INNOWACYJNA GMINA

Projektowanie i optymalizacja

pod redakcją
Mirosława Hajdera

Rzeszów 2015

Monografia

INNOWACYJNA GMINA

Projektowanie i optymalizacja

pod redakcją naukową

Mirosława Hajdera

Rzeszów, 2015

Recenzja naukowa:
Prof. dr hab. inż. Heorhii Loutskii,
dr inż. Paweł Różycki

Redakcja naukowa:
Mirosław Hajder

Redakcja i korekta:
Agnieszka Hajder

Projekt okładki i stron tytułowych:
Andrii Sydorak, Przemysław Nycz

Skład, łamanie i przygotowanie do druku:
Mirosław Hajder, Mariusz Nycz

Pomysł monografii:
Mirosław Hajder, Janusz Kolbusz, Mariusz Nycz

ISBN 978-83-64286-47-6 książka w oprawie miękkiej
ISBN 978-83-64286-48-3 on-line pdf

© Copyright by Wyższa Szkoła Informatyki i Zarządzania z siedzibą w Rzeszowie, 2015

Wydawca:
Wyższa Szkoła Informatyki i Zarządzania z siedzibą w Rzeszowie
35-225 Rzeszów, ul. Sucharskiego 2
www.wsiz.rzeszow.pl
e-mail: wsiz@wsiz.rzeszow.pl, tel. 17-866-11-11
Księgarnia internetowa: www.ksiegarnia.wsiz.pl

WYŻSZA SZKOŁA
INFORMATYKI I ZARZĄDZANIA
z siedzibą w Rzeszowie

SPIS TREŚCI

ROZDZIAŁ 1

Mirosław HAJDER, Piotr HAJDER, Tomasz BARTCZAK: *Komponenty procesu projektowania obiektów i systemów technicznych* 9

ROZDZIAŁ 2

Piotr HAJDER, Mirosław HAJDER, Mariusz NYCZ: *Projektowanie, produkcja i testowanie aplikacji dla urządzeń mobilnych* 29

ROZDZIAŁ 3

Mariola NYCZ, Piotr HAJDER: *Zrównoważony rozwój transportu miejskiego w Rzeszowie na tle innych miast* 49

ROZDZIAŁ 4

Mirosław HAJDER: *Architektura informatyczna bezprzewodowych systemów monitoringu środowiskowego* 61

ROZDZIAŁ 5

Mirosław HAJDER, Janusz KOLBUSZ: *Efektywna lokalizacja węzłów pomiarowych i agregujących w sieciach monitoringu* 75

ROZDZIAŁ 6

Piotr HAJDER, Mariusz NYCZ: *Dekompozycja grafu-modelu sieci pomiarów środowiskowych* 89

ROZDZIAŁ 7

Mariusz NYCZ, Michał WROŃSKI: *Ataki typu odmowa usługi na serwery webowe: podatność, wykrywanie, przeciwdziałanie* 103

ROZDZIAŁ 8

Mirosław HAJDER, Piotr HAJDER: *Etap rekonfiguracji w cyklu życia systemów informatycznych* 119

ROZDZIAŁ 9

Mirosław HAJDER, Piotr HAJDER: *Struktury połączeniowe w dobie sieci obliczeń brzegowych*.....137

WSTĘP

Kolejny, piąty już tom cyklicznej monografii pt. *Innowacyjna gmina* koncentruje się na dwóch obszarach tematycznych. Pierwszy z nich to zastosowanie sieci sensorowych do monitorowania produkcji przemysłowej i rolnej, infrastruktury technicznej, środowiska oraz osób. Drugi wiodący temat to wykorzystanie optymalizacji w produkcji, projektowaniu, budowie oraz eksploatacji obiektów technicznych.

Chociaż w wielu krajach sieci sensorowe są szeroko wykorzystywane już od ponad dekady, w Polsce ich zastosowanie jest raczej symboliczne. Ich istota sprowadza się do zastosowania dużej ilości, gęsto rozmieszczonych, niskokosztowych, miniaturowych urządzeń pomiarowo-sterujących, dokonujących lokalnego pomiaru interesujących nas parametrów. Sieci sensorowe należą do rodziny systemów bezprzewodowych, ich węzły są w pełni bezobsługowe, dysponują autonomicznym zasilaniem i wykorzystywane są głównie do obsługi czujników pomiarowych i elementów wykonawczych. Zapewnienie komunikacji pomiędzy węzłami końcowymi sieci a jej centralnym węzłem jest jednym z najważniejszych realizowanych zadań. Obsługiwane może być również bezpośrednie połączenie pomiędzy węzłami końcowymi sieci, ale zawsze jest to zadanie drugoplanowe. Gęste rozmieszczanie węzłów sieci sensorowej, narzuca m. in. architektura wykorzystywanych urządzeń transmisyjnych, posiadających niewielką wydajność przy małym zużyciu energii. Uniwersalny charakter węzłów pozwala wykorzystywać je do rozwiązywania szerokiego spektrum zadań, w szczególności: monitoringu i sterowania na niedostępnych lub zagrożonych obszarach.

W ostatnim dziesięcioleciu, pojawiło się wiele publikacji, głównie zagranicznych autorów, poświęconych różnym aspektom wykorzystania sieci sensowych. Tematyka prac rozciąga się od zagadnień budowy i zastosowania urządzeń nadawczo-odbiorczych, poprzez architekturę mikrokontrolerów sterujących węzłami wraz z wbudowanym w nie oprogramowaniem, po zagadnienia związane z organizacją pracy sieci, jej projektowaniem oraz poprawą charakterystyk eksploatacyjnych poszczególnych węzłów i sieci jako całości.

W monografii, prezentowane są wyniki badań autorów obejmujących poszukiwanie formalnych metod rozwiązania wybranych problemów projektowania, budowy różnych obiektów technicznych związanych z teleinformatyką . Wszystkie zaprezentowane wyniki zostały zweryfikowane praktycznie i mogą być z powodzeniem wykorzystywane do rozwiązywania innych zbliżonych zadań. Drugim, wiodącym tematem tomu jest wykorzystanie optymalizacji do poprawy efektywności działalności biznesowej, produkcji rolnej oraz projektowania obiektów technicznych. O ile optymalizacja jest obecnie jednym z podstawowych narzędzi pracy naukowców, to jej zastosowanie w biznesie, w szczególności w produkcji rolnej, jest niezadowalające. Aktualnie, metody i środki optymalizacji są szeroko stosowane

przez duże międzynarodowe korporacje oraz średnie przedsiębiorstwa, a ich użycie w dominujących w polskiej gospodarce niewielkich firmach rodzinnych jest znikome. W znacznej mierze, wynika to z oderwania prowadzonych i publikowanych badań naukowych od rzeczywistych potrzeb małego biznesu.

Proponując przedsiębiorcom zastosowanie optymalizacji w działalności biznesowej często spotykamy się z dwoma nieprawdziwymi tezami. Zgodnie z pierwszą, optymalizacja jest złożoną dyscypliną naukową, której wykorzystanie wymaga dogłębnej wiedzy matematycznej. Teza ta była słuszna w latach, kiedy rozwiązanie takich zadań wymagało samodzielnego opracowania i uruchamiania dedykowanych programów komputerowych. Obecnie, dostępna jest ogromna ilość bezpłatnych i komercyjnych uniwersalnych aplikacji, zapewniających nadzwyczaj proste komponowanie procedur poszukiwania rozwiązań zadań optymalizacji. Celowo użyto tutaj słowa *komponowanie*, bowiem najczęściej poszukiwanie rozwiązania nie wymaga napisania chociażby jednego wiersza programu i zazwyczaj ogranicza się do przypisywania odpowiednich funkcji do grupy wybranych komórek tabeli.

Druga błędna teza zakłada, że w małych rodzinnych firmach i gospodarstwach rolnych objętość przetwarzanych danych oraz złożoność procesu decyzyjnego są na tyle znikome, że zastosowanie w nich metod optymalizacji jest zbędne. Również i to twierdzenie jest od podstaw niepoprawne. Optymalizacja wykonywana metodami formalnymi pozwala uwzględnić powiązania pomiędzy wieloma czynnikami wpływającymi na końcową efektywność, co przy zastosowaniu metod intuicyjnych jest absolutnie niemożliwe.

Dotąd, jednym z najważniejszych wymagań stawianych przed autorami rozdziałów było doświadczenie praktyczne w opisywanym przez nich obszarze. Również w bieżącym numerze, każdy z rozdziałów prezentuje rzeczywiste zadanie, którego rozwiązanie znajduje się w obszarze potrzeb i zainteresowań przemysłu, rolnictwa lub administracji. Zachęcamy wszystkich zaciekawionych opisywanymi projektami do kontaktu z autorami, w celu poszukiwania obopólnie korzystnych form współpracy, mającej na celu lepsze dostosowanie prowadzonych badań do potrzeb sfery produkcji i usług oraz do dzielenia się podobnymi doświadczeniami.

Podobnie jak wszystkie poprzednie tomy *Innowacyjnej gminy*, niniejszy jest w pełni bezpłatny i można go odnaleźć w większości bibliotek cyfrowych

Mirosław Hajder

ROZDZIAŁ 1

Komponenty procesu projektowania obiektów i systemów technicznych

Mirosław HAJDER♠, Piotr HAJDER♦, Tomasz BARTCZAK♠

♠*Wyższa Szkoła Informatyki i Zarządzania z siedzibą w Rzeszowie,*
♦*Akademia Górniczo-Hutnicza w Krakowie*
{*Miroslaw.Hajder, Piootr.Hajder, TBartczak10*}*@gmail.com*

Streszczenie

W rozdziale autorzy analizują procedurę projektowania złożonych systemów technicznych, koncentrując się na teleinformatycznych komponentach infrastruktury krytycznej. Rozpoczyna się on od zdefiniowania projektowania jako procesu, porównania różnych jego określeń oraz przedstawienia celowości wykorzystania w nim metod formalnych. Dalej, składniki procesu projektowania przypisano do faz cyklu życia oraz stadiów i etapów projektowania, przeanalizowano ich wzajemne relacje. Szczegółowo przeanalizowano modelowanie jako element procesu projektowania, zaproponowano klasyfikacje modeli oraz metod modelowania. Dalej, usystematyzowano wykorzystywane w trakcie projektowania teleinformatyczne komponenty infrastruktury krytycznej. W podsumowaniu zaprezentowano wyniki badań, dotyczących samodzielnej analizy efektywności rozwiązań projektowych i biznesowych w mikroprzedsiębiorstwach, gospodarstwach rolnych oraz samorządach lokalnych. Rozdział adresowany jest do osób zajmujących się zarządzaniem procedurami inwestycyjnym oraz podejmowaniem decyzji o charakterze biznesowym.

1. Wstęp

W naszym bezpośrednim otoczeniu funkcjonuje ogromna liczba różnorodnych obiektów będących systemami złożonymi. Część z nich, tzw. *systemy ekologiczne*, jest produktem długotrwałej ewolucji środowiska naturalnego, pozostałe zaś, określane jako *systemy techniczne*, powstały dla zaspokojenia potrzeb współczesnej cywilizacji. Analiza systemów ekologicznych wykracza poza obszar tematyczny rozdziału, obejmujący analizę i syntezę obiektów stworzonych ręką człowieka [1], [2], [3], którymi są złożone systemy techniczne.

Systemy techniczne to rezultat świadomych i uporządkowanych działań, których jednym z początkowych etapów jest projektowanie. *Projektowaniem* nazywamy ukierunkowany proces twórczy, mający na celu przygotowanie opisu obiektu lub systemu. Aby osiągnąć najlepsze efekty, podejmując decyzje projektowe, należy rozumnie ograniczać rolę intuicji. Podstawą współczesnego procesu projektowania powinna być *formalizacja*, którą nazywamy ścisłe i bezwzględne przestrzeganie wymogów, określonej ustalonymi normami procedury działania. To właśnie dzięki formalizacji, z projektowania eliminujemy subiektywizm – szkodliwą pochodną intuicji. Formalizacja jest również podstawą szerokiego wykorzystania komputerowych technologii projektowania. Dzięki nim możliwe stało się zwiększenie liczby parametrów i charakterystyk opisujących obiekty, analiza szerszego obszaru alternatywnych rozwiązań, a także poprawa jakości samego projektu oraz przyspieszenie jego przygotowania.

Zainteresowanie autorów teorią projektowania wynika ze znikomego wykorzystania jej metod przez samorządy lokalne, mikroprzedsiębiorstwa oraz gospodarstwa rolne. Niestety, wśród ww. podmiotów, samodzielne planowanie przedsięwzięć biznesowych, a także tworzenie nowych systemów i obiektów technicznych opiera się praktycznie wyłącznie na metodach intuicyjnych. Skutkiem tego są: zwiększone ryzyko prowadzenia działalności biznesowej, niska efektywność wykorzystywanych rozwiązań technicznych, zwiększone koszty ich budowy i eksploatacji. Wyniki badań, zaprezentowane w monografii pokazują, że właściciele niewielkich, prywatnych podmiotów gospodarczych są skłonni samodzielnie weryfikować jakość rozwiązań, oferowanych przez wynajętych projektantów i doradców, nawet jeżeli miałoby się to wiązać z dodatkowymi kosztami. W przypadku samorządów dominuje przekonanie, że powyższe prace są bezsprzecznie celowe, jednak niezbędne czynności powinny być zlecane niezależnym podmiotom zewnętrznym. Przytoczone w kolejnych rozdziałach wyniki utwierdzają nas w przekonaniu, że należy poszerzać wiedzę o projektowaniu, w szczególności wśród osób zatwierdzających projekty i plany biznesowe.

Oczekujemy, że skutkiem lektury rozdziału będzie poprawa konkurencyjności mikroprzedsiębiorstw, gospodarstw oraz samorządów najniższego szczebla, bazująca na zastosowaniu ogólnodostępnych narzędzi wspomagających podejmowanie trafnych decyzji biznesowych i projektowych.

2. Definicje procesu projektowania

Obecnie, trudno wskazać jedną uniwersalną definicję projektowania. Każde z funkcjonujących określeń ściśle wiąże się z konkretnym obszarem: jedne z definicji odzwierciedlają kolejność działań, inne zaś akcentują wymagania jakościowe i ilościowe, stawiane przed projektowanym obiektem lub systemem. Projektowanie może obejmować tworzenia nowego, jak i modernizację istniejącego obiektu lub systemu, zgodnie z wymaganiami stawianymi przez użytkowników technologie lub usługi świadczone z jego zastosowaniem. Potocznie, *projektowanie* definiuje

się jako skończony ciąg powiązanych ze sobą relacjami kroków (czynności), prowadzących do osiągnięcia zamierzonego celu, którym jest przygotowanie opisu obiektu spełniającego zdefiniowane wymagania [4], [5], [6], [7], [8], [9], [10].

Pojęcie projektowania szczegółowo opisują międzynarodowe standardy oraz ich polskie odpowiedniki. Zgodnie z normą ISO 24765, *projektowanie* to proces określenia architektury, komponentów, interfejsów i innych charakterystyk systemu lub jego części. Wynikiem wykonanych prac jest *projekt* będący jednolitym zbiorem modeli, właściwości lub parametrów opisanych w postaci nadającej się do praktycznej realizacji. Powyższe definicje, w równym stopniu odnoszą się do każdego z możliwych wariantów projektowania, od tworzenia nowych obiektów, przez projektowanie rozbudowy istniejących, po ich kompleksową modernizację.

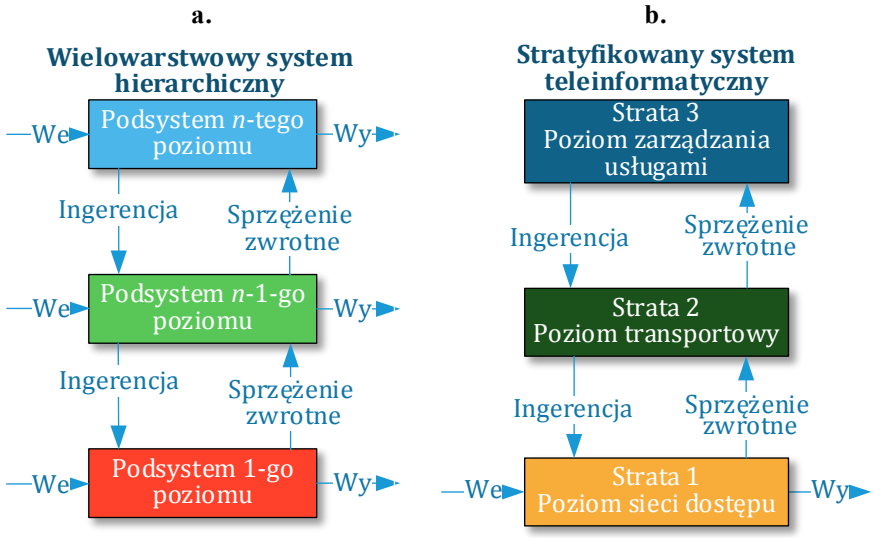
W znanych definicjach zakłada się, że celem projektowania jest opracowanie szczegółowego opisu systemu. Czym zatem jest system, jak można go opisać? Zgodnie z encyklopedyczną definicją, *systemem* nazywamy zbiór elementów z określonymi powiązaniami i relacjami, posiadający całościowy charakter. Już prosta analiza definiuje podstawowe właściwości systemów, którymi są: integralność; względna izolacja od otoczenia; istnienie ograniczonych powiązań ze środowiskiem zewnętrznym; fragmentacja struktury z wyraźnymi powiązaniami pomiędzy częściami; podporządkowanie organizacji systemu określonemu celowi. Z punktu widzenia teorii projektowania, system można zdefiniować również jako środek *osiągnięcia* stawianych celów. Definicja ta jest szczególnie przydatna w przypadku analizy ekonomicznej obiektów lub systemów [1], [2].

Otaczającą nas rzeczywistość należy traktować jako zbiór stale rozwijających się systemów i tworzących je podsystemów. Choć złożoność analizy tych stworzonych ręką człowieka jest wielokrotnie niższa od większości systemów naturalnych, to systemy techniczne stają się coraz bardziej skomplikowane. Tworzące je podsystemy, powiązane są za pomocą rozbudowanych procesów współdziałania, w szczególności wymiany energii, materii i informacji. Z punktu widzenia wzajemnych relacji podsystemów, systemy można sklasyfikować na powiązane sieciowo lub hierarchicznie. Chociaż większość systemów może być opisana za pomocą jednej z powyższych organizacji, z punktu widzenia teorii projektowania szczególnie interesującym jest drugi typ relacji. *Hierarchicznym* nazywać będziemy system złożony ze zbioru wertykalnie powiązanych podsystemów, przy czym w klasycznej postaci każdy z nich przejawia wpływ wyłącznie na sąsiednie warstwy.

