przesłanie danych. Z drugiej strony, optymalna dla minimalizacji opóźnienia, wartość p odpowiada okresowi potrzebnemu do wysłania ramek wygenerowanych dla jednego dowolnego kontraktu. W praktyce wartość powinna być dobrana z tego przedziału mając na uwadze wspomniany kompromis. Oryginalny algorytm zmodyfikowany nieznaczenie i dopasowany jedynie do definicji danych wejściowych i wyjściowych prezentuję jako Algorytm 1. Kontrakt $c_k^{\{s,d\}}$ uznawany jest za nie możliwy do zaplanowania w momencie, gdy brak jest możliwości na umieszczenie w planie o długości pruchu wygenerowanego w ramach niego.

Analiza algorytmu AMUS prowadzi do następujących wniosków:

- Nie ma rozróżnienia pomiędzy ruchem cyklicznym i sekwencyjnym
- Jedynie λ uwzględniono jako wymaganie QoS
- Nie określono jak należy ustalić okres planowania *p*. Na potrzeby implementacji przyjmuję, że *p* zostało określone arbitralnie lub w przypadku jego niezdefiniowania odpowiada okresowi, dla którego kontrakt o najniższej przepływności potrzebuje dokładnie jednej ramki na przesłanie danych.

Algorytm 1 AMUS (Adaptive Multi-hop Scheduling Method)

input:G; RT; C; poutput:SCH/* zbiór harmonogramów dla wszystkich węzłów sieci */1for each $(QP_k|QA_k) \leftarrow c_k^{\{s,d\}} \in C$ do compute przepływność λ 2Posortuj kontrakty $c_k^{\{s,d\}} \in C$ w porządku opadającym po przepływności λ 3 $ACT \leftarrow \{false\}$ /* inicjalizuj tablicę aktywności węzłów */4for each $c_k^{\{s,d\}} \in C$ do5 $|ts \leftarrow 0|$

6		for each $l_j^i \in r_k^{\{s,d\}}$ do				
7		for each $\tau_k^{\{i,j\}}$ within p do				
8				whil	e(ts < T)	
9				if	$\neg ACT[v_i][ts] \land \neg ACT[v_j][ts]$ then /* brak kolizji 1'ego st. */	
10					$VN \leftarrow NE_i \cup NE_j \ /^*$ zbiór sąsiadów dla v_i , $v_j \ */$	
11					for each $ch \in N^i_{CH}$ do	
12					If $\forall v_n \in VN(\neg ACT[v_n][ts][ch])$ then /* brak kolizji 2'ego st. */	
13					$SCH_i \leftarrow SCH_i \cup \left\{ \left(p, c_k^{\{s,d\}}, l_j^i, ts, ch \right) \right\}$	
14					$ACT[v_i][ts][ch] \leftarrow true, ACT[v_j][ts][ch] \leftarrow true$	
15					end	
16					end	
17				er	nd	
18				ts	$t \leftarrow ts + 1 / *$ następna szczelina */	
19				end		
20			er	nd		
21		e	nd			
22	en	d				

5.1.2 Algorytm planowania szczelin z metrykami LLF lub EDF

Metryki LLF (Least Laxity First) oraz EDF (Earliest Deadline First) znane są dobrze w literaturze i stanowią punkt odniesienia dla tworzenia nowych algorytmów [75, 76]. Każde porównanie dostępne w literaturze zakłada odniesienie do tych metryk. Metryki w pierwotnych wersjach prezentowanych w literaturze nie zakładały wykorzystania w sieciach FDMA, gdzie ruch może być dystrybuowany na wielu kanałach. Algorytm 2 wykorzystuje metrykę EDF lub LLF i jest dostosowany do modelowanej sieci TDMA/FDMA. Algorytm unika kolizji 1'ego stopnia poprzez pozostawienie ramki z potencjalną kolizją do kolejnej szczeliny czasowej oraz unika kolizji 2'ego stopnia poprzez śledzenie zajętości kanałów i wykorzystywania innego kanału w przypadku jego dostępności lub analogicznie jak w przypadku kolizji 1'ego stopnia pozostawieniu ramki do kolejnej szczeliny czasowej. Jedyną różnicą pomiędzy EDF i LLF jest metryka użyta do uporządkowania transmisji oczekujących w buforze. W przypadku EDF obliczana jest wartość $ts_deadline_j^k$ zgodnie ze wzorem (4) dla LLF obliczana jest wartość *laxity*^k_{ts} zgodnie ze wzorem (5).

Algorytm 2 Planowanie w oparciu o metryki LLF/EDF

input : G; RT; CP
output : SCH /* zbiór harmonogramów dla wszystkich węzłów sieci */

```
ts \leftarrow 0; ts_{abs} \leftarrow 0; BUF \leftarrow \{\}; ACT \leftarrow \{false\}
1
2
      p \leftarrow LCM(CP. period) \div |ts| /* oblicz okres planowania */
3
      do
           if ts_{abs} \leq T do releaseAtTs(ts, BUF, CP); /* umieść \tau_k^{\{i,j\}} \le BUF */
4
           for each \tau_k^{\{i,j\}} \in BUF do compute laxity-LLF / deadline-EDF
5
           Posortuj \tau_k^{\{i,j\}} \in BUF narastająco po laxity-LLF / deadline-EDF
6
           for each \tau_k^{\{i,j\}} \in BUF do /* dla każdej transmisji */
7
               if \neg ACT[v_i][ts] \land \neg ACT[v_i][ts] then /* brak kolizji 1'ego st. */
8
                   VN \leftarrow NE_i \cup NE_j /* zbiór sąsiadów dla v_i, v_j */
9
                   for each ch \in N_{CH}^i do
10
                       If \forall v_n \in VN(\neg ACT[v_n][ts][ch]) then /* brak kolizji 2'ego st. */
11
                           SCH_i = SCH_i \cup \left\{ \left( p, c_k^{\{s,d\}}, l_j^i, ts, ch \right) \right\}
12
                          ACT[v_i][ts][ch] \leftarrow true, ACT[v_j][ts][ch] \leftarrow true
Przesuń transmisję \tau_k^{\{i,j\}} ramki wzdłuż trasy r_k^{\{s,d\}}
13
14
15
                       end
16
                   end
17
               end
18
            end
           ts_{abs} \leftarrow ts_{abs} + 1; ts \leftarrow ts_{abs} \mod p /* następna szczelina czasowa */
19
20
      while (ts_{abs} \leq T \lor BUF \neq \emptyset)
```

5.1.3 Przegląd innych rozwiązań

Podstawowym standardem warstwy fizycznej dedykowanym do rozwiązań IWN jest IEEE802.15.4 [28]. Standard koncentruje się na definicji warstw fizycznych ale wprowadza również istotne rozwiązania dotyczące warstwy dostępu do medium (MAC). Protokół Time Synchronized Channel Hopping (TSCH) [77, 78], wprowadzony w nowelizacji standardu IEEE802.15.4e, jest dedykowany rozwiązaniom o zwiększonej niezawodności i niskim poborze energii. TSCH łączy dwie techniki wielodostępu do medium TDMA i FDMA oraz wprowadza predefiniowane szablony zmiany kanałów (ang. channel hopping, CH_HOP). Koncepcyjnie TSCH działa analogicznie do zdefiniowanych wcześniej standardów przemysłowych ISA100.11a oraz WirelessHART [9]. Technika CH_HOP została zaproponowana głównie z myślą o sieciach pracujących w układzie kratownicy (ang. mesh network) oraz ograniczeniu wpływu współistnienia z sieciami standardu WIFI oraz BLE [79, 80] wykorzystującymi ten sam podstawowy zakres częstotliwości przeznaczony do zastosowań przemysłowych, naukowych i medycznych (pasmo ISM) [81]. Podstawowe zastosowanie

techniki CH_HOP zakłada wykorzystywanie kolejnych kanałów według określonego wzorca, natomiast standardy takie jak np. ISA100.11a stosują rozszerzoną technikę zakładającą że jakość poszczególnych kanałów radiowych monitorowana jest w sposób ciągły i na podstawie raportów dostarczanych do SM kanały klasyfikowane są jako dobre i złe. Technika, w której łączone jest CH_HOP z dostosowywaniem listy kanałów wykorzystywanych do komunikacji między węzłami na podstawie aktualnych parametrów połączenia, funkcjonuje w literaturze pod nazwę "Adaptive Frequency Hopping". W przypadku standardu ISA100.11a, technika może być stosowana w sposób scentralizowany w ramach procesu planowania szczelin czasowych, natomiast np. w przypadku Bluetooth Low Energy (BLE) realizowana jest w sposób zdecentralizowany na poziomie wymiany informacji między dwoma węzłami. W sieciach IWN, adaptacyjny dobór kanałów w połączeniu z CH_HOP daje dużo lepsze rezultaty niż CH_HOP oparty wyłącznie o predefiniowane wzorce [82, 83].

W [84, 85] przedstawiono algorytm "Traffic Aware Scheduling Algorithm" (TASA), który jest dedykowany do sieci standardu TSCH. TASA zakłada budowanie planu w sposób scentralizowany dla ruchu sekwencyjnego z przewagą komunikacji od węzłów rozproszonych do węzła zbierającego dane (ruch wertykalny). Podstawowa zasada działania algorytmu jest analogiczna do już omówionych AMUS/EDF/LLF, dla każdej kolejnej szczeliny czasowej w oparciu o ramki znajdujące się w węzłach podejmowana jest decyzja o alokacji kanałów dla poszczególnych ramek. W przypadku TASA problem alokacji ramek dla szczeliny na konkretnym kanale radiowym, został formalnie przedstawiony jako skojarzenie i kolorowanie grafu połączeń między sąsiednimi węzłami. Algorytm TASA, działający na menadżerze sieci (SM), przeprowadza procedure ujednolicania grafu i kolorowania dla każdej kolejnej szczeliny czasowej w okresie planowania. Procedura ujednolicania grafu ma na celu znalezienie zbioru linków które są wolne od kolizji 1'ego stopnia (DCFL(k)). Natomiast procedura kolorowania grafu jest odpowiedzialna za znalezienie wszystkich zbiorów linków wolnych od kolizji 2'ego stopnia (ICFL(k)). Każdy zbiór ICFL(k) musi być zaplanowany na innym kanale dla szczeliny czasowej k. Mimo, iż TASA formalnie poprawnie definiuje problem planowania typu TSCH, to efektywnie działanie algorytmu jest zbliżone do działania algorytmów opartych o dystrybucję ruchu w buforach takich jak wcześniej zaprezentowane algorytmy oparte na metrykach EDF/LLF.

W [68] zaprezentowano algorytm SPRF (ang. scheduling of periodic real-time flow) stanowiący rozszerzenie TASA, dedykowany do obsługi ruchu horyzontalnego w wariancie cyklicznym. Zaprezentowano mechanizm adaptacyjnego ustalania priorytetu ruchu bazujący na maksymalnym czasie transmisji i liczbie pozostałych przeskoków potrzebnych na dostarczenie ruchu do węzła docelowego. Algorytm wykorzystuje ten sam mechanizm co TASA w zakresie unikania kolizji 1'ego i 2'ego stopnia. Działanie algorytmu jest efektywnie równoważne do

przedstawionemu wcześniej Algorytm 2 a różnica wynika jedynie z zastosowanej metryki. Proponowana metryka jest bardzo zbliżona do metryki LLF liczonej według wzoru (5), przy czym LLF zakłada, że elementy o mniejszych wartościach (mniejszym stopniu swobody) są alokowane w pierwszej kolejności natomiast, proponowana miara oparta o priorytecie jest de facto odwrotnością liczoną jako $(ts_deadline_j^k - ts + 1)/LLF$ i zakłada alokację w pierwszej kolejności tych o największych wartościach. Mimo, iż algorytm SPRF prezentowany jest jako dedykowany dla ruchu cyklicznego, pominięta jest całkowicie kwestia ustalenia okresu w którym wykonywane jest planowanie.

W [86] zauważono, że wszystkie prezentowane algorytmy planowania scentralizowanego zakładają istnienie globalnej listy kanałów wspólnej dla wszystkich węzłów w sieci. Natomiast w praktycznych zastosowaniach warunki propagacji i użyteczność poszczególnych kanałów radiowych, są różne dla każdego połączenia między węzłami. Dokładnie w ten sposób definiuje to również standard ISA100.11a, gdzie każdy z węzłów sieci ma możliwość określenia listy kanałów które obsługuje, jak również ma możliwość raportowania metryk opisujących stopień niezawodności połączenia na każdym kanale osobno. Zaproponowano algorytm planowania dedykowany do sieci drzewiastych o jednym punkcie odbioru danych (ruch wertykalny).

W [87] został podjęty problem bezpieczeństwa sieci WSN w związku łatwością ujawnienia lokalizacji źródła danych. Autorzy prezentują problem jako "panda i myśliwy", w którym czujniki dedykowane do monitorowania populacji zagrożonych osobników, stanowią jednocześnie źródło informacji o ich położeniu, które może posłużyć do ich unicestwienia. Zaprezentowano koncepcję rozszerzenia planowania ruchu o pseudo-pakiety, których zadaniem jest ukrycie prawdziwej lokalizacji źródła danych. Mimo, iż problem jest marginalny w sieciach IWN to pokazuje różne aspekty zagadnienia planowania.

W [88] przedstawiono problem planowania dodatkowych szczelin czasowych dla ewentualnych retransmisji ramek. Ponieważ wystąpienie retransmisji ramki jest przypadkowe, dodatkowa szczelina nie powinna być dedykowana wyłącznie dla tego ruchu ale współdzielona z innymi, przy jednoczesnym założeniu że ew. retransmisja musi się odbyć przed planowaną transmisją ramki dla kolejnego przeskoku wzdłuż trasy. Autorzy prezentują model pozwalający na ocenę skuteczności przygotowanego planu oraz prezentują algorytm pozwalający na zaplanowanie dodatkowych szczelin w taki sposób aby maksymalizować wykorzystanie dodatkowej współdzielonej szczeliny przygotowanej z myślą o potencjalnej retransmisji. Dodatkowo, zwracają uwagę na konieczność określenia poziomu złożoności operacji ponownego wykonania planu i jego dystrybucji do węzłów w sytuacji np. uszkodzenia węzła sieci. Z punktu widzenia algorytmów planowania zmiana topologii sieci jest traktowana jak zmiana danych wejściowych algorytmu, natomiast w praktyce wiąże się również z

koniecznością zaangażowania wymaganych zasobów w celu dystrybucji nowych konfiguracji do węzłów sieci.

W [63, 64] przedstawiono przegląd różnych rozwiązań dotyczących standardu TSCH w zakresie zarówno planowanie szczelin czasowych jak również sposobów budowania informacji o topologii sieci i innych problemów związanych z sieciami IIoT (synonim IWN). W [89] przedstawiono przegląd algorytmów planowania szczelin czasowych dla sieci WSN (ang. Wireless Sensor Network), które są prekursorem sieci IWN dedykowanym wyłącznie do monitoringu.

5.2 Badania eksperymentalne algorytmów

W rozdziale 5.2.1 prezentuję wyniki eksperymentalne dla algorytmów EDF, LLF, AMUS oraz autorskich rozszerzeń tych algorytmów. Analizując każdy z algorytmów wyciągam wnioski dotyczące jego wad i potencjalnych możliwości poprawy w odniesieniu do zastosowania w IWN. Badam jednocześnie warianty tych algorytmów eliminując zauważone wady lub sugerując zmiany w algorytmach zwiększające ich efektywność. W rozdziale 5.2.2 prezentuję analizę pracy sieci w oparciu o omawiane warianty algorytmów, wykorzystując opracowane środowisko testowe, rozdział 4.1. Analizę prowadzę w oparciu wyniki otrzymane dla opracowanego wektora testowego, rozdział 4.2. Bazując na wynikach porównania algorytmów dostępnych w literaturze oraz ich zmodyfikowanych wariantów uwzględniających sugerowane zmiany, wyciągam wnioski dotyczące autorskich algorytmów planowania dla ruchu cyklicznego i sekwencyjnego. Autorskie rozwiązania prezentuję w kolejnych rozdziałach 5.3 (CAS-PERIOD), 5.4 (CAS-SEQ) oraz algorytm GA-CAS stanowiący całościowe rozwiązanie zdefiniowane problemu 3.2.4 w rozdziałe 6. Opracowany autorski algorytm GA-CAS wykorzystuje autorskie algorytmy planowania CAS-PERIOD i CAS-SEQ.

5.2.1 Wyniki porównania algorytmów planowania szczelin

Algorytm AMUS został użyty w wektorze testowym w dwóch wariantach, gdzie okres planowania *p* został określony arbitralnie jako 1s (AMUS(1)) oraz określony jako okres, dla którego kontrakt o najniższej przepływności potrzebuje dokładnie jednej ramki na przesłanie danych (AMUS). Algorytmy: AMUS, AMUS(1), zostały użyte do planowania ruchu sekwencyjnego, podczas gdy algorytmy: LLF, EDF tylko ruchu cyklicznego. Kontrakty o innym typie ruchu, określane są jako nie zaplanowane (nie wliczają się do ilości zaplanowanych kontraktów). Każdy algorytm planowania uznaje kontrakt za nie możliwy do zaplanowania w momencie, gdy w przyjętym okresie planowania nie ma możliwości zaplanowania transmisji ze względu na kolizję z innymi węzłami. Dodatkowo w algorytmach: LLF, EDF kontrakt zostaje nie zaplanowany, jeżeli:

- Okres planowania (wyrażony w liczbie szczelin) liczony jako najmniejsza wspólna wielokrotność (ang. least common multiple, LCM) zgodnie z opisem algorytmów jest większy od (2²⁰ 1)
- Numer szczeliny, w której transmisja, znajdująca się w buforze, ma zostać przekazana do kolejnego węzła na trasie, jest większy od *ts_deadline^k_j*. Sytuacja stanowi brak możliwości wypełnienia QoS dotyczącego czasu na dostarczenie pakietu.

Część wyników prezentowana jest na wykresach typu "boxplot", gdzie przyjęto:

- Dolna i górna krawędź obszaru zakolorowanego odpowiada wartości dla odpowiednio 1 i 3 kwartyla (Q1, Q3)
- Linia wewnątrz obszaru zakolorowanego odpowiada wartości mediany (2 kwartyl, Q2)
- Dolna i górna linia znajdująca się poza obszarem zakolorowanym to odpowiednio Q1 1.5 · (Q3 – Q1) oraz Q3 + 1.5 · (Q3 – Q1)
- Obserwacje odstające nie są prezentowane na wykresach
- Wartość średnia prezentowana jest w postaci białej kropki

Suma czasów wykonania wszystkich planów w zadanym wektorze testowym dla badanych algorytmów jest równa 1200202,58s i 1152673,39s dla GATEWAY_4B oraz 202803,99s i 180934.29s dla stacji roboczej odpowiednio dla wariantu 1 i 2 generowania topologii sieci. W dodatku B, dla badanych algorytmów zebrałem parametry rozkładów następujących parametrów wygenerowanych planów dla wariantu 1 i 2 oraz dla obu urządzeń:

- Czas wykonania planu Tabela B.7, Tabela B.15, Tabela B.11, Tabela B.16
- Długość planu Tabela B.8, Tabela B.12
- Liczba elementów w planie Tabela B.9, Tabela B.13
- Poziom zaplanowania Tabela B.10, Tabela B.14

Algorytmy planowania szczelin w ruchu cyklicznym dostępne w literaturze [61, 72, 90] opierają się na podobnym wzorcu działania, w którym w głównej pętli, realizowanej do czasu obecności ramek w buforze, realizowane są następujące czynności:

- Uwolnienie ramek w węzłach i przekazanie do bufora
- Aktualizacja kolejności ramek w buforze na podstawie zadanej metryki
- Właściwe przekazanie ramek pomiędzy węzłami, usunięcie ramek dostarczonych do węzłów docelowych oraz umieszczenie szczeliny czasowej w planie.

Różnice sprowadzają się do metryki stosowanej do zadania kolejności ramek w buforze. W analizie posłużyłem się dwiema metrykami EDF i LLF i te algorytmy stanowią punkt wyjścia do obserwacji i analizy działania algorytmów.



Rysunek 5.1 Czas wykonania (rząd 1) oraz poziom zaplanowania (rząd 2) w funkcji liczby węzłów dla wariantu 1 (kolumna 1) i wariantu 2 (kolumna 2) generowania topologii sieci – GATEWAY_4B

Na rysunku 5.1 prezentuję rozkład czasu wykonania oraz poziomu zaplanowania dla topologii sieci generowanej w wariancie 1 oraz 2 dla wszystkich elementów wektora testowego. Poziom zaplanowania obliczany jest jako iloraz sumy zaplanowanych kontraktów do liczby wszystkich kontraktów i wyrażony jest w procentach. Ponieważ analizowane w tym rozdziale algorytmy operują jedynie na wybranym typie kontraktów (cykliczne/sekwencyjne) wszystkie kontrakty o nieobsługiwanym typie są uznawane za niezaplanowane. Maksymalny poziom zaplanowania jest więc równy 50% (wartości dla parametru N_{CP}/N_C są uśredniane co odpowiada sytuacji, gdzie $N_{CP}/N_C = 0.5$). Innymi słowy poziom zaplanowania odnosi się do liczby zaplanowanej przez konkretny algorytm w stosunku do liczby wszystkich kontraktów znajdujących się w wektorze testowym. Biorąc pod uwagę, że w wektorze testowym mamy zarówno kontrakty
cykliczne jak i sekwencyjne a ich średni udział to 50%, w przypadku analizy algorytmu, który operuje wyłączenie na kontraktach cyklicznych, aby uzyskać skuteczność planowania samego algorytmu w odniesieniu do liczby kontraktów które potencjalnie mógł zaplanować, należy prezentowaną wartość poziomu planowania pomnożyć razy 2. Przyjęta tutaj konwencja prezentacji poziomu zaplanowania podyktowana jest koniecznością zachowania spójności z definicją postawionego problemy (3.2.4), w którym optymalizacji i planowaniu poddawany jest ruch zarówno cykliczny jak i sekwencyjny. Prezentowany w rozdziale 6 algorytm GA-CAS stanowi kompleksowe rozwiązanie postawionego problemu i integruje zarówno algorytmy planowania dla ruchu cyklicznego jak i sekwencyjnego. W przypadku algorytmu GA-CAS maksymalny poziom zaplanowania wynosi 100% w odniesieniu do opracowanego wektora testowego, natomiast dla algorytmów które operują na wybranych typie ruchu maksymalny poziom zaplanowania to 50% w odniesieniu do opracowanego wektora testowego.



Rysunek 5.2 Rezerwacja szczelin czasowych TDMA dla algorytmów LLF i EDF przy pakiecie składającym się z 4 ramek i trasie obejmującej 5 przeskoków

Porównując algorytmy EDF i LLF zauważyłem, że LLF daje dużo gorszy poziom zaplanowania (Rysunek 5.1). Przy tym samym okresie planowania wyrażonym w liczbie szczelin czasowych 600917 dla wariantu 1 i 708748 dla wariantu 2, EDF uzyskuje 16,94% zaplanowania podczas gdy LLF jedynie 12,36% dla wariantu 1 (Tabela B.10) i odpowiednio 11,66% oraz 9,12% dla wariantu 2 (Tabela B.14). Różnica jest szczególnie duża dla dużej liczby przeskoków na drodze do węzła docelowego (trasa 2) oraz dużej liczbie kanałów (8 i 16), patrz rysunek A.3 oraz rysunek A.4. Różnica maleje wraz z malejącą liczbą przeskoków i zanika dla liczby przeskoków wynoszącej 1. W przypadku wariantu 1 dla trasy 1 (minimalna liczba przeskoków) poziom zaplanowania to 13,15% oraz 13,03% odpowiednio dla EDF i LLF, przy liczbie węzłów wynoszącej 90 i dostępnej liczbie kanałów 16. Dla tej samej konfiguracji przy trasie 2

(maksymalna niezawodność) poziom zaplanowania dla LLF spada do 11,51% oraz 4,78% odpowiednio dla EDF i LLF. W przypadku wariantu 2 poziom zaplanowania to 11,56% dla obu algorytmów przy trasie 1 oraz 9,62% oraz 3,3% dla trasy 2. Dodatkowo warto zauważyć, że LLF dużo gorzej wykorzystuje dostępne kanały. Dla wariantu 1 przy 90 węzłach w sieci, zmiana poziomu zaplanowania przy zmianie liczby dostępnych kanałów z 2 -> 8 to odpowiednio 8,21% -> 11,65% dla EDF i jedynie 4,6% -> 4,82%. Mimo dużego wzrostu liczby dostępnych kanałów LLF nie pozwala na zaplanowanie znacząco większej liczby kontraktów. Analiza generowanego planu pozwoliła na znalezienie powodu, dla którego LLF daje tym gorszy rezultat im więcej jest węzłów pośredniczących na trasie do węzła docelowego.

Zastosowanie metryki LLF dla ruchu, w którym pakiety podzielone są na ramki oraz transmisja odbywa się przez wiele węzłów pośredniczących prowadzi do generowania planu jak na rysunku 5.2. Dla LLF transmisja realizowana jest w taki sposób, że wszystkie ramki z pakietu przekazywane są w kolejnych szczelinach do pierwszego wezła na trasie, następnie wszystkie ramki przekazywane są do kolejnego węzła i tak dalej aż do osiągniecia węzła docelowego. Taki sposób przekazywania ramek powoduje, że ramki zawsze znajdują się wyłącznie na węzłach będących bezpośrednimi sąsiadami. W takiej sytuacji nie ma możliwości wysyłania ramek na innych kanałach ze względu na kolizję 1'ego stopnia. Odmiennie sytuacja wygląda, jeżeli ramki dystrybuowane sa w taki sposób, aby jak najszybciej co najmniej jedna ramka dotarła do węzła N+2, który jak tylko otrzyma ramkę może na innym kanale rozpocząć jej przekazywanie dalej. Doprowadzenie ramki do węzła N+2 powoduje, że w przypadku transmisji wezeł ten nie będzie źródłem konfliktu 1'ego stopnia. Nawet gdyby wezeł był sasiadem węzła N lub węzła N+1 i miała miejsce kolizja 2'ego stopnia węzeł ten ma możliwość uniknięcia kolizji drugiego stopnia poprzez użycie innego dostępnego kanału. Powodem, dla którego użycie metryki LLF skutkuje takim zachowaniem jest uwzględnienie położenia ramki w samej metryce co powoduje, że w pierwszej kolejności dystrybuowane są ramki znajdujące się najdalej od węzła docelowego (dla tego samego kontraktu). Plan jest generowany optymalnie w przypadku metryki EDF. Metryka EDF uwzględnia $ts_deadline_i^k$ którego wartość jest taka sama dla wszystkich ramek uwolnionych podczas wysyłania pakietu w ramach kontraktu k. Tym samym metryka nie ma wpływu na kolejność dystrybucji ramek w ramach pakietu. Kolejność ramek wynika z implementacji algorytmu i w tym konkretnym przypadku, gdy metryka daje ta sama wartość, ramka dodana jako pierwsza do bufora zawsze znajduje się przed ramką dodaną jako druga. Dostosowanie metryki EDF do ruchu dystrybuowanego w podziale na ramki oraz poprzez wiele przeskoków, wymaga jawnie określenia kolejności porządkowania ramek w ramach metryk o tej samej wartości. Bazując na tym założeniu został powołany w analizie osobny algorytm EDF FIFO dla którego kolejność ramek ustalana jest w pierwszej kolejności z wykorzystaniem metryki EDF a następnie kolejności FIFO, tak by ramka umieszczona wcześniej poruszała się jako pierwsza aż do osiągnięcia węzła docelowego. Należy zwrócić uwagę, że w implementacji algorytmu, ramka umieszczana jest w buforze w momencie jej uwolnienia, przekazywanie ramki pomiędzy węzłami nie powoduje jej usunięcia i ponownego dodania do bufora a jedynie zmianę własności opisujących ramkę (bieżące l_i^i).

Również w przypadku algorytmu LLF możliwe jest zapewnienie poprawnego generowania planu, należy w tym celu przenieść określenie poziomu swobody z transmisji (ramki) na pakiet. Ponieważ pakiet jest pojęciem grupującym ramki nie możemy wprost określić jego położenia, wartości h_{ts}^k . W odniesieniu do pakietu możemy powiedzieć, że został uwolniony oraz że dotarł (lub nie) do węzła docelowego, jego położenie w sieci jest natomiast rozproszone na wiele węzłów biorących udział w transmisji. Proponuję obliczenie poziomu swobody dla pakietu jako:

$$laxity_packet_{ts}^{k} = \left(ts_deadline_{j}^{k} - ts + 1\right) - \left[\overline{h_{ts}^{\{p,k\}}}\right]$$
(29)

gdzie $\overline{h_{ts}^{\{p,k\}}}$ jest średnią ilością pozostających transmisji ramek należących do pakietu p uwolnionego w ramach kontraktu k.

Analogicznie jak w przypadku EDF_FIFO poprawne generowanie planu dla LLF zmodyfikowanego zgodnie ze wzorem (29), wymaga narzucenia porządku drugiego poziomu dla ramek należących do pakietu i tym samym posiadających tą samą wartość miary *laxity_packet*^k_{ts}. Tak zmodyfikowany algorytm LLF otrzymał nazwę LLF_FIFO i dołączony do badanych algorytmów. Wartości poziomu zaplanowania uzyskane dla LLF_FIFO wzrosły w sposób znaczący i oscylują na poziomach zbliżonych do EDF / EDF_FIFO. Algorytm LLF_FIFO uzyskał poziom średniego zaplanowania 16,84% podczas gdy LLF jedynie 12,36% dla wariantu 1 (Tabela B.10) oraz 11,58% i 9,12% odpowiednio dla wariantu 2 (Tabela B.14). Algorytm EDF_FIFO zgodnie z założeniem uzyskał identyczny poziom co EDF, wariant 1 – 16,94% oraz wariant 2 – 11,66%. Dla obu wariantów generowania topologii sieci EDF_FIFO / EDF daje lepsze rezultaty od LLF_FIFO przy czym różnica jest mała. W rozbiciu na trasy wyniki są następujące:

- w przypadku wariantu 1 (Tabela B.17) (90 węzłów, 16 kanałów) dla trasy 1 (minimalna liczba przeskoków) poziom zaplanowania to 13,13% oraz 13,03% odpowiednio dla LLF_FIFO oraz LLF, podczas gdy EDF / EDF FIFO uzyskał 13,15%, dla trasy 2 (maksymalna niezawodność) poziom zaplanowania to 11,36% oraz 4,78% odpowiednio dla LLF_FIFO oraz LLF, podczas gdy EDF / EDF FIFO uzyskał 11,51%.
- w przypadku wariantu 2 (Tabela B.18) (200 węzłów, 16 kanałów) dla trasy 1 (minimalna liczba przeskoków) poziom zaplanowania to 11,56% dla LLF / LLF_FIFO oraz EDF / EDF FIFO, dla trasy 2 (maksymalna niezawodność) poziom zaplanowania

to 9,40% oraz 3,30% odpowiednio dla LLF_FIFO oraz LLF, podczas gdy EDF / EDF FIFO uzyskał 9.52%.

Analiza działania algorytmów pozwala zauważyć, że kontrakty k1 i k2 mogą być planowane niezależnie o ile żaden wezeł stanowiący punkt na trasie kontraktu k1 nie jest bezpośrednim sasiadem żadnego wezła na trasie kontraktu k2. Jest to warunek wystarczający do tego, aby nie mogła zaistnieć kolizja 1'ego oraz 2'ego stopnia między węzłami. Brak jakiejkolwiek kolizji pomiędzy zbiorami kontraktów pozwala traktować zbiory kontraktów niezależnie i daje możliwość generowania planów niezależnie dla każdego ze zbiorów. To założenie pozwala na dokonanie grupowania kontraktów a następnie uruchomienie właściwych algorytmów dla każdej grupy osobno. Faza grupowania kontraktów stanowi element wspólny dla grupy pozostałych czterech wariantów badanych algorytmów rozpoczynających się prefiksem "SEP ". Algorytm SEP EDF FIFO jest wariantem algorytmu EDF FIFO w którym generowanych jest N planów dla N grup powstałych w wyniku grupowania. Mimo iż grupowanie stanowi potencjalnie istotny element, który może podnieść poziom planowania, w praktyce wyodrębnienie >1 grup jest bardzo rzadkie. Ogólny poziom zaplanowania dla SEP EDF FIFO to 16,95%, natomiast dla EDF FIFO wartość jest nieznacznie niższa i wynosi 16,94% dla wariantu 1 oraz 11,66% dla wariantu 2 dla obu odpowiednio. Wariant 1 generowania topologii sieci nie jest reprezentatywny w tym porównaniu, ponieważ generowanie połączeń pomiędzy węzłami jest losowe zgodnie z przyjętą gęstością grafu. W przypadku wariantu 2 algorytmy SEP * mogą dać wymierne korzyści dla większych przestrzeni. Zgodnie z tabelą 4.1 wymiary przestrzeni to [X=50, Y=25, Z=4]m, w odniesieniu do analizy przeprowadzonej w 3.1.2.1, wezły sa w zasięgu widoczności radiowej. Gdyby rozpatrywana przestrzeń miała jeden z wymiarów >100m, grupowanie dałoby lepsze rezultaty. Jednocześnie należy zaznaczyć, że dla większej przestrzeni przy przyjętej liczbie węzłów na m2 całkowita liczba węzłów wzrośnie i należy stwierdzić, że obsługa tak dużej sieci dla SM opartego o GATEWAY 4B nie byłaby możliwa.

Na rysunkach 5.3 (cały wektor) oraz A.3 (wariant 1) i A.4 (wariant 2) prezentuję poziom zaplanowania [%] w funkcji całkowitej liczby kontraktów podlegających planowaniu odpowiednio dla wariantu 1 oraz wariantu 2 generowania topologii sieci. Wyniki w załączniku prezentuję w podziale na liczbę kanałów [2,8,16] oraz trasę [1,2]. Wyraźnie widać, że wyniki tworzą trzy grupy (zaznaczono w pierwszym wierszu):

• Grupa 1 to algorytmy cykliczne LLF i SEP_LLF które uzyskały najniższy poziom zaplanowania wśród wszystkich analizowanych wariantów

- Grupa 2 to algorytmy cykliczne EDF, EDF_FIFO, SEP_EDF_FIFO oraz SEP_LLF_FIFO uzyskane wyniki zaplanowania są wyższe od grupy 1, ale bardzo zbliżone do siebie w ramach grupy 2
- Grupa 3 to algorytmy sekwencyjne AMUS i AMUS(1) różniące się jedynie okresem planowania



Rysunek 5.3 Poziom zaplanowania w funkcji liczby kontraktów

Obserwując w szczególności rysunek A.3 uwagę zwraca niemonotoniczny charakter zależności poziomu zaplanowania w funkcji liczby kontraktów. Powodem jest sposób generowania danych w wektorze testowym. Liczba kontraktów nie jest liczbą uzyskiwaną wprost w wektorze testowym a stanowi wynik pośredni uzależniony od wartości innych parametrów wektora testowego takich jak: liczba węzłów (N_V), stosunek liczby węzłów będących źródłem transmisji do liczby wszystkich węzłów w sieci ($|S|/N_V$) oraz średnia liczba kontraktów przypadająca na jeden węzeł źródłowy ($N_C/|S|$). Jeżeli porównamy tą samą zależność zaprezentowaną w rozbiciu na poszczególne parametry, od których zależy całkowita liczba kontraktów, to monotoniczny charakter funkcji jest zachowany, rysunek B.3 dla wariantu 1 i rysunek B.4 dla wariantu 2 generowania topologii sieci.

Dla trasy z minimalną liczbą przeskoków (trasa 1) wartości dla grupy 1 i 2 są zbliżone. Grupa 2 uzyskuje większy poziom zaplanowania jedynie dla bardzo małej liczby kontraktów (tylko wariant 1). Dla całkowitej liczby kontraktów 6 i liczby dostępnych kanałów 16, algorytm LLF uzyskał 31,88% natomiast SEP_LLF_FIFO 41,25%, podczas gdy już dla 24 kontraktów różnica maleje i uzyskujemy odpowiednio 18,33% oraz 19,27%. Dla trasy z maksymalną niezawodnością (trasa 2) różnica pomiędzy grupą 1 i 2 staje się wyraźna. Różnica rośnie wraz ze wzrostem liczby dostępnych kanałów, co potwierdza wcześniejszą analizę. Algorytmy z

rodziny EDF oraz zmodyfikowane algorytmy LLF pozwalają wykorzystać większą liczbę kanałów w celu dostarczenia pakietu w krótszym czasie, tym samym dając więcej swobody w dostarczaniu ruchu generowanego przez inne kontrakty. Algorytmy z grupy 3 (ruch sekwencyjny) w wariancie 1 uzyskują bardzo podobne poziomy planowania oscylujący w okolicy maksymalnej wartości wynoszącej 50%. Jedynie dla niskiej liczby dostępnych kanałów (2) algorytm AMUS o krótkim okresie planowania wynoszacym 1s uzyskuje gorsze rezultaty. Różnica zwiększa się wraz ze wzrostem liczby kontraktów, dla trasy 2 różnica jest wyraźna już przy 20 kontraktach podczas gdy dla trasy 1 różnica pojawia się w okolicy 80 kontraktów. W przypadku wariantu 2 i większej liczby węzłów charakter tej zależności jest o wiele bardziej wyraźny. Dla małej liczby kanałów (2) i przy dużej liczbie kontraktów, poziom zaplanowania spada do wartości obserwowanych w algorytmach dla ruchu cyklicznego. Dla 16 kanałów bez względu na rodzaj trasy AMUS pozostaje przy maksymalnych wartościach podczas gdy AMUS(1) odchyla się od maksymalnych wartości jedynie dla trasy 2 i przy liczbie kontraktów większej od 160. Analizowane algorytmy cykliczne uzyskują względnie małe poziomy zaplanowania już przy relatywnie małej liczbie kontraktów (węzłów w sieci). Charakter zmiany poziomu zaplanowania jest wykładniczy i uzyskuje nieakceptowalnie małe poziomy już dla około 20 kontraktów. Rozkład poziomu planowania w funkcji liczby węzłów prezentuję na rysunku B.5 dla wariantu 1 i rysunku B.6 dla wariantu 2 generowania topologii sieci.

Analizując poziom zaplanowania dla algorytmów cyklicznych (Tabela B.17) należy zauważyć, że satysfakcjonujący poziom zaplanowania (przyjmujemy >33%) uzyskują jedynie dla bardzo małej całkowitej liczby kontraktów. Algorytm SEP_EDF_FIFO dla sieci z 10 węzłami, 16 dostępnych kanałach i całkowitej liczbie kontraktów wynoszącej 6 (średnia liczba kontraktów cyklicznych to jedynie 3) poziom zaplanowania wynosi 41%. Przy podwojeniu liczby kontraktów przypadających na węzeł źródłowy, poziom dla analogicznej konfiguracji spada do 27,5%. Średni poziom zaplanowania dla sieci składającej się minimalnej liczby węzłów wynoszącej 10, to zaledwie 30%, poziom ten spada gwałtownie i przy 30 węzłach w sieci uzyskujemy 18,16% dla trasy 1 i 16,32 dla trasy 2. Oznacza to, że jedynie 3 z 10 kontraktów dla ruchu cyklicznego podlegających planowaniu zostało zaplanowanych poprawnie. Reszta kontraktów została uznana za niemożliwe do zaplanowania i nie została umieszczona w planie. W sposób oczywisty umieszczenie kontraktu w planie jest warunkiem koniecznym (nie wystarczającym) do uznania kontraktu jako wypełniony zgodnie z definicją postawionego problemu 3.2.4, równanie (12). Algorytmy planowania uznają kontrakt jako nie możliwy do zaplanowania gdy:

1. Okres planowania określany jako najmniejsza wspólna wielokrotność wszystkich kontraktów umieszczonych w planie (LCM) jest większy od maksymalnej wartości przyjętej w implementacji algorytmu.

 Ramka uwolniona w ramach kontraktu nie może zostać dostarczona do węzła docelowego w czasie wynikającym z wymagania QoS dotyczącego maksymalnego opóźnienia (ang. end-to-end latency), Tabela 3.4

Przypadek 2 stanowi kryterium zgodnie z przyjętymi założeniami zdefiniowanego problemu (3.2.4). Zgodnie z wcześniejszymi rozważaniami minimalna możliwa do zagwarantowania wartość $cp_k^{\{s,d\}}$. Deadline wynika z liczby przeskoków na trasie oraz liczby kanałów które można wykorzystać do dystrybucji ramek pakietu. Im żądana w kontrakcie wartość $cp_k^{\{s,d\}}$. Deadline jest bliżej wartości minimalnej, tym istnieje większe prawdopodobieństwo, że ewentualna kolizja w minimalnym planie spowoduje brak możliwości zagwarantowania szczelin czasowych które pozwolą na dostarczenie ramek pakietu przed $ts_deadline_j^k$, równanie (4). Istotą stosowanych metryk jest określenie optymalnej kolejności ruchu.

Przypadek 1, sprowadza się do określenia górnej granicy liczbowej stanowiącej maksymalną długość plan. Zgodnie z przyjętym założeniem okres ruchu cyklicznego stanowiący parametr kontraktu dla standardu ISA100.11a zdefiniowany jest jako niezerowa liczba całkowita. W praktyce przyjmuje bardzo szeroki zakres od 10ms do 32767 s. W badanym wektorze testowym okres jest wartością ze zbioru {10, 11, 12, 14, 16, 20, 25, 33, 50, 100, 200, ..., 9900, 10000} szczelin czasowych. Najmniejsza wspólna wielokrotność dla tak zdefiniowanego zbioru okresów jest o wiele większa niż 2⁶⁴. Na potrzeby implementacji algorytmów przyjęto, że maksymalna długość planu wynosi $(2^{20} - 1)$ szczelin czasowych. Wartość stanowi kompromis pomiedzy długościa planu a czasem jego wykonywania. W literaturze powyższy problem jest pomijany w ewaluacji wyników poprzez odpowiedni dobór okresów dla ruchu podlegającego planowaniu. W artykule [91] przyjęto, że okres przyjmuje jedynie wartości będące kolejnymi potęgami liczby 2, jako 2^i , gdzie *i* przyjął z zakresu <5; 10>. Przyjęcie takich założeń upraszcza problem LCM, natomiast nie wypełnia założeń sieci IWN, gdzie okres może być przyjęty dowolnie we wskazanym zakresie, w tym może obejmować np. same liczby pierwsze. Problem dotyczący ograniczenia LCM wraz z oczekiwanym krótkim czasem wykonania planu stanowią główne ograniczenia w praktycznym wykorzystaniu badanych algorytmów do planownia ruchu cyklicznego.

Na rysunku 5.1 (zbiorczo) oraz rysunkach B.7 i B.8 (z podziałem) prezentuję wyniki czasu wykonania algorytmów na GATEWAY_4B dla odpowiednio wariantu 1 i 2 generowania topologii sieci. Wyniki prezentowane są z podziałem na rodzaj trasy oraz liczbę dostępnych kanałów. Dodatkowo na rysunku B.9 oraz B.10 prezentuję analogiczne wyniki dla wykonania na stacji roboczej. Parametry rozkładu czasu wykonania prezentuję w tabeli B.7 (wariant 1) i B.11 (wariant 2) dla GATEWAY_4B oraz tabeli B.15 (wariant 1) i B.16 (wariant 2) dla stacji roboczej. Analizując wyniki, posługuję się wartościami liczbowymi dla GATEWAY_4B,

wartości dla stacji roboczej są średnio o około 6-6.4 razy krótsze. Średni czas wykonania planów dla wszystkich wektorów testowych w przypadku algorytmu AMUS(1) to 0.02s oraz 0.37s, natomiast dla AMUS to 0.36 oraz 15.96s dla odpowiednio wariantu 1 i 2 generowania topologii sieci. Algorytm AMUS(1) ma stałą długość okresu planowania wynoszącą 100 szczelin czasowych, natomiast AMUS ma długość zmienną określaną na podstawie parametrów planowanych kontraktów. Średnia długość planu dla AMUS, w badanym wektorze wynosi 8555 i 11947 szczelin czasowych dla odpowiednio wariantu 1 (Tabela B.8) i 2 (Tabela B.12). Zestawiając te rozkłady należy stwierdzić, że czas wykonywania planu jest liniowo skorelowany z długością wykonywanego planu. W przypadku algorytmu AMUS dla kontraktów sekwencyjnych okres planowania może być dobrany arbitralnie na względnie niskim poziomie, natomiast w przypadku algorytmów do planowania ruchu cyklicznego okres planowania musi wynikać z najmniejszej wspólnej wielokrotności (LCM), której maksymalną wartość przyjąłem jako $(2^{20} - 1)$. Mimo, iż tak przyjęta wartość nie pozwala na zaplanowanie wszystkich kontraktów znajdujących się w wektorze testowym to i tak czas wykonania algorytmów dla kontraktów cyklicznych jest nieakceptowalnie długi. Średni czas wykonania algorytmów dla wariantu 2 to 54.14s i 54.04s dla odpowiednio SEP EDF FIFO i SEP LLF FIFO. Czas wykonania algorytmów LLF oraz SEP LLF jest krótszy ale wynika to jedynie z dużo mniejszego uzyskiwanego poziomu zaplanowania. Wszystkie algorytmy wykazują dodatnią wykładniczą korelację z wielkością sieci (ilością węzłów w sieci). W przypadku wariantu 2, średni czas wykonania dla SEP (EDF/LLF) FIFO to około 100s przy 200 węzłach i 10s przy 50 węzłach. W przypadku wariantu 1 uzyskałem dla tych algorytmów średnio około 30s przy 90 węzłach i 1s przy 10 węzłach. Rodzaj trasy (liczba przeskoków) w oczywisty sposób wpływa na liczbę generowanych elementów planu, ponieważ wymaga aktywności większej liczby węzłów. Dla wariantu 2 i trasy 2 przy 16 dostępnych kanałach czasy wykonywania są około 2 razy dłuższe niż w analogicznym przypadku dla trasy 1.

Wszystkie analizowane algorytmy dla ruchu cyklicznego nie pozwalają na uzyskanie zadowalających rezultatów z punktu widzenia możliwości ich zastosowania do rozwiązania problemu zdefiniowanego w 3.2.4. Jako powody należy wymienić:

- Niski poziom zaplanowania uzyskiwany już dla względnie małej liczby kontraktów
- Długi czas wykonywania planu, wynoszący średnio ponad 10s dla przyjętej maksymalnej długości planu wynoszącej (2²⁰ – 1) szczelin czasowych.

Długi czas wykonywania nie pozwala na wykonanie w realnym czasie (30s lub 60s) wystarczająco dużej liczby iteracji, tak aby było możliwe całościowe rozwiązanie zdefiniowanego problemu.

5.2.2 Analiza pracy sieci

Wykorzystując opracowane środowisko testowe (rozdział 4.1) dla każdego przygotowanego planu przeprowadzono symulację pracy sieci. Zgromadzone w trakcie symulacji dane pozwalają na określenie poziomu opóźnienia w dostarczaniu pakietów oraz zajętości buforów ramek jakie znajdują się w każdym węźle sieci IWN.



Rysunek 5.4 Rozkład maksymalnego czasu transmisji pakietu w funkcji liczby węzłów dla wariantu 1 i 2 generowania topologii sieci

Kluczowe, z punktu widzenia możliwości zastosowania sieci bezprzewodowej w zastosowaniu przemysłowym, jest ograniczenie maksymalnego czasu transmisji pakietu oraz ograniczenie jego zmienności. Średni maksymalny czas transmisji pakietu dla wszystkich badanych algorytmów (pomijając wariant LLF i SEP LLF, dyskusja 5.2.1) jest zbliżony i wynosi 12 szczelin czasowych dla wariantu 1 (Tabela B.21) oraz 10.26 szczelin dla wariantu 2 (Tabela B.24) generowania topologii sieci. Na rysunku 5.4 prezentuję rozkład maksymalnego czasu transmisji pakietu w funkcji liczby węzłów. Dla wariantu 1 jedynie przy 10 węzłach czas transmisji jest krótszy przy pozostałych rozmiarach sieci czas pozostaje na podobnym poziomie. Dla wariantu 2 we wszystkich trzech badanych rozmiarach sieci średni maksymalny czas jest różny i wynosi 9.01, 9,85 oraz 10,94 szczelin dla algorytmu SEP EDF FIFO. Różnica wartości średniej maksymalnego czasu transmisji pomiędzy algorytmami jest bardzo mała i oscyluje w zakresie pojedynczych części promila na korzyść wariantów EDF, który w każdej konfiguracji algorytmów uzyskuje krótszy czas niż odpowiedni wariant LLF. We wszystkich badanych wariantach rozkład maksymalnego czasu transmisji pakietu jest mocno przesunięty w stronę mniejszych wartości z medianą znajdująca się znacznie poniżej wartości średniej. Jednocześnie wartości maksymalne (bez obserwacji odstających) znajdują się znacznie powyżej wartości średniej, około 3 razy wartość średnia i wynoszą 33 szczeliny wariant 1 oraz 30 szczelin wariant 2 (200 węzłów). W przypadku wariantu 2, aż 25% obserwacji ma czas powyżej 15 szczelin (200 węzłów).

Na rysunku B.11 oraz rysunku B.12 prezentuję wyniki z podziałem na trasę oraz liczbę kanałów odpowiednio dla wariantu 1 i 2 generowania topologii sieci. W przypadku wariantu 1 zaskakująca jest mała różnica w rozkładzie maksymalnego czasu pomiędzy trasami. Średni maksymalny czas dla trasy 1 (SEP EDF FIFO) wynosi 10,49 oraz 13,85 dla trasy 2. Dla wariantu 2 różnica jest zgodna z oczekiwaną i średni maksymalny czas wynosi odpowiednio 6,4 oraz 15,82. Dla wariantu 2 i trasy 1 szerokość rozkładu jest znacznie mniejsza niż w przypadku wariantu 1, co dodatkowo potwierdza, że wariant 1 w zakresie modelowania opóźnienia nie jest właściwym wyborem. W przypadku modelowania sieci, w której większość węzłów znajduje się w zakresie bezpośredniej widoczności (np. hala o powierzchni kilkuset m²), spodziewamy się, że większość tras, będzie miała 1-3 przeskoki a czas transmisji będzie w dużej mierze zależny od ilości ramek w pakiecie. W tym miejscu wyraźnie widać, że wariant 2 jest bardziej odpowiedni do modelowania sieci IWN, mimo że wariant 1 jest często stosowany w literaturze. Dla wariantu 2 i trasy 1 przy wystarczającej, z punktu widzenia ilości ruchu, liczbie kanałów rozkład jest identyczny dla trzech badanych wielkości sieci, co potwierdza powyższe spostrzeżenie. Dłuższe trasy, obecne dla przypadku tras o maksymalnej niezawodności, powodują, że maksymalny czas rośnie wraz ze wzrostem wielkości sieci. Powodem, dla którego rośnie czas dostarczenia pakietu jest rosnąca dla większych sieci średnia liczba przeskoków przypadająca na kontrakt. Liczba kanałów nie wpływa zasadniczo na rozkład czasu transmisji, szczególnie dla wystarczającej liczby kanałów (8 i 16). Dla 2 kanałów, obserwowany krótszy czas związany jest ze znacznie mniejszą liczbą zaplanowanych kontraktów.

Drugim istotnym parametrem, obok czasu transmisji pakietu, jest jego zmienność w odniesieniu do konkretnego kontraktu. Zmienność czasu transmisji jest dominującym elementem wpływającym na jitter. Dążymy, aby wartość tej zmienności wynikająca z planu była zminimalizowana a w praktyce równa 0. Średnia wartość odchylenia standardowego rozkładu czasu transmisji (SEP_EDF_FIFO) wynosi 0,12 szczeliny dla wariantu 1 (Tabela B.20) oraz 0,19 dla wariantu 2 (Tabela B.23). Różnice wartości średniej pomiędzy badanymi algorytmami są małe i należy założyć, że wynikają jedynie z różnic w poziomie zaplanowania. Rozkład odchylenia standardowego (Rysunek B.13, Rysunek B.14) jest w obu wariantach silnie przesunięty w stronę zerowych wartości. Wartość mediany w obu wariantach jest równa 0 a wartość maksymalna (razem z obserwacjami odstającymi) jest wysoka i wynosi (SEP_EDF_FIFO) 7,2 szczeliny dla wariantu 1 oraz 8,92 dla wariantu 2. Dla wszystkich konfiguracji wartość średnia jest powyżej trzeciego kwartyla (Q3).

Odchylenie czasu transmisji pakietu jest spowodowane obecnością kolizji w procesie planowania. Algorytm planowania rozstrzyga kolizję w taki sposób, że jedna z transmisji jest realizowana w późniejszej wolnej szczelinie. Warto zwrócić uwagę, że dla wariantu 2 istnieje znacząca różnica pomiędzy maksymalnym odchyleniem dla algorytmu SEP_EDF_FIFO wynosi 8,92 szczeliny natomiast dla SEP_LLF_FIFO aż 13,46. Również ten parametr potwierdza lepsze dostosowanie miary EDF do zastosowań IWN.

Na rysunku A.5 prezentuję zależność maksymalnego czasu transmisji pakietu w funkcji liczby przeskoków z podziałem na liczbę dostępnych kanałów oraz gęstość ruchu rozumianą jako liczbę kontraktów przypadających na jeden węzeł sieci (iloczyn parametrów $|S|/N_V$ oraz $N_{C}/|S|$ według tabeli 4.1). Warto zauważyć, że gradient zmiany czasu pomiędzy 1 a 2 przeskokiem jest większy niż pomiędzy 2 a kolejnymi. Jeżeli transmisja na trasie o jednym przeskoku zajmuje średnio 5.5 szczeliny to dla dwóch przeskoków jest dwa razy większa i wynosi 11 szczelin. Dla kolejnych przeskoków czas się wydłuża w sposób jednostajny ze średnim nachyleniem dla badanego wektora (16 kanałów, $|S|/N_V = 0.8$, $N_C/|S| = 2$) wynoszącym 1 szczelina na jeden przeskok. Obserwacja pokrywa się ze schematem rezerwacji szczelin prezentowanym wcześniej (Rysunek 5.2), gdzie dla dwóch pierwszych przeskoków nie ma możliwości, za względu na kolizję 1'ego stopnia, na dystrybucję ruchu na wielu kanałach. Takie zachowanie jest charakterystyczne dla węzła z pojedynczym blokiem radiowym, zastosowanie węzłów wyposażonych w dwa i więcej układy nadawczo-odbiorcze pozwoli na wykorzystanie wielu kanałów również dla krótkich tras. Takie węzły pozwoliłby znacznie skrócić czas transmisji, gdzie węzły zamiast wysyłać ramka po ramce mogą wysyłać symbol po symbolu, nie czekając na koniec szczeliny. Budowa takiego układu oraz badanie modelu sieci opartego o węzły z wieloma blokami radiowymi jest zakresem wartym eksploracji z punktu widzenia IWN i będę temu poświęcał kolejne badania w przyszłości. Warto również zauważyć, że liczba kontraktów przypadająca na jeden węzeł w sieci (iloczyn $|S|/N_V$ oraz $N_{C}/|S|$) nie wpływa, w badanym zakresie, znacząco na wydłużenie średniego oraz maksymalnego czasu transmisji. Natomiast ma znaczący wpływ na odchylenie czasu transmisji od wartości oczekiwanej (Rysunek A.6). Różnica jest obserwowana zarówno dla małej jak i dużej liczby przeskoków. Mimo iż kolizje w trakcie planowania są rzadkie i nie wpływają znacząco na wartości graniczne to ich obecność wpływa negatywnie na jitter sprawiając, że wartość czasu transmisji staje się nie deterministyczna.

Istotnym z punktu widzenia pracy sieci elementem węzła jest bufor transmisji. Dla optymalnej wielkości bufora, przepływność sieci wzrasta w sposób znaczący [92]. Zadaniem bufora jest akumulacja ramek generowanych w węźle przez wyższe warstwy (w praktyce warstwa aplikacji, AL i łącza danych DL) oraz ramek generowanych przez inne węzły, w których przekazywaniu, do węzła docelowego, uczestniczy. Sieci IWN pracujące w oparciu o węzły o

ograniczonych zasobach, muszą uwzględniać ograniczenie wielkości bufora. W przypadku węzłów SENSOR NODE 1C oparty o typowy układ z rodziny CC13x2 (TI) rozsądna wielkość bufora to maksymalnie 64 ramki, co przy maksymalnej długości ramki 127B odpowiada około 10kB pamięci RAM. W praktycznych implementacjach stosu komunikacyjnego (np. komercyjny TronixIWN, opracowany w ramach projektu badawczego którego jestem kierownikiem), bufor przechowuje zarówno DPDU w postaci gotowych do transmisji bajtów (zakodowanej zgodnie z formatem ISA100.11a pełnej ramki na warstwie 2, wraz z nagłówkami), jak również zdekodowane wartości potrzebne do poprawnej obsługi elementu znajdującego się w buforze. Przykładowe zmienne przechowywane dodatkowo w stosunku do DPDU to np. priorytet (kolejność elementów), znacznik czasu pozwalający na usunięcie elementu w przypadku utraty ważności lub klucz pozwalający na identyfikację właściwego połączenia l_i^i . Ramka umieszczona w buforze oczekuje na właściwą szczelinę czasową z połączeniem l_i^i , które pozwala na przesłanie ramki do kolejnego węzła na trasie $r_k^{\{s,d\}}$. Długość oczekiwania, czyli czas, w którym ramka znajduje się w buforze jest ściśle powiązany z generowanym planem. Dążymy do tego, aby wielkość bufora ramek pozwoliła na akumulację wszystkich ramek bez konieczności ich odrzucania.

Na rysunku A.7 (góra) przedstawiam procentowy rozkład czasu pracy węzłów w funkcji liczby ramek w buforze transmisji dla ruchu pochodzącego od kontraktów cyklicznych. Wykres został opracowany w oparciu o dane zgromadzone w zbiorze "result_node_agg", zgodnie z opisem w rozdziale 4. Suma liczby szczelin czasowych została znormalizowana do 100%. Wykres prezentuje procentowy czas pracy węzłów sieci przy wskazanej liczbie ramek znajdujących się w buforach. Zgodnie z przyjętą konwencją prezentacji rezultatów, dla przykładowej sieci składającej się z dwóch węzłów, gdzie jeden z węzłów przez 50% czasu pracy sieci ma 1 ramkę w buforze a drugi węzeł przez 100% czasu nie ma żadnej ramki. Na wykresie będzie 25% czasu pracy przy jednej ramce i 75% czasu przy braku ramek. Przykładową wartość 1e-3% przy 5 ramkach należy interpretować w sposób następujący, przez 1 promil czasu pracy węzłów w buforach znajdowało się 5 ramek. Dla wszystkich badanych algorytmów rozkład w buforze jest zbliżony. Badany wektor testowy dla wszystkich konfiguracji potrzebuje 20 elementowy bufor, aby obsłużyć ruch pochodzący od kontraktów cyklicznych.

Na rysunku A.7 (dół) przedstawiam analogiczny rozkład dla grupy algorytmów dedykowanych dla ruchu sekwencyjnego. W przypadku algorytmu AMUS zakładana długość bufora jest niewystarczająca. W przypadku AMUS(1) długość bufora jest wystarczająca, ale zostanie wykorzystany w całości. W rozdziale porównuję algorytmy niezależnie, natomiast w przypadku rozwiązania docelowego, planowany będzie zarówno ruch cykliczny i sekwencyjny.

W takim przypadku dla badanego wektora testowego wskazana długość nie będzie wystarczająca.

Na rysunkach B.15 oraz B.16 przedstawiam rozkład czasu pracy w funkcji liczby ramek w buforze z podziałem na rodzaj trasy i liczbę kanałów. W przypadku obu wariantów wyraźna jest zależność, w której dla trasy o większej liczbie przeskoków (trasa 2) wymagana długość bufora jest krótsza. Dodatkowo w przypadku ruchu sekwencyjnego widać, że większa liczba kanałów wymusza większy rozmiar bufora. Analiza algorytmu AMUS wskazuje, że takie zachowanie wynika z konstrukcji samego algorytmu, który umieszcza kolejno generowany ruch zawsze zaczynając od pierwszej szczeliny, tym samym w pełni wykorzystuje pełne spektrum dostępnych kanałów nie wykorzystując jednocześnie pełnego spektrum dostępnych szczelin. Zmiana sposobu generowania planu może efektywnie przyczynić się do ograniczenia rozmiaru bufora.

5.3 Algorytm CAS-PERIOD

Prezentowany w tym rozdziale autorski algorytm CAS-PERIOD stanowi element całościowego rozwiązania postawionego w 3.2.4 problemu. Algorytm dedykowany jest do wykonania planu szczelin czasowych TDMA/FDMA w zakresie kontraktów dotyczących ruchu cyklicznego. Przyjmując zbiór kontraktów cyklicznych *CP*, zadaniem algorytmu jest znalezienie zbioru *SCH* zawierającego harmonogramy dla wszystkich węzłów sieci $SCH_{\{i\}}^{CP} = \{el \mid el \left(p, c_k^{\{s,d\}}, l_j^i, ts, ch\right) \land c_k^{\{s,d\}} \in CP\};$ spełniające parametry zdefiniowane w kontrakcie $c_k^{\{s,d\}} \in CP$, w szczególności warunek miękki mk_2 odnoszący się do wymaganego maksymalnego czasu transmisji pakietu.

5.3.1 Definicja algorytmu CAS-PERIOD i przyjęte założenia

Analizując wyniki otrzymane z przeprowadzonego badania algorytmów planowania ruchu cyklicznego oraz potrzeby wynikające z definicji postawionego problemu 3.2.4, należy sformułować następujące minimalne założenia dla optymalnego algorytmu planowania ruchu cyklicznego:

- 1. Średni czas wykonania planu dla sieci składającej się ze 100 węzłów dla wariantu 2 generowania topologii sieci musi być poniżej 1s dla GATEWAY_4B. Przyjmuję, że całościowe rozwiązanie zdefiniowanego problemu musi nastąpić w maksymalnym czasie wynoszącym 30s lub 60s, a do rozwiązania konieczne będzie przeprowadzenie kilkunastu prób planowania. Podane czasy są typowymi wartościami dla formowania sieci (np. w procesie JOINING) lub zawarcia kontraktu.
- 2. Średni poziom zaplanowania ruchu cyklicznego dla sieci składającej się ze 100 węzłów dla wariantu 2 generowania topologii przy 16 dostępnych kanałach musi wynosić co najmniej

25% dla badanego wektora testowego (50% licząc ruch cykliczny osobno). Wartość oszacowałem obserwując zachowanie algorytmów planowania EDF/LLF dla przypadków w których nie następuje odrzucanie kontraktów z względu na ograniczenie wynikające z przyjętego maksymalnego okresu planowania ($2^{20} - 1$), patrz rozdział 5.2.1.

- 3. Odchylenie czasu transmisji pakietu od wartości oczekiwanej musi wynosić 0 (składowa jitter wynikająca bezpośrednio z planu). Przyjęcie tego założenia podyktowane jest koniecznością ograniczenia jitter w zastosowaniach IWN. Zakładam że jedynym elementem jitter będzie wpływ synchronizacji węzłów, natomiast wpływ planu zostanie wyeliminowany całkowicie. Zagadnienia dotyczące synchronizacji węzłów sieci w szczególności przy założeniu gwałtownych zmian temperatury zostały omówione w rozdziale 2.2, gdzie proponuję również autorskie metody minimalizacji ich wpływu.
- 4. Maksymalna wartość czasu transmisji pakietu musi pozostać na podobnym poziomie (w stosunku do SEP_EDF_FIFO, rozdział 5.2.2). Założenie podyktowane jest szczególnym znaczeniem maksymalnego czasu transmisji w zastosowaniach IWN, gdzie przyjąłem algorytmy pełnego planowania EDF'LLF jako punkt odniesienia.
- 5. Wymagana długość bufora transmisji musi pozostać na podobnym poziomie (w stosunku do wyników SEP_EDF_FIFO, rozdział 5.2.2). Wielkość bufora transmisji w węźle IWN jest ograniczona małą ilością dostępnej pamięci RAM. Analizując dostępne układy mikrokontrolerów, w szczególności typowe układy TI (CC1352R) należy przyjąć że długość bufora wynosząca 64 (dla ramki o długości 127 bajtów) jest optymalna.



Rysunek 5.5 Rozkład aktywności w szczelinach czasowych dla przypadku potencjalnej kolizji

Na rysunku 5.5 przedstawiam sytuację, w której trzy węzły przesyłają jedną ramkę w ramach trzech kontraktów o różnym okresie uwalniania ruchu. Okresy kontraktów to 6, 12 i 8 szczelin dla odpowiednio węzłów 1, 2 i 3. Algorytmy, w których budowanie optymalnego planu odbywa się w okresie wynikającym z najkrótszego okresu powtarzania LCM(6,8,12) = 24 szczeliny, rozstrzygają kolizję w momencie jej pojawiania się w konkretnej szczelinie. Takie planowanie pozwala na optymalne wykorzystanie wszystkich dostępnych szczelin czasowych, natomiast wprowadza szereg problemów w implementacji (rozdział 5.2.1) oraz wprowadza składową jitter związaną z planem szczelin (rozdział 5.2.2). U podstaw wszystkich zdefiniowanych



Rysunek 5.6 Rozkład aktywności w szczelinach czasowych po rozstrzygnięciu kolizji przy planowaniu w pełnym okresie LCM

problemów w realizacji omawianych algorytmów, jest konieczność planowania w bardzo długim okresie wynikającym z LCM. Długość LCM jest ściśle skorelowana z czasem wykonywania algorytmu a zależność jest wykładnicza. Dla części zastosowań rozwiązaniem długości czasu planowania może być sukcesywne uzupełnianie planu w elementach sieci, w miarę jak kolejne odcinki czasu są zaplanowane. Ograniczenie okresu planowania do wycinka całego zakresu obejmującego 2¹⁶ szczelin pozwala na realizację planu w pożądanym czasie wynikającym z założeń, natomiast wymagałoby aktualizacji planu we wszystkich węzłach sieci co około 10min co jest nieakceptowalne zważywszy na to, że w niektórych aplikacjach, zakładany ruch jest sporadyczny, np. czujniki zagrożenia, a urządzenia są zasilane bateryjnie. Dystrybucja tak dużej ilości ruchu ściśle związanego z rekonfiguracją sieci pochłonie energię potrzebną do realizacji zadań istotnych z punktu widzenia użytkownika.

Prezentowana na rysunku 5.5 kolizja może być rozstrzygnięta w zależności od relacji między węzłami 1 i 2. Jeżeli kolizja jest 2'ego stopnia należy zmienić kanał dla jednego z węzłów w tej konkretnej szczelinie (Rysunek 5.6, górny). Jeżeli kolizja jest 1'ego stopnia dla jednego z węzłów transmisję w tej konkretnej szczelinie należy przesunąć do następnej (lub kolejnej) szczeliny (Rysunek 5.6, dolny). Zauważmy, że kolizję możemy rozstrzygnąć w sposób analogiczny, ale poprzez zastosowanie zmiany dla wszystkich powtórzeń okresu.

Na rysunku 5.7 zmiana kanału nastąpiła we wszystkich aktywnych szczelinach węzła 3, natomiast w przypadku kolizji 1'ego stopnia przesunięcie szczeliny, mimo że kolizja występuje jedynie w 20 szczelinie nastąpiło również analogicznie we wszystkich szczelinach zgodnie z okresem powtarzania. Zauważmy, że taki schemat działania pozwala na ograniczenie liczby elementów w planie. Zamiast jednego długiego planu, o okresie planowania wynikającym z LCM, należy stworzyć 3 plany o okresach planowania odpowiednio 6, 8, 12 i przekazać je do

odpowiednich węzłów. Do zapisania planu w pełnym okresie LCM(6,8,12) (Rysunek 5.6) potrzebnych jest 9 elementów planu (4, 2 i 3 elementy dla kolejnych kontraktów), podczas gdy do zapisu planu o trzech okresach powtarzania 6, 8, 12 (Rysunek 5.7) potrzebne są zaledwie 3 elementy. Przykład wygenerowanych planów dla przedstawionych wariantów prezentuję w tabeli B.25.