Systemy hierarchiczne, podobnie jak wszystkie inne typy systemów, zbudowanych w oparciu o powiązane ze sobą podsystemy, odnosimy do grupy *systemów złożonych* [1], [2], [3]. Dzięki zjawisku synergii, zbiór powiązanych ze sobą podsystemów uzyskuje nowe właściwości, którymi nie dysponują indywidualnie tworzące go elementy. Podobnie jak klasyczne systemy, systemy złożone posiadają następujące cechy: tworzy je znaczna ilość powiązanych i oddziałujących na siebie komponentów; prezentuje je złożony formalny opis; opisuje rozmyte określenie

systemowych funkcji celu działania; charakteryzuje *synergiczność* – system jako całość posiada właściwości, którymi nie dysponuje żaden z jego elementów; są one podatne na oddziaływanie wielu, w szczególności losowych czynników zewnętrznych; wykorzystują człowieka do komunikacji pomiędzy komponentami systemu oraz między systemem, a jego otoczeniem [1], [2], [11]. Z pozoru oczywistą wydaje się teza, że pracochłonność projektowania systemów złożonych jest wielomianową funkcją ilości komponentów oraz skomplikowania wykorzystywanych połączeń. Jednak zwiększenie skali obiektu nie musi powodować wielomianowego wzrostu pracochłonności jego projektowania, czego najlepszym przykładem jest tworzenie systemów hierarchicznych.

Klasyczną organizację systemu hierarchicznego przedstawiono na rys. 1.a. Składa się on ze zbioru podsystemów zorganizowanych w poziomy, każdy z których może posiadać własny opis zachowania w prywatnej przestrzeni zmiennych i parametrów. Podsystem wyższego poziomu koordynuje działanie podsystemu poziomu niższego, co w teorii systemów hierarchicznych nazywamy ingerencją [12]. W większości systemów tej klasy poprawne funkcjonowanie wymaga nie tylko interwencji ukierunkowanej do dołu hierarchii, ale również reakcji skierowanej odwrotnie, co wykonywane jest za pomocą sprzężenia zwrotnego.



Rys. 1. Graficzna ilustracja systemów hierarchicznych: a. Wielowarstwowy system hierachiczny; b. Stratowa reprezentacja rozległego systemu teleinformatycznego

Klasyczna teoria systemów hierarchicznych wyróżnia trzy rodzaje poziomów hierarchii: straty, warstwy i eszelony [12]. W hierarchiach *stratyfikowanych*, każda ze strat opisywana jest za pomocą zbioru modeli, z punktu widzenia wymaganego poziomu abstrakcji, odpowiadających określonemu zachowaniu opisywanego systemu. Na rys. 1.b rozległy system teleinformacyjny został przedstawiony jako spełniający powyższe wymagania, trójpoziomowy system stratyfikowany.

W hierarchiach *warstwowych* zakłada się, że złożone algorytmy funkcjonowania można przedstawić w formie powiązanych sekwencyjnie, prostszych podproblemów, których kompleksowe rozwiązanie jest jednocześnie rozwiązaniem kompletnego problemu źródłowego. W hierarchiach tego typu każdy ze współdziałających podsystemów powinien być dokładnie wydzielony z całości systemu [12]. W hierarchiach *eszelonowych* poszczególne poziomy, reprezentujące fragmenty opisu systemu, złożone ze zbioru wydzielonych, współdziałających ze sobą komponentów, są rozmieszczane w hierarchii tak, że określone elementy są sterowane za pomocą innych komponentów, nazywanych elementami zarządzającymi [11].

Projektowanie systemów hierarchicznych jako całości jest trudne w realizacji. Charakteryzuje je znaczna liczba zmiennych i parametrów, pokaźna ilość krzyżowych powiązań, a także skomplikowanie modeli matematycznych, opisujących procesy interakcji. Jednym z rozwiązań problemu złożoności jest wykorzystanie hierarchiczności struktury poprzez autonomiczne projektowanie podsystemów oraz ich hierarchicznych powiązań. Aby synteza taka była możliwa, konieczne jest spełnienie kilku podstawowych warunków. Po pierwsze, każdy z podsystemów powinien realizować jedną jedyną funkcję, która musi być jasna i zrozumiała, bez względu na jej skomplikowanie. Po drugie, połączenie pomiędzy podsystemami powinno występować tylko w przypadku istnienia powiązań pomiędzy ich funkcjami i musi być maksymalnie proste.

Podział systemu złożonego na straty, warstwy lub eszelony, nazywamy jego *dekompozycją*. Najczęściej, dekompozycja złożonych systemów hierarchicznych grupuje podsystemy w warstwach, nazywanych także poziomami. Ilość poziomów i podsystemów mogą się różnić, jednak podsystemy bezpośrednio wchodzące w różne warstwy powinny wypełniać pełny zakres przewidzianych tam działań. Możliwa jest także pełna rozłączność funkcjonalna poszczególnych podsystemów, tj. każdy z nich jest wykorzystywany jednokrotnie.

Dekompozycja systemów może być doskonałym sposobem zmniejszenia złożoności projektowania. Jeżeli w jej rezultacie powiązane równorzędnie mniejsze elementy zastąpimy zestawem powtarzających się podsystemów, procedura projektowania całości może się uprościć. Wykorzystując specyficzne właściwości systemów hierarchicznych, można zapewnić liniowy, a nie wykładniczy wzrost złożoności projektowania w funkcji rozmiaru systemu. Zauważmy jednak, że projektowanie systemów hierarchicznych budzi istotne wątpliwości. Teoretycznie, hierarchiczność systemu złożonego powoduje wzrost złożoności wzajemnych powiązań pomiędzy komponentami, jak również ilości opisujących go parametrów i charakterystyk. Zazwyczaj, konsekwencją tego są nieakceptowalne złożoności: pamięciowa i czasowa procesu projektowania. Po drugie, wykorzystując metody teorii systemów hierarchicznych [11], [12], [13], projektowanie można podzielić na względnie niezależne etapy wydzielając w nim fragmenty z akceptowalnym poziomem złożoności. Przykładowo, projektując hierarchiczną sieć komputerową, etapy tworzenia rdzenia sieci i sieci dostępowych rozpatrywane są, jako dwa niezależne

procesy projektowe. Niestety, podział zadania na etapy pogarsza jakość projektu – tworzone rozwiązania są często nieoptymalne, wzrasta również prawdopodobieństwo wystąpienia błędu [1], [14]. Pomijając wysoką złożoność, projektowanie jednoetapowe zapewnia uzyskanie optymalnych rozwiązań, odpowiadających rosnącym oczekiwaniom użytkowników. Wymaga ono jednak wykorzystania nowych, często niedostępnych metod i środków modelowania matematycznego i technik przetwarzania komputerowego.

Najczęściej, projektowanie bazuje na metodach będących zestawem zasad tworzenia zbioru elementów systemu i łączenia ich relacjami. Nie istnieje jedna uniwersalna metoda, stosowana do projektowania wszelkich obiektów, a projektowanie tych samych systemów może odbywać się na bazie szerokiego asortymentu metod, różniących się wykorzystywanymi technikami analizy i syntezy [15], [16], [17], [18]. Projektowanie to również proces wielokrokowy, na każdym z kroków którego wykorzystywane są różne metody. Możliwą, aczkolwiek rzadszą, jest również sytuacja, kiedy te same metody stosowane są na różnych krokach procedury projektowej.

3. Fazy, stadia i etapy życia obiektu lub systemu

Dyscypliną naukową, szeroko wykorzystywaną przy syntezie i analizie nowych obiektów i systemów, jest metodologia badań naukowych. Z jej punktu widzenia, dowolny obiekt może znajdować się na różnych fazach cyklu życia oraz stadiach i etapach projektowania. Klasyfikacja tych spośród nich, które powiązane są bezpośrednio z interesującym nas projektowaniem, została pokazana na rys. 2.

FAZY	STADIA	ETAPY
Faza projektowania	Koncepcyjne	Wyszukanie sprzeczności
		Definiowanie problemu
		Określenie problematyki
		Określenie celu
		Wybór kryteriów
	Modelowania	Budowa modeli
		Optymalizacja
		Decyzje projektowe
	Konstruowania	Dekompozycja
		Agregacja
		Badanie warunków
		Tworzenie programu
	Przygotowania technicznego	Tworzenie dokumentacji wykonawczej obiektu
Faza technologiczna	Praktyczna realizacja zaprojektowanego obiektu	
Faza refleksyjna	Porównanie i ocena obiektu	

Rys. 2. Fazy, stadia i etapy życia obiektu

Dla potrzeb niniejszego rozdziału, bazując na metodologii badań naukowych, zmodyfikujemy klasyczne rozumienie cyklu życia. W miejsce tradycyjnych etapów, wyróżnimy trzy podstawowe fazy: projektowania, technologiczną oraz refleksyjną. W trakcie fazy *projektowania*, tworzony jest jednoznaczny opis obiektu, wykorzystywany dalej, w *fazie technologicznej*, do jego fizycznej realizacji. Stosowana do tego dokumentacja wykonawcza opisuje zastosowanie różnorodnych technologii dostępnych inwestorowi.

Interesująca nas faza projektowania rozbita została na cztery stadia: koncepcyjne, modelowania, konstruowania i przygotowania technicznego, stopniowo uszczegóławiające projektowany obiekt lub system. Zgodnie z rys. 2, fazę projektowania rozpoczyna etap *wyszukiwania sprzeczności*, należący do stadium koncepcyjnego danej fazy. W jego trakcie, w analizowanym obszarze, wyszukiwane są sprzeczności w działalności praktycznej. W szczególności określa się przeszkody, utrudniające uzyskanie oczekiwanych rezultatów działalności stosując do tego metody analizy systemowej, pozwalające wykryć nie jedną, a całą grupę sprzeczności oraz powiązań pomiędzy nimi. Jak pokazują wyniki wieloletnich analiz różnych badaczy, sprzeczności należy poszukiwać we wszystkich aspektach funkcjonowania obiektu lub systemu, nawet tych najmniej prawdopodobnych [1], [17]. Problemy pojawiające się na tym stadium projektowania zilustrujemy przykładem dotyczącym sieci komputerowej. Jako bazę sprzętową takich systemów projektanci chętnie wykorzystują urządzenia firmy Cisco. Analiza konkurencyjnych rozwiązań takich producentów jak: 3Com, DLink, Microtek i in. pokazuje daleko idącą zbieżność parametrów funkcjonalnych i charakterystyk użytkowych oferowanego sprzętu, w szczególności w przypadku urządzeń niższej klasy. Chociaż cena urządzeń Cisco jest 2-3 krotnie wyższa niż dostępnych analogów, projektanci wykorzystują je chętniej niż produkty konkurencyjnych firm. Analiza wskazuje dwa powszechnie znane powody takiego postępowania. Po pierwsze, Cisco posiada największą sieć kształcenia w obszarze technologii teleinformatycznych. Szkolenie użytkowników rozpoczyna się już w gimnazjach, kontynuowane jest w szkołach średnich, uczelniach wyższych oraz komercyjnych centrach szkoleniowych. Po drugie, Cisco jako jedna z niewielu firm oferuje pełen zakres sprzętu i technologii, na bazie których można rozwiązać praktycznie każde zadanie, pojawiające się w obszarze teleinformatyki.

Na kolejnym etapie (*definiowanie problemu*) stadium koncepcyjnego fazy projektowania, ze zbioru wykrytych sprzeczności należy wybrać jedną, najważniejszą, tworzącą tzw. sytuację kluczową. Określają ją sprzeczności, których istnienie zostało już udowodnione i zaakceptowane, lecz póki co nie są znane sposoby likwidacji bądź minimalizacji ich wpływu. Kolejnym etapem procedury z rys. 2 jest *określenie problematyki*. W analizie systemowej, problematyka to zbiór zagadnień nierozerwalnie związanych z podlegającym rozwiązaniu problemem. W działaniach praktycznych, konieczność określenia problematyki wynika ze złożoności projektowanych obiektów i systemów, które zazwyczaj składają się z wielu komponentów lub podsystemów. W procesie projektowania konieczne jest uwzględnienie

wszystkich dyscyplin naukowych, zaangażowanych w funkcjonowanie tworzonego systemu. Potrzeba określenia problematyki, to również konsekwencja interdyscyplinarności prac projektowych. Zauważmy, że każdy projektant korzysta w swojej pracy z szeregu dyscyplin pokrewnych wchodząc w ten sposób w wiedzę i interesy innych grup społecznych. Klasyfikując uczestników projektowania, możemy wśród nich wyróżnić: osoby bezpośrednio zaangażowane w proces projektowania (kierownicy działów, wykonawcy i odbiorcy zaangażowani w projekt), w szczególności odpowiedzialne za podejmowanie decyzji; bezpośrednich wykonawców (analitycy, projektanci, programiści, testerzy), których działania są niezbędne do praktycznego rozwiązania zadań projektowych; aktywnych odbiorców, w przyszłości wykorzystujących efekty projektu i zgłaszających uwagi do jego realizacji; pasywnych odbiorców zachowujących się biernie w trakcie realizacji zadania projektowego, a w przyszłości będących konsumentami rezultatów prac projektowych; użytkowników z negatywnym nastawieniem do projektu, przejawiających jawną lub ukrytą wrogość do realizowanego zadania. W powyższej klasyfikacji można uwzględnić również osoby w żaden sposób niepowiązane z realizowanym projektem, mogące z powodów ambicjonalnych wspierać lub utrudniać jego realizację. Z wieloletnich doświadczeń autorów wynika, że najliczniejszą grupę zawsze tworzą przeciwnicy projektu. Nagminnie zdarza się, że przytaczane przez nich argumenty nie mają żadnego uzasadnienia w praktyce.

Określając problematykę (kolejny etap stadium koncepcyjnego), mamy do czynienia z zadaniem grupowego podejmowania decyzji – każdy z członków zespołu projektowego może mieć własny stosunek do konkretnego sposobu rozwiązania zadania. Zauważmy, że rozwiązanie każdego z analizowanych w procesie projektowania problemów może wiązać się z rozwiązaniem istniejących i/lub pojawieniem się nowych problemów. W takim przypadku, wypracowanie wspólnego stanowiska wszystkich członków zespołu może okazać się bardzo trudne.

Kolejnym etapem stadium koncepcyjnego jest *określenie celu* projektowania. Cel wybierany jest na podstawie zdefiniowanego problemu oraz problematyki i w potocznym rozumieniu sprowadza się do określenia sposobów rozwiązania tego pierwszego. Etap ten jest uważany za jeden z najważniejszych w całej fazie projektowania, m. in. dlatego że pozostałe etapy bazują właśnie na nim. W ograniczonym sensie, określanie problemu i celu są do siebie zbliżone. W pierwszym przypadku, określamy niesatysfakcjonujące nas elementy rzeczywistości, w drugim zaś opisujemy nasze wyobrażenie o niej. Określenie celu jest zadaniem, którego pełna formalizacja jest bardzo trudna, wręcz niemożliwa. Zazwyczaj jest ono połączeniem logiki i intuicji, co z uwagi na brak powiązania pomiędzy logicznym i intuicyjnym rozwiązaniem problemu czyni z etapu wyjątkowo złożone zadanie.

Stadium koncepcyjne kończy etap wyboru kryteriów. W działalności praktycznej i naukowej dobór kryteriów jest jedną z najbardziej złożonych prac, wykonywanych w procesie rozwiązania zadań projektowych. Na danym etapie ma miejsce

przejście od celów do kryteriów projektowania, które należy rozpatrywać jako ilościowe modele jakościowych celów. Kryteria powinny maksymalnie odwzorowywać cele i być zbieżne z nimi. Z drugiej jednak strony, kryteria projektowania nie mogą powielać celów, bowiem są one szacowane w odmienny sposób: cele podlegają wyłącznie wyszczególnieniu, z kolei kryteria powinny być wyrażone z użyciem konkretnej skali pomiarowej. Skutki błędnego wyboru kryteriów są trudne do przeszacowania. Świetnym tego przykładem może posłużyć sposób wynagradzania strażaków stosowany w Chicago w latach dwudziestych ubiegłego wieku. Zgodnie z nim, wynagrodzenie zależało od liczby gaszonych pożarów, co spowodowało że strażacy sami wzniecali pożary, aby następnie je ugasić [2].

W pewnym uproszczeniu, kryteria odwzorowują obowiązujące systemy wartości. Istnieje jednak wiele różnorodnych, wykorzystywanych jednocześnie systemów. Zazwyczaj odzwierciedlają one na tyle różne aspekty funkcjonowania, że ich porównanie czy zamiana są niemożliwe. Dlatego, dla większości rozwiązywanych zadań praktycznych dopiero jednoczesne wykorzystanie kilku kryteriów pozwala poprawnie opisać cel projektowania. Oczywiście, nie można wykluczyć, że dla pewnych szczególnych obiektów lub systemów cel wystarczy ocenić za pomocą jednego kryterium, jednak sytuacja taka jest niezmiernie rzadka. W celu ilustracji tego problemu rozważmy przykład systemu wynagradzania nauczycieli. W ramach poszczególnych stopni awansu zawodowego, jedynym kryterium wynagradzania jest liczba przeprowadzonych godzin lekcyjnych. Kryterium to, w jakikolwiek sposób nie uwzględnia jakości kształcenia i wychowania, podobnie jak wysokość dotacji oświatowej czy ilość uczniów w klasie. Bezsprzecznie, system taki powinien być wielokryterialny, jednak można się spodziewać, że i taki nie zadowoli wszystkich zainteresowanych. Innym przykładem systemu, dla którego dobór kryteriów jest wyjątkowo skomplikowany, jest sposób przyjęcia absolwentów szkół ponadgimnazjalnych na studia wyższe. Obecnie, bazuje on na wynikach egzaminu maturalnego. Nikt zapewne nie ma wątpliwości, że system ten nie zapewnia wyboru spośród kandydatów osób, które w przyszłości będą najlepszymi specjalistami w danym obszarze. Można zaryzykować stwierdzenie, że na obecnym poziomie wiedzy i dostępnych możliwościach organizacyjnych, problem ten jest praktycznie nierozwiązywalny.