Rysunek 5.7 Rozkład aktywności w szczelinach czasowych po rozstrzygnięciu kolizji przy planowaniu w minimalnym okresie

Taki sposób planowania prowadzi również do drugiej kluczowej zmiany a mianowicie eliminuje jitter związany z planem. W przypadku węzła 3 w klasycznym rozwiązaniu transmisja jest realizowana 2 razy w czwartej szczelinie okresu powtarzania a raz w piątej szczelinie. Zmiana szczeliny powoduje zmianę czasu dostarczenia pakietu, co w przypadku zastosowań IWN jest zjawiskiem nieporządnym. W proponowanym rozwiązaniu transmisja następuje zawsze w piątej szczelinie co skutecznie eliminuje jitter związany z planowaniem.

Każdy element $el = (p, c_k^{\{s,d\}}, l_j^i, ts, ch)$ planu *SCH* tworzy ciąg arytmetyczny, gdzie *ts* jest pierwszym wyrazem ciągu a *p* jest różnicą ciągu. Warunkiem koniecznym do zaistnienia kolizji między dwoma elementami planów jest istnienie takiego $x, y \in \mathbb{N}_0$, że suma obu ciągów jest równa.

$$ts_1 + x \cdot p_1 = ts_2 + y \cdot p_2 \tag{30}$$

Po przekształceniu otrzymujemy:

$$x \cdot p_1 + y \cdot p_2 = ts_2 - ts_1 \tag{31}$$

Równanie możemy rozwiązać stosując tożsamość Bézouta. Współczynniki Bézouta oraz największy wspólny dzielnik dla pary liczb możemy efektywnie znaleźć poprzez zastosowanie

rozszerzonego algorytmu Euklidesa. Warunkiem istnienia rozwiązania jest następująca podzielność:

$$(ts_2 - ts_1) \mid GCD(p_1, p_2)$$
(32)

gdzie GCD jest największym wspólnym dzielnikiem (ang. greatest common divisor) pary liczb całkowitych.

Zadaniem opracowanego autorskiego algorytmu *Conflict Aware Scheduling* (CAS) jest takie utworzenie *N* planów, aby ich jednoczesna realizacja była wolna od kolizji zarówno 1'ego jak i 2'ego stopnia. Przyjmijmy, że dysponujemy planami $SCH_{\{*,p'\}} \in \{el|el \in SCH \land el. p = p'\}$ oraz $SCH_{\{*,p''\}} \in \{el|el \in SCH \land el. p = p''\}$ zawierającymi elementy planów dla wszystkich węzłów *, o wspólnym okresie powtarzania p', p'', takimi, że są wolne od kolizji 1'ego i 2'ego stopnia wyłącznie w odniesieniu do elementów tego samego planu. Plan $SCH_{\{*,p''\}}$ oraz $SCH_{\{*,p''\}}$ mogą być wykonywane wspólnie jeżeli każda para elementów $SCH_{\{*,p'\}} \times SCH_{\{*,p''\}}$ nie spełnia warunku koniecznego istnienia kolizji, wzór (32).

Zdefiniujmy dwa nowe pojęcia określające swobodę planowania kontraktu: swoboda $(laxity_k^{\{s,d\}})$ i stopień swobody $(laxity_ratio_k^{\{s,d\}})$ planowania. Swoboda planowania dla kontraktu k (33) jest liczbą szczelin czasowych stanowiących różnicę pomiędzy wymaganym czasem dostarczenia pakietu a liczbą szczelin jakie zajmie transmisja pakietu przy braku kolizji. Swoboda planowania określa liczbę szczelin o ile może być wydłużona transmisja, aby wypełnić wymaganie QoS dotyczące czasu transmisji. Liczba zero oznacza brak możliwości przesunięcia planu, każde przesunięcie skutkuje naruszeniem wymagania QoS i uznaniem kontraktu za niemożliwy do zaplanowania. Im liczba jest większa tym istnieje więcej szczelin, które mogą zostać wykorzystane do przesunięcia transmisji w przypadku kolizji. Stopień swobody stanowi iloraz tych dwóch wielkości. Stopień swobody równy 1 oznacza brak swobody w planowaniu, im stosunek jest większy tym większa swoboda w planowaniu.

$$laxity_{k}^{\{s,d\}} = cp_{k}^{\{s,d\}}.$$
 Deadline - $length(h, f, ch)$ (33)

$$laxity_ratio_{k}^{\{s,d\}} = \frac{cp_{k}^{\{s,d\}}.\text{Deadline}}{length(h,f,ch)}$$
(34)

Kluczowym elementem potrzebnym do obliczenia dwóch zdefiniowanych miar jest długość wyrażona w liczbie szczelin czasowych jaką teoretycznie zajmie transmisja pakietu przy braku kolizji. Długość jest funkcją ilości przeskoków na trasie (h), ilości ramek (f) oraz liczby dostępnych kanałów (ch). Wzór uwzględnia dodatkowo w stosunku do wzoru (20) liczbę dostępnych kanałów. Tak zdefiniowana długość transmisji jest minimalną długością transmisji w przypadku, gdy dla każdej transmisji na trasie potencjalnie występuje zawsze kolizja 2'ego

stopnia i do jej uniknięcia konieczna jest transmisja na innym wolnym kanale. Odpowiada to sytuacji, w której węzły uczestniczące w transmisji są bezpośrednimi sąsiadami (Rysunek A.2, a oraz c). Graf połączeń który jest grafem pełnym dla dowolnej trasy spełnia ten warunek. Taki przypadek w praktyce występuje najczęściej i taki obserwuje dla obu tras kontraktów zarówno dla wariantu 1 oraz wariantu 2 generowania topologii sieci.

$$length(h, f, ch) =$$

$$\begin{cases} h \cdot f & h \leq 2 \\ h + 2 \cdot (f - 1) & h > 2 \wedge ch \geq \left\lceil h/_2 \right\rceil & (35) \\ \left\lfloor f/_{ch} \right\rfloor \cdot h + 2 \cdot (ch - 1) & h > 2 \wedge ch < \left\lceil h/_2 \right\rceil \wedge f \mod ch = 0 \\ \left\lfloor f/_{ch} \right\rfloor \cdot h + 2 \cdot [(f \mod ch) - 1] & h > 2 \wedge ch < \left\lceil h/_2 \right\rceil \wedge f \mod ch \neq 0 \end{cases}$$

W praktyce przy długich trasach może się zdarzyć sytuacja, w której faktyczna transmisja jest krótsza od tak obliczonej ze względu na możliwość transmisji ramki w jednej szczelinie, gdy nie ma kolizji 2'ego stopnia. Skrajnym przypadkiem jest sytuacja, w której kolizja 2'ego stopnia nie występuje nigdy i konieczne jest unikanie jedynie kolizji 1'ego stopnia, wzór (20). Taka sytuacja ma miejsce, gdy połączenia pomiędzy węzłami na grafie połączeń, ograniczają się jedynie do analizowanej trasy (Rysunek A.2, b). Podsumowując, należy stwierdzić że wzór (20) określa najmniejszą liczbę szczelin potrzebnych do transmisji pakietu, podczas gdy wzór (34) stanowi uogólnienie i uwzględnia dodatkowo liczbę dostępnych kanałów. Wzór (34) pozwala na oszacowanie oczekiwanej długości transmisji przy znajomości jedynie parametrów wynikających z kontraktu bez znajomości topologii sieci i faktycznego przebiegu trasy $r_k^{\{s,d\}}$.

Algorytm planowania został podzielony na dwie fazy. Faza grupowania (Algorytm 3) w której tworzone są grupy kontraktów, dla których zostaną przygotowane osobne plany oraz fazę planowania (Algorytm 4) w której opracowany jest plan w taki sposób, aby był wolny od kolizji z już utworzonymi planami. Każdy utworzony plan ma własny okres powtarzania. Ewaluacji poddałem pięć wariantów grupowania kontraktów:

- 1. CAS_LATENCY grupę tworzy pojedynczy kontrakt, grupy są uporządkowane narastająco po $cp_k^{\{s,d\}}$. Deadline
- 2. CAS_LAXITY1 grupę tworzy pojedynczy kontrakt, grupy są uporządkowane narastająco po $laxity_k^{\{s,d\}}$
- 3. CAS_LAXITY2 grupę tworzy pojedynczy kontrakt, grupy są uporządkowane narastająco po $laxity_ratio_k^{\{s,d\}}$

- 4. CAS_PERIOD_ASC grupę tworzą kontrakty posiadające wspólny okres $cp_k^{\{s,d\}}$. Period, grupy są uporządkowane narastająco po długości okresu.
- 5. CAS_PERIOD_DESC grupę tworzą kontrakty posiadające wspólny okres $cp_k^{\{s,d\}}$. Period, grupy są uporządkowane opadająco po długości okresu.

Algorytm 3 CAS-PERIOD – faza grupowania

inpu	ut : G; RT; CP
outj	put : SCH /* zbiór harmonogramów dla wszystkich węzłów sieci */
1	$CP_GROUPS \leftarrow prepareContractGroups(CP)$
2	Posortuj CP_GROUPS zgodnie z przyjętym wariantem
3	for each $CP' \in CP_GROUPS$ do
4	<pre>schedulePeriodicCAS(CP'); /* wykonaj pojedynczy plan */</pre>
5	end

Algorytm 4, dla każdej grupy kontraktów CP', obejmującej od 1 do N kontraktów, wykonuje plan o wspólnym okresie powtarzania elementów p. Funkcja releaseAtTs(ts, BUF, CP', RT) dla wszystkich kontraktów w zbiorze CP' umieszcza transmisje $\tau_k^{\{i,j\}}$ w buforze BUF, gdzie i, jsą węzłami pierwszego połączenia l_j^s na trasie $r_k^{\{s,d\}} \in RT$. Transmisja jest uwalniana jeżeli numer szczeliny ts jest równy numerowi szczeliny uwolnienia $ts_release_n^k$, obliczonej dla kontraktu $cp_k^{\{s,d\}}$ zgodnie ze wzorem (3). Planowanie transmisji $\tau_k^{\{i,j\}}$ jako pojedynczego elementu planu SCH prezentuję jako Algorytm 5, który wywoływany jest jako funkcja scheduleElement(G; $\tau_{k}^{\{i,j\}}$; ACT; BUF; ts; p). Plan dla drugiej i kolejnej grupy wykonywany jest w taki sposób, aby nie było kolizji z elementami już wykonanych planów. Zbiór ACT określa aktywności węzłów w przestrzeni ograniczonej trzema wymiarami: węzeł, szczelina czasowa, kanał. Stosuję zapis indeksowania $ACT[v_i][ts][ch]$ odnoszący się do pojedynczego elementu w zbiorze oraz zapis $ACT[v_i][ts]$ który przyjmuje true jeżeli istnieje aktywność w dowolnym elemencie pominiętego wymiaru (w tym przypadku kanale). Funkcja checkScheduleCollision(el; p; SCH) zdefiniowana jako Algorytm 6, zwraca rodzaj kolizji lub sygnalizuje jej brak. Jeżeli istnieje kolizja 1'ego stopnia z elementami innych planów, algorytm postępuje dokładnie tak jak w przypadku kolizji w ramach generowanego planu i próbuje zaplanować kolejną transmisję znajdującą się w buforze. Analogicznie, gdy sygnalizowana jest kolizja 2'ego stopnia z innymi planami, algorytm próbuje zaplanować transmisję na innym kanale. Gdy nie ma kolizji, element jest dodawany do planu nadawcy $SCH_i = SCH_i \cup \{el\}$ a zaplanowana transmisja $au_k^{\{i,j\}}$ jest przesuwana wzdłuż trasy $r_k^{\{s,d\}}$ (zmiana i,j na odpowiadające kolejnemu przeskokowi) lub usuwana z BUF jeżeli odbiorca transmisji

odpowiada węzłowi docelowemu kontraktu. Każda próba zaplanowania aktywności poprzedzona jest weryfikacją czy element aktualnie tworzonego planu nie wchodzi w kolizję z innymi planami w jednym z ich powtórzeń.

Algorytm 4 CAS-PERIOD – faza planowania

input		: G; RT; CP'
outpu	ıt	: SCH /* zbiór harmonogramów dla wszystkich węzłów sieci */
1 t	:s ←	$0; ts_{abs} \leftarrow 0; BUF \leftarrow \{ \}; ACT \leftarrow \{false\}$
2 p) ←	<pre>LCM([CP'.period ÷ ts]) /* oblicz okres planowania */</pre>
3 d	lo	
4	if	$ts_{abs} \leq p \text{ do } releaseAtTs(ts, BUF, CP', RT); /* umieść \tau_k^{\{i,j\}} \le BUF * / CP'$
5	fc	or each $\tau_k^{\{i,j\}} \in BUF$ do oblicz <i>ts_deadline</i> ^k /* wzór (4) */
6	P	osortuj $ au_k^{\{i,j\}} \in BUF$ narastająco po $ts_deadline_n^k \ /^* EDF \ ^*/$
7	fc	or each $\tau_k^{\{i,j\}} \in BUF$ do /* dla każdej transmisji */
8		scheduleElement(G; $\tau_k^{\{i,j\}}$; ACT; BUF; ts; p)
9	E	nd
10	ts	$S_{abs} \leftarrow ts_{abs} + 1$; $ts \leftarrow ts_{abs} \mod p /*$ następna szczelina czasowa */
11 v	while	$e(ts_{abs} \le p \lor BUF \ne \emptyset)$

Zwracam uwage, że zaprezentowany algorytm ma konstrukcję ogólna pozwalająca na dowolne kreowanie grup kontraktów. Jeżeli wygenerujemy grupę składającą się ze wszystkich planowanych kontraktów algorytm zachowa się dokładnie tak jak badany w 5.2 algorytm EDF FIFO, rezultatem wykonania będzie pojedynczy harmonogram o okresie powtarzania wynikającym z LCM dla okresów wszystkich kontraktów. Z drugiej strony, jeżeli wygenerujemy grupy kontraktów, gdzie każda grupa będzie się składać z dokładnie jednego kontraktu (wariant CAS LATENCY, CAS LAXITY1/2) otrzymamy harmonogram dla każdego z kontraktów, gdzie okres powtarzania harmonogramu będzie równy okresowi kontraktu. Skrajny przypadek opisany jako pierwszy posiada teoretyczną możliwość do akumulacji większej ilości ruchu natomiast obarczony jest wszystkimi problemami jakie opisałem podczas analizy wyników tej grupy algorytmów. Drugi z granicznych przypadków posiada mniejszą zdolność akumulacji ruchu natomiast konstrukcja wykonanych planów jest optymalna z punktu widzenia szybkości jej wykonania oraz liczby elementów planu, tym samym jest optymalna z punktu widzenia jej dystrybucji i wykorzystania w węzłach sieci. Pomiędzy tymi dwoma granicznymi przypadkami istnieje cały szereg wariantów liczebności i zawartości grup który może być dopasowany do konkretnej implementacji mając na uwadze opisany kompromis pomiędzy zdolnością planu do akumulacji większej ilości ruchu a korzyści wynikających z jego wielkości i szybkości wygenerowania.

Algorytm 5 CAS – planowanie transmisji

inpu	ut		:	$G; \tau_k^{\{i,j\}}; ACT; BUF; ts; p$			
out	put	t	:	SCH /* zbiór harmonogramów dla wszystkich węzłów sieci */			
1	if	٦Æ	4 <i>C</i> 7	$T[v_i][ts] \land \neg ACT[v_j][ts]$ then /* brak kolizji 1'ego st. */			
2		V	N <	– $NE_i \cup NE_j$ /* zbiór sąsiadów dla v_i , v_j */			
3	for each $ch \in N_{CH}^i$ do						
4			If	$\forall v_n \in VN(\neg ACT[v_n][ts][ch])$ then /* brak kolizji 2'ego st. */			
5				$el \leftarrow \left(p, c_k^{\{s,d\}}, l_j^i, ts, ch\right)$			
6				$result \leftarrow checkScheduleCollision(el, p, SCH)$			
7				<pre>if result == FIRST_DEGREE then /* kolizja 1'ego stopnia */</pre>			
8				return /* węzły są już aktywne w elemencie innego planu */			
9				else if result == SECOND_DEGREE then /* kolizja 2'ego stopnia */			
10				continue /* kanał jest zajęty przez element innego planu */			
11				else			
12				$SCH_i = SCH_i \cup \{el\}; ACT[v_i][ts][ch] \leftarrow true, ACT[v_j][ts][ch] \leftarrow true$			
13				Przesuń transmisję $ au_k^{\{i,j\}}$ ramki wzdłuż trasy $r_k^{\{s,d\}}$			
14				return			
15				end			
16			er	nd			
17		eı	nd				
18	er	nd					

W przypadku sieci IWN w szczególności tam, gdzie węzły zasilane są bateryjnie, urządzenia są aktywne jedynie w ściśle zdefiniowanym ułamku czasu (ang. duty-cycling), tym samym teoretyczna zdolność planu do akumulacji większej ilości ruchu nie jest kluczowa dla tej grupy urządzeń. Poddane ewaluacji warianty grupowania reprezentują przykłady grupowania, które należy zaliczyć do drugiego ze zdefiniowanych granicznych przypadków dla tego algorytmu i zostaną zestawione z pierwszym przypadkiem granicznym równoważnym algorytmowi EDF_FIFO.

Znalezienie kolizji pomiędzy elementem tworzonego planu a innymi już utworzonymi planami (Algorytm 6), polega na sprawdzeniu warunku podzielności (32) dla każdego elementu utworzonych planów. Każdy element planu zawiera informację o połączeniu l_j^i , czyli węzłach nadawcy v_i i odbiorcy v_j . Dla każdego elementu planu sprawdzana jest możliwość wystąpienia kolizji 1'ego (*has1stCollision(el,el'*)) lub 2'ego stopnia (*has2ndCollision(el,el'*)) na podstawie wzajemnej relacji między węzłami w obu elementach planu. Warunki dotyczące węzłów v_i oraz v_j są konieczne aby kolizja była możliwa lecz nie wystarczające. Drugim

warunkiem jest sprawdzenie podzielności zgodnie z równaniem (32). Spełnienie obu warunków jest wystarczające do stwierdzenia kolizji obu elementów i jest sygnalizowane w postaci wartości parametru $COLL \in \{FIRST_DEGREE, SECOND_DEGREE, NONE\}$. Należy zwrócić uwagę, że największy wspólny dzielnik (GCD) może być policzony raz dla każdego z elementów planu $el \in SCH_{\{*,p'\}}$, które posiadają ten sam okres powtarzania (p') i są dla dowolnego węzła (*). Jedynie warunki podzielności i warunki dotyczące możliwości wystąpienia konkretnego rodzaju kolizji należy sprawdzać dla każdego elementu planu.

Algorytm 6 CAS – znajdowanie kolizji między planami

: el /* sprawdzany element nowego harmonogramu */ input /* okres powtarzania nowego harmonogramu */ р /* zbiór harmonogramów dla wszystkich węzłów sieci */ SCH output : COLL /* COLL ∈ {FIRST_DEGREE, SECOND_DEGREE, NONE}*/ for each $SCH_{\{*,p'\}} \in \{el | el \in SCH \land p = p'\}$ do /* wspólny okres p' * / p' = p'1 2 $gcd \leftarrow GCD(p,p')$ 3 for each $el' \in SCH_{\{i',p'\}}$ do /* dla każdego elementu planu */ if has1stCollision(el, el') then /* możliwa kolizja 1'ego stopnia */ 4 If $(el.ts - el'.ts) \mod gcd == 0$ then $COLL \leftarrow FIRST_DEGREE$ 5 else if has2ndCollision(el, el') then /* możliwa kolizja 2'ego stopnia */ 6 7 If $(s.ts - s'.ts) \mod gcd == 0$ then $COLL \leftarrow SECOND_DEGREE$ 8 else $COLL \leftarrow NONE$ 9 10 end 11 end 12 end

5.3.2 Ewaluacja wyników planowania algorytmu CAS-PERIOD

Ewaluacji poddano wszystkie proponowane warianty grupowania: CAS_LATENCY, CAS_PERIOD_ASC i CAS_PERIOD_DESC, CAS_LAXITY1 oraz CAS_LAXITY2. Wszystkie warianty odpowiadają sytuacji, w której tworzone plany mają najmniejszą możliwą długość wynikającą wprost z okresu kontraktów. Jako punkt odniesienia w ewaluacji wykorzystano algorytm SEP_EDF_FIFO który stanowi optymalny wariant badanego algorytmu EDF.

Konwencja prezentacji rezultatów jest analogiczna do użytej w rozdziale 5.2.1. Suma czasów wykonania wszystkich planów w zadanym wektorze testowym dla wszystkich badanych wariantów algorytmu CAS-PERIOD jest równa 18094,52s i 31164,25s dla GATEWAY 4B



Rysunek 5.8 Czas wykonania (rząd 1) oraz poziom zaplanowania (rząd 2) w funkcji liczby węzłów dla wariantu 1 (kolumna 1) i wariantu 2 (kolumna 2) generowania topologii sieci – GATEWAY 4B

oraz 2837,25s i 4983,98s dla stacji roboczej odpowiednio dla wariantu 1 i 2 generowania topologii sieci. W dodatku B przedstawiam parametry rozkładów parametrów wygenerowanych planów dla wariantu 1 i 2 generowania topologii sieci:

- Czas wykonania planu Tabela B.26, Tabela B.34, Tabela B.30, Tabela B.35
- Długość planu Tabela B.27, Tabela B.31
- Liczba elementów w planie Tabela B.28, Tabela B.32
- Poziom zaplanowania Tabela B.29, Tabela B.33

Na rysunku 5.8 prezentuję czas wykonania oraz poziom zaplanowania dla topologii sieci generowanej w wariancie 1 oraz 2 jako rozkład wartości uzyskanych dla wszystkich elementów wektora testowego. Widać wyraźną różnicę na korzyść opracowanego algorytmu CAS-

PERIOD w obu wariantach i porównując obie miary. Czas wykonania spadł o dwa rzędy wielkości dla małych sieci (10 węzłów), ponad jeden rząd wielkości dla sieci zgodnej z założeniami (100 węzłów) i około jeden rząd dla największych sieci składających się z 200 węzłów. W sposób istotny zwiększył się poziom zaplanowania, prawie 1,5 razy dla małych sieci oraz ponad 3 krotnie dla zakładanej wielkości sieci oraz dużych sieci. Warto zwrócić uwagę, że w przypadku wariantu 2 dla CAS_LATENCY, nawet dla największej sieci (200 węzłów), 25% obserwacji ma poziom zaplanowania większy niż 48,79% a w przypadku CAS_LAXITY1 aż 48,96%. Warto w tym miejscu przypomnieć, że zgodnie z przyjętą konwencją zapisu, maksymalnym poziomem zaplanowania jest 50%, ponieważ w wektorze testowym znajdują się zarówno kontrakty cykliczne jak i sekwencyjne w stosunku 50/50%, co rozwinąłem w rozdziale 5.2.1.

W tabeli B.36 oraz B.37 zestawiłem parametry rozkładu poziomu zaplanowania z podziałem jedynie na liczbę węzłów dla odpowiednio wariantu 1 i 2 generowania topologii sieci. Natomiast, w tabelach B.38 oraz B.39 prezentuję parametry rozkładu poziomu zaplanowania z podziałem dodatkowo na liczbę dostępnych kanałów [2,8,16] oraz trasę [1,2]. Porównując badane warianty algorytmu CAS, największy średni poziom zaplanowania dla wszystkich wielkości sieci oraz w obu wariantach generowania topologii sieci zapewnia CAS_LATENCY i drugi w kolejności CAS_LAXITY1. Czas wykonania w przypadku CAS_LATENCY oraz CAS_LAXITY1/2 jest na zbliżonym poziomie i jest nieznacznie krótszy od CAS_PERIOD_DESC co wynika wprost z konstrukcji algorytmu i braku konieczności dodatkowego grupowania, różnica jest natomiast znikoma. Algorytm CAS_PERIOD_ASC uzyskał krótsze czasy wykonania, natomiast różnica wynika z dużo mniejszego poziomu zaplanowania (ilości zaplanowanych kontraktów). Przebieg algorytmów CAS_PERIOD_* jest identyczny, jedyna różnica wynika z przyjętego kierunku sortowania.

Na rysunku A.8 oraz A.9 przedstawiłem średni poziom zaplanowania [%] w funkcji całkowitej liczby kontraktów podlegających planowaniu odpowiednio dla wariantu 1 i wariantu 2 generowania topologii sieci. Natomiast, na rysunku B.19 oraz B.20 prezentuję rozkład poziomu zaplanowania w funkcji liczby węzłów w sieci. Wyniki prezentuję w podziale na liczbę kanałów [2,8,16] oraz trasę [1,2].

W przypadku trasy 1 (z najkrótszą liczbą przeskoków) opracowany algorytm CAS_PERIOD uzyskuje rezultaty kilkukrotnie lepsze od algorytmów z rodziny EDF/LLF. Przy 16 dostępnych kanałach algorytm CAS_PERIOD w wariancie CAS_LATENCY uzyskuje średnio 46,65% dla 30 węzłów i 47,16% dla 90 węzłów w wariancie 1 generowania topologii sieci oraz 49,37% dla 50 węzłów i 49,31% dla 200 węzłów w wariancie 2. Podczas gdy wyniki EDF/LLF są poniżej 15%. W przypadku trasy 2 (z największą niezawodnością) opracowany algorytm

CAS_PERIOD uzyskuje również zdecydowanie lepsze rezultaty, przy czym skala różnicy (16 kanałów) oscyluje od około 3 krotnej przewagi do około 2 krotnej. Przy 16 dostępnych kanałach algorytm CAS_PERIOD w wariancie CAS_LATENCY uzyskuje średnio 36,27% dla 30 węzłów i 34,41% dla 90 węzłów w wariancie 1 generowania topologii sieci oraz 37,01% dla 50 węzłów i 29,67% dla 200 węzłów w wariancie 2. Podczas gdy wyniki EDF/LLF są poniżej 15%. Zdecydowana przewaga algorytmu CAS_PERIOD maleje w przypadku małej liczby kanałów ale dalej pozostaje widoczna na korzyść CAS_PERIOD w każdym z badanych wariantów. Mimo, iż algorytm CAS_PERIOD ma mniejszą zdolność do akumulacji ruchu (co wynika wprost z jego konstrukcji) świetnie wpisuje się parametry ruchu obserwowane w sieciach IWN szczególnie w przypadku węzłów o ograniczonych zasobach zasilanych bateryjnie, gdzie poziom aktywności węzła (ang. duty-cycling) jest niski.

Na rysunku 5.8 (zbiorczo) oraz rysunkach B.21 i B.22 (z podziałem) prezentuję wyniki czasu wykonania algorytmów na GATEWAY_4B dla odpowiednio wariantu 1 i 2 generowania topologii sieci. Wyniki prezentowane są z podziałem na rodzaj trasy oraz liczbę dostępnych kanałów. Dodatkowo na rysunku B.23 oraz B.24 prezentuję analogiczne wyniki dla wykonania na stacji roboczej. Parametry rozkładu czasu wykonania prezentuję w tabeli B.26 (wariant 1) i B.30 (wariant 2) dla GATEWAY_4B oraz tabeli B.34 (wariant 1) i B.35 (wariant 2) dla stacji roboczej. Analizując wyniki (poniżej), posługuję się wartościami liczbowymi dla GATEWAY_4B, wartości dla stacji roboczej są średnio o około 6-6.4 razy krótsze. Średni czas wykonania dla wszystkich wektorów testowych wynosi dla CAS_LATENCY: 0,401s (wariant 1) oraz 2,32s (wariant 2), podczas gdy dla SEP_EDF_FIFO odpowiednio: 16,266s oraz 54,14s. Średnio CAS_PERIOD wykonują plan 40 razy szybciej dla wariantu 1 i 23 razy szybciej dla wariantu 2.

Ze względu na inny sposób działania algorytmu CAS_PERIOD w stosunku do znanych w literaturze, należy zwrócić uwagę na istotne zmniejszenie liczby elementów w planach węzłów $SCH_i = \{el \mid el = (p, c_k^{\{s,d\}}, l_j^i, ts, ch)\}$. Średnia liczba elementów w wygenerowanych planach (*SCH*) dla pełnego wektora testowego dla SEP_EDF_FIFO to: 251945 i 515001 dla odpowiednio wariantu 1 (Tabela B.28) i 2 (Tabela B.32), natomiast dla CAS_LATENCY to: 264 i 460. Jeszcze większa dysproporcja jest obserwowana porównując wartości maksymalne długości planów. Algorytmy z rodziny EDF/LLF generują plan w jednym wspólnym okresie planowania, podczas gdy CAS_PERIOD generuje (w zależności od użytego wariantu grupowania) osobny plan dla każdego okresu powtarzania wynikającego z pojedynczego kontraktu. Ta zmiana pociąga za sobą szereg istotnych korzyści które już omówiłem, takich jak obserwowany zwiększony poziom zaplanowania oraz zdecydowanie krótszy czas wykonywania algorytmu, ale dodatkowo pozwala na przedstawienie wykonanego planu w innej

(bardziej kompaktowej) formie. Taka forma zapisania zaplanowanych czynności jest zgodna ze standardem ISA100.11a i może być efektywnie dystrybuowana do węzłów. Kompaktowy zapis planu pociąga za sobą również mniejszy czas i mniejszy wydatek energetyczny potrzebny na dystrybucję nowych planów do węzłów sieci lub ich aktualizację. Dodatkowo pozwala na łatwiejszy zapis oraz przechowywanie w węzłach sieci które dysponują ograniczonymi zasobami pamięci.

5.3.3 Symulacja pracy sieci dla CAS-PERIOD i ewaluacja wyników

Wykorzystując opracowane środowisko testowe (rozdział 5.2) dla każdego przygotowanego planu przeprowadzono symulację pracy sieci. Symulacja potwierdza poprawność przygotowanych planów a zgromadzone w trakcie symulacji dane pozwalają na określenie poziomu opóźnienia w dostarczaniu pakietów oraz zajętości buforów ramek jakie znajdują się w każdym węźle sieci IWN.



Rysunek 5.9 Rozkład maksymalnego czasu transmisji pakietu w funkcji liczby węzłów dla wariantu 1 i 2 generowania topologii sieci i algorytmów CAS-PERIOD

Na rysunku 5.9 oraz w tabelach B.42 (topologia 1) i B.45 (topologia 2) prezentuję wyniki dotyczące rozkładu maksymalnego czasu transmisji pakietu. Wszystkie parametry prezentowanego rozkładu uległy przesunięciu w stronę mniejszych wartości. Różnica jest szczególnie widoczna dla wariantu 2 generowania topologii sieci przy 200 węzłach, gdzie wartość średnia maksymalnego czasu transmisji (CAS_LATENCY) jest krótsza w stosunku do algorytmu SEP_EDF_FIFO o około 20%. Dodatkowo na rysunkach B.25 oraz B.26 prezentuję rozkład maksymalnego czasu transmisji pakietu z podziałem na rodzaj trasy i liczbę dostępnych kanałów radiowych. Skrócenie maksymalnego czasu transmisji (w stosunku do SEP_EDF_FIFO) jest większe dla trasy 2, gdzie pośredniczy większa liczba węzłów. Mniejsza

liczba dostępnych kanałów jak również większa liczba kontraktów obsługiwana w sieci uwydatniają różnicę w maksymalnym czasie transmisji na korzyść CAS-PERIOD, rysunek A.10. Rozkład średniego czasu transmisji pakietu jest zbliżony we wszystkich porównywanych algorytmach (tabele B.40 oraz B.43).

Najistotniejszą korzyścią stosowania CAS-PERIOD, w odniesieniu do parametrów sieci, jest sprowadzenie do zera składowej jitter wynikającej ze zmienności planu dla konkretnego kontraktu (parametry rozkładu odchylenia standardowego, tabele B.41 oraz B.44). Ta cecha CAS-PERIOD wynika wprost z konstrukcji algorytmu i została potwierdzona w wynikach symulacji. Na rysunku A.11 prezentuję porównanie odchylenia standardowego czasu transmisji pakietu dla CAS-PERIOD (CAS_LATENCY) oraz SEP_EDF_FIFO. Stała wartość czasu transmisji pakietu uwolnionego w związku z konkretnym kontraktem jest szczególnie pożądana w sieciach przemysłowych, gdzie determinizm czasowy i minimalizacja jitter są kluczowe.

Należy zwrócić uwagę, że porównywane wyniki symulacji algorytmów z rodziny CAS_PERIOD i SEP_EDF_FIFO odnoszą się do wygenerowanych planów w których SEP_EDF_FIFO uzyskuje dużo gorsze wyniki dotyczące poziomu zaplanowania (efektywnie mniejszy ruch w sieci) przy jednoczesnej konieczności wykonywania algorytmu w czasie dłuższym o jeden rząd wielkości.

5.4 Algorytm CAS-SEQ

5.4.1 Definicja algorytmu CAS-SEQ i przyjęte założenia

Planowanie wyłącznie ruchu sekwencyjnego jest o wiele mniejszym problemem niż ruchu cyklicznego i efektywnie może być wykonane w IWN przy użyciu przykładowego algorytmu AMUS. Rezultaty otrzymywane dla AMUS, nawet przy krótkim okresie planowania (1s), są zadawalające. W przypadku planowania wyłącznie ruchu sekwencyjnego należy zwrócić uwagę jedynie na dwa aspekty które są często pomijane w literaturze, tj. wpływ nadmiarowości planu na oczekiwany czas dostarczenia pakietu i zajętości bufora ramek w węzłach sieci oraz wpływ kolejności alokacji przestrzeni planowania. W przypadku ruchu sekwencyjnego nie możemy określić dokładnego czasu uwolnienia pakietu ponieważ nie dysponujemy w kontrakcie aperiodycznym informacją o okresie i fazie. Planowanie tego rodzaju ruchu ma większą swobodę w alokacji szczelin czasowych dla transmisji między węzłami. Parametry którymi dysponujemy ograniczają się jedynie do określenia średniej $(ca_k^{\{s,d\}}. Committed_Burst)$ i chwilowej $(ca_k^{\{s,d\}}. Excess_Burst)$ ilości uwalnianych pakietów w jednostce czasu (Tabela 3.2). Znajomość długości pakietów ($ca_k^{\{s,d\}}$. Payload_Size) pozwala na określenie liczby ramek przypadających na okres planowania. Dobór okresu planowania nie jest tutaj kluczowy (jak ma to miejsce w przypadku ruchu cyklicznego) i z przeprowadzonych

analiz (grupa 3, Rysunek A.3 oraz Rysunek A.4) wynika że zarówno planowanie w dłuższym okresie jak i okresie względnie krótkim wynoszącym zaledwie 1s, daje zadawalające rezultaty w IWN. Analizując wyniki otrzymane z przeprowadzonego badania algorytmów planowania ruchu sekwencyjnego (rozdział 5.2.1) oraz zachowanie sieci pracującej w oparciu o generowane plany (rozdział 5.2.2) jak również potrzeby wynikające z definicji postawionego problemu 3.2.4, należy zwrócić uwagę na następujące problemy przy planowaniu wyłącznie ruchu sekwencyjnego z użyciem AMUS:

- Ponieważ celem jest maksymalizacja liczby zawartych kontraktów, planowanie w pierwszej kolejności kontraktów o największej wymaganej przepływności λ (AMUS linia 2, Algorytm 1) nie jest pożądane i należy planować w pierwszej kolejności ruch kontraktów które mają najmniejsze wymagania dotyczące przepływności
- Przestrzeń planowania jest dwuwymiarowa (co wynika wprost z użytego wielodostępu TDMA i FDMA) o osiach odpowiadających długości okresu planowania (czas) oraz liczby dostępnych kanałów (częstotliwość). AMUS zapełnia przestrzeń planowania zawsze zaczynając od pierwszej szczeliny czasowej. Takie budowanie planu prowadzi do sytuacji w której większość elementów planu skoncentrowana jest w okolicach początku okresu planowania pozostawiając plan zlokalizowany bliżej jego końca bez elementów planu, a co za tym idzie bez aktywności węzłów. Niekorzystny efekt widoczny jest szczególnie przy małej liczbie elementów w planie. Zbudowany w ten sposób plan wykazuje tendencję do "przetrzymywania" ramek które mogą zostać uwolnione w trakcie nieaktywności węzłów. Zapełnianie w pierwszej kolejności całej osi wynikającej z długości planu pozwoli na znalezienie szczeliny pozwalającej na transmisję ramki bliżej czasu uwolnienia pakietu.

Mimo, iż planowanie ruchu wyłącznie sekwencyjnego może być efektywnie przeprowadzone z użyciem dostępnych algorytmów to jego planowanie w obecności ruchu cyklicznego napotyka na te same ograniczenia co wyłącznie ruchu cyklicznego. Ruch sekwencyjny w obecności ruchu cyklicznego należy zamodelować jako osobny ruch o wspólnym okresie powtarzania elementów planu, wynikającym z długości okresu planowania przyjętego dla ruchu sekwencyjnego.

Algorytm CAS-SEQ (7) pozwala na efektywne planowanie ruchu sekwencyjnego w obecności ruchu cyklicznego. Głównym elementem algorytmu, podobnie jak w przypadku planowania cyklicznego jest algorytm 5 odpowiedzialny za planowanie pojedynczej transmisji (scheduleElement(G; $\tau_k^{\{i,j\}}$; ACT; BUF; ts; p)) wraz z algorytmem 6 odpowiedzialnym za identyfikację kolizji (checkScheduleCollision(el, p, SCH)). Algorytm planuje kolejno poszczególne kontrakty $ca_k^{\{s,d\}}$, zaczynając od tych z najmniejszą przepływnością (λ). Planowanie odbywa się w pierwszej kolejności wzdłuż osi czasu (ts). Dojście do granicy okresu planowania p powoduje rozpoczęcie planowania ponownie od pierwszej szczeliny, wypełniając kolejno przestrzeń wzdłuż drugiej osi planowania jaką są kanały radiowe.

Algorytm 7 Conflict Aware Scheduling (CAS_SEQ) – faza planowania

input : G; RT; CA; p
output : SCH /* zbiór harmonogramów dla wszystkich węzłów sieci */
1 for each $(QA_k) \leftarrow ca_k^{\{s,d\}} \in CA$ do compute przepływność λ
² Posortuj kontrakty $ca_k^{\{s,d\}} \in C$ w porządku narastającym po przepływności λ
3 $ts \leftarrow 0; BUF \leftarrow \{ \}; ACT \leftarrow \{false\}$
4 for each $ca_k^{\{s,d\}} \in CA$ do
5 $release(BUF, ca_k^{\{s,d\}}); /* umieść \tau_k^{\{i,j\}} w BUF */$
6 for each $\tau_k^{\{i,j\}} \in BUF$ do /* dla każdej transmisji */
7 scheduleElement(G; $\tau_k^{\{i,j\}}$; ACT; BUF; ts; p)
8 end
9 $ts \leftarrow (ts + 1) \mod p /*$ następna szczelina czasowa */
10 end

5.4.2 Ewaluacja wyników planowania i symulacja pracy sieci

Istotnym wkładem algorytmu CAS-SEQ jest zdolność do wykonywania planu w obecności ruchu cyklicznego. Algorytm CAS-SEQ rozpatrywany wyłącznie dla ruchu sekwencyjnego daje bardzo zbliżone wyniki do przykładowego algorytmu AMUS. Wygenerowane plany dla AMUS oraz CAS-SEQ maja ten sam okres planowania przyjęty arbitralnie i wynoszący 1s. Wyniki dotyczące poziomu planowania są nieznaczenie lepsze (tabele B.47 oraz B.49), natomiast czas wykonywania jest wydłużony (tabele B.46 oraz B.48), co wynika zarówno z możliwości unikania kolizji z innymi planami (algorytm 6) jak również samego sposobu planowania poprzez użycie bufora ramek. Zawarta w algorytmie CAS-SEQ autorska metoda unikania kolizji z innymi planami stanowi podstawę do opracowania algorytmu GA-CAS stanowiącego całościowego rozwiązanie postawionego problemu (3.2.4) i pozwala na jednoczesne planowanie ruchu cyklicznego z wykorzystaniem algorytmu CAS-PERIOD i ruchu sekwencyjnego algorytmem CAS-SEQ w ramach jednej sieci IWN.

Wykorzystując opracowane środowisko testowe (rozdział 5.2) dla każdego przygotowanego planu przeprowadziłem symulację pracy sieci. Wykonana symulacja potwierdza poprawność przygotowanych planów.

6 Optymalizacja procesu konfiguracji sieci IWN

Całościowe rozwiązanie zdefiniowanego problemu 3.2.4 wymaga zastosowania efektywnego algorytmu, który we względnie krótkim czasie (30s lub 60s, rozdział 5.3.1) znajdzie optymalne rozwiązanie. Mimo, że istnieją standardy definiujące poszczególne warstwy w stosie komunikacyjnym oraz protokoły wymiany danych między węzłami sieci IWN [93], takie jak ISA100.11a [27] oraz WirelessHART [26], to brak jest w literaturze opracowań dotyczących efektywnych metod konfiguracji poszczególnych warstw w sposób pozwalający na optymalizację działania całej sieci w szczególności w kontekście wypełnienia wymagań QoS (Tabela 3.4).

W rozdziale 6.2 prezentuję definicję opracowanego algorytmu GA-CAS, którego zadaniem jest rozwiązanie zdefiniowanego problemu optymalizacyjnego 3.2.4 w sposób pozwalający na efektywną implementację w środowisku IWN opartym o centralną jednostkę zarządzającą siecią (SM). Implementacja GA-CAS w środowisku rzeczywistym zakłada wykorzystanie GATEWAY_4B oraz rodziny węzłów SN4. Zarówno GATEWAY 4B jak i SN4 należy traktować jako reprezentatywne urządzenia w praktycznych zastosowaniach. Oba urządzenia posiadają mocno ograniczone zasoby w stosunku do pełnionych funkcji. W przypadku GATEWAY 4B ograniczenie wynika z małej ilości pamięci operacyjnej (128/256/512 MB) oraz małej mocy obliczeniowej. W przypadku wezła SN4 ograniczenia odnoszą się dodatkowo do zużycia energii, wynikającego wprost z pracy z wykorzystaniem autonomicznego zasilania bateryjnego. Ze względu na złożoność problemu optymalizacyjnego (3.2.4) oraz w oparciu o przeprowadzoną analizę 6.1 wybrałem algorytm genetyczny (GA) w zakresie poszukiwania optymalnego rozwiązania problemu. Opracowane deterministyczne algorytmy CAS (CAS-PERIOD i CS-SEQ) zostały użyte w zakresie wykonywania planów szczelin czasowych i zintegrowane z wykorzystaniem algorytmu GA. W rozdziale 6.3 prezentuję wyniki badań eksperymentalnych w których porównuje różne warianty GA-CAS (Tabela A.1) i analizuję wpływ parametrów na jego efektywność.

6.1 Analiza możliwości wykorzystania algorytmów genetycznych

W opracowanym algorytmie wykorzystałem algorytm genetyczny (GA), który jest bardzo efektywny w rozwiązywaniu problemów optymalizacyjnych, poprzez szukanie rozwiązania zbliżonego do optymalnego [94, 95]. GA są zdolne do rozwiązywania złożonych problemów optymalizacyjnych które mają dużą przestrzeń szukania rozwiązań [96]. Do podstawowych obszarów stosowania GA, w odniesieniu do sieci bezprzewodowych, według [95] należą: ruting, QoS, równoważenie obciążenia, alokacja kanałów i innych zasobów, rozmieszczenie węzłów, lokalizacja oraz inne. Wiele praktycznych problemów w sieciach bezprzewodowych wymaga optymalizacji wielu parametrów, które mogą rywalizować a nawet potencjalnie

kolidować ze sobą. Ważną cechą GA jest możliwość wspólną optymalizacji wielu parametrów a nawet wielu celów np. poprzez zastosowanie wag [97]. W celu zwiększenia skuteczności i przyspieszenia dochodzenia do celu, populacja GA może okresowo być wzbogacana o materiał genetyczny który W odniesieniu do eksplorowanego problemu, ma większe prawdopodobieństwo na znalezienie optymalnego rozwiązania [98]. Z punktu widzenia zdefiniowanego problemu oraz wykorzystania rozwiązania do zarządzania siecią IWN poprzez implementację w GATEWAY 4B, szczególnie istotna jest zdolność GA do prezentacji aktualnego rozwiązania problemu po określonym czasie (ang. time constrained). Wraz z upływającym czasem rozwiązanie z założenia staje się bardziej dopasowane do celu, natomiast w każdej chwili dysponujemy rozwiązaniem, które nawet gdy jest odległe od optymalnego, może posłużyć do sformowania pierwszej konfiguracji sieci i być aktualizowane w miarę upływającego czasu lub spływających danych wejściowych w postaci raportów z wezłów sieci. Ta zdolność adaptacyjna GA wpisana jest w naturę rodziny algorytmów ewolucyjnych i mimo że nie gwarantuje znalezienia optymalnego rozwiązania w praktyce znajduje rozwiązanie satysfakcjonująco dobre i jednocześnie pozwala na jego znalezienie w relatywnie krótkim czasie [99].

Algorytmy GA z powodzeniem znajdują zastosowanie w rozwiązywaniu problemów routingu w szczególności tych w których rozwiązanie ma wypełniać wiele postawionych warunków (ang. multi-constrained path (MCP)). W [100] wykorzystano GA do rozwiązania problemu znalezienie najkrótszej ścieżki (SP). Zbadano wpływ wielkości populacji oraz sposobu jej inicjalizacji na efektywność znalezienie rozwiązania. Mimo iż w powszechnej świadomości wielkość populacji powinna rosnać wraz ze wzrostem złożoności problemu, który można utożsamiać przykładowo z długością oczekiwanych tras, to satysfakcjonujące rezultaty otrzymuje się przy dużo mniejszej ilości osobników w populacji. W podsumowaniu napisano, że wielkość populacji wpływa pozytywnie na efektywność znalezienia rozwiązania, ale jest okupiona dużym kosztem zarówno w odniesieniu do czasu (złożoności obliczeniowej) jak i potrzebnej pamięci. Autorzy zestawiają swoje wyniki z algorytmem Dijkstry i wymieniają dwie główne zalety użycia GA w postawionym problemie. Algorytm jest mniej wrażliwy na różne topologie i konfiguracje sieci oraz czas potrzebny do znalezienia rozwiązania nie rośnie znacząco wraz ze wzrostem wielkości sieci. Zaprezentowano również eksperymentalny wzór pozwalający na oszacowanie efektywnej wielkości populacji dla przypadków, w których krzyżowanie zdecydowanie dominuje nad mechanizmem mutacji. Trasę zakodowano w chromosomie o zmiennej długości w taki sposób, że kolejne geny zawierają numer węzła zaczynając od węzła źródłowego w pierwszym genie i kończąc na węźle docelowym w ostatnim. Ta zaproponowana metoda kodowania trasy jest aktualnie powszechnie stosowana i stanowi naturalny zapis łatwo reprezentowalny w algorytmach.

W [101] badano efektywność zastosowania algorytmów GA do rozwiązania problemu najkrótszej ścieżki (SP) w zmiennym środowisku. W praktycznych zastosowaniach topologia sieci ulega ciągłym zmianom ze względu na przemieszczanie się węzłów, ograniczenia wynikające z żywotności baterii, zmian warunków propagacji fal radiowych, etc. Algorytmy pracujące w takich warunkach muszą efektywnie reagować na takie zmiany. Mimo, że istnieje grupa deterministycznych algorytmów rozwiązywania problemu SP: algorytm Dijkstry, algorytm przeszukiwanie wszerz (ang. breadth-first search, BFS), algorytm Bellmana-Forda, itp. Wszystkie te algorytmy mają wielomianową złożoność czasową. Są skuteczne w stacjonarnych sieciach bezprzewodowych lub przewodowych. Wykazuja jednak niedopuszczalnie wysoką złożoność obliczeniową dla komunikacji w czasie rzeczywistym obejmującej szybko zmieniające się topologie sieci [100]. Podstawowy schemat zastosowania GA w opisanym środowisku zakłada restart algorytmu przy każdej zmianie i generowanie nowych pokoleń zaczynając od inicjalizacji pokolenia zerowego. Takie rozwiązanie jest nieefektywne, ponieważ nie wykorzystuje już zgromadzonych informacji oraz nie zapewnia wystarczająco szybkiej zbieżności w osiągnieciu zadawalającego rezultatu. Autorzy badają dwa modele będące odpowiedzią na opisany problem: tzw. "imigrantów" w populacji [102-104] oraz zapamiętywanie już otrzymanych rezultatów w celu późniejszego użycia [105]. Połączenie obu metod znalazło zastosowanie w [106]. Koncepcja imigrantów podobnie jak sam algorytm genetyczny, została zaczerpnięta z obserwacji otaczających nas zjawisk. W miarę tworzenia kolejnych pokoleń, populacja staje się mniej zróżnicowana co sprzyja dochodzeniu do wartości optymalnej w przypadku niezmieniającego się celu. Same krzyżowanie i mutacja zapewniają jedynie ograniczone zróżnicowanie i dedykowane są unikaniu lokalnych minimów. W celu dynamicznego zareagowania na pojawiającą się zmianę celu poprzez zmianę otoczenia konieczne jest zwiększenie zróżnicowania poprzez dopuszczenie świeżego materiału genetycznego w postaci imigrantów, którzy wnoszą niejako zróżnicowanie między kolejnymi pokoleniami w ramach populacji. Z drugiej strony, koncepcja oparta na zapamiętywaniu optymalnych rezultatów bazuje na obserwacji, że zmiany środowiska zachodzą cyklicznie i mimo że aktualnie cel znajduje się w zupełnie innym miejscu niż poprzednio istnieje duża szansa, że za jakiś czas środowisko ustali się w warunkach już występujących i poprzednio znalezione optimum będzie mogło być wykorzystane i przyspieszy znalezienie nowego optymalnego rozwiązania. W [101] potwierdzono te założenia i skonkludowano stwierdzeniem, że obie metody sprzyjają zwiększeniu wydajności algorytmów GA, przy czym imigranci uwypuklają swoją przewagę w przypadku zmian nie okresowych natomiast schemat oparty na pamięci w przypadku zmian cyklicznych.

W [107] zaproponowano połączenie zastosowania wielu populacji [108] oraz imigrantów w celu zwiększenia efektywności działania GA do rozwiązania SP w zmiennym środowisku.

Klasyczny wariant GA ma pojedynczą populację przeszukującą całą przestrzeń potencjalnych rozwiązań. Podejście wielopopulacyjne próbuje podzielić przestrzeń poszukiwań na kilka części, a następnie wykorzystuje kilka małych populacji do przeszukiwania ich oddzielnie. Zaproponowano by populacja główna była odpowiedzialna za szukanie nowych rozwiązań w szerokiej przestrzeni, podczas gdy dodatkowe populacje są odpowiedzialne za szukanie rozwiązań w okolicach znalezionego optimum. Dodatkowo mała liczba imigrantów jest dodawana do populacji głównej w celu zwiększenia jej różnorodności. Zaproponowane połączenie dwóch technik zwiększa jakość dopasowania proponowanego rozwiązania a otrzymane rezultaty są lepsze niż w przypadku GA z restartem oraz GA z losowymi imigrantami. Autorzy konkludują stwierdzeniem, że rozwiązanie oparte na wielu populacjach pomaga w osiągnięciu zadawalającego rezultatu w przypadku zmiennego środowiska.

Również inne optymalizacyjne problemy dotyczące grafów są rozwiązywane za pomocą GA, np. problem komiwojażera [109].

Druga ważna grupa zastosowań GA jest alokacja zasobów w celu wypełnienia wymagań QoS. W [110] wykorzystano GA do kontroli mocy nadajnika w kognitywnych układach radiowych przy jednoczesnym wypełnieniu wymagań QoS i optymalizacji dostępu do medium. Użytkownicy drugoplanowi moga jedynie nadawać, jeżeli wypełnią wymaganie QoS, którym jest ustalony poziom stosunku sygnału do zakłóceń plus szum (ang. signal-to-interference-plusnoise ratio, SINR). Sieć drugoplanowa dąży do ograniczenia zakłóceń generowanych w sieci pierwszoplanowej przy jednoczesnym wypełnieniu progowych wymagań QoS dla własnych użytkowników. Autorzy proponują dwie alternatywne funkcje przystosowania. Jedna oparta na funkcji kary [111] w celu implementacji warunków i dażąca do minimalizacji zużycia energii sieci drugoplanowej. Druga wykorzystuje wielokryteriowa funkcję przystosowania która wspólnie optymalizuje pojemność sieci i zużycie energii. W [110] wygenerowano populacje startową całkowicie losowo, natomiast do selekcji osobników wykorzystano metodę ruletki, gdzie prawdopodobieństwo wybrania osobnika jako rodzica jest zależne od jego pozycji w populacji. Wykorzystano najmniej złożoną metodę krzyżowania polegającą na wybraniu z równym prawdopodobieństwem materiału genetycznego jednego z rodziców, natomiast mutacja genotypu jest całkowicie losowa, gdzie do materiału genetycznego rodzica dodawany jest losowy wektor. Przy takiej konstrukcji funkcji mutacji zaobserwowano duży jej wpływ na osiągane wyniki, efektywność spada wraz ze wzrostem prawdopodobieństwa mutacji. Najmniejszym i zarazem najlepszym poziomem prawdopodobieństwa mutacji jest 10%, dla każdej wartości większej uzyskano gorsze rezultaty. W dużej mierze podyktowane to jest niedopasowaną funkcją mutacji do problemu optymalizacyjnego. GA uzyskuje najlepsze rezultaty, jeżeli każda z prowadzonych operacji, krzyżowanie, mutacja, selekcja dopasowana jest do natury optymalizowanego problemu, tak aby przestrzeń poszukiwań była ograniczana

fizyczną naturą zjawiska. Autorzy nie analizują otrzymanych rezultatów z uwzględnieniem reżimu czasowego, natomiast zauważają, że dla bardzo małych populacji wyniki są mocno nie optymalne, jakość otrzymanych rozwiązań rośnie wraz ze wzrostem populacji, natomiast dla większych populacji GA potrzebuje o wiele więcej czasu do znalezienia optymalnego rozwiązania oraz że istnieje pewna wielkość populacji (45 w badanych przypadku) gdzie przy dalszym wzroście populacji nie widać wzrostu poziomu dopasowania. Wnioski są zbieżne z prezentowanymi w [100] I są szczególnie istotne w przypadku generowania rozwiązania problemu w reżimie czasowym, co ma miejsce w badanym przeze mnie zastosowaniu.

W [112] zaproponowano rozwiązanie problemu wielu wymagań QoS przy znalezieniu trasy typu unicast za pomocą GA. Przywołano podział metryk na addytywne, multiplikatywne i oparte o najmniejszą wartość, przypomniano, że klasa problemu została potwierdzona jako NP-zupełny [60] i jego rozwiązanie nie jest możliwe w rozsądnym czasie z punktu widzenia aplikacji wrażliwych na opóźnienia. Zaproponowana funkcja celu stanowi iloczyn trzech odrębnych funkcji: funkcja czasu transmisji między punktami docelowymi trasy, gdzie wartość jest odwrotnie proporcjonalna dla wartości tego czasu; funkcja kary, która ma spełniać rolę warunku granicznego i eliminować rozwiązania, które przekraczają maksymalną wartość czasu transmisji oraz funkcja naprawy, której zadaniem jest faworyzowanie węzłów o większej ilości energii pozostającej w baterii. Autorzy wykorzystują mechanizm dwupunktowego krzyżowania chromosomów, gdzie dwa losowo wybrane punkty w chromosomach rodziców stanowią punkty ich złączenie, tak że łańcuch od początku do punkt 1 oraz od punkt 2 do końca brany jest z chromosomu jednego rodzica a fragment pomiędzy punktami z chromosomu drugiego rodzica. W ciekawy sposób rozwiązano funkcję mutacji, pozostawiając losowość w zakresie wyboru chromosomu, który podlega mutacja natomiast samą mutację dostosowano do natury rozwiązywanego problemu. W procesie mutacji zamianie ulegają węzły z najmniejszą i największą ilością energii pozostającej w baterii. Zastosowanie takiego operatora przyspiesza dojście do optymalnego rozwiązania przy jednoczesnym zmniejszeniu różnorodności przeszukiwanego obszaru. Pozostaje otwarte pytanie czy operator mutacji jest właściwym miejscem do takiej operacji i czy takie przekształcenie nie powinno być realizowane na poziomie wzrostu osobnika w momencie kształtowania fenotypu, pozostawiając operator mutacji dedykowany jedynie do zwiększenia różnorodności bieżacej populacji. Warto w tym miejscu zwrócić uwage, że dobór parametrów i technik stosowanych w GA jest kompromisem pomiędzy zdolnością do eksploracji oraz zdolnością do eksploatacji. Eksploracja, która obejmuje poszukiwanie rozwiązań w przestrzeniach jeszcze nie znanych, może być utożsamiana ze zdolnością GA do szybkiej reakcji na zmiany warunków. Natomiast eksploatacja obejmuje coraz bliższe dochodzenie do obranego celu i może być utożsamiana ze zdolnością GA do szybkiego znalezienia rozwiązania przy poprawnie obranym celu. Każdy z elementów składowych GA takich jak: selekcja, krzyżowanie, mutacja, imigranci, pamięć, wzrost, etc., dedykowany jest określonym zadaniom w ramach tak zdefiniowanego kompromisu i wychodzenie poza ich role tworzy jedynie niepotrzebny chaos pojęciowy. Podsumowując, zaprezentowane rezultaty potwierdzają większą skuteczność algorytmu opartego o GA w stosunku do innych porównywanych.

W [94] zaprezentowano algorytm oparty na GA który znajduje r-możliwych tras minimalizując wszystkie k-metryki QoS. Zaprezentowana funkcja dopasowania ma charakter uniwersalny i uwzględnia obecność k-metryk QoS na l-linkach w ramach r-możliwych tras. Autorzy uwzględniają jedynie metryki addytywne, przy czym należy zwrócić uwagę, że każda metryka multiplikatywna może być łatwo przekształcona do metryki addytywnej z wykorzystaniem własności logarytmu. Ponieważ zakres wartości każdej metryki QoS może się różnić w zależności od typu metryki, zwykłe zsumowanie wszystkich metryk QoS spowodowałoby, że wynik byłby zdominowany przez metryki QoS, które mają większe wartości w porównaniu z innymi. Dlatego przed dodaniem metryk QoS każda metryka musi zostać znormalizowana. Autorzy kodują trasę w sposób jak [100] rozróżniając dwie metody kodowania: binarne (ogólne) i uwzględniające kodowanie dopasowane do rozwiązywanego problemu. Populacja startowa jest inicjalizowana w pełni losowo z grafu połączeń. Do selekcji wykorzystano metodę turniejowa a operację krzyżowania wykonano jako 1-punktową z uwzględnieniem funkcji naprawy eliminującej ewentualne powstałe pętle. Zaproponowana metoda mutacji zakłada losowy wybór punktu mutacji w chromosomie, od którego fragment ścieżki do węzła docelowego generowany jest losowo z grafu połaczeń. Autorzy prezentuja wyniki i prowadza interesująca dyskusję dotyczącą doboru parametrów GA takich jak: prawdopodobieństwo mutacji, wielkość populacji, liczba pokoleń. Dobór wielkości populacji i liczby pokoleń jest kompromisem pomiędzy poziomem dopasowania a czasem potrzebnym do wykonania algorytmu. Wyższe wartości co do zasady dadzą lepszy rezultat, ale kosztem dłuższego czasu wykonywania obliczeń. Autorzy zauważają, że o ile ścieżka między węzłami istnieje to do jej znalezienia nie potrzebna jest duża liczba pokoleń a wzrost dopasowania przy dużej liczbie pokoleń nie jest znaczący. Czas wykonywania algorytmu rośnie liniowo wraz ze wzrostem liczby pokoleń. Poziom dopasowania rośnie wraz ze wzrostem wielkości populacji, ale wzrost jest logarytmiczny, tj. przyrost dopasowania maleje wraz ze wzrostem liczebności populacji. Natomiast czas wykonywania rośnie wykładniczo wraz ze wzrostem liczebności populacji. Autorzy zauważają, że istnieje pewna wielkość prawdopodobieństwa mutacji, dla której poziom dopasowania jest maksymalny a czas wykonywania minimalny przy stałej liczbie pokoleń. Wielkość sieci wpływa znacząco na czas wykonywania algorytmu przy stałej liczbie pokoleń różnica pomiędzy siecią 8x8 a 12x12 węzłów jest 6 krotna.

W [22] wykorzystano GA do optymalizacji parametrów sieci WirelessHART. Zaprezentowany mechanizm adaptacyjny pozwala menadżerowi sieci na aktualizację w czasie rzeczywistym wybranych parametrów sieci w oparciu o dane dotyczące wymagań QoS. Parametry podlegające optymalizacji to: poziom współczynnika poprawnie dostarczonych ramek (ang. *Packet Delivery Ratio*, PDR) przy którym link uznawany jest za użyteczny do komunikacji; liczba dostępnych kanałów oraz liczba powtórzeń dostarczenia ramki. Problem optymalizacyjny został przedstawiony jako wspólna optymalizacja wielu celów. Zaprezentowana funkcja dopasowania dąży do minimalizacji całkowitego czasu transmisji, maksymalizacji czasu życia oraz maksymalizacji niezawodności sieci. Zaprezentowany mechanizm adaptacyjny pracuje w formie sprzężenia zwrotnego, które bazując na zestawach danych zgromadzonych z węzłów sieci przy odpowiadających im parametrach sieci. Budując kolejne zestawy danych wybiera wśród nich te które dają najlepszy wynik dopasowania. Zaprezentowana koncepcja stanowi interesujące uzupełnienie algorytmów znajdujących właściwą konfigurację parametrów sieci i w praktycznych scenariuszach może być wykorzystana do optymalizacji wybranych parametrów sieci.

W [99] wykorzystano GA do generowania planu szczelin czasowych oraz tablicy routingu w przypadku zastosowań w sieciach komórkowych w sytuacji gdy ruch jest przekazywany pomiędzy kolejnymi przekaźnikami w celu maksymalizacji przepływności oraz minimalizacji zużycia energii. Wartość zaproponowanej funkcji celu jest wprost proporcjonalna do logarytmu przepływności i odwrotnie proporcjonalna do liczby przeskoków oraz zagregowanej energii potrzebnej do osiągnięcia punktu dostępu. Zaproponowany algorytm wykorzystuje genotyp zarówno do zakodowania informacji o planie jak i trasie ruchu. Autorzy wykorzystują pojęcie fenotypu jedynie do zdekodowania zapisanej w genotypie informacji (numer linku). Wyniki pokazują, że zaproponowany algorytm zapewnia rozwiązanie bliskie optymalnemu w czasie o rząd wielkości krótszym niż algorytm oparty o przeszukiwanie wyczerpujące (ang. exhaustive search).

Problemy optymalizacyjne w sieciach IWN są również rozwiązywane z wykorzystaniem innych niż GA technik. W [113] zaprezentowano algorytm wykorzystujący uczenie przez wzmacnianie (ang. Q-learning) dedykowany do znalezienia tras routingu w IWN zarządzanych centralnie np. standardu WirelessHART.

W [17] zaprezentowano algorytm *SchedEx-GA* wykorzystujący GA i dostarczający międzywarstwową konfigurację dla sieci IWN np. dla standardu WirelessHART [26]. Centralny menedżer na bieżąco otrzymuje aktualizacje dotyczące jakości komunikacji na poszczególnych kanałach między sąsiednimi węzłami, które to informacje bezpośrednio odzwierciedlają stan sieci. W przypadku wystąpienia zmian, nowa konfiguracja jest przygotowywana i sugerowana
menedżerowi w celu jej wdrożenia. Algorytm SchedEx-GA w zakresie planowania szczelin czasowych wykorzystuje algorytm SchedEx [72] wraz jego rozszerzeniem [114] ale dodatkowo wykorzystuje GA do przygotowania tablicy routingu i całościowego rozwiązania problemu. Zaproponowana funkcja celu minimalizuje liczbę naruszeń wymagania dotyczącego całkowitego czasu transmisji pomiędzy węzłami docelowymi. Wykorzystano kodowanie genotypu jak w [100] przy czym genotyp zawiera informację o wszystkich trasach w sieci a nie jedynie o jednej. Dodatkowo zastosowano pojęcie fenotypu jako elementu stanowiącego przekształcenie genotypu. To fenotyp stanowi całościowe rozwiązanie problemu i to on podlega ocenie z użyciem funkcji dopasowania. Przekształcenie genotypu w fenotyp polega na wykonaniu planu szczelin czasowych z wykorzystaniem algorytmu SchedEx. Wykorzystano populację startową generowaną losowo ale dodatkowo zasilono ją osobnikiem z genotypem utworzonym z tras o najmniejszej liczbie przeskoków [115]. Ta technika zwana "sianiem" (ang. seeding), pozwala na uzyskanie szybszej zbieżność do celu i jest znana w literaturze [116]. Należy stwierdzić, że stosowanie tej techniki jest w pełni uzasadnione w przypadku, jeżeli natura problemu optymalizacyjnego pozwala na zlokalizowanie potencjalnych obszarów, w których rozwiązanie problemu istnieje z dużym prawdopodobieństwem. W procesie selekcji osobników do procesu rekombinacji genetycznej wykorzystuje metodę turniejową. Operator krzyżowania korzysta z metody 1-punktowej, przy czym krzyżowane są wszystkie trasy zawarte w genotypie, tak że trasa o numerze 1 z rodzica 1 krzyżowana jest z trasą 1 rodzica 2.

Zgodnie z moją najlepszą posiadaną wiedza opracowanie [17] jest jedynym, które zakłada skalowalne i całościowe rozwiązanie problemu znalezienia konfiguracji sieci WSN lub IWN zarządzanej centralnie na podstawie informacji uzyskanych od węzłów (np. dotyczące jakości połączeń pomiędzy sąsiadami) i wykorzystujące TDMA jako technikę współdzielenia dostępu do medium. Mimo, iż ogólnie sformułowane cele się pokrywają, to sformułowanie problemu i zastosowane metody oraz algorytmy znacząco od siebie odbiegają. Poniżej w sposób porównawczy prezentuję podstawowe różnice na poziomie funkcjonalnym pomiędzy prezentowanym w niniejszym opracowaniu rozwiązaniem a tym prezentowanym w [17].

- Odmienna definicja celu optymalizacyjnego. SchedEx-GA minimalizuje liczbę naruszeń wymagania QoS dotyczącego czasu transmisji, podczas gdy GA-CAS maksymalizuje liczbę zawartych kontraktów z uwzględnieniem ich priorytetów i przy spełnieniu wymagań QoS;
- GA-CAS uwzględnia dodatkowo bilans energetyczny sieci na podstawie modelu energetycznego węzłów, który jest podstawą do określenia możliwości wywiązania się z zobowiązania kontraktowego wynikające z parametru dotyczącego czasu obowiązywania kontraktu;

- GA-CAS zakłada, że komunikacja z węzłem realizowana jest w oparciu o ważny kontrakt zawarty z managerem sieci. Kontrakt ustanawia parametry połączenie takie jak: wymagania QoS, priorytet, czas trwania, typ ruchu i inne opisujące ruch i zależne od jego typu, szczegółowo opisane w tabeli 3.4. SchedEx-GA tworzy pojęcie logicznego czujnika jako elementu składowego czujnika fizycznego;
- GA-CAS opracowany został w celu obsługi zarówno ruchu cyklicznego jak również sekwencyjnego z gwarantowanymi parametrami QoS, natomiast SchedEx-GA dostosowany jest do ruchu sekwencyjnego oraz ruchu cyklicznego bez gwarantowanego czasu dostarczenia pakietu;
- SchedEx-GA wykorzystuje heurystyczny układ planowania [72] natomiast GA-CAS deterministyczny;
- Układ planowania zastosowany w SchedEx-GA uwzględnia jedynie niezawodność jako warunek w procesie planowania, natomiast GA-CAS zarówno niezawodność jak i maksymalne opóźnienie między węzłami końcowymi;
- GA-CAS dedykowany jest dla sieci IWN zarządzanych centralnie z przewagą ruchu horyzontalnego [10, 117] (Przemysł 4.0, M2M). SchedEx-GA dedykowany jest dla sieci WSN zarządzanych centralnie z przewagą ruchu wertykalnego z małą liczbą węzłów zbierających dane (ang. sink node);
- SchedEx-GA opracowano z myślą o standardzie WirelessHART podczas gdy GA-CAS o standardzie ISA100.11a [93].