Finalnym produktem stadium koncepcyjnego jest zadanie techniczne, jednoznacznie opisujące cele i kryteria projektowania. Stadium koncepcyjne może być wykonywane samodzielnie przez inwestora, może być również zlecone specjalistycznej firmie. W obu przypadkach, zalety i wady takiego rozwiązania są nader oczywiste. Wykorzystanie firmy zewnętrznej gwarantuje *powiew świeżości* w podejściu do rozwiązania zadania projektowego, prawdopodobne jest jednak nieuwzględnienie wybranych specyficznych wymagań użytkownika, których realizacja opera się na zastosowaniu rozwiązań charakterystycznych dla branży. Z kolei, samodzielne przygotowanie jest zazwyczaj bardzo zachowawcze i często prowadzi do powielenia błędów, charakterystycznych dla poprzedniego rozwiązania zadania

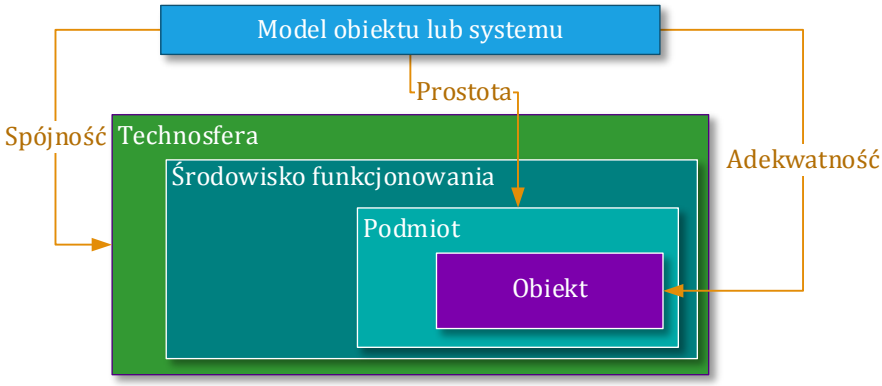
projektowego. Zdaniem autorów, celowym jest zlecenie prac projektowych podmiotowi zewnętrznemu z samodzielną, dogłębną analizą efektywności zaproponowanych rozwiązań.

Zgodnie z podejściem z rys. 2, kolejnym stadium fazy projektowania jest *modelowanie*, w którym wyróżniamy trzy etapy: budowa modeli, optymalizacja oraz podejmowania decyzji projektowych. Zgodnie z encyklopedyczną definicją modelowanie to metoda badania obiektów za pomocą budowy i analizy ich modeli. W modelowaniu podstawą badania obiektu jest model, będący swoistą kopią bazowego obiektu. W większości przypadków, model posiada inną przyrodę niż modelowany obiekt. Przykładowo, modelem zjawisk zachodzących w sieci zasilającej może być układ równań różniczkowo-całkowych, będący w istocie ich matematycznym opisem.

Modelowanie w procesie projektowania posiada trzy podstawowe funkcje: opisową, prognostyczną oraz normatywną. Funkcja *opisowa* ma na celu wyjaśnienie obserwowanych realnych zjawisk, odpowiadających na pytania związane z funkcjonowaniem obiektu lub systemu. W ten sposób modele stają się elementami odwzorowania teorii naukowych i można przypisywać im element poznawczy. Funkcja *prognostyczna* zapewnia przewidywanie przyszłych właściwości i stanów modelowanych obiektów lub systemów. Dzięki temu, udaje się przedłużyć ich czas życia, a także optymalizować koszty ich modernizacji. Projektowanie powinno uwzględniać nie tylko uwarunkowania o charakterze ekonomiczno-technicznym, ale również normatywnym. Oznacza to, że procedura projektowania powinna dodatkowo respektować punkt widzenia podmiotu zlecającego, wyrażający się jego interesami i preferencjami. Za spełnienie powyższych wymagań odpowiada m. in. normatywna funkcja procesu modelowania.

Podstawą omawianych procesów jest model. Zgodnie z definicją encyklopedyczną, *modelem* nazywamy system, którego analiza jest środkiem uzyskania informacji o innym systemie. W języku potocznym, modelem określamy reprezentację pewnego rzeczywistego procesu, urządzenia lub koncepcji, która jest traktowana jako wirtualna reprezentacja rzeczywistego podmiotu, która została stworzona za pomocą przeznaczonych specjalnie do tego celu narzędzi. Aby określić właściwości, którymi powinien charakteryzować się model, przeanalizujmy relacje pomiędzy uczestnikami procesu modelowania, którymi są: *model*, badany *obiekt*, *fragment technosfery*, w którym funkcjonuje rzeczywisty obiekt oraz *podmiot* tworzący bądź wykorzystujący model do własnych celów. Relacje pomiędzy technosferą, środowiskiem, podmiotem i modelem zostały przedstawione na rys. 3. Podstawową cechą każdego modelu, określającą relacje pomiędzy nim a opisywanym przez niego obiektem jest adekwatność, która odzwierciedla stopień odwzorowania obiektu przez model. Adekwatny model jest zawsze *wystarczająco* kompletny, dokładny i prawdziwy. Kluczem powyższych rozważań jest słowo *wystarczająco*. Jednym z celów przyświecających modelowaniu jest uzyskanie wiedzy o rzeczywistym obiekcie lub systemie w krótkim czasie i po niskich kosztach. W skrajnym

przypadku, można wyobrazić sobie sytuację, że wysoka adekwatność modelu zapewni pełną zgodność modelu i rzeczywistego obiektu, jednak koszty procesu modelowania będą porównywalne z kosztami jego realizacji, co wyklucza celowość modelowania. Modele, w swej istocie zakładają potrzebę pewnego przybliżenia odwzorowania rzeczywistości, gwarantującego znacznie niższe koszty modelowania niż fizycznej realizacji. Tak więc, poziom adekwatności może być określany stopniem uproszczenia procesu modelowania względem fizycznej realizacji obiektu.



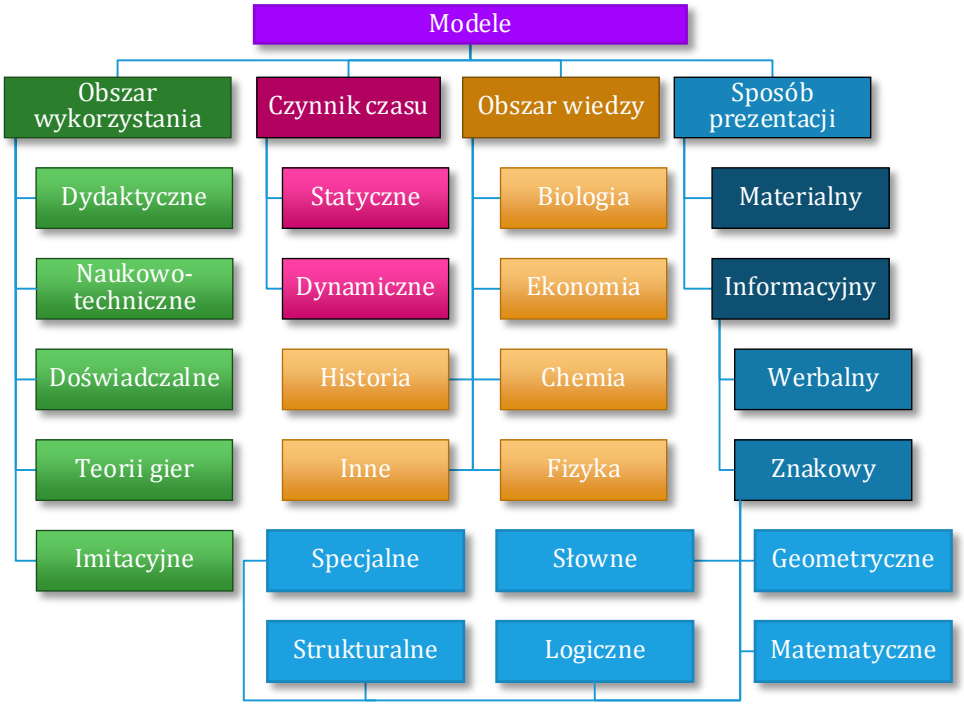
Rys. 3. Relacje pomiędzy elementami procesu modelowania i stawiane wymagania

Drugim, istotnym wymaganiem, stawianym przed modelami, jest ich *prostota*. Wiąże ona model z tworzącym i wykorzystującym go podmiotem oraz określa przejrzystość zawartego w modelu opisu mechanizmu funkcjonowania obiektu lub systemu. W procesie modelowania, na bazie zdefiniowanych wcześniej metod i środków, w oparciu o właściwości modelu określa się interesujące nas charakterystyki rzeczywistego obiektu. Zazwyczaj, prosty model oferuje nie tylko przejrzystość opisu, ale również łatwość jego rozwiązania. Jednak jego prostota, nie zawsze gwarantuje łatwość określenia charakterystyk rzeczywistego obiektu. Przykładowo, opis zjawisk zachodzących w obiekcie za pomocą układów równań różniczkowych może być stosunkowo prosty, co jednak nie oznacza, że poszukiwanie ich rozwiązania będzie łatwe.

Każdy model, podobnie jak opisywany przez niego obiekt bądź system, funkcjonuje w określonej części technosfery. W obu przypadkach, występujące relacje mają charakter dwustronny, co oznacza, że na funkcjonowanie obiektu jak i modelu przejawiają wpływ na zjawiska zachodzące w technosferze. Ma miejsce także oddziaływanie odwrotne: obiekt i model wpływają na technosferę, modyfikując ją realnie bądź wirtualnie. Prawdziwa jest zatem teza, że relacje pomiędzy modelem i środowiskiem mają charakter dwustronny. Odbywają się one za pomocą tzw. *punktów styku*, których przykładami mogą posłużyć: w systemach monitoringu środowiska – sensory pomiarowe; w systemach bankowych – bankomaty i komputery użytkowników korzystających z *e*-bankowości itp. Z punktu widzenia liczby i asortymentu punktów styku z technosferą, realne systemy istotnie różnią się od modeli.

O ile w pierwszym przypadku ich liczba i asortyment wynikają z realnych warunków eksploatacji, to w przypadku modelu ich ilość i rodzaje podlegają minimalizacji. Wynika to z dwóch podstawowych uwarunkowań: po pierwsze, nie każdy z punktów styku przejawia istotny wpływ na adekwatność modelu; po drugie: zwiększenie ich liczby ogranicza prostotę i przyczynia się do wzrostu nakładów niezbędnych w procesie modelowania.

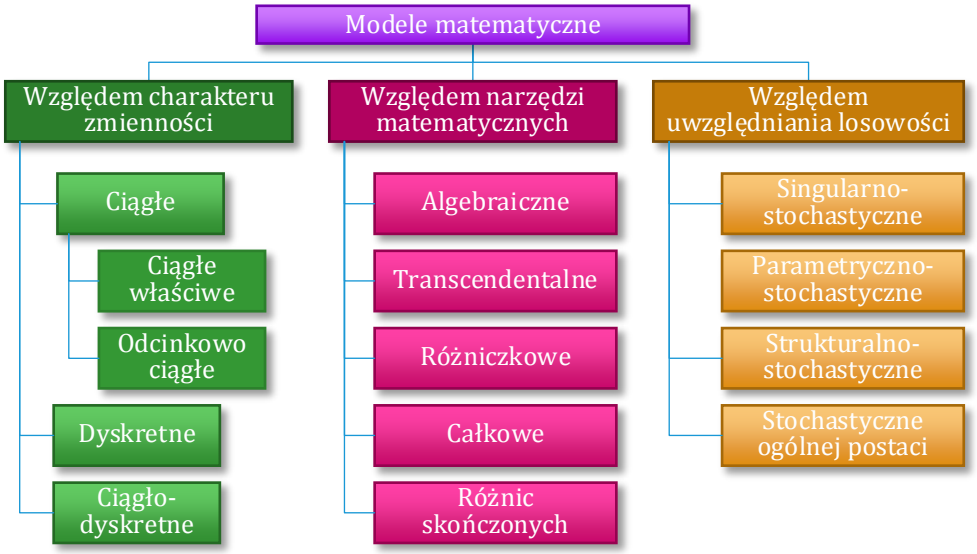
Znanych jest wiele różnorodnych klasyfikacji modeli, uwzględniających różnorakie kryteria podziału. Przykładową dwupoziomową klasyfikację hierarchiczną przedstawiono na rys. 4.



Rys. 4. Ogólna klasyfikacja modeli

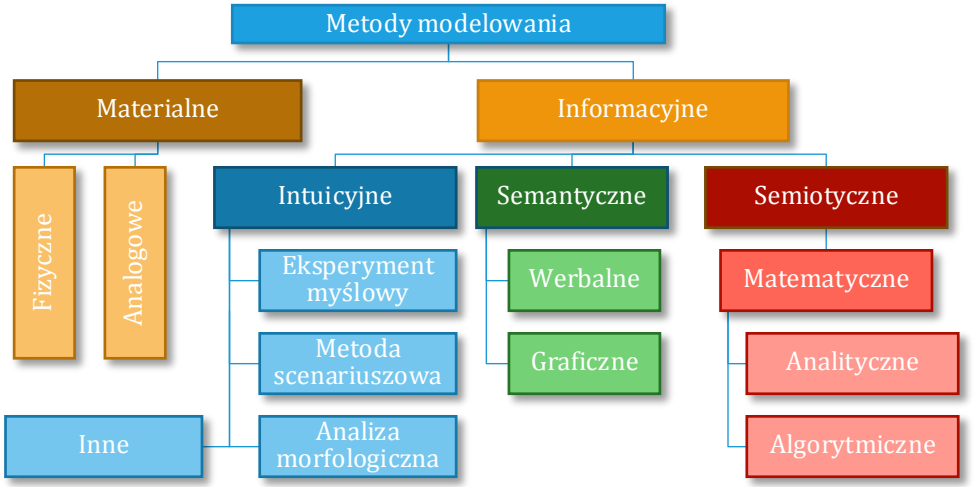
Jako bazowe, zastosowano tutaj podziały modeli względem: obszaru wykorzystania, czasu, obszaru wiedzy oraz sposobu prezentacji. Czytelnicy monografii najszerzej wykorzystywać będą dynamiczne modele naukowo-techniczne z obszaru biologii i ekonomii, bazujące na matematycznym sposobie przedstawienia. Modele te będą fragmentem aplikacji, która z wykorzystaniem intuicyjnego interfejsu przeprowadzi użytkownika przez cały proces modelowania tworzonego projektu.

Jednym z najszerzej wykorzystywanych typów są modele matematyczne, których klasyfikację przedstawiono na rys. 5. Do ich zapisu stosuje się język wszechstronnie pojętej matematyki. Szerokie zastosowanie grupy wiąże się głównie z masowością dostępu do komputerów oraz szerokim asortymentem bazujących na matematyce narzędzi informatycznych, dostępnych praktycznie na każdej platformie.



Rys. 5. Klasyfikacja modeli matematycznych

Istnieje wiele metod modelowania, których przykładową klasyfikację przedstawiono na rys. 6.



Rys. 6. Przykładowa klasyfikacja metod modelowania

Modelowanie materialne polega na przygotowaniu, w miejsce oryginału, specjalnej formy obiektu lub systemu, podlegającego badaniom. Jeżeli w analizie modelu wykorzystane zostaną te same zjawiska fizyczne co w rzeczywistym obiekcie, to modelowanie takie nazywać będziemy *fizycznym*. Modelowanie takie jest nadal wykorzystywane przy projektowaniu środków komunikacji (samochodów, samolotów, statków) oraz wybranych obiektów inżynieryjnych (budowli hydrotechnicz-

nych). Chociaż powyższe techniki są sukcesywnie wypierane przez metody matematyczne, ociekanie skrzydeł samolotu czy aerodynamikę samochodu nadal bada się w tunelach aerodynamicznych w oparciu o przygotowane wcześniej modele fizyczne. W przypadku czytelników monografii wykorzystanie materialnego modelowania fizycznego jest wątpliwe.

Modelowanie analogowe to modyfikacja modelowania fizycznego, w którym realny proces fizyczny zachodzący w obiekcie, jest zamieniany innym procesem, wygodniejszym z punktu widzenia prowadzenia procesu modelowania. Analogią taką można posłużyć się, na przykład, badając rozkład temperatury w metalowej belce, rozkład napięć w układzie elektrycznym, czy przepływ danych w sieci teleinformatycznej. W pierwszym z ww. przykładów rzeczywisty proces przewodzenia ciepła można zamienić przewodnictwem elektryczności w modelu. Taka analogia opiera się na słusznym założeniu, że równania przewodnictwa cieplnego i rozkładu potencjału elektrycznego można sprowadzić do identycznej postaci. Wykorzystując układy elektryczne modeluje się również niestacjonarne rozkłady temperatur w pomieszczeniach i szereg innych zjawisk, zachodzących w rzeczywistych obiektach (ociekanie cieczy i gazów, przemieszczanie się domieszek w roztworze itp.), bez konieczności ich budowy.

Istota *modelowania informacyjnego* polega na badaniu utworzonych obiektów wirtualnych, opisanych dla potrzeb modelowania za pomocą metod i środków nieznanych w obiekcie rzeczywistym. Pierwszy rodzajem jest modelowanie intuicyjne, nieznajdujące uzasadnienia w którejś z formalnych logik prezentacji badanego obiektu, bowiem modelowany obiekt może nie poddawać się formalizacji, bądź projektant jest zdania, iż taka formalizacja jest zbędna. Modelowanie intuicyjne odbywa się w świadomości człowieka i najczęściej ma formę eksperymentów myślowych, scenariuszy lub gier. Opiera się ono na gromadzonym przez wiele dziesięcioleci doświadczeniu projektantów. Wykorzystuje się w nim wiedzę empiryczną, nabytą w trakcie eksperymentów lub podczas obserwacji zjawisk, zachodzących w obiekcie przy tym ich fizyczna interpretacja nie jest wymagana.

Pierwszą z wyszczególnionych na rys. 6 metod modelowania intuicyjnego jest eksperyment myślowy. Z filozoficznego punktu widzenia jest on metodą poznania otaczającej nas rzeczywistości za pomocą wirtualnego wyobrażenia bazującego na hipotezie, teorii bądź zasadzie. Na eksperymentach myślowych opierali swoje badania już starożytni uczeni, jednak najszersze wykorzystanie znalazł on w okresie Odrodzenia. W owym czasie wykorzystywali go m. in. Galileusz, Kartezjusz, Newton i Leibniz. Dwudziestowiecznym uczonym, który najszerzej korzystał z eksperymentów myślowych był Albert Einstein.

Nauka zna wiele przykładów takich eksperymentów. Jeden z najbardziej znanych przypisywany jest Lukrecjuszowi i dotyczy nieskończoności Wszechświata. W swoim dziele „O naturze wszechrzeczy" (*łac*. *De rerum natura*), aby udowodnić postawioną wyżej tezę, wykorzystuje on następujące rozumowanie: *Jeżeli granica Wszechświata istnieje to można do niej strzelać z łuku. Jeśli strzała przeleci na wylot*

to granicę tę nie można nazywać granicą, jeżeli zaś odbije się od niej, to u analizowanej kosmicznej ściany musi być druga strona, co oznacza że poza rozpatrywaną granicą istnieje kontynuacja przestrzeni. W obu przypadkach nie udaje się osiągnąć końca Wszechświata, co oznacza że jest on nieskończony. Innym, powszechnie znanym eksperymentem myślowym jest zdefiniowane w 1935 roku doświadczenie pod nazwą *kot Schrödingera*. Z uwagi na niegodne traktowanie w nim kotów – najsympatyczniejszych domowych futrzaków, nie będziemy go przytaczać. Do najbardziej znanych eksperymentów myślowych w obszarze informatyki zaliczamy: maszynę Turinga; problem dwóch generałów, problem czytelników i pisarzy oraz problem ucztujących filozofów.

Eksperymenty myślowe na bazie współczesnej, pełniejszej analizy danych empirycznych, pozwalają nam uzyskać istotne informacje, zazwyczaj będące rezultatem nowego, nietypowego spojrzenia na nie. Chociaż są one wykorzystywane głównie w fizyce, matematyce, różnych rodzajach psychologii, w historii, ekonomice, marketingu i epidemiologii, znajdują również zastosowanie w projektowaniu obiektów i systemów technicznych, gdzie wykorzystuje się je do uwzględniania zachowania użytkowników.

Metoda scenariuszowa opiera się na przygotowaniu i uzgodnieniu pisemnych opisów projektowanego obiektu bądź systemu. Początkowo, opis ten zawierał logiczną sekwencję zachodzących w czasie zdarzeń, wynikających z funkcjonowania modelowanej struktury lub możliwe sposoby rozwiązania stojącego przed nią problemu. Z czasem, zrezygnowano z czasu jako obowiązkowego elementu procesu modelowania i scenariuszem stał się dowolny dokument, zawierający opis rozwiązania konkretnego problemu, bez względu na formę jego zapisu.

Scenariusze tworzone są przez zespół ekspertów, którzy powinni przewidzieć nie tylko zachowanie obiektu bądź systemu, ale również najważniejsze jego ilościowe charakterystyki ekonomiczno-techniczne. Jeżeli dysponują oni informacjami o podobnych projektach, wskazaną jest ich analiza statystyczna. Na etapie tworzenia scenariuszy należy określić ogólne zasady rozwoju obiektu lub systemu, przeanalizować czynniki wewnętrzne i zewnętrzne, wpływające na jego rozwój i cele funkcjonowania, a także wykonać obszerne badania literaturowe w danym obszarze. Podstawową trudnością w tworzeniu scenariuszy jest zagwarantowanie ich realności, w przeciwnym przypadku będą one wyłącznie zbiorem niczym nieuzasadnionych prognoz. Aby uniknąć tego, zaleca się rozważyć optymistyczne i pesymistyczne scenariusze funkcjonowania projektowanej struktury zakładając, że jego rzeczywiste działanie znajdować się będzie pomiędzy nimi. Scenariusze pozwalają wyłącznie uzyskiwać wstępną wiedzę o obiekcie lub systemie i powinny stanowić podstawę dla dalszych prac projektowych.

Inną metodą modelowania intuicyjnego jest, opracowana przez szwajcarskiego astrofizyka Fritza Zwicky, *metoda morfologiczna*. Choć przypisujemy ją do modelowania intuicyjnego, wykorzystuje ona metody analityczne. Termin morfologia, wy-

wodzi się z biologii i oznacza dziedzinę, zajmującą się badaniem struktury wewnętrznej obiektów i systemów. Metoda ta opiera się na systematycznym poszukiwaniu maksymalnej liczby wariantów realizacji obiektu lub systemu poprzez kombinatoryczne łączenie ich elementów konstrukcyjnych. Rozwiązywane zadanie dzieli się różnymi sposobami na części i rozpatruje w różnych aspektach jego funkcjonowania. Do grupy metod projektowania intuicyjnego możemy zaliczyć również metody: strukturalizacji oraz drzewa celów.

Wszystkie powyższe metody mogły być wykorzystywane zarówno przez pojedynczego projektanta, jak i ich zespół. Istnieją również metody przeznaczone do pracy zespołowej. Odnosimy do nich: gry operacyjne, burzę mózgów, metodę delficką oraz metodę synektyczną, których szczegółowe opisy można znaleźć w literaturze [5].

Niezależną grupę metod modelowania tworzą *metody semantyczne*, znajdujące zastosowanie dla określonego zbioru założeń początkowych. Najczęściej, założenia te przyjmują formę hipotez, tworzonych w procesie obserwacji obiektu modelowania bądź jego analogów. W odróżnieniu od metod intuicyjnych, metody semantyczne bazują na znajomości wewnętrznych mechanizmów funkcjonowania modelowanych obiektów, możliwe jest również precyzyjne odtworzenie działań konkretnego obiektu dla innych obiektów bądź ich systemów. Metody werbalnego modelowania semantycznego bazują na zestawach słów, na bazie których mogą być definiowane stwierdzenia, oceny i wnioski, dotyczące badanego obiektu. Dla modelowania graficznego, wykorzystując takie nośniki informacji jak papier, kalka techniczna lub monitor komputera sporządza się rysunki techniczne, schematy funkcjonalne, wykresy zależności parametrów i in., ilustrujące funkcjonowanie modelowanego obiektu lub systemu.

Najbardziej sformalizowanym typem modelowania jest *modelowanie semiotyczne*, oparte na ogólnie znanych słowach, zwrotach, zapisach, rysunkach oraz wszelkiego typu znakach, powiązanych pomiędzy sobą zbiorem ściśle określnych zasad, opisujących interpretację zarówno pojedynczych elementów modelu, jak i ich grup. *Modelowanie matematyczne* jest specyficznym rodzajem modelowania semiotycznego, w którym do opisu i badania oryginalnego obiektu zastosowano narzędzia matematyki. Szerokie zastosowanie tej grupy metod należy łączyć z rozwojem informatyki i metod rozwiązania za jej pomocą zadań matematycznych. Dzięki dostępności taniego sprzętu komputerowego oraz pakietów matematycznych, a także błyskawicznemu wzrostowi wydajności urządzeń obliczeniowych, ten rodzaj modelowania stał się podstawowym elementem większości prac projektowych. *Analityczne* metody modelowania matematycznego zakładają wykorzystanie formalnego opisu modelowanego obiektu lub systemu. Do tego celu, najczęściej stosowane są układy równań diofantycznych, całkowych, różniczkowych oraz różniczkowo-całkowych, które są komponowane i rozwiązywane w procesie modelowania. Niestety, w komputerach cyfrowych większość wykorzystywanych układów równań (pomijając diofantyczne) rozwiązuje się za pomocą złożonych czasowo

metod przybliżonych. Co prawda, oferują one bardzo dokładne ich rozwiązania, ale niezbędne do tego czasy są w wielu przypadkach nieakceptowalne. W *algorytmicznych* metodach modelowania matematycznego opis obiektu lub systemu realizowany jest za pomocą metod charakterystycznych dla informatyki. Sposób ich działania może być przedstawiony za pomocą dowolnej z metod opisu algorytmów, a sama złożoność procesu modelowania jest w większości przypadków stosunkowo niewysoka. Metody tej grupy świetnie nadają się do modelowania w obszarze zarządzania i informatyki, gdzie nie występuje potrzeba opisu przebiegu zjawisk fizycznych, opisywanych za pomocą wspomnianych wcześniej układów równań.

Kolejnym etapem stadium modelowania jest *optymalizacja*. Polega ona na poszukiwaniu pośród wielu wariantów rozwiązania zadania projektowego, najlepszego pod względem spełnienia określonych wcześniej kryteriów. Ona oraz kolejne etapy fazy projektowania przedstawione zostaną w kolejnym rozdziale monografii.

4. Podsumowanie

Złożoność współczesnych obiektów technicznych osiągnęła poziom uniemożliwiający ich optymalne projektowanie wyłącznie na bazie metod intuicyjnych. Projektantowi tworzącemu obiekt trudno jest uwzględnić wszystkie uwarunkowania funkcjonowania skomplikowanej struktury, decydujące o efektywności jej eksploatacji. Pomimo jednoznaczności powyższego stwierdzenia, realizacja wielu złożonych projektów odbywa się z pominięciem jakichkolwiek formalizacji, w szczególności tych bazujących na technologiach komputerowych. Nierzadko narzędzia informatyki stosowane są wyłącznie do przygotowania części graficzno-opisowej dokumentacji wykonawczej obiektu. Na szczęście, powyższe zjawisko dotyczy, przede wszystkim, niewielkich, nieskomplikowanych obiektów, projektowanych przez jednoosobowe biura, na zlecenie niewielkich podmiotów gospodarczych, gospodarstw rolnych, rzadziej samorządów gminnych. Przygotowanie dokumentacji dużych, rozbudowanych konstrukcji znacznie szerzej bazuje na narzędziach CAD/CAM/CAE, wprowadzających formalizację do procesu projektowania.

Istotnym ograniczeniem szerokiego zastosowania komputerowych metod wspomagania projektowania są koszty zakupu specjalistycznego oprogramowania. Zakup taki jest opłacalny wyłącznie przy pełnym portfelu zamówień, jednoczesnej realizacji wielu zleceń oraz kilkuosobowej obsadzie stanowiska ze specjalistycznymi narzędziami informatycznymi. Dla jednoosobowych firm projektowych, koncentrujących się na niewielkich zleceniach, wydatek kilkunastu, a nierzadko kilkudziesięciu tysięcy złotych jest ekonomicznie nieuzasadniony. Firmom takim trudno jest wykorzystać możliwości, oferowane przez takie narzędzia.

Powyższe obserwacje potwierdzają wcześniejszą tezę, że samodzielna weryfikacja jakości wykonania prac zlecanych przez mikro przedsiębiorstwa, gospodarstwa rolne oraz samorządy najniższego szczebla jest zasadna. W większości realizowane

inwestycje mają niewielki rozmiar, a prace projektowe wykonywane są przez niewielkie, zazwyczaj jednoosobowe biura projektowe, w niemal zerowym stopniu korzystające z komputerowego wspomagania procesu projektowania. Istnieje zatem uzasadnione podejrzenie, że w przedstawionych projektach znajdą się elementy, które można poprawić, udoskonalić.

Z wielokrotnym projektowaniem niektórych obiektów wiąże się również druga wątpliwość. Przedstawiona na rys. 2 koncepcja cyklu życia, wyróżniająca fazę projektowania, prezentuje składowe cyklu z punktu widzenia metodologii badań naukowych. Należy zatem odpowiedzieć na pytanie, czy zaprezentowaną tam sekwencję procedur należy wykorzystywać również do tworzenia projektów prostych, powtarzalnych obiektów. Jak pokazuje doświadczenie, w takim przypadku, wykonywane prace projektowe ograniczają się najczęściej do stadium przygotowania technicznego fazy projektowania obejmującego tworzenie dokumentacji wykonawczej obiektu.

W celu odpowiedzi na postawione wyżej pytanie rozważmy przykład projektowania wielkopowierzchniowych szklarni, zlokalizowanych w różnych miejscach na terenie kraju. Teoretycznie, projektowane obiekty będą w zupełności powtarzalne i ograniczenie się do stadium *przygotowania technicznego* można uważać za zasadne. Realizacja konkretnego obiektu może różnić się sposobem posadowienia, określonym przez lokalne warunki geologiczne. Dopuszczalna jest również zmiana kształtów geometrycznych obiektu, wynikająca z rozmiarów działki, dostępnej inwestorowi. Można przytoczyć jeszcze wiele innych szczegółów konstrukcyjnych obiektu, które mogą być doprecyzowane dopiero na etapie przygotowania technicznego. Istnieje jednak wiele innych detali konstrukcyjnych i wykonawczych, które powinny być analizowane na wcześniejszych stadiach fazy projektowania. Przykładowo, przynależność obszaru, na którym zlokalizowana zostanie szklarnia do konkretnej strefy klimatycznej może, w przypadku jej całorocznej eksploatacji, odgrywać istotną rolę w wyborze sposobu jej ogrzewania. Jak pokazują szacunki autorów, różnica w zapotrzebowaniu na energię cieplną dla różnych stref mogą w warunkach Polski sięgać 40%. Zauważmy, że dobór źródła ciepła powinien odbywać się na etapie optymalizacji stadium modelowania fazy projektowania cyklu życia obiektu. Podobnie, decyzje dotyczące uwzględnienia w projekcie takich charakterystyk jak: osiągana grubość pokrywy śnieżnej, szybkość wiatrów na danym obszarze, czy dostępność lokalnych źródeł energii cieplnej powinny być podejmowane na wstępnych etapach fazy projektowania.

Powyższe spostrzeżenia dotyczą praktycznie każdego typu projektowanego obiektu. Dla przykładu, przeanalizujmy fazę projektowania cyklu życia produktu z obszaru elektroniki konsumenckiej. Istotnym parametrem eksploatacji urządzeń elektronicznych jest temperatura oraz wilgotność powietrza, w którym ma funkcjonować projektowane urządzenie. O ile w umiarkowanej strefie klimatycznej odprowadzenie ciepła czy zabezpieczenie elementów elektronicznych przed korozją

ma charakter drugoplanowy, to w przypadku urządzeń pracujących w strefie tropikalnej musi mieć charakter pierwszoplanowy. Rozważania dotyczące projektowania obiektów i systemów są kontynuowane w części II artykułu, w jednym z kolejnych rozdziałów monografii.

Bibliografia

[1] Ch. S. Wasson, *System Analysis, Design, and Development: Concepts, Principles, and Practices*. New Jersey, USA: Wiley-Interscience, 2005.

[2] R. B. Northrop, *Introduction to Complexity and Complex Systems*. New York, USA: CRC Press, 2010.

[3] N. Boccara, *Modeling Complex Systems*, 2nd ed. New York, USA: Springer, 2010.

[4] M. P. Clark, *Networks and Telecommunications: Design and Operation*. Chichester, UK: John Wiley & Sons, 1991.

[5] S. Evans, *Telecommunications Network Modeling, Planning and Design*. London, UK: The Institution of Electrical Engineers, 2003.

[6] V. Alwayn, *Optical Network Design and Implementation*. Indianapolis, USA: Cisco Press, 2004.

[7] W. L. Butler, "Hints for computer system design," *SIGOPS Oper. Syst. Rev.*, vol. 17, no. 5, pp. 33-48, May 1983.

[8] L. Kleinrock, "Analytic and simulation methods in computer network design," in *Proceedings of the May 5-7, 1970, spring joint computer conference*, Atlantic City, New Jersey, 1970, pp. 569-579.

[9] F. Larumbe and B. Sansò, "Optimal Location of Data Centers and Software Components in Cloud Computing Network Design," in *Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, Washington, DC, 2012, pp. 841-844.

[10] R. Nance, R. Moose, and R. Foutz, "A statistical technique for comparing heuristics: an example from capacity assignment strategies in computer network design," *J Commun. ACM*, vol. 30, no. 5, pp. 430-442, May 1987.

[11] N. J. Smith and A. P. Sage, *An introduction to hierarchical systems theory*. Dallas, USA/Texas: Information and Control Sciences Center, SMU Institute of Technology, 2005.

[12] M. D. Mesarovic, D. Macko, and Y. Takahara, *Theory of hierarchical, multilevel systems*. New York, USA: Academic Press, 1970.

[13] T. L. Saaty, *Fundamentals of Decision Making with the Analytic Hierarchy Process*. Pittsburgh, USA: RWS Publications, 2000.

[14] A. Levitin, *Introduction to the Design and Analysis of Algorithms*. Phenix, USA: Addison Wesley, 2006.

[15] J. D. McCabe, *Network Analysis, Architecture and Design*, 3rd ed. New York, USA: Morgan Kaufmann, 2007.

[16] J. Mendez-Rangel and C. Lozano-Garzon, "A network design methodology proposal for E-health in rural areas of developing countries," in *6th Euro American Conference on Telematics and Information Systems*, Valencia, 2012, pp. 1-7.

[17] R. V. Slyke. (2001, Nov.) Network planning and design. [Online]. ftp://ftp.shore.net/members/ws/Support/BDC/nd.pdf

[18] D. L. Spohn, *Data Network Design*. New York, USA: McGraw-Hill, 2002.

ROZDZIAŁ 2

Projektowanie, produkcja i testowanie aplikacji dla urządzeń mobilnych

Piotr HAJDER♣, Mirosław HAJDER♦, Mariusz NYCZ♠
♣*Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie,*
♦*Wyższa Szkoła Informatyki i Zarządzania z siedzibą w Rzeszowie*
♠*Politechnika Rzeszowska im. Ignacego Łukasiewicza*
Piootr.Hajder@gmail.com, Miroslaw.Hajder@gmail.com, MarNycz@gmail.com

Streszczenie

W rozdziale, skoncentrowano się na przedstawieniu teoretycznych i praktycznych podstaw systemów testowania – końcowego etapu fazy produkcji oprogramowania. Rozdział składa się z dwóch części: w pierwszej przedstawiono wybraną wiedzę dotyczącą procedury produkcji i eksploatacji aplikacji mobilnych, druga zaś opisuje przygotowane przez autorów narzędzie wspomagające ich testowanie. Część pierwszą rozpoczyna uporządkowanie terminologii w zakresie mobilności: zdefiniowano i porównano różne znaczenia terminu mobilność, wskazano to które stanowić będzie podstawę dalszych rozważań. Dalej, przedstawiono etapy produkcji aplikacji tej klasy, wskazując na szczególną rolę testowania w zapewnianiu jej jakości, a także rozważono typy operacji testowania. Cały rozdział kończą propozycje autorów dotyczące dalszych prac w danym obszarze.