6.2 Definicja algorytmu GA-CAS i przyjęte założenia

6.2.1 Reprezentacja problemu w algorytmie

Rozwiązanie problemu optymalizacyjnego w GA reprezentowane jest w postaci fenotypu (ang. phenotype). Fenotyp stanowi przekształcenie genotypu (ang. genotype), który stanowi reprezentację genetyczną każdego osobnika. Genotyp stanowi zbiór informacji opisujących osobnika, podczas gdy fenotyp jest przekształceniem genotypu podlegającym ocenie za pomocą funkcji przystosowania (funkcji celu). Genotyp jest elementem, który jest przekazywany z pokolenia na pokolenie w procesie krzyżowania, natomiast fenotyp jest przekształceniem genotypu indywidualnym dla osobnika i realizowanym w procesie jego "wzrostu" przez analogię dostosowania jego indywidualnych cech do środowiska zewnętrznego w którym wzrasta.

Podstawowy schemat działania GA [118] obejmuje następujące fazy:

- 1. Inicjalizacja populacji początkowej
- 2. Selekcja
- 3. Krzyżowanie (ang. crossover)

- 4. Mutacja (ang. mutation)
- 5. Utworzenie fenotypu
- 6. Obliczenie funkcji przystosowania

Sieć IWN została zamodelowana zgodnie z opisem 3.2.3. Zadaniem algorytmu jest znalezienie optymalnego rozwiązania problemu zdefiniowanego w 3.2.4 przy zadanych warunkach. Rozwiązaniem zdefiniowanego problemu jest znalezienie takiej konfiguracji sieci conf = $\{RT, SCH\}$, gdzie RT jest zbiorem wszystkich tras routingu a SCH jest zbiorem wszystkich harmonogramów transmisji dla wszystkich węzłów sieci, która będzie optymalna z punktu widzenia funkcji celu i wypełni postawione twarde warunki. Tak zdefiniowany problem możemy zdekomponować i przedstawić w postaci dwóch problemów wzajemnie zależnych od siebie i wymagających wspólnej oceny. Pierwszym problemem jest znalezienie takiego RT które wypełnia warunki miękkie mk_0, mk_1 zgodnie z równaniami (15) oraz (16). Drugim jest przekształcenie RT -> SCH oraz wypełnienie pozostałych warunków: miękkiego mk_2 (tylko ruch cykliczny, równanie (15)) oraz twardych k_0, k_1, k_2 , równanie (12).

Zbiór wszystkich tras routingu spełniających warunki mk_0, mk_1 oraz częściowo mk_2 został zakodowany jako genotyp. Fenotyp jest całościowym rozwiązaniem problemu w postaci konfiguracji sieci $conf = \{RT, SCH\}$. Utworzenie fenotypu realizowane jest poprzez wykonanie opracowanego i przedstawionego w rozdziałach 5.3 oraz 5.4 algorytmu planowania CAS wykorzystując otrzymaną i zapisaną w genotypie RT. Umieszczenie algorytmu planowania na poziomie przekształcenia genotyp → fenotyp jest realizowane jak w [17]. Zbiór genotyp i odpowiadający mu fenotyp tworzą osobnika w populacji. Inicjalizacja osobnika w wyniku tworzenia populacji startowej lub w wyniku krzyżowania zakłada każdorazowo inicjalizację genotypu, natomiast fenotyp tworzony jest w wyniku przekształcenia genotyp \rightarrow fenotyp. W strukturze genotypu wyodrębniono |C| chromosomów, gdzie |C| jest liczbą wszystkich kontraktów. Chromosomy o numerach od 1 do |C| zawierają geny odnoszące się do trasy $r_k^{\{s,d\}} \in RT$ kolejnych kontraktów zgłaszanych do zawarcie z SM przez węzły sieci. Długości chromosomów od 1 do |C| są zmienne [119] a każdy gen zawiera informację o kolejnym węźle na trasie. Pierwszy gen zawiera numer węzła będącego źródłem transmisji w ramach kontraktu a ostatni gen zawiera numer wezła będącego wezłem docelowym. Liczba genów w chromosomach odpowiada liczbie przeskoków na trasie dla odpowiadającego kontraktu. Taki sposób kodowania tras w chromosomach jest powszechnie stosowany w GA [94, 100] i zgodnie z taksonomią według [95] nosi nazwę "bezpośrednie kodowanie wartości". Oczywiście istnieją inne metody kodowania [95]: binarne, drzewiaste, etc., natomiast zgodnie z aktualnym konsensusem prezentowanym w literaturze bezpośrednie kodowanie wartości jest najlepiej dopasowane do problemów bazujących na różnych wariantach problemu najkrótszej

trasy i powinno być stosowane wszędzie tam gdzie operatory krzyżowania i mutacji dostosowane są do natury problemu czyli bazują na zapisanych informacjach a nie jedynie na danych [120]. W chromosomach są zapisywane jedynie trasy "możliwe" czyli spełniające warunki graniczne:

- trasa nie zawiera pętli, co oznacza, że każdy węzeł na trasie może wystąpić raz;
- maksymalna długość trasy (chromosomu) to $N_V = |V|$;
- trasa spełnia warunek mk_0 odnoszący się do poprawności wygenerowanej trasy routingu;
- trasa spełnia warunek mk_1 odnoszący się do wymaganej niezawodności, który jest spełniony, jeżeli niezawodność ρ'_k dostarczenia ruchu dla kontraktu k jest nie mniejsza niż niezawodność ρ_k zadana w wymaganiach QoS kontraktu;
- trasa spełnia częściowo warunek mk₂ w zakresie warunku koniecznego, ale nie wystarczającego, który do weryfikacji wymaga jedynie znajomości trasy a który szczegółowo zdefiniowano i opisano w 3.2.4.;

W zależności od konfiguracji algorytm w różnych miejscach korzysta z dwóch wariantów najkrótszej trasy, analogicznie do tych które stanowią TRASA 1 i 2 w wygenerowanych danych będących elementem wektora testowego na potrzeby badania algorytmów planowania szczelin czasowych, rozdział 4.2.2.

- Trasa 1 (MIN-HOP), odpowiada trasie z najmniejszą liczbą przeskoków, generowana jest z wykorzystaniem algorytmu przeszukiwania wszerz (ang. breadth-first search, BFS). Wykorzystano funkcję *boost::breadth_first_search()* z biblioteki BOOST.
- Trasa 2 (MAX-RELIABILITY), odpowiada trasie z najwyższym całkowitym prawdopodobieństwem dostarczenia pakietu PDR. W tym wypadku użyto algorytmu Dijkstry z zastrzeżeniem, że mnożenie prawdopodobieństw PDR dla poszczególnych krawędzi przekształcono w dodawanie ich logarytmów. Zastosowana w algorytmie waga dla każdej krawędzi jest równa $PDR_{LOG} = -1 \times \ln(1 PER)$. Wykorzystano funkcję *boost::dijkstra shortest paths()* z biblioteki BOOST.

Opracowany algorytm wykorzystuje również trasy wygenerowane losowo (RANDOM). Do generowania losowych tras wykorzystałem algorytm bazujący na zaproponowanym w [100] i wykorzystanym również w [94]. Implementacja własna bazuie funkcii na boost::loop erased random walk() z biblioteki BOOST. W każdym momencie, gdy mowa jest o generowaniu tras MIN-HOP, MAX-RELIABILITY, RANDOM, trasy generowane są w oparciu o aktualny dla SM graf połączeń. Graf połączeń zaimplementowany jest jako lista sąsiedztwa adjacency list Implementacja (ang. graph). wykorzystuje boost::adjacency list<boost::vecS, boost::vecS, boost::bidirectionalS, VertexSM, EdgeSM>,

gdzie zbiory wierzchołków i zbiory krawędzi wierzchołków zostały zaimplementowane jako tablice o zmiennej długości (std::vector).

Jeżeli wygenerowana dla kontraktu na podstawie aktualnego grafu połączeń trasa z najmniejszą możliwą liczbą przeskoków (Trasa 1) nie zapewnia możliwości dostarczenie pakietu w wymaganym czasie zgodnie z wzorem (20), kontrakt uznawany jest za niemożliwy do realizacji i wykluczany jest z puli kontraktów stanowiących efektywnie dane wejściowe dla algorytmu. Analogicznie w odniesieniu do niezawodności, jeżeli wygenerowana dla kontraktu na podstawie aktualnego grafu połączeń trasa z największą możliwą niezawodnością (Trasa 2) nie zapewnia wystarczającego poziomu niezawodności wymaganego przez kontrakt, kontrakt uznawany jest za niemożliwy do realizacji i wykluczany jest z niemożliwy do realizacji i wykluczany jest z puli kontraktów stanowiących efektywnie dane wejściowe dla algorytmu.

Opracowany algorytm kończy działanie i dostarcza optymalny rezultat kierując się dwoma warunkami dotyczącymi: maksymalnej liczby pokoleń, maksymalnego czasu przetwarzania. W przeważającej większości opracowań dotyczących GA, wykorzystywane jest jedynie ograniczenie dotyczące liczby pokoleń, w którym z góry określona jest liczba pokoleń po wygenerowaniu których algorytm ma zwrócić aktualne najlepiej przystosowane rozwiązanie. Niestety, sposób ten mimo naturalnej łatwości implementacji w GA, nie ma praktycznego zastosowania, w interesującym nas problemie, w którym konfiguracja sieci musi zostać dostarczona do węzłów w maksymalnym z góry określonym czasie (ang. timeout). Wszystkie żądania przesyłane z węzłów dotyczące chęci ustanowienie kontraktu mają ustalony maksymalny czas na otrzymanie odpowiedzi od SM. Przekroczenie tego czasu skutkuje przerwaniem procedury i rozpoczęciem jej na nowo. Bazując na tych ustaleniach, algorytm posługuje się jednocześnie drugim ograniczeniem dotyczącym maksymalnego czasu na znalezienie rozwiązania. Czas ten w praktycznych zastosowaniach wynosi 30s lub 60s. Algorytm udostępnia rozwiązanie po przekroczeniu jednego z dwóch warunków. W ewaluowanych wynikach badań warunki zostały ustalone jako: 40 pokoleń lub 30s. Wartość dotycząca ograniczenia czasu jest krótszą z dwóch występujących w praktyce i oczekuję, że pozwoli ona na uwypuklenie zjawisk które mają miejsce w analizowanych przypadkach. Liczba 40 jest względnie małą liczbą pokoleń dla której w wielu opracowań obserwujemy wyraźne spowolnienie dochodzenia do celu. Można powiedzieć, że poniżej tej wartości każde kolejne pokolenie powoduje zasadniczą poprawę dopasowania, powyżej tej wartości ten przyrost jest zasadniczo mniejszy. Oczywiście granica jest umowna i w praktyce znajduje się gdzieś pomiędzy 20-100 pokoleń w zależności od szeregu innych parametrów, wpływających na gradient zbieżności.

Implementacja opracowanego algorytmu udostępnia na interfejsie dwie główne metody pozwalające na interakcję:

- GA-CAS::init(timeout)
- GA-CAS::processGenerationNo(no)

Metoda GA-CAS::init(timeout) odpowiedzialna jest za sprawdzenie warunków granicznych dotyczących kontraktów, inicjalizację populacji startowej tworzącej pokolenie 0 oraz "wzrost" osobników, poprzez utworzenie fenotypu w wyniku wykonania mapowania genotyp→fenotyp wraz z obliczeniem wartości funkcji dopasowania dla każdego z osobników. Ponieważ wykonanie opisanych operacji może trwać dłużej niż określony graniczny czas przekazany jako argument timeout, populacja może zawierać mniej niż NUMBER OF INDIVIDUALS osobników, gdzie postąpieniem po którym sprawdzany jest warunek dotyczący przekroczenia czasu jest wykonanie operacji na pojedynczym osobniku. Metoda GA-CAS::processGenerationNo(no) odpowiedzialna jest za przygotowywanie kolejnych pokoleń populacji w wyniku zastosowania operatorów selekcji i rekombinacji genetycznej (krzyżowanie oraz mutacja) oraz wzrostu osobników poprzez mapowanie genotyp-fenotyp i obliczenie funkcji przystosowania. Warto w tym miejscu zwrócić uwagę, że przed fazą wzrostu następuje autorska metoda estymacji wartości funkcji dopasowania na podstawie samego genotypu. Jeżeli wartość oszacowanej wartości funkcji dopasowania jest mniejsza od wartości funkcji dopasowania "najgorszego" dorosłego członka populacji ("dorosły" osobnik jest członkiem populacji poddanym fazie wzrostu, czyli mapowania genotyp→fenotyp), osobnik jest pozbawiany fazy wzrostu i nie dodawany do populacji. Metoda szacowania wartości funkcji dopasowania na podstawie jedynie genotypu została opisana w dalszej części rozdziału.

6.2.2 Operatory genetyczne

6.2.2.1 Populacja startowa

Na początku tworzona jest populacja startowa składająca się z *NUMBER_OF_INDIVIDUALS* liczby unikatowych osobników. Osobnik uznawany jest za unikatowy, jeżeli w populacji nie istnieje osobnik, który ma identyczne chromosomy dla każdego z kontraktów. Ponieważ utworzony osobnik może nie być dodany do populacji ze względu na brak unikatowości, zaimplementowany jest mechanizm ponawiający tworzenie osobnika, przy czym maksymalna liczba ponowień jest równa maksymalnej liczebności populacji. Maksymalna liczba wygenerowanych osobników podczas inicjalizacji to 2 x *NUMBER_OF_INDIVIDUALS*. Chromosom może być zainicjalizowany wyłącznie z "możliwą" trasą. Populacja inicjalizowana jest w jednym z 5 wariantów, które w różnym stopniu wykorzystują technikę "siania" [115] do stworzenie zbioru potencjalnych rozwiązań dopasowanego do oczekiwanego rozwiązania. Wybór typu inicjalizacji POPULATION_INIT realizowany jest na poziomie kompilacji tak aby istnienie wyboru nie wpływało na czas wykonywania algorytmu.

- 1. POPULATION_INIT::RANDOM trasy dla każdego kontraktu zakodowane w chromosomach każdego z osobników generowane są wyłącznie losowo.
- POPULATION_INIT::MIN_RANDOM dodatkowo w stosunku do metody pierwszej, pierwszy osobnik w populacji ma wszystkie chromosomy wygenerowane jako Trasa 1 (MIN-HOP)
- POPULATION_INIT::MAX_RANDOM dodatkowo w stosunku do metody pierwszej, pierwszy osobnik w populacji ma wszystkie chromosomy wygenerowane jako Trasa 2 (MAX-RELIABILITY)
- POPULATION_INIT::MIN_MAX_RANDOM dodatkowo w stosunku do metody pierwszej, pierwszy osobnik w populacji ma wszystkie chromosomy wygenerowane jako Trasa 1 (MIN-HOP) a drugi osobnik w populacji ma wszystkie chromosomy wygenerowane jako Trasa 2 (MAX-RELIABILITY)
- POPULATION_INIT::MIXED sposób generowania trasy dla każdego kontraktu zakodowanego w chromosomach każdego z osobników populacji losowany jest z równym prawdopodobieństwem spośród trzech możliwych: Trasa 1 (MIN-HOP), Trasa 2 (MAX-RELIABILITY) oraz trasa losowa (RANDOM).

6.2.2.2 Selekcja

Zadaniem selekcji jest wybór osobników z bieżącej populacji w celu przeprowadzenia rekombinacji genetycznej i utworzenia genotypu nowych osobników, potencjalnie stanowiących nowe pokolenie. Operator selekcji zapewnia osobnikom z większym poziomem dopasowania większe szanse wyboru i tym samym większe szanse na przekazanie fragmentu własnego genotypu do nowo powstającego osobnika. Oczekujemy, że rodzice z wysokim poziomem dopasowania będą mieli potomstwo o jeszcze większym poziomie dopasowania. Istnieje kilka rodzajów operatorów selekcji [95, 121] wśród nich: metoda turniejowa, ruletka, elitaryzm oraz różne ich warianty. W algorytmie wykorzystałem metodę turniejową z dwoma graczami, głównie ze względu na małą złożoność obliczeniową i częste wykorzystanie w algorytmach GA dotyczących problemu SP [94]. W tym wariancie, wybieramy losowo dwóch osobników z populacji a następnie wybieramy tego który ma większą wartość funkcji przystosowania (rozdział 6.2.3). Wybór drugiego rodzica wykonywany jest w analogicznie. Para rodziców może być wybrana tylko raz przy tworzeniu nowego pokolenia. Osobniki do losowane z "matecznika" pool), selekcji są (ang. mating utworzonego Z NUMBER OF INDIVIDUALS osobników z najwyższym poziomem przystosowania. W wyniku selekcji następnie rekombinacji genetycznej а tworzonych jest NUMBER OF INDIVIDUALS nowych osobników.

6.2.2.3 Krzyżowanie

Jedną z operacji wykonywanych w procesie rekombinacji genetycznej jest krzyżowanie. Zadaniem operatora jest wykorzystanie genotypu obu rodziców i utworzenie z nich genotypu nowego osobnika. Krzyżowanie uzupełnione mutacją tworzą fundament rekombinacji, pozwalający na efektywne szukanie rozwiązań w rozległych przestrzeniach. W wyniku krzyżowania wykorzystujemy cechy najlepszych osobników w celu ich połączenie i utworzenia nowego o jeszcze lepszym od rodziców poziomie przystosowania. Najczęstszym operatorem krzyżowania jest jedno lub dwu-punktowy, gdzie punkt odnosi do miejsca połączenia w chromosomach rodziców. W algorytmie zaimplementowałem dwa operatory krzyżowania:

- CROSSOVER_TYPE:: SINGLE_POINT (Jedno-punktowy) dla każdej z odpowiadających sobie par chromosomów losowo z równym prawdopodobieństwem wybierany jest pojedynczy punkt złączenia. Nowo utworzony chromosom składa się z fragmentu od początku chromosomu pierwszego rodzica do punktu złączenia oraz fragmentu od punktu złączenia do końca chromosomu drugiego rodzica.
- CROSSOVER_TYPE::SIMPLE (Prosty) dla każdej z odpowiadających sobie par chromosomów rodziców z równym prawdopodobieństwem wybierany jest chromosom jednego z rodziców i umieszczany w genotypie potomka.

Operator krzyżowania "prosty" nie tworzy nowych tras a jedynie wykorzystuje już istniejące i tworzy nowe osobniki z nowymi ich kombinacjami. Jest to najprostsza z możliwych form operatora z najniższą możliwą złożonością obliczeniową. Stanowi punkt porównania dla bardziej złożonego operatora jedno-punktowego, którego zadaniem jest tworzenie nowych tras w wyniku wymiany fragmentów chromosomów. Zastosowanie operatorów opartych o większą liczbę punktów złączenia odrzuciłem ze względu na spodziewaną długość trasy, która wynosi od jednego do kilku punktów przeskoku oraz o ilość wspólnych węzłów w parach chromosomów które mogą posłużyć jako potencjalne punkty złączenie. W przeważającej liczbie przypadków istnieje jeden punkt. Potencjalnym punktem złączenia pary chromosomów są unikatowe pozycje takich samych węzłów w chromosomach rodziców. Węzły źródłowy oraz docelowy nie są brane pod uwagę. Taki mechanizm krzyżowania jest analogiczny do opisanego w [94, 100, 122]. Posługując się przykładem, numery węzłów dla zwiększenia czytelności zostały zastąpione kolejnymi literami alfabetu, załóżmy że mamy taką parę odpowiadających sobie chromosomów u rodziców:

Rodzic 1 -chromosom k: [A F T I K U V Z]

Rodzic 2 – chromosom k: [A W D P Q I M Z]

Węzły A oraz Z są odpowiednio węzłem źródłowym oraz docelowym. Wspólnym węzłem jest I, jest to jedyny potencjalny punkt złączenia. W wyniku krzyżowania powstanie chromosom k ich potomka:

Dziecko – chromosom k: [A F T I M Z]

Istnieje możliwość, że w parze chromosomów nie ma wspólnych węzłów, w takim przypadku stosowana jest metoda "prosta" i wybierany jest pełen chromosom jednego z rodziców. Zauważmy również, że w wyniku operacji może powstać trasa zawierająca pętle (mimo że chromosomy rodziców nie zawierają pętli), w takim przypadku zgodnie z [94, 100], należy znaleźć powtarzjące się węzły i usunąć zgodnie z zaproponowanym algorytmem. Po znalezieniu i usunięciu pojedynczej pętli konieczne jest powtórzenie przeszukiwania w celu usunięcia kolejnych pętli, do czasu aż trasa będzie wolna od petli. W implementacji algorytmu zastosowałem własne autorskie rozwiązanie eliminujące pętle w operacji krzyżowania i nie wymagające przeprowadzenia krzyżowania w dwóch przebiegach: krzyżowanie, naprawa. Załużmy istnienie dwóch par chromosomów u rodziców:

Rodzic 1 - chromosom k: [A R S F D G]

Rodzic 2 – chromosom k: [A B C **D** E **F** G]

Istnieją dwa wspólne węzły tworzące potencjalne punkty złączenia, węzeł F oraz D. Złączenie zgodnie z zaprezentowaną metodą może potencjalnie utworzyć dwa chromosomy:

Dziecko' – chromosom k: [A R S F G]

Dziecko'' – chromosom k: [A R S F **D** E F G]

Chromosom dla dziecko' jest wolny od pętli, natomiast dla dziecko'' zawiera pętle [**F** D E **F**], którą należy usunąć w wyniku naprawy trasy jak w przytoczonych publikacjach. Należy zauważyć, że do powstania pętli dochodzi w sytuacji gdy w rodzicu 2 istnieje inny punkt złączenie po prawej stronie podczas gdy w rodzicu 1 ten punkt jest po lewej stronie od wybranego punktu złączenia. Konkludując poczynioną obserwację, autorskie rozwiązanie zakłada, że jako punkty złączenia wybierane są jedynie takie które spełniają postawiony warunek. Przebieg opracowanego algorytmu jest następujący:

- przechodząc od końca kolejno po węzłach chromosomu k rodzica 2;
- dla każdego węzła chromosomu k rodzica 2 szukam kolejno w chromosomie k rodzica 1 takiego samego węzła;
- dla każdej znalezionej pary takich samych węzłów w chromosomach obu rodziców sprawdzam poniższy warunek;
- jeżeli pozycja poprzednio znalezionego punktu złączenia w chromosomie k rodzica 1 znajduje się przed aktualnie znalezioną pozycję to pomijam znalezioną parę;

• w przeciwnym wypadku znalezione pozycje w chromosomach obu rodziców traktuje jako potencjalny punkt złączenia obu chromosomów.

Zaprezentowana oryginalna metoda pozwala na efektywne generowanie nowych tras wolnych od pętli i nie wymaga zastosowania funkcji naprawy trasy. Jeżeli operator krzyżowania złącza rodziców w odwrotnej kolejności, pierwszy fragment z rodzica 2 a drugi z rodzica 1, to w zaprezentowanym algorytmie należy zamienić występowanie rodziców, tam gdzie mowa o rodzicu 2 zastosować rodzica 1 i odwrotnie.

Wybór sposobu krzyżowania CROSSOVER_TYPE realizowany jest na poziomie kompilacji tak aby istnienie wyboru nie wpływało na czas wykonywania algorytmu.

6.2.2.4 Mutacja

Zadaniem mutacji jest wprowadzenie zróżnicowania w populacji. Mutacja pozwala na eksplorację obszarów przestrzeni poszukiwań, które do tej pory nie były znane. Mutacja pozwala na unikanie lokalnych minimum i daje możliwość znalezienie globalnego optimum rozwiązywanego problemu. chromosom określone Każdy W genotypie ma prawdopodobieństwo wystąpienia mutacji. Dobór właściwego poziomu mutacji jest istotny z punktu widzenia zapewnienia właściwego kompromisu pomiędzy coraz lepszym przystosowaniem do optymalnego rozwiązania a eksploracją w celu jego znalezienia. Istnieje wiele rodzajów mutacji stosowanych w GA [95] a jedynie kilka jest dostosowanych do rodzaju informacji zapisanej w badanym chromosomie. Ze względu na to że numery węzłów oraz połączenia między nimi są ściśle ograniczone poprzez aktualny graf połączeń, operator mutacji musi uwzględniać ten fakt i mutacja choć losowa powinna modyfikować chromosomy w taki sposób aby w dalszym ciągu utworzona trasa była "możliwa". Innymi słowy jak najwięcej warunków dotyczących trasy powinno być spełnionych po wykonaniu mutacji, aby proces był skuteczny a powstała trasa nie za każdym razem musiała być odrzucana ze względu na brak spełnienia warunków granicznych. Użyty operator mutacji wykorzystuje rozszerzoną koncepcję zaprezentowaną w [100] i użytą również w [94]. W oparciu o zdefiniowane prawdopodobieństwo mutacji dla każdego chromosomu określamy czy podlega mutacji. Jeżeli chromosom podlega mutacji, losowo z równym prawdopodobieństwem wybierany jest punkt mutacji jako pozycja na trasie z wyłączeniem węzłów źródłowego i docelowego. Poczynając od punktu mutacji do węzła docelowego generowana jest losowo nowa trasa. W opracowanym algorytmie operator został znacząco rozszerzony i poziom mutacji określany jest jako zbiór trzech wartości {MUTATION MIN, MUTATION MAX, MUTATION RANDOM}. Każda z wartości interpretowana jest jako prawdopodobieństwo wystąpienia konkretnego typu mutacji i jest wyrażona w procentach. Suma trzech wartości nie może więc przekroczyć 100% a maksymalna wartość oznacza, że mutacja występuje zawsze. Trzeci z zaproponowanych typów mutacji (MUTATION_RANDOM) odpowiada operatorowi zaproponowanemu w [94] i wykorzystuje sposób generowania trasy losowej (RANDOM). Mutacja typu MUTATION_MIN generuje nową trasę od punktu mutacji do punktu docelowego jako trasę 1 (MIN_HOP). Natomiast, mutacja typu MUTATION_MAX generuje nowy fragment trasy jako trasę 2 o maksymalnej niezawodności (MAX-RELIABILITY). Mutacja oparta o generowanie fragmentu trasy jako trasa 1 lub trasa 2 nie zostały użyte w znanych mi pozycjach literaturowych i stanowią proponowane unikalne rozwiązanie operatora mutacji. W części dotyczącej ewaluacji wyników zostaną zestawione różne typu mutacji w celu ich porównania.

Ze względu na to, że generowana trasa stanowi kontynuację istniejącej trasy algorytmy generujące muszą być zmodyfikowane w stosunku do prezentowanych przy okazji opisu dotyczącego reprezentacji problemu. Konieczność unikania pętli powoduje, że algorytmy muszą mieć świadomość odwiedzenia węzłów tworzących fragment trasy nie podlegający mutacji, tak aby użyty wcześniej węzeł nie mógł być użyty ponownie i aby nowo wygenerowany fragment nie zawierał węzłów użytych w niezmienianym fragmencie trasy.

Biorąc pod uwagę powyższe założenie dotyczące generowania losowych fragmentów tras, konieczne było opracowanie własnego algorytmu który pozwala na określenie już odwiedzonych węzłów (nie zmieniany fragment trasy). W przypadku generowania fragmentu chromosomu w wariancie trasa 1 oraz 2 wykorzystałem odpowiednio funkcje *boost::breadth_first_visit()* oraz *boost::dijkstra_shortest_paths_no_init()*, które są wariantami wcześniej wymienionych funkcji ale pozwalają na określenie już odwiedzonych wierzchołków grafu.

6.2.2.5 Wzrost populacji

Większość rozwiązań GA wykorzystuje jedynie mechanizmy rekombinacji i pomijają fazę wzrostu osobnika. Faza wzrostu jest procesem w którym osobniki nabywają cech indywidualnych wynikających ze środowiska w którym bytują. Genotyp jest zbiorem informacji (genów) opisujących osobnika i przekazywanych z pokolenie na pokolenie w procesie rekombinacji genetycznej. Natomiast fenotyp jest pojęciem szerszym i obejmuje zarówno informacje genetyczne jak również cechy osobnika nabyte w wyniku przebywania w określonym środowisku. Możemy więc powiedzieć że organizm "rodzi się" z indywidualnym genotypem i w trakcie życia nabywa cechy, zależne od środowiska, które dostosowują go do środowiska w którym wzrasta. W trakcie wzrostu osobnika w środowisku bytowania, następuje przekształcenie genotyp→fenotyp i to fenotyp jest pełnym zbiorem cech osobnika które decydują o stopniu jego dostosowania do środowiska. Przekształcenie to jest bardzo rzadko spotykane w opracowaniach dotyczących problemów optymalizacyjnych. W zbiorczym opracowaniu [95] dotyczącym wykorzystania GA w WSN słowo fenotyp się nie pojawia.

W opracowanym algorytmie wykorzystuję pojęcia fenotypu i wykorzystuję przekształcenie genotyp—fenotyp [17]. Ocena osobnika z wykorzystaniem funkcji przystosowania wykonywana jest w oparciu o fenotyp zawierający zbiór pełnych informacji o konfiguracji conf = {RT, SCH}. Ponieważ, genotyp zawiera jedynie możliwe trasy dla wszystkich kontraktów RT, to do oceny potrzebna jest znajomość drugiego z elementów konfiguracji czyli planu połączeń SCH. Wykonanie przekształcenia $RT \rightarrow SCH$ z wykorzystaniem aktualnego grafu połączeń zakodowałem jako przekształcenie genotyp—fenotyp i będę je nazywał "wzrostem" osobnika w odróżnieniu od "rozmnażania", które utożsamiam jedynie z operacją powstania nowego genotypu (rekombinacja genetyczna). Przekształcenie $RT \rightarrow SCH$ jest operacją planowania i wykorzystuje dwa, opracowane i przebadane we wcześniejszej części publikacji, algorytmy CAS_PERIOD i CAS_SEQ odpowiednio dla kontraktów dotyczących ruchu cyklicznego i ruchu sekwencyjnego. Sekwencja algorytmu dokonującego opisanego przekształcenia jest następująca:

- 1. Dla wszystkich kontraktów cyklicznych o najwyższym priorytecie wykonaj planowanie wykorzystując algorytm CAS_PERIOD.
- 2. Operację z punktu 1 powtórz kolejno dla zbiorów kontraktów o coraz niższych priorytetach.
- 3. Dla wszystkich kontraktów sekwencyjnych (acyklicznych) o najwyższym priorytecie wykonaj planowanie wykorzystując algorytm CAS_SEQ.
- 4. Operację z punktu 3 powtórz kolejno dla zbiorów kontraktów o coraz niższych priorytetach.

Oczywiście należy tutaj zaznaczyć, że każda z operacji planowania wykonana w powyższym algorytmie jest wykonywana ze świadomością planu wykonanego w kroku poprzednim. Innymi słowy, każde wywołanie algorytmów CAS_PERIOD i CAS_SEQ tworzy nowy zestaw planów uwzględniając już utworzone plany. Takie działanie jest immanentną cechą opracowanych algorytmów, które wykonują plan unikając kolizji z innymi już zaplanowanymi elementami. Ostatnią operacją opisanego przekształcenia $RT \rightarrow SCH$ jest dokonanie oceny fenotypu z zastosowaniem funkcji przystosowania. Analizując przedstawioną sekwencję algorytmu planowania należy zauważyć że ruch cykliczny dla wszystkich priorytetów planowany jest przed ruchem sekwencyjnym, takie zachowanie jest celowe i podyktowane dodatkowymi warunkami którymi jest obwarowane planowanie cykliczne w szczególności dotyczącymi maksymalnego czasu dostarczenia pakietu do węzła docelowego. Należy również zauważyć że mimo iż planowanie wykonywane jest w takiej kolejności to ocena uzależniona jest wyłącznie od priorytetu bez uwzględnienia rodzaju ruchu. Dzięki takiemu zapisaniu problemu algorytmy planowania mają większą szansę na zaplanowanie ruchu cyklicznego który jest "cięższy" do zaplanowania ze względu na dodatkowe warunki a jednocześnie dają szansę GA na premiowanie kontraktów o wyższym priorytecie bez względu na rodzaj.

6.2.3 Funkcja przystosowania

Funkcja przystosowania pozwala na określenie wartości poziomu przystosowania osobnika i jest kluczowym elementem GA pozwalającym na znalezienie optymalnych rozwiązań. Zadaniem funkcji przystosowania jest zapewnienie przekazywania do kolejnych pokoleń osobników z najwyższym poziomem dopasowania. Funkcja musi być tak skonstruowana aby "punktować" te rozwiązania które są najbliżej optymalizacyjnego celu.

Wartość funkcji przystosowania obliczana jest dla każdego nowego osobnika populacji z wykorzystaniem bezpośrednio funkcji optymalizacyjnej zdefiniowanej w 3.2.4 jako równanie (12) wraz z równaniami zależnymi. Wartość obliczana jest na podstawie informacji zawartej w fenotypie osobnika. Definicja funkcji zakłada jej ogólny charakter i określa wagi na podstawie liczby kontraktów o niższym priorytecie. W opracowanym algorytmie na potrzeby efektywnej implementacji tej funkcji, została przyjęta maksymalna liczba możliwych kontraktów dla każdego z priorytetów wynosząca 255 oraz maksymalna liczba priorytetów zgodnie ze standardem ISA100.11a (4 priorytety). Dzięki takiemu założeniu wagi dla poszczególnych priorytetów są stałe i wynoszą [0xFFFFF, 0xFFF, 0xFFF, 0] zaczynając od kontraktów o najwyższym priorytecie. Wartość funkcji dopasowania zapisana jest w postaci 32 bitowej liczby całkowitej bez znaku, gdzie 0 oznacza osobnika o najniższym poziomie dopasowania.