1. Wprowadzenie

Analiza publikacji z obszaru aplikacji mobilnych pokazuje istotną niekonsekwencję terminologiczną: pojęcie *mobilność* wykorzystywane jest w kilku odmiennych znaczeniach. Na początek, rozważmy problem z punktu widzenia użytkownika końcowego. Z jego punktu widzenia, *aplikacją mobilną* nazywamy produkt uruchamiany na urządzeniach przenośnych, najczęściej smartfonach lub tabletach. W tym przypadku, mobilność definiuje lokalizację funkcjonowania aplikacji. Zdecydowanie rzadziej, termin *mobilność* wykorzystuje się w odniesieniu do przenoszenia produktu pomiędzy różnymi typami urządzeń wyposażonych w te same platformy operacyjne, lecz różne sprzętowe. Najrzadziej, słowa *mobilność* używa się w do opisu produktu, pozwalającej na nieskrępowane przenoszenie aplikacji pomiędzy różnymi platformami sprzętowymi, czasami dodatkowo i operacyjnymi. Drugi oraz

trzeci rodzaj mobilności można urzeczywistnić metodami pośrednimi, tj. za pomocą narzędzi zapewniających taką przenaszalność. Ich przykładem jest oprogramowanie *Docker* opracowane dla Linuxa i przeznaczone do automatyzacji wdrażania oraz zarządzania aplikacjami przy wirtualizacji na poziomie systemu operacyjnego [1], [2], [3]. Pozwala ono umieścić aplikację z całym jej otoczeniem i relacjami w kontenerze, który może być przeniesiony na dowolny Linux z zagwarantowaną obsługą *cgroups* w jądrze. W dalszej części rozdziału, wykorzystywana będzie pierwsza definicja mobilności.

Nieco inne spojrzenie na mobilność aplikacji prezentuje teoria jakości oprogramowania [4], [5]. Jej interpretacja jest interesująca również dlatego, że kryterium jakości odgrywa kluczową rolę przy ocenie praktycznie każdego produktu. Zgodnie z klasyczną definicją, jakość oprogramowania to parametr określający zakres włączenia w aplikację zestawu funkcji niezbędnych do zwiększenia efektywności produktu w czasie cyklu jego życia. Dla dowolnego wyrobu, którego elementem jest oprogramowanie, można zdefiniować trzy rodzaje specyfikacji wymagań, którymi są odpowiednio: *wymagania funkcjonalne*, *jakościowe* i *programowo-sprzętowe*. Kryterium jakości łączy w sobie wszystkie parametry oraz istotne cechy produktu, odnoszące się do wypełnienia wymagań zdefiniowanych we właściwej specyfikacji [6], [7].

Najczęściej, jakość oprogramowania definiują dwie składowe: jakość procedur jego opracowania oraz samego produktu [8], [9]. Zauważmy, że produkcja oprogramowania bazuje na takich komponentach jak: technologie; środki techniczne; zasoby ludzkie; organizacja procesu produkcyjnego oraz sprzęt pomocniczy, których jakość bezpośrednio przekłada się na jakość wytwarzanego oprogramowania. Oprócz tego, o jakości produktu decydują takie czynniki jak: przejrzystość i adekwatność przygotowanej dokumentacji; formalizacja nadzoru nad produkcją; kompleksowość testowania. Znane i wykorzystywane obecnie modele jakości, definiowane są najczęściej zestawem parametrów i powiązań pomiędzy nimi, które stanowią bazę zarówno dla wyszczególnienia wymagań jakościowych, jak również określenia wartości wskaźnika jakości. Nierzadko, model jakości definiowany jest przez strukturalizowany zbiór właściwości, niezbędnych do realizacji postawionych celów.

Pojęcie *mobilności*, w różnym znaczeniu, występuje w większości znanych definicji jakości oprogramowania. W modelu *J.A. Mc Calla* pojawia się w przenaszalności – jednym z trzech podstawowych kryteriów określenia i identyfikacji jakości [10], [11]. Mobilność, wraz z wielokrotnym wykorzystaniem oraz zgodnością funkcjonalną są podstawowymi czynnikami zapewniającymi przenaszalność aplikacji na inne platformy sprzętowe i programowe [12]. Model *B.W. Boehma* jest często traktowany jako rozwinięcie modelu *Mc Calla*, uzyskane poprzez uogólnienie wykorzystywanych w nim charakterystyk [13]. Oba modele mają charakter hierarchiczny, przy czym model *Boehma* reprezentuje hierarchia o większej głębokości. Również w jego przypadku *mobilność* rozumiana jest jako niezależność sprzętowa

oraz programowa autonomiczność. Zbliżone definicje mobilności spotykamy także w modelach jakości *C. Ghezzi*, *A. Sharma* oraz *G.R. Dromeya* [14]. W normach ISO/IEC, dotyczących jakości oprogramowania (ISO/IEC 9126 oraz ISO/IEC 25010) mobilność rozumiana jest jako przenaszalność, charakteryzująca się, co najmniej adaptowalnością, instalowalnością oraz zastępowalnością [15], [16].

Aby zrozumieć aktualność tematyki badań, w szczególności kluczową rolę jakości oprogramowania w sprawnym funkcjonowaniu społeczeństwa, przedstawmy miejsce aplikacji mobilnych w życiu społeczno-gospodarczym. Początek XXI wieku, głównie dzięki rozwojowi technologii informacyjnych, może być odbierany jako czas znaczących zmian w życiu publicznym społeczeństwa, w szczególności w obszarach polityki, gospodarki i kultury. Istotne zmiany zachodzą również w systemie zarządzania, gdzie znaczna część spraw urzędowych, przenoszona jest bezpośrednio w obszar wirtualny, do sieci Internet [17]. Masowość wykorzystania technik informacyjnych m. in. do załatwiania spraw urzędowych, wymaga istotnej zmiany wymagań stawianych przed przeznaczonymi do tego aplikacjami. O ile w przypadku ludzi wykształconych, dzięki oparciu edukacji na osobo-zorientowanym i kompetencyjnym podejściu do nauczania, udało się rozwinąć u nich takie cechy jak: twórcze myślenie, zdolność uczenia się, umiejętność stawiania celów i podejmowania efektywnych decyzji, sprzyjających szerokiemu wykorzystaniu nowoczesnych technologii. Szczególną rolę odgrywa tutaj zorientowane na działalność praktyczną podejście kompetencyjne. Niestety, w przypadku osób starszych, a także młodzieży z ograniczonym dostępem do technik komputerowych i sieci Internet, pojawia się problem efektywnego wykorzystania *e-narzędzi*. Jakość oprogramowania, jego niezawodność oraz czytelność i intuicyjność interfejsów odgrywają kluczową rolę w skutecznym propagowaniu szerokiego wykorzystania aplikacji informatycznych. Dlatego, jakość należy uznać za jedno z najważniejszych kryteriów oceny współczesnego oprogramowania [4], [18].

Rozdział monografii prezentuje wyniki prac autorów w obszarze budowy, wdrażania i eksploatacji aplikacji mobilnych. Najszerzej zaprezentowane zostały metodyki projektowania, produkcji i testowania aplikacji. Doświadczenia uzyskane w trakcie praktycznego zastosowania metodyk, narzędzi i środków potwierdzają oczekiwania dotyczące poprawy jakości produkcji aplikacji. Uzyskane wyniki były weryfikowane, m. in. w trakcie zajęć dydaktycznych (głównie projekty semestralne i dyplomowe studentów) na trzech uczelniach.

2. Środowisko operacyjne i narzędziowe aplikacji mobilnych

Obecnie, na rynku systemów operacyjnych dla smartfonów, znaczącą pozycję posiadają 3 produkty: Android, iOS oraz Windows Phone. *Android* to obecnie podstawowy system operacyjnym dla sieci obliczeń brzegowych, w tym Internetu rzeczy. Dominuje on wśród smartfonów, tabletów, zegarków elektronicznych, telewizorów, lodówek, czajników itp. Podstawą tworzenia aplikacji w środowisku Androida

jest język Java, uzupełniony bogatym zestawem bibliotek. Środowiskiem przygotowania aplikacji mogą być Android SDK, Android Studio lub pakiet Eclipse z zainstalowaną wtyczką ADT [19], [20], [21]. Produkcja aplikacji możliwa jest w środowisku Windows, Mac OS oraz Linux. Aktualną wersją systemu jest wersja 7.

System *iOS* jest produktem firmy Apple, przeznaczonym dla urządzeń mobilnych iPhone, iPod touch oraz iPad. Jego cechą szczególną jest zamkniętość narzędziowa – produkcja aplikacji możliwa jest wyłącznie w środowisku systemu operacyjnego Mac OS. Do przygotowania aplikacji stosuje się dwa języki *Objective-C* oraz *Swift*. Kod napisany z wykorzystaniem tego ostatniego, bezproblemowo współpracuje z aplikacjami zrealizowanymi w języku *C* oraz *C++*.

Kolejny z systemów, *Windows Phone* [22], nigdy nie dorównał poziomem wykorzystania dwóm pierwszym. Tworzenie aplikacji opiera się na języku C#. Dostępna jest bezpłatna wersja Visual Studio Express 2012 for Windows Phone.

Porównanie trzech podstawowych środowisk operacyjnych dla urządzeń mobilnych przedstawiono w tabeli 1.

Tabela 1. Porównanie najważniejszych systemów operacyjnych urządzeń mobilnych

Charakterystyka	**System operacyjny**		
	Android	**iOS**	**Windows Phone**
Typ dostępności kodu źródłowego systemu	Otwarty	Zamknięty	Zamknięty
Użyteczność	Prosty i zrozumiały intuicyjnie interfejs użytkownika. Brak istotnych ograniczeń.	Prosty i zrozumiały intuicyjnie interfejs użytkownika. Występują istotne ograniczenia.	Prosty i zrozumiały intuicyjnie interfejs użytkownika. Brak istotnych ograniczeń.
Wydajność w skali makro	Czas przygotowania aplikacji – wysoki. Czas nauki narzędzi – krótki.	Czas przygotowania aplikacji – umiarkowany. Czas nauki narzędzi: *Objective C* – długi; *Swift* – krótki.	Czas przygotowania aplikacji – umiarkowany. Czas nauki narzędzi – krótki.
Wygoda tworzenia i uruchomienia aplikacji	Oferowane narzędzia zapewniają jej wysoki poziom.	Oferowane narzędzia zapewniają jej wysoki poziom.	Oferowane narzędzia zapewniają jej wysoki poziom.
Dostępność programistów na rynku pracy	Aplikacje tworzone na bazie języka Java. Dostępność – nadzwyczajna	*Objective-C* – wąski krąg programistów; Swift – szeroki krąg programistów	Podstawą produkcji oprogramowania jest język C#. Dostępność programistów – szeroka.

Wydajność systemu operacyjnego w skali mikro	Czas ładowania części operacyjnej i aplikacyjnej – krótki. Czas reakcji systemu na zapytania użytkownika systemu operacyjnego – krótki. Możliwe opóźnienia w przypadku obróbki złożonych żądań pochodzących od aplikacji.	Czas ładowania części operacyjnej i aplikacyjnej – krótki. Czas reakcji systemu na zapytania użytkownika systemu operacyjnego – krótki. Możliwe opóźnienia w przypadku obróbki złożonych żądań pochodzących od aplikacji.	Czas ładowania części operacyjnej i aplikacyjnej – krótki. Czas reakcji systemu na zapytania użytkownika systemu operacyjnego – krótki. Możliwe opóźnienia w przypadku obróbki złożonych żądań pochodzących od aplikacji.

Zdefiniujmy charakterystyki wykorzystywane w powyższej tabeli. *Typ dostępności kodu źródłowego systemu* określa dostępność do treści programów, elementów zapewniających korzystanie z jądra systemu operacyjnego. *Użyteczność* to wizualna i funkcjonalna kompletność samego systemu i aplikacji opracowywanych za jego pomocą. *Wydajność w skali makro*, opisuje mierzone w osobodniach nakłady na przygotowanie identycznych aplikacji w różnych systemach operacyjnych [23]. *Wygoda tworzenia i uruchomienia aplikacji* odzwierciedla poziom zaawansowania narzędzi przeznaczonych do tworzenia, uruchomienia i eksploatacji aplikacji. *Dostępność programistów na rynku pracy* szacuje względną liczbę osób znających narzędzia wykorzystywane do tworzenia aplikacji. *Wydajność systemu operacyjnego w skali mikro* definiuje szybkość działania aplikacji w środowisku systemu, w szczególności oceniane są: czas ładowania systemu operacyjnego i wszystkich programów towarzyszących; czas reakcji na żądanie użytkownika; charakterystyki czasowe przetwarzania rozbudowanych zapytań [23].

Procentowy udział w rynku różnych systemów operacyjnych został przedstawiony w tabeli 2. Wykres ten przygotowano na podstawie danych udostępnionych w [24].

Tabela 2. Udział systemów mobilnych operacyjnych w rynku

System operacyjny	**Udział w zadanym czasokresie w %**				
	01.2014	**07.2014**	**01.2015**	**07.2015**	**01.2016**
Android	7,15	9,15	13,63	18,04	20,40
Windows XP	24,56	19,48	13,59	11,24	8,04
Windows 7	46,80	47,85	45,11	42,16	35,39
Windows 8	8,52	11,34	15,46	16,58	13,67
Windows 10	0,00	0,00	0,00	0,14	9,23
iOS	2,49	2,60	3,33	3,63	4,78
Linux	1,23	1,40	1,53	1,57	2,17

Windows Phone	0,00	0,00	0,45	1,13	1,39
Inne	9,25	8,18	6,90	5,51	4,93

W porównaniu z tabelą 2 założono, że dane o Windows 8 obejmują zarówno wersję 8.0 jak i 8.1. Statystyki opisujące Windows Phone uwzględniają wyłącznie wersje 8.1 oraz 10. Z analizy wykresu nasuwa się wniosek: udział *innych* produktów w rynku zmniejsza się i już obecnie ma pomijalnie mały poziom.

Dane dotyczące wyłącznie udziału wybranych systemów w rynku smartfonów zostały zaprezentowane tabeli 3. W statystyce pominięto produkty, których udział w rynku jest śladowy (Symbian [25], BlackBerry OS). Z przedstawionych danych wynika, że systemem, który bezsprzecznie dominuje na rynku jest Android, sprzedawany obecnie w wersji 6.0.

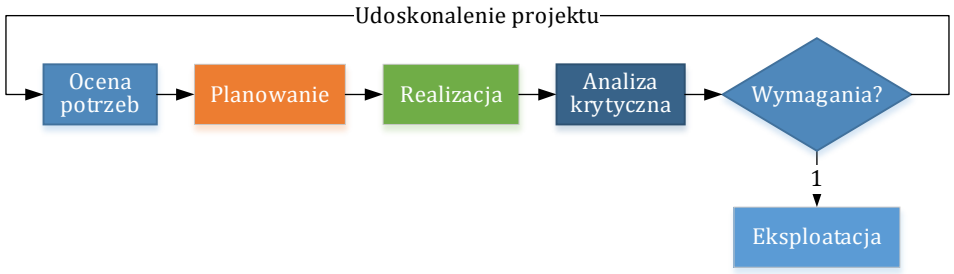
Tabela 3 Udział wybranych systemów operacyjnych w rynku telefonów

System operacyjny	**Udział w zadanym czasokresie w %**				
	01.2014	**07.2014**	**01.2015**	**07.2015**	**01.2016**
Android	74,18	77,91	78,30	76,40	77,72
iOS	25,82	22,09	19,11	19,12	16,99
Windows Phone	0,00	0,00	2,59	4,48	5,29

Analizując powyższe dane stwierdzono, że rynek systemów operacyjnych dla telefonów komórkowych jest dwubiegunowy. Zdaniem większości analityków tendencja ta będzie się jeszcze pogłębiać, a stan ten jest korzystny – zbytnie rozdrobnienie rynku doprowadziłoby do ograniczenia nakładów na rozwój oprogramowania. Oczywiście, dla użytkowników opłacalne byłoby umocnienie się pozycji systemu Windows Phone, jednak tendencja taka jest, zdaniem analityków, mało prawdopodobna.

3. Procedura realizacji aplikacji mobilnej

Korzystając z górnego poziomu abstrakcji, interaktywną realizację aplikacji mobilnej można przedstawić jako cykliczną sekwencję czterech kroków, pokazanych na rys. 1. Pierwszy z nich – *ocena potrzeb* – ma za zadanie określenie celowość realizacji projektu, w szczególności jego wartość komercyjną, socjalną, ekologiczną, dydaktyczną itp. Analizie podlega również rynek usług, w tym potrzeby użytkownika oraz dostępność podobnych produktów. Przeprowadzone badania pozwalają uniknąć sytuacji, kiedy na rynku istnieje już podobny produkt, co wiąże się z ryzykiem braku zainteresowania budowaną aplikacją.



Rys. 1. Cykliczna procedura realizacji aplikacji mobilnej

Następnym etapem jest *planowanie*, będące w istocie analizą informacji zebranych w poprzednim kroku procedury. Na tym etapie: definiuje się kamienie milowe projektu; szacuje niezbędne zasoby sprzętowe i narzędziowe; określa pracochłonność realizacji; gromadzi zasoby wiedzy niezbędnej do jego wykonania, a także określa pożądane charakterystyki projektu. Kolejny etap – *realizacja* – jest podstawowym krokiem analizowanego cyklu. Pierwszym z rozwiązywanych w jego trakcie zadań jest przygotowanie modelu logicznego realizacji produktu, a następnie sukcesywne tworzenie wszystkich komponentów wraz z oceną ich zgodności. Ostatnim etapem cyklu jest *analiza krytyczna*, bazująca na usystematyzowanym gromadzeniu informacji o funkcjonowaniu przyszłego produktu. Zbierane są zarówno opinie twórców projektu, jak i jego użytkowników. Uzyskana w ten sposób wiedza jest wykorzystywany do dalszego doskonalenia projektu. Na danym etapie może okazać się, że cel stawiany przez Zamawiającego nie został osiągnięty w satysfakcjonującym stopniu. Analiza krytyczna pozwala ustalić, które z fragmentów produktu odpowiadają zdefiniowanym wymaganiom, które zaś wymagają poprawy. Jeżeli przygotowana aplikacja nie spełnia oczekiwań klienta, powyższy proces jest powtarzany. Cykliczność omawianego procesu pozwala poprawić jakość końcowego produktu i lepiej dostosować go do oczekiwań klienta. Procedura projektowania może być powtarzana wielokrotnie, dzięki czemu produkt będzie systematycznie doskonalony. Jeżeli analiza krytyczna pokazuje, że aplikacja satysfakcjonuje odbiorcę, przechodzimy do etapu *eksploatacji*, w czasie którego testowanie może (aczkolwiek nie musi) być kontynuowane. W celu praktycznego zastosowania, sekwencja działań przedstawiona na rys. 1 powinna być uszczegółowiona, co pokazano w dalszej części rozdziału, a także w dostępnej literaturze [26], [27].

4. Testowanie aplikacji mobilnych

Dla większości współczesnej populacji, aplikacje mobilne stanowią nieodłączną część codziennego życia. Świadomi tego usługodawcy (banki, operatorzy telekomunikacyjni, firmy usługowe, sprzedawcy dóbr konsumpcyjnych), ale również organy administracji publicznej traktują aplikacje mobilne jako jeden z podstawowym obszarów realizacji swojej działalności. Dzięki temu, aplikacje mobilne charakteryzuje coraz szersza, zwiększająca się dynamicznie funkcjonalność, czego

konsekwencją jest ciągle rosnący strumień danych, w szczególności zapytań generowanych przez użytkowników urządzeń mobilnych i wbudowanych (Internet rzeczy) oraz udzielanych na nie odpowiedzi. W rezultacie, intensywność przetwarzania informacji, w szczególności na serwerach obsługujących aplikacje, istotnie wzrosła. Zwiększa się również nierównomierność wykorzystania kanałów komunikacyjnych, a także dominacja w sieci usług izochronicznych [28].

Problemem nabierającym nowego znaczenia jest także bezpieczeństwo informacji, którego zagrożenia rosną chociażby z powodu zwiększania masowości przetwarzania [29]. Nie bez znaczenia jest także możliwość nieprawidłowej obsługi aplikacji przez samego klienta, a także brak weryfikacji przez producenta aplikacji błędnych danych i działań wykonywanych przez użytkownika. Ponieważ większość użytkowników dysponuje ograniczoną wiedzą z zakresu cyberbezpieczeństwa, błędne funkcjonowanie jest w analizowanym przypadku szczególnie prawdopodobne. Prawdopodobieństwo bezbłędnego działania zespołu człowiek–aplikacja mobilna radykalnie zwiększa *wieloaspektowe testowanie*, wykonywane na etapach jej produkcji i eksploatacji [30].

Zgodnie z bazową definicją, *testowanie* to procedura sprawdzenia zgodności między rzeczywistym a oczekiwanym zachowaniem programu przeprowadzana w ściśle określony sposób, w specjalnie stworzonych, sztucznych warunkach [31]. Testowanie, podobnie jak cały proces programowania został ujednolicony za pomocą norm i standardów. Obowiązujące obecnie standardy testowania to ISO/IEC/IEEE 29119 oraz ISO12207 [32], [33]. Są one jednoznaczną instrukcją opisującą organizację, wykonanie i dokumentowanie procedur testowania. Wykorzystywane obecnie modele i kryteria są przydatne do testowania logicznych struktur programów, dynamiki ich zachowania, sekwencji operacji interfejsu graficznego, nie pozwalają one jednak jednoznacznie definiować wymagań stawianych przed procesem testowania. Jednym z tego powodów jest różnorodność środowisk mobilnych, wymagająca uwzględnienia środowiska operacyjnego, architektury urządzenia, przeglądarki, specjalistycznych API [33].

Rozważmy dokładniej różnice w testowaniu tradycyjnych aplikacji i programów mobilnych, rozpoczynając od zbadania różnic pomiędzy oboma typami oprogramowania. Pomimo widocznych postępów, urządzeń mobilnych nie wyróżniają znaczne zasoby obliczeniowe, charakterystyczne dla większości komputerów stacjonarnych. Testowanie aplikacji stacjonarnej zależy od wykorzystywanego modelu. Przykładowo, jeżeli zastosowanie znajduje technologia *cienkiego klienta*, głównie odbywa się ono na centralnym serwerze aplikacji. W przypadku programów mobilnych dominuje testowanie na urządzeniach końcowych, tj. smartfonach, tabletach, smartwatchach użytkownika. Do komunikacji z użytkownikiem aplikacje mobilne korzystają z ekranów o mniejszym rozmiarze i rozdzielczości niż aplikacje stacjonarne, różne są także podstawowe funkcjonalności urządzeń. W przypadku programów mobilnych, podstawowym zadaniem urządzenia jest nawiązywanie, utrzymywanie i obsługa połączeń tekstowych, głosowych i obrazowych pomiędzy

parą lub grupą abonentów. W przypadku komputerów stacjonarnych, funkcjonalności te są trzeciorzędne [7], [34].

Aplikacje mobilne funkcjonują w bardzo zawężonym zbiorze środowisk operacyjnych. Jak wynika z przedstawionych danych zbiór ten zawęża się faktycznie do dwóch elementów: systemów Android oraz iOS [35]. Dla aplikacji stacjonarnych zbiór środowisk operacyjnych jest znacznie szerszy i składa się z 6-7 elementów. Systemy operacyjne dla środowisk mobilnych starzeją się moralnie znacznie szybciej niż systemy stacjonarne.

Sposoby integracji z siecią Internet urządzeń mobilnych i stacjonarnych istotnie różnią się. W pierwszym przypadku do tego celu wykorzystuje się technologie 3G, 4G, Wi-Fi, drugim zaś technologie przewodowe Ethernet oraz Wi-Fi [36]. Wykorzystywane rozwiązania różnią się zarówno możliwościami technicznymi, jak i kosztami. Dodatkowo, połączenie urządzeń mobilnych nie ma stałego charakteru – urządzenia te są zmuszone permanentnie wyszukiwać sieć, a uzyskiwane połączenia mogą się różnić nie tylko z uwagi na zastosowana technologię, ale również warunki atmosferyczne, odległość od stacji bazowej czy zagęszczenie użytkowników. Dlatego, w odróżnieniu od aplikacji stacjonarnych, dla których możliwości komunikacyjne są stabilne, programy mobilne powinny być testowane dla różnych przepustowości łącza [37], [38].

Cechą wspólną aplikacji mobilnych jest szerokie zastosowanie nowoczesnych technik multimedialnych, co poprawia atrakcyjność jej komunikowania się z użytkownikiem. Wymaga to jednak jednoczesnego utrzymywania wielu kanałów wejściowych (klawiatura, dotyk, głos, gesty itp.) podlegających niezależnemu testowaniu. W tym obszarze możliwości aplikacji stacjonarnych są znacznie uboższe [38]. Powyższe (i inne) różnice w funkcjonowaniu obu typów aplikacji powodują, iż narzędzia stosowane do testowania programów stacjonarnych, nie zawsze są przydatne dla aplikacji mobilnych.

W przypadku aplikacji mobilnych, na dobór metod i środków testowania istotny wpływ przejawia także ich architektura. W ogólnym przypadku, wyróżniamy trzy alternatywne architektury: mobilne aplikacje webowe, aplikacje natywne oraz aplikacje hybrydowe [39], [40]. *Mobilna aplikacją webową* jest w istocie strona internetowa uruchamiana na urządzeniu użytkownika za pomocą mobilnej przeglądarki. Zaletami rozwiązania są: wysoka wydajność w skali makro; dostęp w technologii cienkiego klienta zapewniający minimalne obciążenie urządzenia mobilnego; prosta aktualizacja aplikacji oraz brak potrzeby instalacji. Z drugiej jednak strony, architektura nie oferuje autonomicznych funkcji, w porównaniu do pozostałych zapewnia ograniczoną funkcjonalność, pojawiają się kłopoty z redystrybucją, bowiem *Google Play* i *App Store* takowej nie obsługują. *Aplikacja natywna* jest przygotowywana wyłącznie dla wskazanej platformy operacyjnej, najczęściej Android i iOS. Do zalet tego typu rozwiązania należy zaliczyć: pracę w trybie autonomicznym; efektywne wykorzystanie funkcji urządzenia, w tym endemicznych; rozbudowany interfejs użytkownika; możliwość szerokiego zastosowania technologii

push. Główną wadą tego typu aplikacji są zwiększone koszty, zarówno opracowania jak i obsługi [41]. Ostatni typ aplikacji – *aplikacje hybrydowe* – są połączeniem obu poprzednich typów. Można zdefiniować je jako odwzorowanie zawartości mobilnej aplikacji webowej w formacie aplikacji natywnej. Najważniejsze zalety architektury to: pełne wykorzystanie możliwości urządzeń mobilnych; własna przeglądarka; prosta dystrybucja, a także niskie koszty przygotowania i obsługi [40].

Powyższe ustalenia pokazują, iż proces testowania należy do najważniejszych etapów całego cyklu życia produktu programistycznego. Dlatego, istotnym jest określenie jego miejsca w całym cyklu życia aplikacji. Aktualnym jest pytanie: czy z punktu widzenia cyklicznej procedury produkcji aplikacji mobilnej z rys. 1, testowanie powinno być częścią etapu realizacji projektu, czy też jego analizy krytycznej. Obecnie, do rąk odbiorcy trafia komercyjny produkt z wyeliminowaną maksymalną liczbą błędów i satysfakcjonującym poziomem użyteczności. Można więc wnioskować, że testowanie jest fragmentem etapu realizacji aplikacji. Jednak z doświadczeń autorów, a także analiz innych badaczy [34], [42] wynika, że:

1. Testowanie produktu powinno opierać się zarówno na sekwencjach testowych tworzonych manualnie, jak i generowanych automatycznie;
2. Procedury testowania powinny być wbudowane w każdy etap cyklu życia produktu;
3. Wydzielenie niezależnego etapu testowania poprzedzającego eksploatację produktu jest pożądane.

Rozważmy dalej testowanie aplikacji mobilnej, przed dopuszczeniem do dystrybucji, co zgodnie z rys. 1 może zostać wykonane na etapie analizy krytycznej. Jedną z podstawowych podejmowanych wtedy decyzji jest określenie czy testowanie bazować będzie na rzeczywistych urządzeniach, czy też na symulatorze [43]. Z punktu widzenia efektywności testowania, wykorzystanie do tego rzeczywistych urządzeń mobilnych jest rozwiązaniem, zapewniającym maksymalną skuteczność działań. Po pierwsze, ten rodzaj testowania gwarantuje najwyższą dokładność rezultatów, tj. wykrywane błędy pojawiają się rzeczywiście, a nie są efektem zdarzeń zachodzących w symulatorze. Po drugie, wchodzące przerwania (wywołania, SMS-y) oraz charakterystyczne właściwości sprzętu, takie jak działania sensorów wewnętrznych, lokalizacji geograficznej, wiadomości typu *push*, pojemności akumulatora testuje się prosto i skutecznie. Wszelkie procedury odbywają się w realnych warunkach, a pojawiające się błędy są zazwyczaj w 100% przenoszone pomiędzy różnymi egzemplarzami tego samego typu urządzenia [44].

Z uwagi na obszerność rynku urządzeń mobilnych, wykorzystanie podejścia nie jest trywialne. Inżynier stoi przed trudnym wyborem urządzenia wykorzystywanego do testowania. Testowanie dla wszystkich urządzeń dostępnych na rynku jest z oczywistych powodów niemożliwe, zaś przeprowadzenie testowania dla szerokiego ich zbioru wiąże się z nieakceptowalnymi kosztami. Dlatego, dobór urządzeń do grupy testowanych odbywa się na podstawie trzech prostych zasad [45]:

1. Wykorzystuje się urządzenia najpopularniejsze na rynku;

2. Urządzenia wchodzące w skład grupy powinny być zarządzane różnymi systemami operacyjnymi;
3. Należy testować urządzenia wyposażone w ekrany o różnej rozdzielczości, procesory z różną mocą obliczeniową, dysponujące różną pamięcią operacyjną.

Alternatywnym rozwiązaniem jest zastosowanie emulatorów lub symulatorów. *Emulacja* to odtwarzanie działania całego programu lub systemu (nie zaś ich fragmentów) z zachowaniem kluczowych właściwości i zasad funkcjonowania. Emulacja wykonuje kod programu w typowym dla niego środowisku, składającym się z tych samych elementów, co emulowany obiekt. Z kolei *symulacja* to czysto wirtualne odtwarzanie aplikacji na silniku specjalnego programu. Symulacja wyłącznie imituje wykonanie programu i w całości ma wirtualny charakter [46]. Choć oba powyższe procesy wyglądają podobnie, różnice pomiędzy nimi są bardzo wyraźne i dotyczą oferowanego poziomu wiarygodności. Emulacja naśladuje rzeczywiste zjawiska zachodzące w programie, symulacja zaś wyłącznie właściwości obiektu. Innymi słowy, emulator dokonuje zamiany rzeczywistego urządzenia na wirtualne. Z kolei symulator nie tworzy wirtualnej kopii urządzenia, a ogranicza się wyłącznie do udostępniania środowiska identycznego funkcjonującemu na oryginalnym urządzeniu. Z powyższych ustaleń wynika, że do testowania mobilnych witryn lepiej jest stosować emulatory. Z kolei symulatory lepiej sprawdzają się w procesie testowania aplikacji mobilnych. O popularności obu rozwiązań zdecydowały również takie czynniki jak: prostota instalacji, szybkość procedury testowania, niskie koszty realizacji procedur testowania. Z drugie jednak strony, metody te nie zapewniają kompleksowego testowania produktów mobilnych [43], [47].

Wbrew panującemu przekonaniu, testowanie jest złożonym procesem, w skład którego wchodzi kilkanaście różnych procedur. Zakres testowania zależy w pierwszej kolejności od skali wykorzystania aplikacji oraz jej przewidywanych zastosowań. Wykonanie kilkunastu procedur dla aplikacji używanej wyłącznie przez jej twórcę jest w większości przypadków nieuzasadnione. Z drugiej jednak strony, jeżeli aplikacja ta jest elementem infrastruktury krytycznej należy zapewnić maksymalne prawdopodobieństwo jej poprawnego funkcjonowania [30], [48], [49].

Praktycznie w każdym przypadku, wykonuje się trzy typy procedur testowania: testowanie funkcjonowania; testowanie interfejsu użytkownika oraz testowanie wrażliwości obciążeniowej. *Testowanie funkcjonalne* weryfikuje zgodność działania aplikacji ze zdefiniowanymi wcześniej wymaganiami. Technologia testowania funkcjonalnego silnie zależy od rodzaju weryfikowanej aplikacji. Po pierwsze, decydować o tym będzie typ aplikacji określający jego funkcję biznesową, inaczej testowane będą funkcjonalności aplikacji obsługujących bankowość elektroniczną, handel elektroniczny, rozrywkę, sieci socjalne, *m*-edukację. Metody i środki testowania będą zależeć również od docelowych odbiorców aplikacji, którymi mogą być informatycy, bankowcy, nauczyciele oraz użytkownicy niekomercyjni. Na jej przebieg przejawia wpływ również sposób dystrybucji aplikacji. W zakres testowania

funkcjonalnego wchodzą: testowanie pól; funkcjonalności; przerwań, także permanentna weryfikacja odpowiedzi użytkownika; testowanie poprawek oraz zasobów urządzenia. W ramach testowania funkcjonalnego, dodatkowo wykonuje się weryfikację: procedury uruchamiania i zamykania aplikacji; poprawności dołączenia użytkowników; bezbłędności komunikatów diagnostycznych; efektywności wykorzystania energii, a także parametrów zapewniających poprawne funkcjonowanie w środowisku sieci socjalnych [45], [50].

Testowanie *interfejsu użytkownika*, fragmentu aplikacji odpowiadającego za jej kontakt z klientem końcowym, ma zapewnić wysoką jakość elementu aplikacji, często postrzeganego jako czynnik wyznaczający poziom satysfakcji klienta [18]. Ten etap obejmuje takie działania jak: analiza funkcjonalno-technicznych wymagań stawianych przez interfejsem użytkownika; opracowanie wymagań oraz planów testowania interfejsu użytkownika; wykonanie przykładowych testów oraz analiza ich przebiegu; określenie kompletności pokrycia interfejsu użytkownika wymaganiami; sporządzanie raportów o problemach dotyczących rozbieżności w zachowaniu systemu i wymaganiach; przygotowanie raportów w sytuacji, kiedy dla pewnych fragmentów interfejsu opisy funkcjonowania nie są zdefiniowane. Działaniami najczęściej wykonywanymi na tym etapie są: weryfikacja funkcjonowania aplikacji z wyświetlaczami o różnej rozdzielczości oraz na urządzenia z różną architekturą; ocena strony estetycznej produktu; kontrola poprawności wyświetlania reklam (czy ich wyświetlanie nie utrudnia pracy z aplikacją) [51].

Innym przykładem szczególnie istotnej procedury testowania są *badania obciążeniowe* [52]. Zauważmy, iż mimo gwarantowanego determinizmu, funkcjonowanie aplikacji w warunkach minimalnego i maksymalnego obciążenia może się istotnie różnić. Chociaż błędne rezultaty funkcjonowania pojawiają się skrajnie rzadko, to ich brak lub zbyt długi czas oczekiwania na nie jest zjawiskiem powszechnym. Dlatego, komponenty serwerowe aplikacji poddawane są testowaniu imitującemu stan wysokiego obciążenia, wywołanego przez znaczną liczbę użytkowników. Głównymi celami badań obciążeniowych są: dobór poprawnych dla danej aplikacji i jej użytkowników zasobów obliczeniowych oraz ich konfiguracji; optymalizacja wydajności aplikacji, włączając w to konfigurację serwerów, klientów oraz optymalizację samego kodu; oszacowanie działania i wydajności aplikacji w przypadku wdrażania nowych wersji produktu lub poprawek do istniejących; oszacowanie działania i wydajności aplikacji na etapie wdrożenia do eksploatacji. Działaniami standardowo wykonywanymi na tym etapie są: sprawdzenie czasu reakcji aplikacji na różne typy żądań obsługi, w celu stwierdzenia iż spełnia ona wymagania akceptowalne przez użytkownika; testowanie funkcjonowania aplikacji przy obciążeniach kilkukrotnie przewyższających obciążenie nominalne. Poprzez śledzenie pracy aplikacji obciążonej nominalnie: badania jest stabilność jej funkcjonowania; określane jest jej zachowanie w przypadku większego od nominalnego napływu danych wejściowych; szacowana jest liczba użytkowników jednoczesnych, mogących efektywnie wykorzystywać aplikację [53].