6.2.4 Szacowanie wartości funkcji przystosowania na podstawie genotypu

Do określenia wartości funkcji przystosowania poza znajomością genotypu osobnika, potrzebna jest znajomość pełnego fenotypu, czyli wykonania operacji potencjalnie najbardziej złożonej, wymagającej zastosowania algorytmów CAS_PERIOD i CAS_SEQ. Analizując zachowanie algorytmu oraz właściwości funkcji przystosowania zaproponowałem autorskie rozwiązanie które pozwala na oszacowanie wartości tej funkcji na podstawie samego genotypu. W procesie selekcji branych jest *NUMBER_OF_INDIVIDUALS* osobników o najwyższym poziomem przystosowania a zatem nowy osobnik którego wartość funkcji przystosowania jest niższa od najniższej wartości występującej w populacji, zostanie z populacji usunięty lub może do niej nie być w ogóle dodawany. Szacunkowa wartość która pozwoliłaby na określenie górnej wartości funkcji przystosowania będzie miała taką własność, że przy porównaniu z najmniej przystosowanym dorosłym osobnikiem w populacji pozwoli na podjęcie decyzji o porzuceniu wzrostu osobnika. Porzucenie osobnika zaoszczędzi czas na znalezienie potencjalnego rozwiązania, którego wartość funkcji przystosowania nie będzie na pewno wyższa po wykonaniu mapowaniu fenotypu. Innymi słowy oszacowana wartość ma taką własność że jest zawsze większa lub równa wartości docelowej funkcji przystosowania obliczonej z

wykorzystaniem fenotypu. Własność taką ma wartość obliczona bezpośrednio z genotypu zakładająca, że każdy kontrakt mający trasę w genotypie zostanie poprawnie zaplanowany. Obliczenie wartości szacunkowej funkcji przystosowania sumuje, z uwzględnieniem wag wynikających z priorytetów kontraktów, te kontrakty które mają trasy (co należy czytać jako mają "możliwe" trasy). Każdy kontrakt który nie ma trasy jest sumowany z wartością 0. Efektywność zaproponowanej metody jest tym większa im mniejsza jest liczebność populacji ze względu na większe prawdopodobieństwo utworzenie osobnika o wartości niższej niż występująca w populacji. Efektywność jest również większa dla pierwszych pokoleń ponieważ wtedy różnorodność proponowanych rozwiązań jest również zwiększona. Te właściwości pokrywają się z postawionymi celami, które zakładają dostarczenie rozwiązania w z góry ograniczonym czasie, tym samym stosowana będzie mała liczebność populacji i mała liczba pokoleń, zgodnie z analizą literaturową dokonaną na wstępie.

6.2.5 Implementacja algorytmu i notacja zapisu wersji algorytmu

GA-CAS został zaimplementowany w środowisku rzeczywistym w języku C++. Pełna implementacja algorytmu łącznie z algorytmami planowania obejmuje około 5 tysięcy linii kodu. GA-CAS wykorzystuje następujące biblioteki programistyczne:

- C++17 STD
- BOOST 1.68 (https://www.boost.org; biblioteki: LOG, GRAPH) w zakresie operacji na grafach oraz integracji mechanizmu logowania zdarzeń
- TronixIWN w zakresie definicji podstawowych obiektów warstwy aplikacyjnej standardu ISA100.11a oraz stosu komunikacyjnego; biblioteka opracowana w ramach projektu badawczo-rozwojowego (współfinansowanego w ramach projektu EU RPMA.01.02.00-14-9551/17-00), którego jestem kierownikiem

Opracowany algorytm GA-CAS pozwala na szeroką parametryzację. Implementacja algorytmu pozwala na określenie wartości parametrów w taki sposób aby samo istnienie wyboru nie wpływało na czas wykonywania algorytmu. Wszystkie parametry są uwzględniane na poziomie kompilacji algorytmu (ang. compile-time) i z punktu widzenia wykorzystania algorytmu dostępnych jest wiele wersji algorytmu każda o innym zestawie parametrów. Listę wszystkich parametrów które pozwalają na konfigurację algorytmu prezentuję w tabeli 6.1 wraz z krótkim opisem i odniesieniem do miejsca gdzie można znaleźć szczegółowy opis.

Nawa	Opis
POPULATION_INIT	Sposób inicjalizacji populacji, rozdział 6.2.2.1
IndividualsNo	Liczebność populacji, szczegóły rozdział 6.2.2.1, 6.2.2.2
InVitroGenNo	Liczba pokoleń in-vitro, rozdział 6.3.1
CROSSOVER_TYPE	Typ krzyżowania, szczegóły rozdział 6.2.2.3

MutationMin, MutationMax, MutationRandom	Procentowy poziom prawdopodobieństwa wystąpienia mutacji określonego typu, rozdział 6.2.2.4
CAS_TYPE	Grupowania i sortowania kontraktów w CAS_PERIOD, rozdział 5.3.1
CONTENTION	Typ rywalizacji o medium w algorytmie CAS_SEQ, rozdział 5.4.1
TRAFFIC	Poziom nadmiarowości planu w algorytmie CAS_SEQ, rozdział 5.4.1

Tabela 6.1 Lista parametrów (ang. compile-time) dla algorytmu GA-CAS

Bazując na zestawie parametrów algorytmu każdy wariant jest identyfikowany poprzez nazwę kodową obejmująca nazwę algorytmu wraz z zestawem parametrów, przykładowa nazwa kodowa to: "GA-CAS_LATENCY_MIN_MAX_10_20_S_0_0_10_DISABLED_EXCESS". Nazwa kodowa którą się posługuje konstruowana jest według poniższego wzorca, gdzie w nawiasach klamrowych znajdują się nazwy właściwych parametrów które podmieniane są na ich wartości w docelowej nazwie kodowej.

GA-CAS_{CAS_TYPE}_{POPULATION_INIT}_{IndividualsNo}_{InVitroGenNo}_{CROSSO VER_TYPE}_{MutationMin}_{MutationMax}_{MutationRandom}_{CONTENTION}_{TRAF FIC}

Dopuszczam również stosowanie fragmentów nazwy kodowej przy jednoczesnym określeniu we wspólnym opisie wartości parametrów, które nie zostały ujawnione w prezentowanym fragmencie. W szczególności dotyczy to parametrów *CAS_TYPE*, *CONTENTION*, *TRAFFIC*, które jeżeli zostały pominięte w nazwie kodowej mają wartości domyślne odpowiednio: *LATENCY*, *DISABLED*, *EXCESS*. Prezentowanie skróconej nazwy kodowej wykorzystuje szczególnie na wykresach ze względu na konieczność utrzymania kompaktowej formy opisu.

6.3 Badania eksperymentalne algorytmu GA-CAS

Ewaluacji poddano 124 warianty algorytmu GA-CAS, których nazwy kodowe umieściłem w tabeli A.1. Sumaryczny czas wykonywania optymalizacji dla wszystkich wariantów (przy ograniczeniu czasu wykonywania 30s) oraz dla pełnego wektora testowego wynosi odpowiednio dla wariantu 1 oraz 2 generowania topologii sieci:

- GATEWAY_4B 14194666s oraz 5444881s
- Stacja robocza 14092564s oraz 5774251s

Czas wykonania pojedynczej optymalizacji mierzony jest jako czas przygotowania konfiguracji sieci (conf = {RT, SCH}) na podstawie danych opisujących stan sieci G i wniosków zawarcia kontraktów C, znajdujących się w elemencie wektora testowego przy użyciu jednej wersji algorytmu. Pojedyncza optymalizacja wykonywana jest zawsze na pojedynczej jednostce arytmetyczno-logicznej (MCU) bez wykorzystywania zrównoleglenia przetwarzania.



Rysunek 6.1 Czas wykonywania optymalizacji algorytmem GA-CAS $(RANDOM_{10}_{0}S_{0}_{0}_{30})$ z podziałem na operacje w funkcji liczby węzłów dla obu wariantów generowania topologii sieci

Opracowane środowisko testowe (rozdział 4) wykorzystuje zrównoleglenie przetwarzania wektora testowego w celu przyspieszenia generowania wyników (stacja robocza) ale przy zachowaniu powyższego założenia dotyczącego pojedynczej optymalizacji.

Opracowany algorytm GA-CAS pracuje w podwójnym reżimie. Rezultat wykonania optymalizacji w postaci rozwiązania o największym poziomie przystosowania jest dostarczany po wygenerowaniu 40 pokoleń lub upływie czasu 30s (lub 60s). Różnice między wariantem 30s i 60s są oczywiście znaczące z punktu widzenia uzyskiwanego poziomu dopasowania natomiast charakter zależności oraz wartość poznawcza dla analizy jest identyczna. Z tego też punktu widzenia prezentując rezultaty ograniczam się jedynie do wariantu 30s, który z punktu widzenia zastosowania IWN jest bliższy praktycznym scenariuszom. Rezultaty wygenerowałem i prezentuję w oparciu o pracę na dwóch urządzeniach: GATEWAY_4B oraz

stacja robocza. Pomiędzy czasami wykonywania w obu tych środowiskach istnieje przybliżona zależność. Algorytm uruchomiony na stacji roboczej potrzebuje około 5,5-6 razy mniej czasu na wykonanie identycznej optymalizacji. Dlatego też prezentowany wariant wykonania algorytmów na stacji roboczej dla ograniczenia czasowego 30s możemy w przybliżeniu utożsamiać z wykonaniem na GATEWAY_4B dla ograniczenia 180s. Oczywiście obie architektury są zupełnie inne i zależność ta nie jest całkowicie liniowa, niemniej jednak dla celów porównawczych możemy się posługiwać takim uproszczeniem. Ze względu na zastosowane podwójne ograniczenie, zrozumienie zachowania algorytmów i zachodzących procesów wymaga znajomości rozkładu dwóch wielkości podlegających ograniczeniom, czyli czasu wykonania oraz liczby pokoleń.

Analizę rozpoczynam od prezentacji wyników dotyczących rozkładu czasu wykonania optymalizacji opracowanym algorytmem GA-CAS w wersji (RANDOM_10_0_S_0_0_30) w funkcji liczby węzłów dla obu wariantów generowania topologii sieci, patrz rysunek 6.1. Mierzony jest całkowity czas wykonania optymalizacji oraz osobno czasy dla trzech wydzielonych grup operacji:

- Inicjalizacja wszystkie operacje związane z inicjalizacją populacji
- Rozmnażanie wszystkie operacje związane z rekombinacją genetyczną czyli krzyżowanie i mutacja
- Wzrost wszystkie operacje związane z mapowaniem genotyp->fenotyp

Grupa "inne" obejmuje wszystkie inne operacje nie wymienione powyżej i jej wartość stanowi różnicę między całkowitym czasem wykonania optymalizacji i czasami wykonania w ramach wydzielonych grup. Ponieważ czas wykonania stanowi jedno z ograniczeń dla dostarczenia rozwiązania przez algorytm, tam gdzie widoczne jest ograniczenie wynikające z czasu, niezbędny do pełnego obrazu jest rozkład drugiego z parametrów podlegających ograniczeniu. Rozkład liczby pokoleń w funkcji liczby kontraktów z podziałem na sposób inicjalizacji populacji, prezentuję na rysunku A.12 oraz rysunku A.13, odpowiednio dla wersji 1 i 2 generowania topologii sieci. Rozkład został wykonany dla GA-CAS w wersji *_10_0_S_0_0_30, gdzie gwiazdką oznaczono sposób inicjalizacji populacji prezentowany na osobnym wykresie w odpowiadającym wierszu.

Czas wykonania na GATEWAY_4B stanowi główne ograniczenie już przy liczbie 50 węzłów, natomiast na stacji roboczej przy 90-100 węzłów (Rysunek 6.1). Dla wariantu 1 generowania topologii sieci średnia liczba pokoleń dla GATEWAY_4B spada poniżej 40 dla 24 kontraktów podczas gdy dla stacji roboczej dla około 40 kontraktów. W przypadku wariantu 2 na GATEWAY_4B już dla konfiguracji z najmniejszą badaną liczbą kontraktów wynoszącą 30, średnia liczba pokoleń jest mniejsza od limitu 40, podczas gdy na stacji roboczej ograniczenie

obserwowane jest przy 60 kontraktach. Zaprezentowany obraz jednoznacznie pokazuje, że wygenerowanie 40 zakładanych pokoleń jest możliwe jedynie w przypadku małych sieci i niewielkiej liczbie kontraktów, w pozostałych przypadkach dostarczenie optymalnego rozwiązania musi się odbyć przy ograniczeniu czasem i niewielkiej liczbie pokoleń. Dla największej badanej sieci w przypadku wariantu 1 dostarczenie rezultatu musi się odbyć po około 5-8 pokoleniach dla wariantu 2 średnio 2-3 pokoleń. Dlatego też, już na tym etapie analizy można zauważyć, że dobór populacji startowej oraz jej wielkość mogą być kluczowe w efektywnym dostarczeniu rozwiązań.

Pomiar czasu wykonania z podziałem na grupy operacji pozwala na dokonanie kolejnych istotnych obserwacji. Czas potrzebny na wykonanie operacji z grupy "inne" wynoszący średnio 10-20ms jest pomijalnie mały, co potwierdza zasadność i poprawność wydzielenia pozostałych grup. Czas potrzebny na inicjalizację populacji rośnie wykładniczo wraz ze wzrostem wielkości sieci i nie jest dominujący dla sieci o wielkości do 100 węzłów. W przypadku badanej sieci 200 węzłów (wariant 2) inicjalizacja populacji (10 osobników) w przypadku GATEWAY 4B zajmuje już na tyle dużo czasu, że przeprowadzenie pozostałych operacji nie jest możliwe. Analizując zachowanie algorytmu należy zauważyć, że optymalizacja w tym zakresie może nastąpić poprzez zastosowanie odpowiednio dobranej mniejszej populacji lub ograniczenie gęstości grafu połączeń. Zastosowanie mniejszej populacji jest dość intuicyjne i daje możliwość ograniczenia czasu jedynie w niewielkim zakresie. Wariant 2 sieci zakłada określoną gęstość urządzeń w przestrzeni, taki sposób dość dobrze oddaje scenariusze występujące w praktyce, natomiast w takim przypadku każdy węzeł ma gorsze lub lepsze (różne poziomy RSSI) połaczenie praktycznie z każdym innym węzłem. Sytuacja sprowadza się do przypadku analizy grafu połączeń o dużej gęstości. Rozwiązaniem może być zmniejszenie gęstości grafu poprzez ewidencjonowanie jedynie N najlepszych połączeń z sąsiadami. Dodatkowo istnieje dość spory margines poprawy w implementacji sposobu inicjalizacji populacji, np. poprzez utworzenie i zapamiętanie drzew rozpinających.

Porównując rozkłady liczby pokoleń dla różnych sposobów inicjalizacji (Rysunek A.12, Rysunek A.13), możemy zauważyć, że algorytm przy inicjalizacji MIN_RANDOM i RANDOM pozwala na wygenerowanie większej liczby pokoleń niż pozostałe sposoby. Najmniej pokoleń generowanych jest przy inicjalizacji typu MAX_RANDOM. Liczba generowanych pokoleń spada wykładniczo wraz ze wzrostem liczby obsługiwanych kontraktów.

W tych przypadkach gdzie dostarczenie rozwiązania odbyło się po wygenerowaniu co najmniej kilkunastu pokoleń czas potrzebny na *"wzrost"* jest dominujący i zbliżony do czasu całkowitego, kolejna co do wielkości grupa operacji *"rozmnażanie"*, potrzebuje o średnio jeden

rząd mniej czasu. Różnica pomiędzy czasem potrzebnym na "*rozmnażanie*" a czasem potrzebnym na "*wzrost*" jest tym większa im więcej pokoleń zostało wygenerowanych. W przypadku wariantu 1 dla 30 węzłów (GATEWAY_4B) różnica jest ponad 25-krotna i dla 90 węzłów jest około 8-krotna. Obserwacja prowadzi do wniosku, że operacje związane z wykonywaniem planów, są dominujące w pracy algorytmu GA-CAS. Dzieje się tak mimo zastosowania autorskich algorytmów planowania szczelin czasowych, które pozwoliły na radykalne skrócenie czasu potrzebnego na wykonanie planu (patrz 5.3 oraz 5.4). W dalszej części opracowania przedstawię autorskie rozwiązanie stanowiące próbę odpowiedzi na poczynioną obserwację. Zaproponowane rozwiązanie zakłada wykorzystanie autorskiej metody nazwanej "in-vitro" której celem jest zwiększeniem poziomu przystosowania otrzymywanego rozwiązania wykorzystując do tego pokolenia pozbawione fazy "wzrostu" i ocenę z wykorzystaniem autorskiej miary pozwalającej na estymację poziomu dopasowania osobnika bez tworzenia fenotypu.

W przypadku algorytmów GA jest szereg parametrów których zrozumienie i właściwy dobór jest kluczowy do osiągnięcia zadawalających rezultatów optymalizacji. Wszystkie parametry dla opracowanego algorytmu GA-CAS zestawiłem w tabeli 6.1. Poniżej prezentuję otrzymane wyniki optymalizacji i porównuje wpływ szeregu parametrów takich jak: sposób inicjalizacji populacji startowej, typ operatora krzyżowania, wielkość populacji oraz współczynnik mutacji.

Na rysunku 6.2 prezentuję otrzymany poziom dopasowania w funkcji liczby kontraktów, uzyskany dla wariantu * 10 0 * 0 0 30, gdzie gwiazdką oznaczyłem parametry dla których prezentuję wyniki na osobnych wykresach. W kolumnach znajdują się dane dla wariantu 1 oraz wariantu 2 generowania topologii sieci. Wiersz 1 obejmuje dane uzyskane bezpośrednio po wygenerowaniu pokolenia startowego (pokolenie 0). Bezpośrednio po inicjalizacji populacji następuje faza "wzrostu" obejmująca utworzenie fenotypu i dokonanie oceny. Te trzy elementy wykonane łącznie tworzą pokolenie 0. Wiersz 1 prezentuje dane jedynie dla GATEWAY 4B, natomiast dane dla stacji roboczej nie odbiegają w znaczący sposób. Zgodnie z rysunkiem 6.10 w przypadku wariantu 1 rozkład czasu wykonania fazy "inicjalizacja" nie zbliża się do ograniczenia czasowego, natomiast w przypadku wariantu 2 jedynie dla 200 wezłów istnieje mała liczba elementów wektora testowego dla których faza "inicjalizacja" podlega skróceniu ze względu na reżim czasowy. W takim przypadku generowana jest mniejsza populacja co będzie stanowiło o nieznacznie większych wartościach uzyskanych dla pokolenia 0 w przypadku optymalizacji na stacji roboczej dla której nawet przy 200 wezłach faza nie podlega ograniczeniom. Wiersze 2 i 3 prezentują wyniki uzyskane po zakończeniu optymalizacji dla odpowiednio GATEWAY 4B oraz stacja robocza. Używam określenia "pokolenie: max" ponieważ pokolenie na którym zakończyła się optymalizacja wynika z drugiego ograniczenia jakim jest limit czasu wynoszący 30s. W celu określenia numeru pokolenia na którym



Rysunek 6.2 Poziom dopasowania uzyskany w wyniku optymalizacji algorytmem GA-CAS (* $_10_0 *_0_30$) w funkcji liczby kontraktów dla obu wariantów generowania topologii sieci oraz z podziałem na sposób inicjalizacji populacji i użyty operator krzyżowania

skończyła się optymalizacja, należy skonfrontować wyniki z rozkładem liczby pokoleń na rysunkach A.12 oraz A.13. Zaprezentowana konwencja wykresu będzie stosowana również w dalszej części tego rozdziału przy analizie wpływu innych parametrów.

Używany poziom dopasowania jest wartością procentową uzyskaną w wyniku normalizacji wartości funkcji oceny otrzymanego rozwiązania. Normalizacja wykonywana jest poprzez dzielenie otrzymanej wartości przez wartość funkcji oceny przy założeniu zawarcia wszystkich kontraktów. Maksymalna wartość dopasowania wynosi 100%, każda wartość mniejsza od 100% oznacza że część kontraktów nie może zostać zawarta. Dodatkowo w tabelach B.50 oraz

B.51 prezentuję rozkład wartości poziomu dopasowania dla tego samego wariantu ale z podziałem na liczbę węzłów w sieci.

Sposób inicjalizacji populacji startowej w radykalny sposób wpływa na rezultat otrzymany jedynie z pokolenia 0. Podstawowym wariantem, popularnie stosowanym w literaturze [100, 101, 107, 110, 112], jest inicjalizacja populacji wyłącznie losowo (RANDOM). Wyniki otrzymane dla tego typu inicjalizacji po wygenerowaniu jedynie pokolenia 0 znacząco odbiegają od wyników otrzymanych w innych typach inicjalizacji. Wszystkie inne badane typy zakładają wykorzystanie techniki "siania" i wykorzystują inne dane do wzbogacenia populacji startowej. Najlepszy rezultat dla pokolenia 0 w przypadku obu wariantów generowania topologii sieci uzyskuje inicjalizacja MIN-MAX

Wynik otrzymany na koniec optymalizacji w przypadku wariantu 1 generowania topologii sieci stawia typ inicjalizacji RANDOM w środku otrzymanych rezultatów. Natomiast dla wariantu 2, wyłącznie losowa inicjalizacja populacji, uzyskała najgorsze rezultaty dla dużych sieci oraz znalazła się w środku otrzymanych rezultatów w przypadku sieci poniżej 100 węzłów. Ten typ inicjalizacji populacji wymaga większej liczby pokoleń aby dojść do porównywalnych, z innymi typami, rezultatów. W przypadku wariantu 1 generowania topologii sieci najlepsze rezultaty optymalizacji otrzymałem dla inicjalizacji MIN. Następnie bardzo zbliżone rezultaty uzyskały RANDOM, MIXED oraz MIN-MAX, przy czym wraz z malejącą liczbą pokoleń inicjalizacja MIN-MAX wybija się z grupy i prezentuje lepsze wyniki, w końcowej fazie zbliżając się do rezultatów MIN. Inicjalizacja typu MAX mimo, iż daje zadawalające rezultaty dla pokolenia 0 to wraz ze wzrostem liczby generowanych pokoleń zmienia swoją pozycję na tle pozostałych typów i ostatecznie plasuje się w gronie najgorszych. Dla obu wariantów generowania topologii sieci można zauważyć prawidłowość że różnice między poszczególnymi typami są tym większe im większa jest liczba obsługiwanych kontraktów (większa sieć).

Drugim parametrem GA-CAS który można porównać na rysunku 6.2 jest rodzaj użytego operatora krzyżowania (P – 1-punktowy; S – prosty). Różnica w przypadku zastosowania skrajnie innych operatorów krzyżowania nie jest znacząca. Istnieją maksymalnie 1% różnice między wynikami przy zastosowaniu obu operatorów, natomiast charakter tych zmian jest dość przypadkowy i oscyluje raz na korzyść jednego a raz na korzyść drugiego operatora. Oczywiście różnice będą się uwypuklały na korzyść operatora 1-punktowego dla mniejszych wartości współczynnika mutacji (0_0_30 w analizowanym przypadku), którego wpływ na poziom dopasowania zaprezentuję w dalszej części. Zaproponowany operator "prosty" stanowi punkt odniesienia jako najprostszy możliwy do zastosowania. W badanym scenariuszu zależy na jak najszybszym dojściu do optymalnego rozwiązania, dlatego dominującym operatorem rekombinacji powinna być mutacja, tak aby eksplorować jak największą przestrzeń



Rysunek 6.3 Poziom dopasowania uzyskany w wyniku optymalizacji algorytmem GA-CAS (*_*_0_S_0_0_30) w funkcji liczby kontraktów dla obu wariantów generowania topologii sieci oraz z podziałem na sposób inicjalizacji populacji i wielkość populacji

potencjalnych rozwiązań. Krzyżowanie powinno dominować w sytuacji gdy algorytm GA-CAS będzie dążył do dalszej optymalizacji uzyskanych wcześniej rezultatów. Docelowa konstrukcja algorytmu GA-CAS powinna uwzględniać ten fakt adaptacyjnie dostosowując stosunek między operatorem krzyżowania i mutacji do fazy pracy w której algorytm się znajduje, np. wykorzystując do tego stosunek liczby nowych kontraktów do liczby wszystkich kontraktów w optymalizowanej sieci.

Na rysunku 6.3 prezentuję wpływ wielkości populacji na uzyskiwany poziom dopasowania. Wyniki prezentuję dla różnych sposobów inicjalizacji populacji dla wariantu *_*_0_S_0_0_30,

gdzie gwiazdką oznaczyłem parametry prezentowane na osobnych wykresach. W przypadku pokolenia 0, obserwuję pozytywny wpływ dla tych sposobów inicjalizacji gdzie losowe generowanie trasy ma dominujący wpływ na wynik dopasowania. Większa liczba osobników daje większe prawdopodobieństwo znalezienia dopasowanego rozwiązania. W przypadku typu inicjalizacji z "sianiem", takich jak: MIN, MAX oraz MIN-MAX o poziomie dopasowanie w znacznym stopniu decydują osobniki odpowiednio z trasami MIN-HOP (trasa 1) i MAX-RELIABILITY (trasa 2). W takim przypadku większa liczba losowo wygenerowanych osobników nie wpływa na poziom dopasowania, istnieje mała szansa na wygenerowanie losowego osobnika o wyższym poziomie dopasowania od osobników MIN lub MAX "wstrzykniętych" do populacji. Wyniki otrzymane na koniec optymalizacji w sposób jednoznaczny prezentują negatywny wpływ wielkości populacji. Dla wszystkich badanych sposobów inicjalizacji populacji, wyniki otrzymane dla 20 osobników w populacją są gorsze od wyników uzyskanych dla populacji składającej się z 10 osobników. W ogólnej świadomości istnieje przekonanie, że większa liczba osobników i większe liczba pokoleń wpływają pozytywnie na efektywność algorytmu. Takie ogólne stwierdzenie nie potwierdza się w przypadku pracy algorytmu w reżimie czasowym. Ponieważ zarówno wielkość populacji jak i liczba pokoleń determinuje czas wykonywania optymalizacji, należy stwierdzić że przy ograniczonym czasie, przy wzroście wielkości populacji zasadniczo maleje liczba generowanych pokoleń. Wśród tych dwóch parametrów większa liczba wygenerowanych pokoleń ma bardziej pozytywny wpływ na rezultat niż większa populacja. Częstsze wykonywanie selekcji osobników do kolejnego pokolenie pozwala na uzyskanie większego gradientu zbieżności do celu optymalizacyjnego.

Na rysunku 6.4 prezentuję wpływ współczynnika mutacji określonego typu na uzyskiwany prezentuję dla W poziom dopasowania. Wyniki algorytmu wariancie MIN-MAX 10 0 S * * *, gdzie gwiazdką oznaczyłem wartości współczynnika mutacji prezentowane na osobnych wykresach. Nie zerowe wartości dla jednego z typów oznaczaja, że pozostałe typy mutacji mają współczynniki 0. Innymi słowy prezentuję wpływ poziomu mutacji określonego typu. Porównuję mutację typu RANDOM zaprezentowaną w [100] oraz wykorzystaną również w [94], z autorskimi typami operatorów MIN oraz MAX. Dodatkowo na rysunku A.14 prezentuje wpływ mutacji w konfiguracji o mieszanych typach będących połaczeniem MIN oraz RANDOM. Mimo iż operator mutacji nie ma wpływu na rezultaty pokolenia 0 (operator mutacji wykorzystywany jest do generowania osobników kolejnych pokoleń), to na wykresie prezentuję wyniki pokolenia 0 ze względu na spójność z poprzednio prezentowanymi wykresami oraz w celu zasygnalizowania użytego konkretnego sposobu inicjalizacji populacji (MIN-MAX).



Rysunek 6.4 Poziom dopasowania uzyskany w wyniku optymalizacji algorytmem GA-CAS (MIN-MAX_10_0_S_*_*_*) w funkcji liczby kontraktów dla obu wariantów generowania topologii sieci oraz z podziałem na typ i współczynnik mutacji

Otrzymane wyniki pokazują zasadniczy wpływ użytego typu mutacji oraz jej poziomu na uzyskiwane wartości dopasowania. Mutacja ma tym większy wpływ na rezultat im więcej pokoleń użytych zostało w osiągnięciu rezultatu. W przypadku wariantu 1 generowania topologii sieci przy pracy na stacji roboczej, wykresy dla poszczególnych mutacji są równo oddalone od siebie, ponieważ generowana jest ta sama liczba pokoleń (brak ograniczenia limitem czasu). W takim przypadku należy stwierdzić, że wielkość sieci i rosnąca liczba kontraktów nie wpływają na procentową różnicę między mutacjami różnych typów oraz poziomów. W przypadku wariantu 2 generowania topologii sieci, gdzie wyraźne jest

ograniczenie liczby generowanych pokoleń różnice wynikające z typu oraz poziomu mutacji zanikają wraz z malejącą liczbą pokoleń. Dla GATEWAY_4B proces zanikania różnic przebiega na krótszym zakresie kontraktów, podczas gdy na stacji roboczej proces zanikania jest wydłużony, co dobrze koreluje z prezentowanym rozkładem liczby pokoleń na rysunkach A.12 oraz A.13.

Opracowany autorski operator MIN uzyskuje najlepsze rezultaty dla obu topologii zarówno na GATEWAY 4B jak i na stacji roboczej. Operator RANDOM znajduje się w środku stawki i najgorsze rezultaty uzyskuje operator MAX. W przypadku operatorów MIN oraz RANDOM wpływ prawdopodobieństwa mutacji jest dodatnio skorelowany z rezultatem. Najlepsze rezultaty uzyskałem dla największego badanego poziomu mutacji wynoszącego 90%. W przypadku operatora MAX korelacja jest ujemna, co oznacza, że najlepsze rezultaty uzyskałem dla najmniejszego z badanych współczynników mutacji wynoszącego 10%, zwiększenie poziomu mutacji wpływało negatywnie na otrzymywane rezultaty w każdym z badanych wariantów. Na rysunku A.14 przedstawiłem porównanie mieszanych operatorów MIN oraz RANDOM w stosunku 80/10% oraz 60/30% w obie strony oraz jako punkt odniesienia najlepszych wariantów uzyskanych dla operatorów jednorodnego typu czyli 90% MIN i 90% RANDOM. Wyniki uzyskane dla badanych operatorów mieszanych mieszczą się w przedziale wartości wyznaczonym poprzez operatory jednorodne pozostawiają operator jednorodny MIN 90% jako najlepszy na przestrzeni całego wektora testowego. Operator 80 0 10 w którym 10% prawdopodobieństwa z puli operatora MIN przekazałem operatorowi RANDOM znajduje się nieznacznie poniżej wartości uzyskiwanej dla operatora jednorodnego MIN 90%. W praktycznych scenariuszach będę stosował ten operator ze względu na cel nadrzędny operatora mutacji jakim jest eksploracja przestrzeni, starając się tym samym uniknąć potencjalnie napotkanych lokalnych minimów. Zasadne jest również stosowanie adaptacyjnego doboru poziomu prawdopodobieństwa mutacji, stosując wysokie wartości w pierwszej fazie pracy GA-CAS i wartości mniejsze gdy zmiana warunków jest nieznaczna lub gdy warunki są ustalone.

6.3.1 Analiza metody in-vitro

Opracowana autorska faza rozwoju "in-vitro" stanowi odpowiedź na zauważoną w ewaluacji wyników różnicę w czasie wykonania pomiędzy fazami "rozmnażanie" a "wzrost". W tych przypadkach gdzie dostarczenie rozwiązania odbyło się po wygenerowaniu co najmniej kilkunastu pokoleń czas potrzebny na "wzrost" jest dominujący i zbliżony do czasu całkowitego. Operacje w ramach fazy "*rozmnażanie*", potrzebują o średnio jeden rząd mniej czasu. Faza rozwoju "*in-vitro*" pozwala na generowanie zdefiniowanej liczby pokoleń w których osobniki nie poddawane są pełnemu rozwojowi, poprzez całkowite wyeliminowanie fazy "*wzrost*". Faza "*in-vitro*" następuje bezpośrednio po inicjalizacji populacji startowej, a po jej zakończeniu następuje generowanie pokoleń "*pełnych*". Pokolenie "*in-vitro*", w

porównaniu z pokoleniem "pełnym", rezygnuje z wykonywania mapowanie genotyp→fenotyp, które to przekształcenie opiera się o najbardziej kosztowną czasowo operację planowania szczelin czasowych. Zarówno generowanie "pełnych" pokoleń jak również pokoleń "in-vitro" podlega ograniczeniu wynikającemu z liczby generowanych pokoleń (osobno definiowane dla każdego, Tabela 6.1) jak również ograniczeniu limitem czasu wykonywania. W celu zwiększenia różnorodności populacji 0 dla fazy pełnego "wzrostu", po zakończeniu generowania pokoleń "in-vitro" do populacji stanowiącej ostatnie pokolenie "in-vitro", analogicznie jak w przypadku inicjalizacji populacji startowej, "dosiewane" są ponownie osobniki z wszystkimi chromosomami wygenerowanymi jako Trasa 1 (MIN-HOP) oraz Trasa 2 (MAX-RELIABILITY) dokładnie te same które zostały dodane dla odpowiedniego sposobu inicjalizacji populacji startowej. Operacja "dosiania" populacji realizowana jest w inicjalizacji następujących sposobach populacji startowej POPULATION INIT:: {MIN RANDOM; MAX RANDOM; MIN MAX RANDOM}.

Ocena osobnika, która jest kluczowym elementem generowania kolejnego pokolenia, zakłada znajomość pełnych informacji o osobniku w tym jego pełnego fenotypu. Generowanie pokoleń "in-vitro" wymaga więc zastosowania metody szacowania wartości funkcji przystosowania na podstawie informacji dostępnych w genotypie. Do szacowania wartości funkcji przystosowania wykorzystuję autorską miarę przystosowania genotypu. Konstrukcja funkcji optymalizacji jest analogiczna jak zdefiniowano w (12) natomiast nie są weryfikowane warunki twarde: k_0, k_1, k_2 , ponieważ odnoszą się do wygenerowanego planu szczelin czasowych, którym nie dysponujemy w fazie "in-vitro". Zmianie ulega wartość wagi przyznawanej "możliwemu" do zawarcia kontraktowi. Wzór (13) zdefiniowany w rozdziale 3.2.4 przyjmuje zmodyfikowaną postać:

$$o_q(RT) = \sum_{\substack{c_k^{\{s,d\}} \in C \\ k}} \begin{cases} ec_k^{\{s,d\}} + laxity_k^{\{s,d\}} & if soft\left(c_k^{\{s,d\}}, conf\right) = 1 \\ 0 & otherwise \end{cases}$$
(36)

gdzie $ec_k^{\{s,d\}}$ jest bazową wagą przyznawaną "możliwemu" kontraktowi a $laxity_k^{\{s,d\}}$ definiowana jest zgodnie ze wzorem (33), przy czym wartości powyżej 0xFF są przyjmowane jako równe 0xFF. Wagi kontraktów o kolejnych wyższych priorytetach liczone są według zależności:

$$ec_{k}^{\{s,d\}} = \left\{ \sum_{x \in \{x \in C \mid x. \text{ priority} = c_{k}^{\{s,d\}}. \text{ priority} - 1 \}} ec_{x}^{\{s,d\}} \right\} \times \left(\sum_{x \in \{x \in C \mid x. \text{ priority} = c_{k}^{\{s,d\}}. \text{ priority} \}} 0xFF \right) + 1$$

$$(37)$$

Dodatkowo w stosunku do mechanizmu wag uzależnionych od priorytetu kontraktu, który jest wykorzystany w pierwotnej funkcji optymalizacyjnej, stosowana jest miara $laxity_k^{\{s,d\}}$. Zadaniem miary $laxity_k^{\{s,d\}}$ jest premiowanie tych tras które są "możliwe" i mają największą swobodę w planowaniu. Większa wartość swobody oznacza większe prawdopodobieństwo powodzenia w procesie planowania szczelin, ze względu na to, że istnieje więcej potencjalnych szczelin w których ruch może być zaplanowany zanim naruszony zostanie warunek dotyczący maksymalnego czasu transmisji pakietu. Waga została skonstruowana w taki sposób, że waga kontraktu o priorytecie p jest większa od sumy wag wszystkich możliwych do zawarcia kontraktów o priorytecie p – 1, gdzie p – 1 oznacza priorytet o 1 stopień niższy od priorytetu p na skali porównawczej. Dodatkowo waga zakłada że suma wartości przyznawanych za swobodę w planowaniu dla wszystkich kontraktów o priorytecie p jest mniejsza od bazowej wagi $ec_k^{\{s,d\}}$ dla tego priorytetu. Interpretacja definicji matematycznej jest następująca:

- Jeden zawarty kontrakt o wyższym priorytecie jest "wart" więcej niż wszystkie kontrakty o niższych priorytetach
- Zawarty kontrakt o swobodzie 0 jest "wart" więcej niż suma punktów za swobodę pozostałych kontraktów o tym samym priorytecie
- Kontrakt o wyższej swobodzie jest "wart" więcej od kontraktu o niższej swobodzie planowania.

W przypadku kontraktu cyklicznego we wzorze (15) nie jest weryfikowany warunek miękki mk_2 odnoszący się do czasu transmisji pakietu. Rolę tego warunku przejmuje miara "swobody" użyta do określenia wartości ważonej przyznawanej kontraktowi, wzór (36). Na poziomie implementacji, analogicznie jak w przypadku obliczania wartości funkcji przystosowania, poczyniłem założenia dotyczące maksymalnej liczby kontraktów w danym priorytecie i liczby priorytetów. Założenie pozwala na obliczenie wartości przy wykonaniu jednego przebiegu i sprowadza się do określenia stałej bazowej wagi dla każdego priorytetu.



Rysunek 6.5 Poziom dopasowania uzyskany w wyniku optymalizacji algorytmem GA-CAS (*_10_* $S_0_0_30$) w funkcji liczby kontraktów dla obu wariantów generowania topologii sieci oraz z podziałem na sposób inicjalizacji i liczbę pokoleń "in-vitro"

Na rysunku 6.5 prezentuję otrzymany poziom dopasowania w funkcji liczby kontraktów, uzyskany dla wariantu $*_10_*_S_0_0_30$ z fazą "in-vitro", gdzie gwiazdką oznaczyłem parametry dla których prezentuję wyniki na osobnych wykresach. Dane prezentuję w analogiczny sposób jak na poprzednich wykresach, z tą różnicą że wiersze 1 oraz 3 prezentują wyniki uzyskane bezpośrednio po wygenerowaniu wszystkich (możliwych ze względu na

ograniczenie limitem czasu) pokoleń "in-vitro" (pokolenie 0). Bezpośrednio po inicjalizacji populacji następuje faza "in-vitro" w której generowane są kolejne pokolenia pozbawione fazy "wzrost". Po wygenerowaniu wymaganej liczby pokoleń "in-vitro" lub przekroczeniu limitu czasu, ostatnie wygenerowane pokolenie "in-vitro" poddawane jest fazie "wzrost", obejmującej utworzenie fenotypu i dokonanie pełnej oceny. Wygenerowane pokolenie staje się pokoleniem 0 dla fazy pełnego "wzrostu". Wiersz 1 prezentuje dane pokolenia 0 jedynie dla GATEWAY_4B, natomiast dane dla stacji roboczej prezentowane są w wierszu 3. Zaprezentowanie, w osobnych wierszach, danych dla pokolenia 0 podyktowane jest dużą różnicą w rozkładach liczby generowanych pokoleń "in-vitro" dla obu środowisk uruchomieniowych.

Rozkład liczby pokoleń "in-vitro" oraz pokoleń "pełnych" w funkcji liczby kontraktów z podziałem na sposób inicjalizacji populacji, prezentuję na rysunkach A.15 oraz A.16, odpowiednio dla wersji 1 i 2 generowania topologii sieci. Rozkład został wykonany dla GA-CAS w wersji *_10_20_S_0_0_30, gdzie gwiazdką oznaczono sposób inicjalizacji populacji prezentowany na osobnym wykresie w odpowiadającym wierszu. W odróżnieniu od rozkładu liczby pokoleń prezentowanego wcześniej, prezentuję dwa osobne rozkłady pokoleń osobno dla fazy "in-vitro" i "pełnych" pokoleń obejmujących fazę "wzrost". W przypadku wariantu 1 generowania topologii sieci, liczba generowanych pokoleń "in-vitro" jest zawsze równa zadanej maksymalnej wynoszącej 20 (nie następuje ograniczenie limitem czasu), natomiast liczba pokoleń "pełnych" jest o średnio 1-5 pokoleń mniejsza od przypadku bez fazy "in-vitro". Różnica jest większa dla większej liczby kontraktów (wielkości sieci), co jest zgodne z wcześniejszymi wnioskami dotyczącymi czasów wykonywania faz: "rozmnażanie" oraz "wzrost". Dla wariantu 2 generowania topologii sieci, ograniczeniu podlega zarówno liczba generowanych pokoleń "in-vitro" jak również "pełnych". Suma średniej liczby pokoleń dla obu faz jest około dwa razy większa od liczby pokoleń generowanych bez fazy "in-vitro".

Prezentowane na rysunku 6.5 wyniki jednoznacznie potwierdzają że zastosowanie autorskiej fazy "in-vitro" w opracowanym algorytmie GA-CAS pozwala na uzyskanie lepszych rozwiązań w każdym z badanych przypadków oprócz wariantu z inicjalizacją populacji MIN-MAX. W przypadku zarówno GATEWAY_4B jak i stacji roboczej, najbardziej optymalne rozwiązania są generowane w wariancie inicjalizacji populacji MIN_RANDOM przy wielkości populacji "in-vitro" równej 20 pokoleń. Dla fazy "in-vitro" obejmującej 10 pokoleń wyniki są nieznaczenie (1-2%) gorsze od przypadku z liczbą 20 pokoleń. Różnice na korzyść "in-vitro" są większe w przypadku wariantu 2 generowania topologii sieci oraz w przypadku optymalizacji z użyciem mniej wydajnego urządzenia, GATEWAY_4B. Porównując działanie GA-CAS bez i z fazą "in-vitro" największy wzrost poziomu dopasowania na koniec optymalizacji jest obserwowany w przypadku inicjalizacji populacji RANDOM oraz MIXED,

gdzie dla GATEWAY 4B różnice w skrajnych przypadkach sięgają 10%. Oczywiście dla rezultatów uzyskiwanych bezpośrednio z pokolenia 0, przyrosty są o wiele większe i dla Inicjalizacji wyłącznie losowej sięgają kilkudziesięciu procent. Przypadek z inicjalizacją populacji MIN-MAX zachowuje się odmiennie, korelacja dopasowania z pokoleniem "invitro" jest ujemna. Przypadek ten pokazuje że istnieją sytuację w której generowanie pokoleń "in-vitro" może doprowadzić do rozwiązania gorszego od tego które znajduje się w populacji startowej. Sytuacja występuje dla MIN-MAX ponieważ ta populacja startowa ma największy poziom dopasowania w momencie startu. W celu uniknięcia takich przypadków integralnym elementem fazy "in-vitro" jest faza "dosiewania" populacji, która po przeprowadzeniu generowania "in-vitro" wstrzykuje do otrzymanej populacji te same osobniki które zostały wstrzyknięte w populacji startowej (MIN-HOP, MAX-RELIABILITY). Mimo iż na poziomie koncepcyjnym przewidziałem ta sytuację, to w implementacji algorytmu dla "dosiewania" założyłem limit czasowy, który eliminuje ten mechanizm jeżeli został przekroczony maksymalny limit czasu w fazie "in-vitro". Warunek ten nie jest konieczny ze względu na znikomy czas potrzebny na "dosiewanie" zwłaszcza, że wykorzystujemy już wcześniej wygenerowane osobniki, które powinny zostać zapamiętane na poziome "siania" populacji startowej i bez konieczności ich ponownego generowania, wykorzystane w fazie "dosiewania". Obecność fazy "dosiewania" bez względu na limit czasu, skutecznie eliminuje te przypadki i prezentowana na wykresach ujemna korelacja dla przypadku MIN-MAX znika. Na prezentowanych rysunkach umieściłem oryginalne dane aby dodatkowo zwrócić uwage na znaczenie techniki "dosiewania" w ramach fazy "in-vitro".

7 Podsumowanie

Praca dotyczy zagadnień optymalizacji konfiguracji sieci IWN składającej się z węzłów o ograniczonych zasobach (sprzętowych i energetycznych). W sposób szczegółowy została omówiona tematyka badań oraz stan literatury. Przegląd literatury na poziomie podstawowym przedstawiono w rozdziale 1.1 oraz rozszerzony i dopasowany do specyfiki poszczególnych problemów w rozdziałach 5.1, 6.1 oraz artykułach [19, 23, 24]. Zaprezentowana w rozdziale 1.2 teza pracy została potwierdzona w wyniku udokumentowanych badań a opracowane modele, metody i algorytmy zostały zaimplementowane z wykorzystaniem elementów sprzętowych wytworzonych w ramach projektu badawczo-rozwojowego autora pracy. Do najważniejszych osiągnięć pracy należy:

- Sformułowanie kompleksowego problemu optymalizacyjnego dotyczącego przygotowania konfiguracji IWN uwzględniającego wymagania QoS i profile energetyczne węzłów;
- Opracowanie deterministycznych algorytmów planowania szczelin czasowych CAS-PERIOD i CAS-SEQ wykorzystujących oryginalny mechanizm (CAS) unikający kolizji;
- 3. Opracowanie heurystycznego algorytmu GA-CAS, kompleksowej optymalizacji konfiguracji IWN, integrującego algorytmy CAS-PERIOD i CAS-SEQ
- 4. Zbadanie wybranych algorytmów planowania szczelin czasowych dostępnych w literaturze, analiza i opracowanie autorskich wariantów;
- Opracowanie metodyki oraz środowiska badań eksperymentalnych z wykorzystaniem prototypów rzeczywistych urządzeń opracowanych w ramach komercyjnego projektu badawczo-rozwojowego;
- 6. Analiza parametrów sieci i opracowanie reprezentatywnych wektorów testowych dla dwóch wariantów generowania topologii sieci;
- 7. Opracowanie modeli energetycznych węzłów IWN;
- Opracowanie profili zużycia energii i zależności czasowych wsparte opracowaną metodyką pomiarową dla węzłów IWN w praktycznych scenariuszach pracy (badania eksperymentalne);
- Opracowanie efektywnej metody i algorytmów pozwalających na aktywne ograniczenie wpływu dryftu zegara czasu rzeczywistego w przypadku gwałtownych zmian temperatury otoczenia dla węzłów o ograniczonych zasobach w szczególności ograniczonym bilansie energetycznym;
- Zbadanie wpływu temperatury otoczenia (badania eksperymentalne) na zdolność węzła do wypełnienia wymagań QoS;

7.1 Osiągnięcia pracy i wnioski

Opracowana kompleksowa metoda pozwala na efektywne przygotowanie i optymalizację między-warstwowej konfiguracji sieci IWN, składającej się z węzłów o ograniczonych zasobach, na podstawie wielowymiarowych wymagań QoS zgłaszanych przez procesy aplikacyjne (warstwy aplikacyjne). Kontrakt, uzgadniany z managerem sieci (SM), definiuje parametry QoS dotyczące wymiany ruchu pomiędzy węzłem wnioskującym o jego zawarcie a dowolnym węzłem docelowym. Opracowana metoda uwzględnia zarówno ruch cykliczny jak również ruch sekwencyjny dla których definiuje dedykowane parametry jakościowe. W literaturze problemy dotyczące obu rodzajów ruchu opisywane są osobno i brak jest metod pozwalających na kompleksowe podejście. Problem znalezienia konfiguracji IWN został formalnie sformułowany jako problem optymalizacyjny w rozdziale 3.2.4 w sposób obejmujący oba rodzaje ruchu, które posiadają dedykowane zestawy parametrów QoS. Dla obu rodzajów ruchu zaproponowałem modele opisujące generowanie pakietów, przy czym ruch sekwencyjny opisany jest poprzez dwa parametry (średnia i maksymalna przepływność) i modelowany jako zmodyfikowany rozkład Bernoulliego (pominięte w literaturze w odniesieniu do IWN). Ponieważ w praktycznych zastosowaniach ruch w sieciach przemysłowych należy pogrupować ze względu na priorytet, w sformułowanym problemie zaproponowałem autorską miarę opartą o wagę kontraktu pozwalającą na ocenę z uwzględnieniem priorytetu. Prezentowane metody zostały opracowane dla sieci w których urządzenia komunikują się bezpośrednio ze sobą (ruch horyzontalny) i realizują zadania autonomicznie (M2M), podczas gdy większość opracowań w literaturze, koncentruje się na algorytmach dedykowanych dla ruchu do sieci szkieletowej (ruch wertykalny).

Istotnym aspektem opracowanego rozwiązania zdefiniowanego problemu jest uwzględnienie, że zarówno węzły jak również SM są urządzeniami o ograniczonych zasobach. Ograniczenia dotyczą zarówno dostępnej pamięci operacyjnej, mocy obliczeniowej, energii oraz są na tyle istotne, że determinują dobór właściwych metod i algorytmów. Prezentowane założenie wymagało autorskiego podejścia w wielu obszarach opracowanych metod. Wymagane było opracowanie modelu energetycznego sieci IWN bazując na pomiarach fizycznych układów wykorzystywanych w realizacji projektu B+R (dofinansowanego w ramach RPMA.01.02.00-14-9551/17-00). Do charakteryzacji wykorzystano układy CC13x2 produkcji TI. Opracowany model energetyczny (rozdział 2) został uwzględniony w opracowanej metodzie optymalizacyjnej i stanowi bazę do wypełnienia jednego z wymagań QoS dotyczącego czasu obowiązywania kontraktu. Występujące, w elementach sieci, ograniczenia wymagają autorskiego podejścia do analizy efektywności metod optymalizacyjnych. W opracowanym modelu położyłem duży nacisk na zdolność znalezienia rozwiązania w ściśle określonym czasie, który nazwałem reżimem czasowym (rozdział 6.3). Prezentowane w dysertacji wyniki

dotyczą wykonywania opracowanych i porównywanych algorytmów na fizycznych elementach IWN (referencyjne) i obejmują również rozkłady czasów wykonywania. W przypadku algorytmu GA-CAS (rozdział 6), który stanowi całościowe rozwiązanie postawionego problemu, wyniki dotyczące efektywności znalezienia rozwiązania są prezentowane przy dodatkowym ograniczeniu wynikającym z maksymalnego czasu wykonywania algorytmu co odpowiada praktycznym scenariuszom.

Występujące ograniczenia oraz konieczność implementacji opracowanych rozwiązań w połączeniu z realizacją wzorcowych elementów IWN (w ramach RPMA.01.02.00-14-9551/17-00) narzuciły konieczność opracowania autorskiego modelu synchronizacji węzłów sieci przy narażeniu na gwałtowne zmiany temperatury. Zbadałem wpływ temperatury otoczenia (badania eksperymentalne) na zdolność węzła do wypełnienia wymagań QoS. Poczynione obserwacje w połączeniu z wykonaną charakteryzacją węzłów sieci pozwoliły na opracowanie autorskiej metody i algorytmów pozwalających na aktywne ograniczenie wpływu dryftu zegara czasu rzeczywistego w przypadku gwałtownych zmian temperatury otoczenia dla węzłów o ograniczonych zasobach w szczególności ograniczonym bilansie energetycznym. Opracowana metoda wykorzystuje dwa tryby LINEAR/EXP oraz regresję liniową w celu ograniczenia dryftu przy zachowaniu korzystnego bilansu energetycznego. Skuteczność metody potwierdziłem poprzez praktyczną implementację oraz serię badań symulacyjnych.

Rozwiazanie postawionego problemu optymalizacyjnego 3.2.4 dla IWN zdefiniowanego w 3.2.3 wymagało opracowania autorskiej metody z wykorzystaniem heurystycznego algorytmu GA-CAS oraz algorytmów planowania szczelin czasowych dla ruchu cyklicznego CAS-PERIOD i ruchu sekwencyjnego CAS-SEQ. Zaproponowany algorytm GAS-CAS wykorzystuje algorytm genetyczny wraz z autorską fazą IN-VITRO. Wybór algorytmu genetycznego poprzedziłem dogłębną analizą 6.1. Zadaniem GA-CAS jest dostarczenie między-warstwowej (SCH - warstwa 2; RT-warstwa 3) konfiguracji sieci na podstawie zgłaszanych kontraktów ($c_k^{\{s,d\}} \in C$) i znajomości grafu połączeń G. Autorskie algorytmy CAS-PERIOD i CAS-SEQ wykorzystują opracowany mechanizm planowania unikający kolizji bez konieczności wykonywania planu w długim okresie wynikającym z wielokrotności okresów kontraktów. Zaproponowane rozwiązanie w sposób znaczący ogranicza czas wykonywania planu oraz buduje plany w sposób umożliwiający ich efektywną dystrybucję do węzłów sieci (kompaktowość planu). Korzyści wynikające ze stosowania prezentowanych metod są więc wielowymiarowe. Pozwalają na wykonywanie planów w realnym czasie oraz ich dystrybucję do węzłów przy korzystnym bilansie energetycznym. W ramach dysertacji wykonałem implementację algorytmów GA-CAS oraz CAS-PERIOD/SEQ w środowisku rzeczywistym i poddałem ewaluacji w oparciu o dedykowane do pełnienia funkcji SM w sieci IWN urządzenie, GATEWAY_4B. Ewaluacja algorytmów obejmuje zarówno wyniki dotyczące samego procesu planowania (wykorzystania algorytmów), jak również symulacji pracy sieci w oparciu o przygotowane przez algorytmy konfiguracje sieci.

Symulacja sieci IWN realizowana w oparciu o przygotowaną przez opracowane algorytmy konfigurację wymagała opracowania autorskiego symulatora. Przeprowadzone symulacje potwierdziły zarówno poprawność przygotowanych konfiguracji jak również dostarczyły danych niezbędnych do analizy pracy samej sieci w oparciu o rozkłady czasów dostarczania pakietów oraz ilości ramek w buforach węzłów. Ewaluację opracowanych algorytmów realizowałem w oparciu o autorskie wektory testowe uwzględniające dwa warianty generowania topologii sieci oraz reprezentatywny zestaw parametrów odpowiadających praktycznym scenariuszom pracy. W literaturze częściej występuje wariant 1 (matematyczne generowanie grafu) podczas gdy wariant 2 (przestrzenne rozmieszczenie węzłów) daje bardziej praktyczne rezultaty. Ewaluacja w oparciu o oba warianty pozwala na porównania wyników, co nie jest prezentowane w literaturze. Realizacja wariantu 2 wymagała analizy wpływu przestrzennego rozmieszczenia węzłów na parametry toru radiowego oraz oczekiwany poziom niezawodności transmisji pakietu. Opracowane środowisko testowe pozwala na zrównoleglone wykonywanie badanych algorytmów dla poszczególnych elementów wektora testowego. W ramach prowadzonych badań wykorzystano wiele losowo generowanych sieci dla zadanych w pojedynczym elemencie wektora testowego parametrów. Środowisko przygotowano do pracy wielowątkowej i istnieje możliwość uruchomienia badań przy zastosowaniu specjalistycznych środowisk wieloprocesorowych (w badaniach wykorzystano 12 równoległych strumieni przetwarzania). Zastosowanie środowiska o większej skalowalności pozwoli na skrócenie badania pojedynczego wektora testowego lub umożliwi rozszerzenie wektora testowego lub zwiększenie liczby losowo generowanych konfiguracji sieci (ma znaczenie przy większej liczbie węzłów).

Opracowane w ramach dysertacji metody i algorytmy mają praktyczny wymiar i pozwalają na bezpośrednie zastosowania w środowisku przemysłowym. Użyte definicje kontraktów w tym wymagania QoS, priorytety i rodzaje ruchu są zgodne ze standardem ISA100.11a. Użyteczność i efektywność metod została potwierdzona w serii badań w środowisku symulującym warunki rzeczywiste i poparta licznymi eksperymentami. Opracowane w ramach dysertacji metody i algorytmy zostały zaimplementowane oraz wdrożone w środowisku rzeczywistym z wykorzystaniem urządzeń wytworzonych w prowadzonym przeze mnie projekcie badawczo-rozwojowym (dofinansowany w ramach RPMA.01.02.00-14-9551/17-00, 1,5 mln PLN).

7.2 Kierunki dalszych badań

Sformułowany w pracy problem optymalizacyjny dotyczący znalezienia konfiguracji sieci IWN (rozdział 3.2) oraz opracowane metody i algorytmy (GA-CAS, CAS-PERIOD, CAS-SEQ) mają duży potencjał praktycznej implementacji w sieciach IWN. W ramach dalszej ewaluacji opracowanych metod dokonam analizy danych pozyskanych, w ramach wykonanych wdrożeń, w dłuższym okresie funkcjonowania sieci IWN.

Opracowany algorytm GA-CAS i metoda optymalizacji konfiguracji sieci koduje w chromosomie tablicę routingu *RT* i mapuje w fenotypie plan transmisji *SCH*, które są kluczowe dla efektywnej konfiguracji IWN. W ramach dalszych badań należy rozszerzyć zakres informacji kodowanych w chromosomie [22] i podlegających optymalizacji o zagadnienia:

- 1. Retransmisję ramek z użyciem tego samego linku nadajnik-odbiornik;
- 2. Redundancję linku lub fragmentu trasy węzeł źródłowy-docelowy;
- 3. Dobór poziomu PDR dla którego link klasyfikowany jest jako dobry;
- 4. Dopasowanie mocy nadajnika dla konkretnego linku;

Dostępne układy transceiverów radiowych mogą pracować w szerokim zakresie częstotliwości (5GHz, 2.4GHz, 868MHz, 433MHz), szerokim zakresie przepływności i innych parametrów determinujących warstwę fizyczną (PHY). Zdolność sieci IWN do pracy z wykorzystaniem wielu alternatywnych konfiguracji warstw fizycznych zwiększa możliwości funkcjonalne i pozwala na podniesienie niezawodności komunikacji. Zdolność sieci IWN do optymalizacji konfiguracji przy wielu wykorzystywanych warstwach fizycznych, jak również zdolność węzłów do autonomicznego wyboru warstwy fizycznej która jest wykorzystywana do transmisji, przyczyni się do zwiększenia efektywności sieci IWN. Potencjalnie zakresem badań w tym obszarze może być wykorzystanie multi-path TCP (MPTCP) [123].

Wykorzystanie topologii kratownicy w sieciach IWN pozwala na zwiększenie autonomiczności. W praktyce liczba przeskoków na drodze do węzła docelowego jest mocno ograniczona, głównie ze względu na dodatkowe opóźnienie. Rozwiązaniem może być wykorzystanie węzłów z wieloma transceiverami i zastąpienie przekazywania całych ramek, przekazywaniem pojedynczych symboli. Opracowanie i analiza modelu sieci IWN wykorzystującego węzły z wieloma blokami radiowymi oraz implementacja takich układów w praktyce jest zakresem wartym eksploracji w odniesieniu do IWN i będę temu poświęcał uwagę w przyszłości.

8 Literatura

- Chen, B., Wan, J., Shu, L., Li, P., Mukherjee, M., & Yin, B. (2017). Smart Factory of Industry 4.0: Key Technologies, Application Case, and Challenges. *IEEE Access*, 6, 6505–6519. doi:10.1109/ACCESS.2017.2783682
- Masood, T., & Sonntag, P. (2020). Industry 4.0: Adoption challenges and benefits for SMEs. *Computers in Industry*, 121, 103261. doi:10.1016/J.COMPIND.2020.103261
- Ribeiro, L., & Bjorkman, M. (2018). Transitioning from Standard Automation Solutions to Cyber-Physical Production Systems: An Assessment of Critical Conceptual and Technical Challenges. *IEEE Systems Journal*, 12(4), 3816–3827. doi:10.1109/JSYST.2017.2771139
- Trappey, A. J. C., Trappey, C. V., Hareesh Govindarajan, U., Chuang, A. C., & Sun, J. J. (2017). A review of essential standards and patent landscapes for the Internet of Things: A key enabler for Industry 4.0. *Advanced Engineering Informatics*, 33, 208–229. doi:10.1016/j.aei.2016.11.007
- Yu, W., Liang, F., He, X., Hatcher, W. G., Lu, C., Lin, J., & Yang, X. (2018). A Survey on the Edge Computing for the Internet of Things. *IEEE Access, SPECIAL SECTION ON MOBILE EDGE COMPUTING*, 6, 6900–6919. doi:10.1109/ACCESS.2017.2778504
- Pan, J., & McElhannon, J. (2018). Future Edge Cloud and Edge Computing for Internet of Things Applications. *IEEE Internet of Things Journal*, 5(1), 439–449. doi:10.1109/JIOT.2017.2767608
- Li, X., Li, D., Wan, J., Vasilakos, A. V., Lai, C. F., & Wang, S. (2017). A review of industrial wireless networks in the context of Industry 4.0. *Wireless Networks*, 23(1), 23– 41. doi:10.1007/s11276-015-1133-7
- Dalmarco, G., Ramalho, F. R., Barros, A. C., & Soares, A. L. (2019). Providing industry
 4.0 technologies: The case of a production technology cluster. *The Journal of High Technology Management Research*, 30(2), 100355.
 doi:10.1016/J.HITECH.2019.100355
- Wollschlaeger, M., Sauter, T., & Jasperneite, J. (2017). The Future of Industrial Communication: Automation Networks in the Era of the Internet of Things and Industry 4.0. *IEEE industrial electronics magazine*, *11*(1), 17–27. doi:10.1109/MIE.2017.2649104
- 10. Zeid, A., Sundaram, S., Moghaddam, M., Kamarthi, S., & Marion, T. (2019).
Interoperability in Smart Manufacturing: Research Challenges. *Machines 2019, Vol. 7, Page 21, 7*(2), 21. doi:10.3390/MACHINES7020021

- Amodu, O. A., & Othman, M. (2018). Machine-to-Machine Communication: An Overview of Opportunities. *Computer Networks*, 145, 255–276. doi:10.1016/J.COMNET.2018.09.001
- Wagner, T., Herrmann, C., & Thiede, S. (2017). Industry 4.0 Impacts on Lean Production Systems. *Procedia CIRP*, 63, 125–131. doi:10.1016/J.PROCIR.2017.02.041
- Huang, V. K. L., Pang, Z., Chen, C. J. A., & Tsang, K. F. (2018). New trends in the practical deployment of industrial wireless: From noncritical to critical use cases. *IEEE Industrial Electronics Magazine*, 12(2), 50–58. doi:10.1109/MIE.2018.2825480
- Yuan, Y., Yuan, H., Ho, D. W. C., & Guo, L. (2020). Resilient Control of Wireless Networked Control System under Denial-of-Service Attacks: A Cross-Layer Design Approach. *IEEE Transactions on Cybernetics*, 50(1), 48–60. doi:10.1109/TCYB.2018.2863689
- Tan, J., Liu, A., Zhao, M., Shen, H., & Ma, M. (2018). Cross-layer design for reducing delay and maximizing lifetime in industrial wireless sensor networks. *Eurasip Journal* on Wireless Communications and Networking, 2018(1), 1–26. doi:10.1186/S13638-018-1057-X/TABLES/7
- She, C., Duan, Y., Zhao, G., Quek, T. Q. S., Li, Y., & Vucetic, B. (2019). Cross-Layer Design for Mission-Critical IoT in Mobile Edge Computing Systems. *IEEE Internet of Things Journal*, 6(6), 9360–9374. doi:10.1109/JIOT.2019.2930983
- Dobslaw, F., Zhang, T., & Gidlund, M. (2016). QoS-Aware Cross-Layer Configuration for Industrial Wireless Sensor Networks. *IEEE Transactions on Industrial Informatics*, *12*(5), 1679–1691. doi:10.1109/TII.2016.2576964
- Njah, Y., & Cheriet, M. (2021). Parallel Route Optimization and Service Assurance in Energy-Efficient Software-Defined Industrial IoT Networks. *IEEE Access*, 9, 24682– 24696. doi:10.1109/ACCESS.2021.3056931
- Kozłowski, A., & Sosnowski, J. (2019). Energy efficiency trade-off between dutycycling and wake-up radio techniques in IoT networks. *Wireless Personal Communications*, 106. doi:10.1007/s11277-019-06368-0
- Aijaz, A., & Raza, U. (2017). DeAMON: A Decentralized Adaptive Multi-Hop Scheduling Protocol for 6TiSCH Wireless Networks. *IEEE Sensors Journal*, 17(20), 6825–6836. doi:10.1109/JSEN.2017.2746183

- Salcic, Z.;;, Wang, K., Wireless, I.-K. A., Delicato, F. C., Lin, F., Pires, P. F., ... Wang, K. I.-K. (2022). Adaptive Wireless Network Management with Multi-Agent Reinforcement Learning. *Sensors 2022, Vol. 22, Page 1019, 22*(3), 1019. doi:10.3390/S22031019
- Shi, J., & Sha, M. (2019). Parameter Self-Configuration and Self-Adaptation in Industrial Wireless Sensor-Actuator Networks. *Proceedings - IEEE INFOCOM*, 2019-*April*, 658–666. doi:10.1109/INFOCOM.2019.8737467
- 23. Kozłowski, A., & Sosnowski, J. (2021). Adaptive time offset compensation for TDMA industrial wireless networks. *Ad Hoc Networks*, *119*. doi:10.1016/j.adhoc.2021.102507
- 24. Kozłowski, A., & Sosnowski, J. (2018). Evaluating energy consumption in wireless sensor networks applications. *Proceedings of SPIE, Photonics Applications in Astronomy, Communications, Industry, and High Energy Physics Experiments.* doi:10.1117/12.2501347
- Boubiche, D. E., Pathan, A. S. K., Lloret, J., Zhou, H., Hong, S., Amin, S. O., & Feki, M. A. (2018). Advanced Industrial Wireless Sensor Networks and Intelligent IoT. *IEEE Communications Magazine*, 56(2), 14–15. doi:10.1109/MCOM.2018.8291108
- Industrial communication networks Wireless communication network and communication profiles - WirelessHARTTM (IEC 62591:2016). (2016). Retrieved from https://webstore.iec.ch/publication/24433
- 27. International Electrotechnical Commission (IEC). (2015). Industrial networks Wireless communication network and communication profiles ISA 100.11 a (IEC 62734:2014).
- IEEE Std 802.15.4TM-2015, IEEE Standard for Low-Rate Wireless Networks. (2015).
 IEEE Computer Society. doi:10.1109/IEEESTD.2016.7460875
- 29. Kozłowski, A., & Sosnowski, J. (2017). Analysing efficiency of IPv6 packet transmission over 6LoWPAN network. *Proceedings of SPIE, Photonics Applications in Astronomy, Communications, Industry, and High Energy Physics Experiments, 10445*, 104451J. doi:10.1117/12.2280699
- Cheng, L., Niu, J., Luo, C., Shu, L., Kong, L., Zhao, Z., & Gu, Y. (2018). Towards minimum-delay and energy-efficient flooding in low-duty-cycle wireless sensor networks. *Computer Networks*, 134, 66–77. doi:10.1016/J.COMNET.2018.01.012
- Carrano, R. C., Passos, D., Magalhaes, L. C. S., & Albuquerque, C. V. N. (2014). Survey and taxonomy of duty cycling mechanisms in wireless sensor networks. *IEEE Communications Surveys and Tutorials*, 16(1), 181–194.

doi:10.1109/SURV.2013.052213.00116

- 32. CC26x0, CC13x0 SimpleLinkTM Wireless MCU Power Management Software Development Reference Guide. (2017). Retrieved from http://www.ti.com/lit/ug/swra486a/swra486a.pdf
- Pérez-Solano, J. J., & Felici-Castell, S. (2015). Adaptive time window linear regression algorithm for accurate time synchronization in wireless sensor networks. *Ad Hoc Networks*, 24(PA), 92–108. doi:10.1016/j.adhoc.2014.08.002
- Wu, Y.-C., Chaudhari, Q., & Serpedin, E. (2011). Clock Synchronization of Wireless Sensor Networks. *IEEE Signal Processing Magazine*, 28(1), 124–138. doi:10.1109/MSP.2010.938757
- Zhang, X., Chen, H., Lin, K., Wang, Z., Yu, J., & Shi, L. (2019). RMTS: A robust clock synchronization scheme for wireless sensor networks. *Journal of Network and Computer Applications*, 135, 1–10. doi:10.1016/j.jnca.2019.02.028
- Mani, S. K., Durairajan, R., Barford, P., & Sommers, J. (2018). An architecture for IoT clock synchronization. In *ACM International Conference Proceeding Series* (pp. 1–8). New York, NY, USA: Association for Computing Machinery. doi:10.1145/3277593.3277606
- Li, J., Mechitov, K. A., Kim, R. E., & Spencer, B. F. (2016). Efficient time synchronization for structural health monitoring using wireless smart sensor networks. *Structural Control and Health Monitoring*, 23(3), 470–486. doi:10.1002/stc.1782
- Yiğitler, H., Badihi, B., & Jäntti, R. (2020). Overview of Time Synchronization for IoT Deployments: Clock Discipline Algorithms and Protocols. *Sensors*, 20(20), 5928. doi:10.3390/s20205928
- Elsts, A., Fafoutis, X., Duquennoy, S., Oikonomou, G., Piechocki, R., & Craddock, I. (2018). Temperature-resilient time synchronization for the internet of things. *IEEE Transactions on Industrial Informatics*, 14(5), 2241–2250. doi:10.1109/TII.2017.2778746
- Schmid, T., Charbiwala, Z., Shea, R., & Srivastava, M. B. (2009). Temperature compensated time synchronization. *IEEE Embedded Systems Letters*, 1(2), 37–41. doi:10.1109/LES.2009.2028103
- Skiadopoulos, K., Tsipis, A., Giannakis, K., Koufoudakis, G., Christopoulou, E., Oikonomou, K., ... Stavrakakis, I. (2019). Synchronization of data measurements in wireless sensor networks for IoT applications. *Ad Hoc Networks*, 89, 47–57.