Jedną z pierwszych procedur testowania, wykonywanych w trakcie przygotowania procesu produkcyjnego jest *weryfikacja dokumentacji* [54]. Zdaniem większości autorów, testowanie rozpoczyna się już na etapie produkcji oprogramowania [31]. Testerzy otrzymują mapy nawigacyjne, wzory ekranów, opis sytuacji awaryjnych, które poddawane są analizie pod kątem spójności i kompletności. Wszelkie sprzeczności pojawiające się w dowolnym z aspektów funkcjonowania aplikacji, powinny być wykryte i usunięte jeszcze przed rozpoczęciem kodowania. Na tym etapie definiowane i analizowane są plany testów, scenariusz testowania, macierze śledzenia [7], [55].

Istotnym typem testowania jest *weryfikacja użyteczności*, zorientowana na zapewnienie komfortu korzystania z aplikacji [56]. Celem samym w sobie jest w tym przypadku opracowanie zgodnego ze standardami, intuicyjnego interfejsu. Jako kluczowe parametry traktuje się wygodę, sprawność oraz efektywność korzystania z programu. Przykładami działań wykonywanych przez testera są: określenie optymalnego rozmiaru czcionki; weryfikacja poziomu zapełnienia menu kontekstowego; sprawdzanie synchronizacji działań użytkownika i aplikacji; określanie szybkości reakcji aplikacji na żądanie użytkownika [57].

Kolejnym rodzajem procedury testowania jest *weryfikacja zgodności*, wykonywana w celu zapewnienia optymalnej wydajności programu na różnych urządzeniach, z uwzględnieniem takich parametrów jak zasoby obliczeniowe urządzenia, rozdzielczość i rozmiar ekranu, wersje systemu operacyjnego i oprogramowania narzędziowego. W szczególności, prowadzone są działania, których celem jest zapewnienie poprawnej pracy w różnych środowiskach operacyjnych, na różnych przeglądarkach, w różnych konfiguracjach baz danych [39], [58].

Nadzwyczaj istotnym, chociaż często pomijanym, rodzajem testowania jest *badanie bezpieczeństwa* aplikacji, mające na celu testowanie bezpieczeństwa systemów informacyjnych wykorzystujących aplikację, a także analizę zagrożeń związanych z koniecznością zapewnienia kompleksowego podejścia do ochrony informacji [59]. W tym zakresie, wykonywane są następujące działania: weryfikowane jest odporność danych użytkownika na ataki pochodzącymi z sieci; sprawdzana jest skuteczność systemu uwierzytelniania użytkowników, odporność przed atakami na dostępność; sprawdzane są skutki buforowania plików, interakcje plików systemowych, luki w zabezpieczeniach [60].

Do rzadziej wykonywanych typów testowania możemy zaliczyć *testowanie odtworzenia* (sprawdzenie aplikacji z punktu widzenia skuteczności jej odtworzenia po awariach sprzętu, komunikacji lub oprogramowania) oraz *testowanie lokalizacji* (weryfikacja dostosowania aplikacji do różnic kulturowych użytkowników) [32]. Niektórzy z autorów rozróżniają jeszcze *testowanie beta* będące badaniem pierwszej publicznej wersji aplikacji. Bazuje ono na znacznej liczbie testerów o różnych kompetencjach, wiedzy, upodobaniach. Wymaga ono jednak przygotowania kompletnej wersji aplikacji [61].

Każdy z powyższych etapów testowania może opierać się specjalistycznych narzędziach. Przykładowo, jeżeli program ma być poddany testom w chmurze, do tego celu można zastosować następujące narzędzia: *Xamarin Test Cloud*, *Perfecto Mobile Continuous Quality Lab*, *Keynote Mobile Testing*; testując funkcjonalność aplikacji można skorzystać z *Appium*, *Selendroid*, *Robotium*, *Ranorex*; testowanie użyteczności może opierać się na: *Zoom*, *Loop11*; z kolei testowanie interfejsu użytkownika na *FitNesse*, *iMacros*, *Jubula*, *LoadUI*; do testowania zgodności przygotowano następujące narzędzia: *BrowserStack*, *CrossBrowserTesting*, *Litmus*, *Browsera*, *Rational Clearcase*, *Ghostlab*; badania obciążeniowe mogą zostać oparte na następujących produktach: *NeoLoad by Neotys*, *Apteligent*, *New Relic*, a testowanie bezpieczeństwa na: *CS*, *OWASP Zed Attack Proxy*, *Veracode*, *Google Nogotofail* oraz *SQL Map*. Pozostałe rodzaje testowania można oprzeć na następujących produktach: testowanie lokalizacji – *Ubertesters*, *eggPlant*; testowanie zmian – *Appium*, *Robotium*, *Ranorex*; testowanie beta – *HockeyApp*, *Ubertesters*, *TestFlight*. Wyczerpujące informacje o powyższych produktach dostępne są w Internecie

Na koniec, wyjaśnimy różnice pomiędzy testowaniem manualnym i automatycznym. Wielu spośród współczesnych specjalistów wyraża przekonanie, że manualne testowanie aplikacji wkrótce zaniknie. Jednak szczegółowa analiza tej tezy skłania do przeciwstawnego wniosku: dla każdej aplikacji testowanie automatyczne może być pominięte, jednocześnie rezygnacja z testowania manualnego jest niemożliwa. Za wykorzystaniem testowania manualnego przemawiają: wysoka elastyczność i lepsze modelowanie działań użytkownika. Ponadto, z krótkoterminowej perspektywy realizacji pojedynczych zadań jest tańsze niż testowanie automatyczne. Z drugiej jednak strony, testowanie manualne nie jest powtarzalne – powtórne testowanie aplikacji jest równie pracochłonne jak pierwsze, wszelkie działania wykonywane są stosunkowo wolno, a niektóre z wykonywanych testów (w szczególności, badania obciążeniowe) nie mogą być zrealizowane w trybie testowania manualnego [62], [63]. Do najważniejszych zalet testowania automatycznego można odnieść: wysoką szybkość wykonania procedur; powtórne wykonanie procedur testowania jest bezproblemowe, wysoki stopień podobieństwa wybranych testów dla szerokiej grupy aplikacji; formalizacja zapisów wyników testowania; niski koszt jednostkowy przy masowym użyciu. Wadami testowania automatycznego są: istnienie ograniczeń na pewne typy testów, wysoka pracochłonność przygotowania procesu testowania [5].

Chociaż z powyższej krótkiej analizy wynika, że zalety testowania automatycznego znacznie przeważają nad jego wadami, automatyzacja testowania aplikacji mobilnych rodzi nowe problemy, związane z brakiem standaryzacji mobilnej infrastruktury testowej, języków scenariuszy testowania i protokołów zapewniających powiązanie pomiędzy instrumentami testowania, a platformami sprzętowymi, operacyjnymi i narzędziowymi [62]. Z uwagi na częste wnoszenie poprawek do aplikacji mobilnych, niezbędna jest efektywna struktura programowo-sprzętowa, zezwalająca na wielokrotne wykonanie testów, wspierająca skalowalną automaty-

zację testowania. W praktyce sprowadza się to do organizacji chmur diagnostycznych, zawierających wszystkie elementy niezbędne do: dołączenia szerokiego zbioru modeli urządzeń, budowy chmur emulacyjnych pozwalających testować aplikacje mobilne, opracowania zunifikowanych programów zarządzających procedurami testowania, wspierających skalowalną automatyzację testów [64], [65].

5. Podsumowanie i dalsze prace

Analizy zaprezentowane w rozdziale skłaniają do następujących wniosków:

1. Zaproponowane modyfikacje terminologii, polegające na rozróżnieniu pojęć mobilności i przenaszalności pozwalają uniknąć nieporozumień przy syntezie i analizie aplikacji. Ich zastosowanie w publikacjach naukowo-technicznych należy uważać za wskazane;
2. Jeżeli środki dostępne wykonawcy są niewystarczające do wyprodukowania aplikacji przenaszalnej pomiędzy popularnymi środowiskami operacyjnymi, sukces komercyjny aplikacji jest najbardziej prawdopodobny, jeżeli będzie ona dostępna dla systemu operacyjnego Android;
3. Jako dominujące platformy operacyjne obsługujące aplikacje mobilne można uznać Android i iOS, przy czym, rola tej drugiej systematycznie maleje. Udział pozostałych systemów w tynku można uznać za śladowy;
4. Testowanie jest procesem wielopłaszczyznowym, wykonywanym zarówno manualnie jak i automatycznie. Testowanie automatyczne należy uznać za istotne rozszerzenie manualnego, które go jednak nie zastępuje. Badanie aplikacji we wszystkich płaszczyznach istotnie podwyższa koszty jej produkcji i nie w każdym przypadku jest konieczne;
5. Testowanie produktu należy opierać na sekwencjach testowych tworzonych manualnie oraz generowanych automatycznie. Procedury testowania o różnym zakresie funkcjonalnym i złożoności, powinny być wbudowane w każdy etap cyklu życia produktu.

W drugiej części rozdziału, zaprezentowano wyniki prac autorów poświęconych budowie kompleksowego systemu wspomagania testowania aplikacji mobilnych. W szczególności, omówiono metodologiczne podstawy jego funkcjonowania, możliwości funkcjonalne oraz korzyści z jego zastosowania.

Bibliografia

[1] T. Zhang, J. Gao, J. Cheng i T. Uehara, „Compatibility Testing Service for Mobile Applications," w *2015 IEEE Symposium on Service-Oriented System Engineering*, San Francisco, 2015.

[2] S. R. Choudhary, „Cross-platform testing and maintenance of web and mobile applications," w *ICSE Companion 2014 Companion Proceedings of the 36th International Conference on Software Engineering*, Hyderabad, 2014.

[3] K. Matthias i S. P. Kane, Docker: Up and Running: Shipping Reliable Containers in Production, Sebastopol: O'Reilly Media, 2015.

[4] A. Gillies, Software Quality: Theory and Management (3rd edition), lulu.com, 2011.

[5] D. Galin, Software Quality Assurance: From theory to implementation, Harlow: Pearson Education, 2004.

[6] „ISO/IEC 25063: Inżynieria systemów i oprogramowania -- Wymagania jakościowe oraz ocena systemów i oprogramowania (SQuaRE) -- Wspólny format przemysłowy (CIF) dotyczący użyteczności: Opis kontekstu użytkowania," Polski Komitet Normalizacyjny, 2014.

[7] K. Wiegers i J. Beatty, Software Requirements, third edition, Redmond: Microsoft Press, 2014.

[8] J. Beatty i A. Chen, Visual Models for Software Requirements, Redmond: Microsoft Press, 2012.

[9] S. H. Wang, D. Samadhiya i D. Chen, „Software Quality: Role and Value of Quality Models," *International Journal of Advancements in Computing Technology,* tom 3, nr 6, pp. 65-74, 2011.

[10] J. A. McCall, Factors in Software Quality. Volume I. Concepts and Definitions of Software Quality, PN, 1977.

[11] L. Chen, M. A. Babar i B. Nuseibeh, „Characterizing Architecturally Significant Requirements," *IEEE Software,* tom 30, nr 2, pp. 38-45, 2013.

[12] W. B. Frakes i K. Kang, „Software Reuse Research: Status and Future," *IEEE Transactions on Software Engineering,* tom 31, nr 7, pp. 529-536, 2005.

[13] B. Boehm, „A Spiral Model of Software Development and Enhancement," *IEEE Computer,* tom 21, nr 5, pp. 61-72, 1988.

[14] R. G. Dromey, „A Model for Software product Quality," *IEEE Transactions on Software Engineering,* tom 21, nr 2, pp. 146-162, 1995.

[15] „ISO/IEC 9126 Software engineering — Product quality," ISO, 2001.

[16] „ISO/IEC 25010: Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models," ISO, 2011.

[17] IAB, „Wpływ Internetu na Gospodarkę w Polsce," 2014.

[18] E. N. McKay, UI is Communication: How to design intuitive, user-centered interfaces by focusing on effective communication, Waltham: Morgan Kaufmann, 2013.

[19] E. Brunette, Eclipse IDE: Pocket Guide, Sebastopol: O'Reilly Media, 2005.

[20] N. Smyth, Android Studio Development - Android 6th Edition, CreateSpace, 2015.

[21] M. Gargenta i M. Nakamura, Learning Android. Develop Mobile Apps Using Java and Eclipse. 2nd Edition., O'Reilly Media, 2011.

[22] A. Whitechapel i S. McKenna, Windows Phone 8 Development Internals, Sebastopol: O'Reilly, 2013.

[23] M. Hajder, H. Loutskii i W. Stręciwilk, Informatyka. Wirtualna podróż w świat systemów i sieci komputerowych., I red., M. Hajder, Red., Rzeszów: Wydawnictwo Wyższej Szkoły Informatyki i Zarządzania, 2002.

[24] „Systemy operacyjne," Geminus, 01 01 2016. [Online]. Available: http://ranking.pl/pl/rankings/operating-systems.html.

[25] D. W. Wood, Smartphones and beyond: Lessons from the remarkable rise and fall of Symbian, 2014.

[26] M. Hajder, „Architektura wielopoziomowej sieci komunikacyjnej," w *Studium rozwoju sieci szerokopasmowej dla województwa Podkarpackiego na lata 2007-2013*, M. Stroński, Red., Poznań, Poznańskie Centrum Superkomputerowo-Sieciowe, 2008, pp. 11-50.

[27] M. Hajder, „Rekonfiguracja i rozbudowa sieci szerokopasmowej," w *Studium rozwoju sieci szerokopasmowej dla województwa Podkarpackiego na lata 2007-2013*, M. Stroiński, Red., Poznań, Poznańskie Centrum Superkomputerowo-Sieciowe, 2008, pp. 203-221.

[28] Ericsson, „Ericsson Mobility Report: On the Pulse of the Network Society," Ericsson, Sztokholm, 2015.

[29] M. Maffei i E. Tuosto, Trustworthy Global Computing, Rzym: Springer, 2014.

[30] J. S. Park, A. T. Suresh i G. An, „A Framework of Multiple-Aspect Component-Testing for Trusted Collaboration in Mission-Critical Systems," w *International Conference on Collaborative Computing: Networking, Applications and Worksharing*, Atlanta, 2006.

[31] J. A. Whittaker, Exploratory Software Testing: Tips, Tricks, Tours and Techniques to Guide Test Design, Stoughton: Pearson Education, 2010.

[32] „ISO/IEC/IEEE 29119: Software and systems engineering -- Software testing -- Part 1: Concepts and definitions," ISO, 2013.

[33] „ISO/IEC 12207: Systems and software engineering -- Software life cycle processes," ISO, 2008.

[34] J. Gao, X. Bai, W.-T. Tsai i T. Uehara, „Mobile Application Testing: A Tutorial," *Computer,* tom 47, nr 2, pp. 46-55, 2014.

[35] „Operating system market share," Netapplications, 01 01 2016. [Online]. Available: https://www.netmarketshare.com/operating-system-market-share.aspx?qprid=8&qpcustomd=1&qpsp=2016&qpnp=1&qptimeframe=Y.

[36] A. Kumar, Y. Liu i J. Sengupta, „Evolution of Mobile Wireless Communication Networks: 1G to 4G," *International Journal on Electronics & Communication Technology,* tom 1, nr 1, pp. 68-72, 2010.

[37] G. Boudour, C. Teyssié i Z. Mammeri, „Bandwidth Reservation for Heterogeneous Traffics in Mobile Wireless Networks," w *Fifth Advanced International Conference of Telecommunications*, Wenecja, 2009.

[38] N. C. Zakas, „The Evolution of Web Development for Mobile Devices," *ACM Queue,* tom 11, nr 2, 2013.

[39] P. Gokhale i S. Singh, „Multi-platform strategies, approaches and challenges for developing mobile applications," w *International Conference on Circuits, Systems, Communication and Information Technology Applications (CSCITA)*, Mumbai, 2014.

[40] N. Gok i N. Khanna, Building Hybrid Android Apps with Java and JavaScript, Sebastopol: O'Reilly, 2013.

[41] N. P. Huy i D. van Thanh, „Developing apps for mobile phones," w *7th International Conference on Computing and Convergence Technology (ICCCT)*, Seul, 2012.

[42] E. Alves, P. Machado, T. Massoni i S. Santos, „A Refactoring-Based Approach for Test Case," w *8th International Workshop on Automation of Software Test (AST)*, San Francisco, 2013.

[43] S. J. Levulis i D. J. Harris, „Are all tests equal? A comparison of emulator and device testing for mobile usability evaluation," w *Proceedings of the Human Factors and Ergonomics Society 59th Annual Meeting*, 2015.

[44] Segue Technologies Inc., Planning Your Mobile Application, Arlington, 2016.

[45] D. Knott, Hands-On Mobile App Testing: A Guide for Mobile Testers and Anyone Involved in the Mobile App Business, Crawfordsville: Pearson Education, 2015.

[46] J. Banks, J. S. Carson II, B. L. Nelson i D. M. Nicol, Discrete-Event System Simulation, Pearson Education, 2014.

[47] G. Miura, „Pushing the Boundaries of Traditional Heritage Policy: maintaining long-term access to multimedia content," *IFLA Journal,* tom 33, nr 4, pp. 323-326, 2007.

[48] M. Ouyang, „Review on modeling and simulation of interdependent critical infrastructure systems," *Reliability Engineering & System Safety,* tom 121, pp. 43-60, 2014.

[49] G. Pye i M. J. Warren, „Critical Infrastructure Systems Modelling: Benchmarking CPNTools," w *Proceedings of The 8th Australian Information Warfare and Security Conference* , 2007.

[50] R. Black, Advanced Software Testing Vol. 1: Guide to the ISTQB Advanced Certification as an Advanced Test Analyst, Santa Barbara: Rocky Nook, 2015.

[51] B. Laboon, A Friendly Introduction to Software Testing, CreateSpace, 2015.

[52] B. Wescott, The Every Computer Performance Book, 2013.

[53] I. Molyneaux, The Art of Application Performance Testing: From Strategy to Tools, Sebastopol: O'Reilly Media, 2015.

[54] D. L. Parnas, „Software Documentation and the Verification Process," w *Computer Aided Verification. CAV 2001. Lecture Notes in Computer Science*, Heidelberg, 2001.

[55] L. Crispin i J. Gregory, Agile Testing: A Practical Guide for Testers and Agile Teams, Pearson Education, 2009.

[56] C. J. Mueller, D. Tamir i O. V. Komogortsev, „An Economical Approach to Usability Testing," w *33rd Annual IEEE International Computer Software and Applications Conference*, Seattle, 2009.