doi:10.1016/j.adhoc.2019.03.002

- Wang, H., Shao, L., Li, M., Wang, B., & Wang, P. (2018). Estimation of Clock Skew for Time Synchronization Based on Two-Way Message Exchange Mechanism in Industrial Wireless Sensor Networks. *IEEE Transactions on Industrial Informatics*, 14(11), 4755–4765. doi:10.1109/TII.2018.2799595
- 43. Qiu, T., Zhang, Y., Qiao, D., Zhang, X., Wymore, M. L., & Sangaiah, A. K. (2018). A robust time synchronization scheme for industrial internet of things. *IEEE Transactions* on *Industrial Informatics*, *14*(8), 3570–3580. doi:10.1109/TII.2017.2738842
- Huang, G., Zomaya, A. Y., Delicato, F. C., & Pires, P. F. (2014). Long term and large scale time synchronization in wireless sensor networks. *Computer Communications*, 37, 77–91. doi:10.1016/j.comcom.2013.10.003
- Schenato, L., & Fiorentin, F. (2011). Average TimeSynch: A consensus-based protocol for clock synchronization in wireless sensor networks. *Automatica*, 47(9), 1878–1886. doi:10.1016/j.automatica.2011.06.012
- 46. Huan, X., & Kim, K. S. (2020). On the practical implementation of propagation delay and clock skew compensated high-precision time synchronization schemes with resource-constrained sensor nodes in multi-hop wireless sensor networks. *Computer Networks*, *166*, 106959. doi:10.1016/j.comnet.2019.106959
- Xing, Y., Chen, Y., Yi, W., & Duan, C. (2015). Time Synchronization for Wireless Sensor Networks Using Adaptive Linear Prediction. *International Journal of Distributed Sensor Networks*, 2015(February). doi:10.1155/2015/917042
- Candell, R., Kashef, M., Liu, Y., Lee, K. B., & Foufou, S. (2018). Industrial wireless systems guidelines: Practical considerations and deployment life cycle. *IEEE Industrial Electronics Magazine*, *12*(4), 6–17. doi:10.1109/MIE.2018.2873820
- Akesson, B., Nasri, M., Nelissen, G., Altmeyer, S., & Davis, R. I. (2020). An Empirical Survey-based Study into Industry Practice in Real-time Systems. *Proceedings - Real-Time Systems Symposium*, 2020-Decem, 3–11. doi:10.1109/RTSS49844.2020.00012
- Shi, H., Zheng, M., Liang, W., & Zhang, J. (2019). A real-time transmission scheduling algorithm for industrial wireless sensor networks with multiple radio interfaces. *IEEE Vehicular Technology Conference*, 2019-April. doi:10.1109/VTCSpring.2019.8746481
- Lee, T. H., Chiang, H. S., Chang, L. H., Hsieh, M. C., Wen, C. H., & Yap, K. M. (2013). Modeling and performance analysis of route-over and mesh-under routing schemes in 6LoWPAN. *Proceedings - 2013 IEEE International Conference on Systems, Man, and*

Cybernetics, SMC 2013, 3802-3806. doi:10.1109/SMC.2013.649

- Tang, L., Wang, K. C., Huang, Y., & Gu, F. (2007). Channel characterization and link quality assessment of IEEE 802.15.4-compliant radio for factory environments. *IEEE Transactions on Industrial Informatics*, 3(2), 99–110. doi:10.1109/TII.2007.898414
- 53. Tanghe, E., Joseph, W., Verloock, L., Martens, L., Capoen, H., Van Herwegen, K., & Vantomme, W. (2008). The industrial indoor channel: Large-scale and temporal fading at 900, 2400, and 5200 MHz. *IEEE Transactions on Wireless Communications*, 7(7), 2740–2751. doi:10.1109/TWC.2008.070143
- Díez, V., Arriola, A., Val, I., & Velez, M. (2020). Reliability evaluation of point-to-point links based on IEEE 802.15.4 physical layer for IWSAN applications. AEU -International Journal of Electronics and Communications, 113. doi:10.1016/j.aeue.2019.152967
- Agrawal, P., Ahlen, A., Olofsson, T., & Gidlund, M. (2014). Long term channel characterization for energy efficient transmission in industrial environments. *IEEE Transactions on Communications*, 62(8), 3004–3014. doi:10.1109/TCOMM.2014.2332876
- Audeoud, H. J., & Heusse, M. (2018). Quick and efficient link quality estimation in wireless sensors networks. 2018 14th Annual Conference on Wireless On-Demand Network Systems and Services, WONS 2018 - Proceedings, 2018-Janua(Section II), 87– 90. doi:10.23919/WONS.2018.8311667
- 57. Damsaz, M., Guo, D., Peil, J., Stark, W., Moayeri, N., & Candell, R. (2017). Channel modeling and performance of Zigbee radios in an industrial environment. *IEEE International Workshop on Factory Communication Systems - Proceedings, WFCS*. doi:10.1109/WFCS.2017.7991975
- 58. Kurt, S., & Tavli, B. (2017). Path-Loss Modeling for Wireless Sensor Networks. *Ieee Antennas & Propagation Magazine*, 59(February), 18–37.
- Wassie, D. A., Rodriguez, I., Berardinelli, G., Tavares, F. M. L., Sørensen, T. B., & Mogensen, P. (2018). Radio Propagation Analysis of Industrial Scenarios within the Context of Ultra-Reliable Communication. *IEEE Vehicular Technology Conference*, 2018-June(1), 1–6. doi:10.1109/VTCSpring.2018.8417469
- 60. Wang, Z., & Crowcroft, J. (1996). Quality-of-service routing for supporting multimedia applications. *IEEE Journal on Selected Areas in Communications*, *14*(7), 1228–1234. doi:10.1109/49.536364

- 61. Ergen, S. C., & Varaiya, P. (2010). TDMA scheduling algorithms for wireless sensor networks. *Wireless Networks*, *16*(4), 985–997. doi:10.1007/s11276-009-0183-0
- Padrah, Z., Pastrav, A., Palade, T., Ratiu, O., & Puschita, E. (2021). Development and Validation of an ISA100.11a Simulation Model for Accurate Industrial WSN Planning and Deployment. *Sensors 2021, Vol. 21, Page 3600, 21*(11), 3600. doi:10.3390/S21113600
- 63. Kharb, S., & Singhrova, A. (2019). A survey on network formation and scheduling algorithms for time slotted channel hopping in industrial networks. *Journal of Network and Computer Applications*, *126*, 59–87. doi:10.1016/J.JNCA.2018.11.004
- 64. Teles Hermeto, R., Gallais, A., & Theoleyre, F. (2017). Scheduling for IEEE802.15.4-TSCH and slow channel hopping MAC in low power industrial wireless networks: A survey. *Computer Communications*, *114*, 84–105. doi:10.1016/J.COMCOM.2017.10.004
- Duquennoy, S., Al Nahas, B., Landsiedel, O., & Watteyne, T. (2015). Orchestra: Robust Mesh Networks Through Autonomously Scheduled TSCH. In SenSys '15: Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems (pp. 337–350). doi:10.1145/2809695.2809714
- Osman, M., & Nabki, F. (2021). OSCAR: An Optimized Scheduling Cell Allocation Algorithm for Convergecast in IEEE 802.15.4e TSCH Networks. *Sensors 2021, Vol. 21, Page 2493, 21*(7), 2493. doi:10.3390/S21072493
- Kotsiou, V., Papadopoulos, G. Z., Chatzimisios, P., & Theoleyre, F. (2020). LDSF: Low-Latency Distributed Scheduling Function for Industrial Internet of Things. *IEEE Internet* of Things Journal, 7(9), 8688–8699. doi:10.1109/JIOT.2020.2995499
- Shi, K., Zhang, L., Qi, Z., Tong, K., & Chen, H. (2019). Transmission scheduling of periodic real-time traffic in IEEE 802.15.4e TSCH-Based industrial mesh networks. *Wireless Communications and Mobile Computing*, 2019. doi:10.1155/2019/4639789
- Nguyen, V. D., Sharma, S. K., Vu, T. X., Chatzinotas, S., & Ottersten, B. (2021). Efficient Federated Learning Algorithm for Resource Allocation in Wireless IoT Networks. *IEEE Internet of Things Journal*, 8(5), 3394–3409. doi:10.1109/JIOT.2020.3022534
- Ren, J., Yu, G., He, Y., & Li, G. Y. (2019). Collaborative Cloud and Edge Computing for Latency Minimization. *IEEE Transactions on Vehicular Technology*, 68(5), 5031– 5044. doi:10.1109/TVT.2019.2904244

- Sun, W., Liu, J., Yue, Y., & Zhang, H. (2018). Double Auction-Based Resource Allocation for Mobile Edge Computing in Industrial Internet of Things. *IEEE Transactions on Industrial Informatics*, 14(10), 4692–4701. doi:10.1109/TII.2018.2855746
- Dobslaw, F., Zhang, T., & Gidlund, M. (2014). End-to-End Reliability-Aware Scheduling for Wireless Sensor Networks. *IEEE Transactions on Industrial Informatics*, *12*(2), 758–767. doi:10.1109/TII.2014.2382335
- 73. Jin, Y., Kulkarni, P., Wilcox, J., & Sooriyabandara, M. (2016). A centralized scheduling algorithm for IEEE 802.15.4e TSCH based industrial low power wireless networks. *IEEE Wireless Communications and Networking Conference, WCNC*, (Wcnc). doi:10.1109/WCNC.2016.7565002
- Palattella, M. R., Accettura, N., Grieco, L. A., Boggia, G., Dohler, M., & Engel, T. (2013). On optimal scheduling in duty-cycled industrial iot applications using IEEE802.15.4e TSCH. *IEEE Sensors Journal*, 13(10), 3655–3666. doi:10.1109/JSEN.2013.2266417
- Subramanian, A., Garcia, M. J., Callaway, D. S., Poolla, K., & Varaiya, P. (2013). Realtime scheduling of distributed resources. *IEEE Transactions on Smart Grid*, 4(4), 2122– 2130. doi:10.1109/TSG.2013.2262508
- 76. Chilukuri, S., Piao, G., Lugones, Di., & Pesch, Di. (2021). Deadline-Aware TDMA Scheduling for Multihop Networks Using Reinforcement Learning. 2021 IFIP Networking Conference, IFIP Networking 2021. doi:10.23919/IFIPNETWORKING52078.2021.9472801
- 77. T. Watteyne, E., Palattella, M., & Grieco, L. (2015). Using IEEE 802.15.4e Time-Slotted Channel Hopping (TSCH) in the Internet of Things (IoT): Problem Statement. Internet Engineering Task Force (IETF). Retrieved from https://datatracker.ietf.org/doc/html/rfc7554
- Scanzio, S., Vakili, M. G., Cena, G., Demartini, C. G., Montrucchio, B., Valenzano, A., & Zunino, C. (2020). Wireless Sensor Networks and TSCH: A Compromise between Reliability, Power Consumption, and Latency. *IEEE Access*, 8, 167042–167058. doi:10.1109/ACCESS.2020.3022434
- 79. Carhacioglu, O., Zand, P., & Nabi, M. (2018). Cooperative Coexistence of BLE and Time Slotted Channel Hopping Networks. *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC, 2018-Septe.* doi:10.1109/PIMRC.2018.8580918

- Veisi, F., Nabi, M., & Saidi, H. (2020). Coexistence Analysis of Multiple Asynchronous IEEE 802.15.4 TSCH-Based Networks. *IEEE Access*, 8, 150573–150585. doi:10.1109/ACCESS.2020.3016770
- Zhang, L., Liang, Y. C., & Xiao, M. (2019). Spectrum Sharing for Internet of Things: A Survey. *IEEE Wireless Communications*, 26(3), 132–139. doi:10.1109/MWC.2018.1800259
- 82. Gomes, R. D., Alencar, M. S., Queiroz, D. V, Fonseca, I. E., & Benavente-Peces, C. (2017). Comparison between Channel Hopping and Channel Adaptation for Industrial Wireless Sensor Networks. In *Proceedings of the 6th International Conference on Sensor Networks (SENSORNETS 2017)*. Porto, Portugal. doi:10.5220/0006206800870098
- Dakdouk, H., Tarazona, E., Alami, R., Féraud, R., Papadopoulos, G. Z., & Maillé, P. (2018). Reinforcement learning techniques for optimized channel hopping in IEEE 802.15.4-TSch networks. *MSWiM 2018 Proceedings of the 21st ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, 18, 99–107. doi:10.1145/3242102.3242110
- 84. Palattella, M. R., Accettura, N., Dohler, M., Grieco, L. A., & Boggia, G. (2012). Traffic aware scheduling algorithm for reliable low-power multi-hop IEEE 802.15.4e networks. *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC*, 327–332. doi:10.1109/PIMRC.2012.6362805
- Palattella, M. R., Accettura, N., Grieco, L. A., Boggia, G., Dohler, M., & Engel, T. (2013). On optimal scheduling in duty-cycled industrial iot applications using IEEE802.15.4e TSCH. *IEEE Sensors Journal*, 13(10), 3655–3666. doi:10.1109/JSEN.2013.2266417
- Shi, H., Zheng, M., Liang, W., & Zhang, J. (2019). Convergecast Scheduling for Industrial Wireless Sensor Networks with Local Available Channel Sets. *IEEE Sensors Journal*, 19(22), 10764–10772. doi:10.1109/JSEN.2019.2929672
- He, Y., Han, G., Xu, M., & Martinez-Garcia, M. (2021). A Pseudo-Packet Scheduling Algorithm for Protecting Source Location Privacy in the Internet of Things. *IEEE Internet of Things Journal*. doi:10.1109/JIOT.2021.3117957
- Yang, D., Xu, Y., Wang, H., Zheng, T., Zhang, H., Zhang, H., & Gidlund, M. (2015). Assignment of segmented slots enabling reliable real-time transmission in industrial wireless sensor networks. *IEEE Transactions on Industrial Electronics*, 62(6), 3966– 3977. doi:10.1109/TIE.2015.2402642

- Sood, T., & Sharma, K. (2018). A comparative analysis on the Scheduling Algorithms for Wireless Sensor Networks. 2018 13th International Conference on Industrial and Information Systems, ICHS 2018 - Proceedings, 236–245. doi:10.1109/ICHNFS.2018.8721433
- 90. Yan, M., Lam, K. Y., Han, S., Chan, E., Chen, Q., Fan, P., ... Nixon, M. (2014). Hypergraph-based data link layer scheduling for reliable packet delivery in wireless sensing and control networks with end-to-end delay constraints. *Information Sciences*, 278, 34–55. doi:10.1016/j.ins.2014.02.006
- Saifullah, A., Xu, Y., Lu, C., & Chen, Y. (2010). Real-time scheduling for WirelessHART networks. *Proceedings - Real-Time Systems Symposium*, 150–159. doi:10.1109/RTSS.2010.41
- 92. Pathak, A. A., & Deshpande, V. S. (2015). Buffer management for improving QoS in WSN. 2015 International Conference on Pervasive Computing: Advance Communication Technology and Application for Society, ICPC 2015. doi:10.1109/PERVASIVE.2015.7087068
- 93. Petersen, S., & Carlsen, S. (2011). WirelessHART Versus ISA100.11a: The Format War Hits the Factory Floor. *IEEE Industrial Electronics Magazine*, 5(4), 23–34. doi:10.1109/MIE.2011.943023
- Yussof, S., & See, O. H. (2010). A robust GA-based QoS routing algorithm for solving multi-constrained path problem. *Journal of Computers*, 5(9), 1322–1334. doi:10.4304/jcp.5.9.1322-1334
- Mehboob, U., Qadir, J., Ali, S., & Vasilakos, A. (2016). Genetic algorithms in wireless networking: techniques, applications, and issues. Soft Computing (Vol. 20). Springer Berlin Heidelberg. doi:10.1007/s00500-016-2070-9
- 96. Dasgupta, D., & Michalewicz, Z. (2013). *Evolutionary Algorithms in Engineering Applications*. Springer Science & Business Media.
- 97. Konak, A., Coit, D. W., & Smith, A. E. (2006). Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering and System Safety*, 91(9), 992– 1007. doi:10.1016/J.RESS.2005.11.018
- Chen, S., Newman, T. R., Evans, J. B., & Wyglinski, A. M. (2010). Genetic algorithmbased optimization for cognitive radio networks. *33rd IEEE Sarnoff Symposium 2010, Conference Proceedings*. doi:10.1109/SARNOF.2010.5469780
- 99. Lorenzo, B., & Glisic, S. (2013). Optimal routing and traffic scheduling for multihop

cellular networks using genetic algorithm. *IEEE Transactions on Mobile Computing*, *12*(11), 2274–2288. doi:10.1109/TMC.2012.204

- Ahn, C. W., & Ramakrishna, R. S. (2002). A genetic algorithm for shortest path routing problem and the sizing of populations. *IEEE Transactions on Evolutionary Computation*, 6(6), 566–579. doi:10.1109/TEVC.2002.804323
- 101. Yang, S., Cheng, H., & Wang, F. (2010). Genetic algorithms with immigrants and memory schemes for dynamic shortest path routing problems in mobile ad hoc networks. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, 40(1), 52–63. doi:10.1109/TSMCC.2009.2023676
- Tinós, R., & Yang, S. (2007). A self-organizing random immigrants genetic algorithm for dynamic optimization problems. *Genetic Programming and Evolvable Machines*, 8(3), 255–286. doi:10.1007/S10710-007-9024-Z/TABLES/7
- Yu, X., Tang, K., Chen, · Tianshi, Yao, X., Yu, X., Tang, · K, … Cercia, X. Y. (2008). Empirical analysis of evolutionary algorithms with immigrants schemes for dynamic optimization. *Memetic Computing 2008 1:1*, 1(1), 3–24. doi:10.1007/S12293-008-0003-6
- 104. Yu, X., Tang, K., & Yao, X. (2008). An immigrants scheme based on environmental information for genetic algorithms in changing environments. 2008 IEEE Congress on Evolutionary Computation, CEC 2008, 1141–1147. doi:10.1109/CEC.2008.4630940
- Branke, J. (1999). Memory enhanced evolutionary algorithms for changing optimization problems. *Proceedings of the 1999 Congress on Evolutionary Computation, CEC 1999*, 3, 1875–1882. doi:10.1109/CEC.1999.785502
- Yang, S. (2008). Genetic algorithms with Memory- and elitism-based immigrants in dynamic environments. *Evolutionary Computation*, 16(3), 385–416. doi:10.1162/EVCO.2008.16.3.385
- 107. Cheng, H., & Yang, S. (2010). Multi-population Genetic Algorithms with Immigrants Scheme for Dynamic Shortest Path Routing Problems in Mobile Ad Hoc Networks. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 6024 LNCS(PART 1), 562–571. doi:10.1007/978-3-642-12239-2 58
- Branke, J., Kaussler, T., Smidt, C., & Schmeck, H. (2000). A Multi-population Approach to Dynamic Optimization Problems. *Evolutionary Design and Manufacture*, 299–307. doi:10.1007/978-1-4471-0519-0_24

- 109. Michalak, K. (2021). Evolutionary Algorithm Using Random Immigrants for the Multiobjective Travelling Salesman Problem. *Procedia Computer Science*, 192, 1461– 1470. doi:10.1016/J.PROCS.2021.08.150
- 110. López, R. B., Sanchez, S. M., Fernandez, E. M. G., Souza, R. D., & Alves, H. (2014). Genetic algorithm aided transmit power control in cognitive radio networks. *Proceedings* of the 2014 9th International Conference on Cognitive Radio Oriented Wireless Networks and Communications, CROWNCOM 2014, 61–66. doi:10.4108/ICST.CROWNCOM.2014.255393
- 111. Yeniay, Ö. (2005). Penalty Function Methods for Constrained Optimization with Genetic Algorithms. *Mathematical and Computational Applications 2005, Vol. 10, Pages 45-56, 10*(1), 45–56. doi:10.3390/MCA10010045
- 112. Yen, Y. S., Chao, H. C., Chang, R. S., & Vasilakos, A. (2011). Flooding-limited and multi-constrained QoS multicast routing based on the genetic algorithm for MANETs. *Mathematical and Computer Modelling: An International Journal*, 53(11–12), 2238– 2250. doi:10.1016/J.MCM.2010.10.008
- 113. Kunzel, G., Indrusiak, L. S., & Pereira, C. E. (2020). Latency and Lifetime Enhancements in Industrial Wireless Sensor Networks: A Q-Learning Approach for Graph Routing. *IEEE Transactions on Industrial Informatics*, 16(8), 5617–5625. doi:10.1109/TII.2019.2941771
- 114. Dobslaw, F., Zhang, T., & Gidlund, M. (2015). Latency improvement strategies for reliability-aware scheduling in industrial wireless sensor networks. *International Journal of Distributed Sensor Networks*, 2015. doi:10.1155/2015/178368
- 115. De Couto, D. S. J., Aguayo, D., Bicket, J., & Morris, R. (2005). A high-throughput path metric for multi-hop wireless routing. *Wireless Networks 2005 11:4*, 11(4), 419–434. doi:10.1007/S11276-005-1766-Z
- Friedrich, T., & Wagner, M. (2015). Seeding the initial population of multi-objective evolutionary algorithms: A computational study. *Applied Soft Computing*, *33*, 223–230. doi:10.1016/J.ASOC.2015.04.043
- 117. Al-Fuqaha, A., Khreishah, A., Guizani, M., Rayes, A., & Mohammadi, M. (2015). Toward better horizontal integration among IoT services. *IEEE Communications Magazine*, 53(9), 72–79. doi:10.1109/MCOM.2015.7263375
- 118. Kramer, O. (2017). *Genetic Algorithm Essentials* (Vol. 679). Cham: Springer International Publishing. doi:10.1007/978-3-319-52156-5

- Lamini, C., Benhlima, S., & Elbekri, A. (2018). Genetic Algorithm Based Approach for Autonomous Mobile Robot Path Planning. *Procedia Computer Science*, *127*, 180–189. doi:10.1016/J.PROCS.2018.01.113
- 120. Cooper, P. (2017). Data, information, knowledge and wisdom. *Anaesthesia & Intensive Care Medicine*, 18(1), 55–56. doi:10.1016/J.MPAIC.2016.10.006
- Dumitrescu, D., Lazzerini, B., & Jain Lakhmi C. (2000). Evolutionary computation. CRC Press, Inc. Retrieved from https://dl.acm.org/doi/book/10.5555/350139
- 122. Munetomo, M., Yamaguchi, N., Akama, K., & Sato, Y. (2001). Empirical investigations on the genetic adaptive routing algorithm in the internet. *Proceedings of the IEEE Conference on Evolutionary Computation, ICEC, 2,* 1236–1243. doi:10.1109/CEC.2001.934332
- 123. Morawski, M., & Ignaciuk, P. (2022). Choosing a Proper Control Strategy for Multipath Transmission in Industry 4.0 Applications. *IEEE Transactions on Industrial Informatics*, 18(6), 3609–3619. doi:10.1109/TII.2021.3105499

A. Dodatki



A.1. Struktura zbioru danych dla środowiska badań eksperymentalnych

Rysunek A.1 Model koncepcyjny zbiorów i ich związków, przedstawiający środowisko testowe wykorzystywane w ewaluacji algorytmów planowania oraz optymalizacji wraz z symulacją



A.2. Plan aktywności szczelin czasowych TDMA

Rysunek A.2 Plan aktywności szczelin czasowych TDMA dla różnych wariantów sieci



A.3. Komplementarna dokumentacja wyników eksperymentalnych

Rysunek A.3 Poziom zaplanowania w funkcji liczby kontraktów dla wariantu 1 generowania topologii sieci z podziałem na trasę oraz liczbę dostepnych kanałów



Rysunek A.4 Poziom zaplanowania w funkcji liczby kontraktów dla wariantu 2 generowania topologii sieci z podziałem na trasę oraz liczbę kanałów



Rysunek A.5 Maksymalny czas transmisji pakietu w funkcji liczby przeskoków dla wariantu 2 generowania topologii sieci z podziałem na liczbę kanałów oraz $|S|/N_V$ i $N_C/|S|$



Rysunek A.6 Odchylenie standardowe czasu transmisji pakietu w funkcji liczby przeskoków dla wariantu 2 generowania topologii sieci z podziałem na liczbę kanałów oraz $|S|/N_V$ i $N_C/|S|$



Rysunek A.7 Rozkład procentowy czasu pracy węzła sieci w funkcji liczby ramek w buforze transmisji dla wariantu 1 i 2 generowania topologii sieci – kontrakty cykliczne (góra) oraz sekwencyjne (dół)



Rysunek A.8 Poziom zaplanowania w funkcji liczby kontraktów dla wariantu 1 generowania topologii sieci



Rysunek A.9 Poziom zaplanowania w funkcji liczby kontraktów dla wariantu 2 generowania topologii sieci



Rysunek A.10 Maksymalny czas transmisji pakietu w funkcji liczby przeskoków dla wariantu 2 generowania topologii sieci z podziałem na liczbę kanałów oraz $|S|/N_V$ i $N_C/|S|$ przy CAS-PERIOD



Rysunek A.11 Odchylenie standardowe czasu transmisji pakietu w funkcji liczby przeskoków dla wariantu 2 generowania topologii sieci z podziałem na liczbę kanałów oraz $|S|/N_V$ i $N_C/|S|$ przy CAS-PERIOD



Rysunek A.12 Rozkład liczby pokoleń dla optymalizacji GA-CAS (*_10_0_S_0_0_30) w funkcji liczby kontraktów dla wariantu 1 generowania topologii sieci z podziałem na sposób inicjalizacji populacji



Rysunek A.13 Rozkład liczby pokoleń dla optymalizacji GA-CAS (*_ $10_0S_0_30$) w funkcji liczby kontraktów dla wariantu 2 generowania topologii sieci z podziałem na sposób inicjalizacji populacji



Rysunek A.14 Poziom dopasowania uzyskany w wyniku optymalizacji algorytmem GA-CAS (MIN-MAX_10_0_S_*_*_*) w funkcji liczby kontraktów dla obu wariantów generowania topologii sieci oraz z podziałem na różne warianty mieszanej mutacji



*Rysunek A.15 Rozkład liczby pokoleń "pełnych" oraz "in-vitro" dla optymalizacji GA-CAS (*_10_20_S_0_0_30) w funkcji liczby kontraktów dla wariantu 1 generowania topologii sieci z podziałem na sposób inicjalizacji populacji*



*Rysunek A.16 Rozkład liczby pokoleń "pełnych" oraz "in-vitro" dla optymalizacji GA-CAS (*_10_20_S_0_0_30) w funkcji liczby kontraktów dla wariantu 2 generowania topologii sieci z podziałem na sposób inicjalizacji populacji*

A.4. Lista wariantów algorytmu GA-CAS

/**

* Crossover: simple */ GA_CAS_LATENCY_RANDOM_10_0_S_0_0_30, GA_CAS_LATENCY_MIN_10_0_S_0_0_30, CA_CAS_LATENCY_MAX_10_S_0_0_30

 $\begin{array}{l} GA_CAS_LATENCY_MAX_10_0_S_0_0_30,\\ GA_CAS_LATENCY_MIXED_10_0_S_0_0_30,\\ \end{array}$

GA_CAS_LATENCY_MIN_MAX_10_0_S_0_0_10, GA_CAS_LATENCY_MIN_MAX_10_0_S_0_0_30, GA_CAS_LATENCY_MIN_MAX_10_0_S_0_0_90,

GA_CAS_LATENCY_MIN_MAX_10_0_S_10_0, GA_CAS_LATENCY_MIN_MAX_10_0_S_30_0, GA_CAS_LATENCY_MIN_MAX_10_0_S_90_0,

GA_CAS_LATENCY_MIN_MAX_10_0_S_0_10_0, GA_CAS_LATENCY_MIN_MAX_10_0_S_0_30_0, GA_CAS_LATENCY_MIN_MAX_10_0_S_0_90_0,

GA_CAS_LATENCY_MIN_MAX_10_0_S_10_10, GA_CAS_LATENCY_MIN_MAX_10_0_S_20_20_20, GA_CAS_LATENCY_MIN_MAX_10_0_S_30_30,

GA_CAS_LATENCY_MIN_MAX_10_0_S_80_0_10, GA_CAS_LATENCY_MIN_MAX_10_0_S_60_0_30, GA_CAS_LATENCY_MIN_MAX_10_0_S_10_0_80, GA_CAS_LATENCY_MIN_MAX_10_0_S_30_0_60,

/**

* Population: 20

*/

GA_CAS_LATENCY_RANDOM_20_0_S_0_0_30, GA_CAS_LATENCY_MIN_20_0_S_0_0_30, GA_CAS_LATENCY_MAX_20_0_S_0_0_30, GA_CAS_LATENCY_MIXED_20_0_S_0_0_30, GA_CAS_LATENCY_MIN_MAX_20_0_S_0_0_30,

/**

* Crossover: single point

*/

 $\begin{array}{l} GA_CAS_LATENCY_RANDOM_10_0_P_0_0_30,\\ GA_CAS_LATENCY_MIN_10_0_P_0_0_30,\\ GA_CAS_LATENCY_MAX_10_0_P_0_0_30,\\ GA_CAS_LATENCY_MIXED_10_0_P_0_0_30,\\ \end{array}$

GA_CAS_LATENCY_MIN_MAX_10_0_P_0_0_10, GA_CAS_LATENCY_MIN_MAX_10_0_P_0_0_30, GA_CAS_LATENCY_MIN_MAX_10_0_P_0_0_90,

GA_CAS_LATENCY_MIN_MAX_10_0_P_10_0, GA_CAS_LATENCY_MIN_MAX_10_0_P_30_0, GA_CAS_LATENCY_MIN_MAX_10_0_P_90_0_0,

GA_CAS_LATENCY_MIN_MAX_10_0_P_0_10_0, GA_CAS_LATENCY_MIN_MAX_10_0_P_0_30_0, GA_CAS_LATENCY_MIN_MAX_10_0_P_0_90_0,

GA_CAS_LATENCY_MIN_MAX_10_0_P_10_10, GA_CAS_LATENCY_MIN_MAX_10_0_P_20_20_20, GA_CAS_LATENCY_MIN_MAX_10_0_P_30_30,

GA_CAS_LATENCY_MIN_MAX_10_0_P_80_0_10, GA_CAS_LATENCY_MIN_MAX_10_0_P_60_0_30, GA_CAS_LATENCY_MIN_MAX_10_0_P_10_0_80, GA_CAS_LATENCY_MIN_MAX_10_0_P_30_0_60, /**

* In-Vitro: 10 */

 $\begin{array}{l} GA_CAS_LATENCY_RANDOM_10_10_S_0_0_30,\\ GA_CAS_LATENCY_MIN_10_10_S_0_0_30,\\ GA_CAS_LATENCY_MAX_10_10_S_0_0_30,\\ GA_CAS_LATENCY_MIXED_10_10_S_0_0_30,\\ GA_CAS_LATENCY_MIN_MAX_10_10_S_0_0_30,\\ \end{array}$

**		
*	In-Vitro:	20

```
*/
```

GA_CAS_LATENCY_RANDOM_10_20_S_0_0_30, GA_CAS_LATENCY_MIN_10_20_S_0_0_30, GA_CAS_LATENCY_MAX_10_20_S_0_0_30, GA_CAS_LATENCY_MAX_10_20_S_0_0_30,

GA_CAS_LATENCY_MIN_MAX_10_20_S_0_0_10, GA_CAS_LATENCY_MIN_MAX_10_20_S_0_0_30, GA_CAS_LATENCY_MIN_MAX_10_20_S_0_0_90,

GA_CAS_LATENCY_MIN_MAX_10_20_S_10_0, GA_CAS_LATENCY_MIN_MAX_10_20_S_30_0, GA_CAS_LATENCY_MIN_MAX_10_20_S_90_0,

GA_CAS_LATENCY_MIN_MAX_10_20_S_0_10_0, GA_CAS_LATENCY_MIN_MAX_10_20_S_0_30_0, GA_CAS_LATENCY_MIN_MAX_10_20_S_0_90_0,

GA_CAS_LATENCY_MIN_MAX_10_20_S_10_10_10,

/**

* In-Vitro: 10; Crossover: single point */

GA_CAS_LATENCY_RANDOM_10_10_P_0_0_30, GA_CAS_LATENCY_MIN_10_10_P_0_0_30, GA_CAS_LATENCY_MAX_10_10_P_0_0_30, GA_CAS_LATENCY_MAX_10_10_P_0_0_30,

GA_CAS_LATENCY_MIN_MAX_10_10_P_0_0_30,

/**

* In-Vitro: 20; Crossover: single point */

GA_CAS_LATENCY_RANDOM_10_20_P_0_0_30, GA_CAS_LATENCY_MIN_10_20_P_0_0_30, GA_CAS_LATENCY_MAX_10_20_P_0_0_30, GA_CAS_LATENCY_MIXED_10_20_P_0_0_30,

GA_CAS_LATENCY_MIN_MAX_10_20_P_0_0_10, GA_CAS_LATENCY_MIN_MAX_10_20_P_0_0_30, GA_CAS_LATENCY_MIN_MAX_10_20_P_0_0_90,

GA_CAS_LATENCY_MIN_MAX_10_20_P_10_0_0, GA_CAS_LATENCY_MIN_MAX_10_20_P_30_0_0, GA_CAS_LATENCY_MIN_MAX_10_20_P_90_0_0,

GA_CAS_LATENCY_MIN_MAX_10_20_P_0_10_0, GA_CAS_LATENCY_MIN_MAX_10_20_P_0_30_0, GA_CAS_LATENCY_MIN_MAX_10_20_P_0_90_0,

Tabela A.1 Lista wariantów algorytmu GA-CAS poddanych ewaluacji