[57] M. Lv, W. Hou i Z. Chunjing, „Research of usability test mode based on the implicit user behavior lib," w *2008 9th International Conference on Computer-Aided Industrial Design and Conceptual Design*, Kuming, 2008.

[58] A. Mesbah i M. R. Prasad, „Automated cross-browser compatibility testing," w *33rd International Conference on Software Engineering (ICSE)*, Honolulu, 2011.

[59] B. C. Zapata i A. H. Ninirola, Testing and Securing Android Studio Applications, Birmingham: Packt, 2014.

[60] D. Chell, T. Erasmus, S. Colley i O. Whitehouse, The Mobile Application Hacker's Handbook, Wiley, 2015.

[61] M. Fine, Beta Testing for Better Software, Wiley & Sons, 2002.

[62] A. Leitner, I. Ciupa i B. Meyer, „Reconciling Manual and Automated Testing: The AutoTest Experience," w *40th Annual Hawaii International Conference on System Sciences*, Waikoloa, 2007.

[63] J. Seon-Jae, K. Hae-Geun i C. Youn-ky, „Manual Specific Testing and Quality Evaluation for Embedded Software," w *Seventh IEEE/ACIS International Conference on Computer and Information Science*, Portland, 2008.

[64] D. Huizinga i A. Kolawa, Automated Defect Prevention: Best Practices in Software Management, Wiley-IEEE Computer Society Press, 2007.

[65] D. Graham i M. Fewster, Experiences of Test Automation: Case Studies of Software Test Automation, Crawfordsville: Pearson Education, 2012.

Rozdział 3

Zrównoważony rozwój transportu miejskiego w Rzeszowie na tle innych miast

Mariola NYCZ♦, Piotr HAJDER♣
♦Politechnika Rzeszowska,
♣*Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie,*
m_nycz@prz.edu.pl, Piootr.Hajder@gmail.com

Streszczenie

W artykule przedstawiono podstawowe informacje, dotyczące problematyki zrównoważonego rozwoju, w kontekście funkcjonowania transportu miejskiego. Omówiono ideę zrównoważonego rozwoju transportu miejskiego jako polityki wspierania środków alternatywnych do transportu indywidualnego i przedstawiono wybrane rozwiązania zaimplementowane w Rzeszowie. Oceniono poziom intensywności podejmowanych w tym zakresie działań w Rzeszowie na tle innych miast. Analizy przeprowadzono w oparciu o dane prezentowane przez GUS w dziale dotyczącym wskaźników zrównoważonego rozwoju.

1. Wprowadzenie

Rozwój stanowi zasadniczy cel działań podejmowanych przez rządy państw (także władze miast czy regionów), bo to właśnie rozwój decyduje o poziomie życia mieszkańców i w konsekwencji ich przyszłych wyborach politycznych. Koncepcja zrównoważonego rozwoju jest propozycją świadomego, a także odpowiedzialnego życia społecznego i indywidualnego, opartego o harmonię z otoczeniem społecznym i przyrodniczym uwzględniającym ograniczenia ekologiczne i oczekiwania społeczeństwa. Właściwe zdefiniowanie kierunków rozwoju zrównoważonych rozwiązań w transporcie miejskim i ich wdrożenie może znacząco przyczynić się do polepszenia poziomu jakości życia mieszkańców miast.

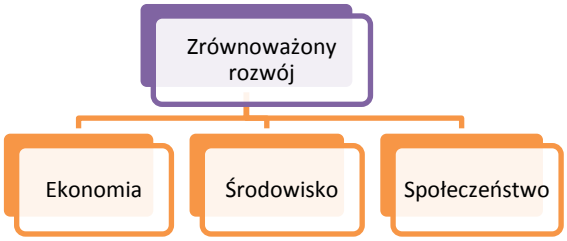
W kolejnej części przedstawiono historię koncepcji zrównoważonego rozwoju, jej początek i ewolucję na przestrzeni lat. Podrozdział trzeci i czwarty poświęcono zrównoważonemu rozwojowi miast, zdefiniowaniu pojęcia zrównoważonego rozwoju transportu poprzez zaprezentowanie rozwiązań, które zostały zaimplementowane w Rzeszowie wraz z inwestycjami planowanymi w najbliższym czasie. Ostatni, piąty podrozdział stanowi analizę mierzalnych i porównywalnych rozwiązań jakie wdrożono w Rzeszowie w porównaniu z innymi większymi miastami w Polsce w obszarze działań na rzecz zrównoważonego rozwoju.

Przedmiot szczegółowej analizy stanowi długość wprowadzonych buspasów na terenie miast w latach 2013-2015 oraz długość funkcjonujących ścieżek rowerowych w latach 2011-2015. Dynamika zmian tych wielkości jest pewnym mierzalnym wyznacznikiem działań prowadzanych w ostatnich latach w ramach zrównoważonego rozwoju transportu miejskiego. Analiza ma dać odpowiedź na pytanie o wielkość zróżnicowania działań na rzecz zrównoważonego rozwoju w polskich miastach, a także pozwolić na ocenę ich efektywności w Rzeszowie na tle innych miast. Dane wykorzystane w analizie pochodzą z baz Głównego Urzędu Statystycznego.

2. Koncepcja zrównoważonego rozwoju w świetle dokumentów

W raporcie, z przyjętej w Sztokholmie w 1972 roku Deklaracji Konferencji Narodów Zjednoczonych, podjęto się zdefiniowania pojęcia „zrównoważonego rozwoju" (ang. *Sustainable Development*) jako prawa do zaspokajania aspiracji rozwojowych przez obecną generację, bez ograniczania praw dla przyszłych pokoleń w kwestii ich potrzeb rozwojowych. Rozwój cywilizacyjny i gospodarczy nie powinien przebiegać kosztem wyczerpania zasobów nieodnawialnych czy niszczeniem środowiska tak, aby przyszłe pokolenia też posiadały prawo do rozwoju [1].

Oficjalne przyjęcie do realizacji koncepcji zrównoważonego rozwoju nastąpiło na drugim „Szczycie Ziemi" (konferencja ONZ „Środowisko i rozwój") w Rio de Janeiro w czerwcu 1992 roku. Uchwalono wówczas rezolucję zwaną „Agenda 21", przedstawiającą zastosowanie praktyczne zrównoważonego rozwoju na kilku płaszczyznach: przestrzennej, ekonomicznej, ekologicznej, społecznej i polityczno-instytucjonalnej [2]. Funkcjonowanie koncepcji zrównoważonego rozwoju sprowadza się do osiągnięcia trzech zasadniczych celów przedstawionych na rys. 1.



Rys. 1. Zasadnicze cele koncepcji zrównoważonego rozwoju [1]

Dziesięć lat później, w Johannesburgu odnowiono postanowienia z Konferencji w Rio de Janeiro, zaś przełom tysiącleci był okazją dla podkreślenia roli 189 członków Organizacji Narodów Zjednoczonych (ONZ) w kształtowaniu świata. 8 września 2000 roku przyjęta została Deklaracja Milenijna Narodów Zjednoczonych gdzie określono cele rozwoju będące zobowiązaniami mającymi pozwolić na skuteczne stawianie czoła wyzwaniom XXI wieku, a wdrażanie postawionych celów miało następować do końca 2015 roku. Warto wspomnieć, że w roku 2012 przypadła dwudziesta rocznica Konferencji Narodów Zjednoczonych określana Szczytem Ziemi Rio+20. Konferencja miała na celu pobudzenie politycznego zainteresowania

zrównoważonym rozwojem, dokonanie oceny postępów, zidentyfikowanie luk powstałych podczas wdrażania założonych koncepcji, a także wskazanie nowych kierunków i wyzwań na przyszłość [3].

Nowe Cele Zrównoważonego Rozwoju, przedstawione na rys. 2, zostały zaprezentowane w Agendzie na Rzecz Zrównoważonego Rozwoju 2030 („Przekształcenie naszego świata 2030 r."). Szczyt Zrównoważonego Rozwoju w Nowym Jorku, który odbył się we wrześniu 2015 r., to ostatnie istotne wydarzenie dotyczące rozważanej tematyki, które miało charakter globalny. Agenda została przyjęta jednogłośnie przez wszystkie 193 państwa członkowskie.

Reasumując, nowe cele są efektem długich negocjacji członków ONZ. Stanowią wspólne dobro publiczne wypracowane i zaakceptowane przez wszystkie kraje członkowskie z uwzględnieniem ich różnorodnej specyfiki, poziomu rozwoju i potencjału. Zestaw siedemnastu globalnych celów jest uniwersalny, a jednocześnie są to cele zintegrowane i niepodzielne, zharmonizowane z wymiarami zrównoważonego rozwoju.

Rys. 2. Nowe cele zrównoważonego rozwoju [4]

Od początku formułowania koncepcji zrównoważonego rozwoju państwa europejskie brały czynny udział w opracowywaniu i wdrażaniu światowych inicjatyw mających promować i umożliwiać zrównoważony rozwój w kontekście społecznym, ekonomicznym i ekologicznym. Do ważniejszych inicjatyw można zaliczyć Traktat Amsterdamski (1997 r.), Strategię Lizbońską (2000), Strategię Zrównoważonego rozwoju z Goeteborga (2001) z późniejszymi zmianami oraz Strategię Europa 2020 (przyjętą w czerwcu 2010 r.). Poza wymienionymi dokumentami, koncepcja zrównoważonego rozwoju pojawia się również i innych opracowaniach m.in. w Białej Księdze, Zielonej Księdze.

Podstawowe priorytety zaproponowane w celu osiągnięcia założeń „Strategii Europa 2020" stanowią [5]:

- wzrost inteligentny, czyli rozwój gospodarki uwarunkowany na wiedzę i innowacje;
- wzrost zrównoważony, gospodarka wspierana poprzez efektywne korzystanie z zasobów, przyjazne środowisku i konkurencyjne;
- wzrost sprzyjający integracji społecznej, a więc zapewniający spójność terytorialną i społeczną z wysokim poziomem zatrudnienia.

Do przedstawienia obrazu osiągniętych postępów w kierunku zrównoważonego rozwoju, w odniesieniu do założeń i celów zawartych w strategii na rzecz zrównoważonego rozwoju, opracowany został właściwy system wskaźników Unii Europejskiej. System ten jest zbudowany z następujących obszarów tematycznych: rozwój społeczno-ekonomiczny, włączenie społeczne, zrównoważona produkcja i konsumpcja, zdrowie publiczne, zmiany demograficzne, zmiany klimatu oraz energia, zasoby naturalne, zrównoważony transport, globalne partnerstwo i dobre rządzenie. Eurostat gromadzi odpowiednie dane na bieżąco i systematycznie je udostępnia.

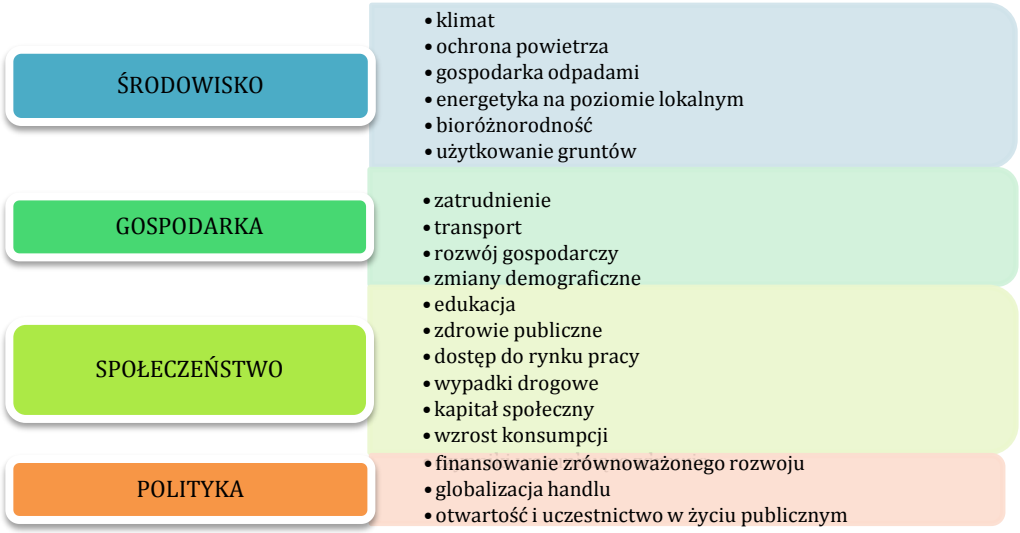
Zrównoważony rozwój jest problemem wielowymiarowym, stąd zoperacjonalizowanie tegoż pojęcia przez między innymi: ekonomistów, filozofów, ekologów, socjologów czy politologów na potrzeby swoich dziedzin.

3. Zrównoważony rozwój miast

Obecnie światowy rozwój ma swój początek w miastach. Znaczenie miast w rozwoju zarówno gospodarczym, jak i społecznym jest kluczowe. Obecnie około osiemdziesiąt procent Europejczyków mieszka w miastach (w Polsce jest to ok. 60%, przy czym od kilku lat udział mieszkańców miast powoli spada). Wśród polskich miast można znaleźć takie, które pod względem jakości życia dorównują miastom Europy Zachodniej.

Miasto stanowi jednostkę administracyjną składającą się z elementów, pomiędzy którymi zachodzą wielorakie i dynamiczne reakcje. Miasto to jednostka działająca na określonym terenie i w zadanym otoczeniu, która musi być traktowana, jako dynamiczna i funkcjonalna całość. Dlatego też traktuje się je jako system terytorialny. W zrównoważonym rozwoju miast należy uwzględniać wszystkie relacje zachodzące w tym systemie wraz z zachodzącymi w nim zmianami.

Według rankingu „Zrównoważony rozwój miast" liderem pod tym względem, wśród objętych analizą 66 miast okazała się Warszawa. Spośród elementów składowych tego rankingu można wyróżnić: środowisko, politykę, gospodarkę i społeczeństwo (niektóre z tych wymiarów zaprezentowano na rys. 3).



Rys. 3. Kryteria oceny zrównoważonego rozwoju miast w czterech wymiarach: środowisko, gospodarka, społeczeństwo i polityka

W „Raporcie miast uczących się", opracowanym w 2015 roku, również przygotowanym przez Politykę Insight na zamówienie Polskiej Fundacji im. Roberta Schumana i Fundacji Konrada Adenauera w Polsce, Rzeszów dla odmiany zajął wysokie, drugie miejsce. Kwalifikacji miast w rankingu dokonano na podstawie różnorodnych danych, przekształconych we wskaźniki, według których oceniane były miasta. Wśród kategorii oceny wyróżniono m.in. infrastrukturę produkcyjną (zasoby infrastrukturalne umożliwiające badania i tworzenie innowacji), kapitał ludzki (wykwalifikowani pracownicy na rynku lokalnym), współpracę między mieszkańcami i firmami oraz łatwość prowadzenia biznesu (efektywne funkcjonowanie instytucji publicznych, z którymi kontaktują się przedsiębiorcy). Mianem miast uczących się określa się te miasta, które funkcjonują jako „magazyny" pomysłów i nauki, dbające przy tym o środowisko i infrastrukturę, ułatwiającą przepływ wiedzy, stanowiącą motor wzrostu gospodarczego [6].

Rzeszów w tym rankingu opisany został jako miasto, które swoją przyszłość wiąże z technologiami, skąd jego nazwa: „Stolica nowych technologii". Ponadto, Rzeszów jest drugim miastem po Lizbonie, pod względem zapotrzebowania na informatyków. Opracowanie podkreśla także, że Rzeszów może być jednym z nielicznych miast, w którym w najbliższych latach można spodziewać się wzrostu liczby mieszkańców. Obok Warszawy i Poznania, to właśnie w stolicy Podkarpacia powstają firmy najnowocześniejsze, zatrudniające najlepszych pracowników, są to intensywnie rozwijające się ekonomicznie przedsiębiorstwa.

Rozwój transportu znacząco wpłynął na zwiększenie wykorzystywania nieodnawialnych źródeł energii, co niesie konsekwencje dla obecnych i przyszłych pokoleń. Unia Europejska wielokrotnie przedstawia i podkreśla istotność właściwych

działań w dziedzinie polityki transportowej koncentrując się przede wszystkim na rozwiązaniach przyjaznych środowisku i realizujących koncepcję zrównoważonego rozwoju [7].

Po przystąpieniu Polski do Unii Europejskiej zaczęły powstawać krajowe dokumenty dotyczące transportu i zrównoważonego rozwoju (m.in. Polityka Transportowa Państwa na lata 2006-2025, Strategia Rozwoju Kraju 2007-2015, Polska 2030 Wyzwania Rozwojowe i inne).

Ustawa o publicznym transporcie zbiorowym (z 1 marca 2011 r.) wprowadziła obowiązek sporządzania planów transportowych przez ministra (ds. transportu) oraz przez samorządy wojewódzkie i powiaty liczące ponad 80 tys. mieszkańców, miasta (powyżej 50 tys. mieszkańców), a także związki międzygminne (liczące ponad 80 tys. mieszkańców) [8]. Tak duży nacisk kładziony przez Unię Europejską na rozwój publicznego transportu zbiorowego wynika ze szczególnej jego roli w walce o ochronę środowiska, przeciwdziałaniu kongestii czy ograniczaniu liczby wypadków samochodowych. Jak widać, to właśnie wspieranie transportu zbiorowego stanowi najważniejszy element polityki zrównoważonego rozwoju [9].

4. Zrównoważony transport miejski w Rzeszowie

Zrównoważony rozwój transportu jest warunkiem wzrostu dobrobytu społecznego i rozwoju społeczno-gospodarczego. Dziś żyjemy szybciej niż jeszcze kilka lat temu, podróżujemy także szybciej, częściej i wygodniej, mając dostęp do nowych systemów i środków transportu. Zdefiniowanie pojęcia zrównoważonego transportu (działań) w odniesieniu do zrównoważonego rozwoju to [2]:

- zaspokajanie potrzeby mobilności wśród współczesnego pokolenia;
- odpowiednie wykorzystanie zasobów nieodnawialnych i korzystanie z zasobów odnawialnych;
- transport spełniający normy emisji hałasu i substancji szkodliwych;
- gwarantujący dostępność i różnorodność środków transportu;
- wpływający na poprawę jakości życia mieszkańców;
- transport przystępny cenowo;
- bezpieczny transport dla zdrowia ludzkiego i ekosystemów;
- minimalizujący potrzebę wykorzystania ziemi.

Rzeszów, podobnie jak inne miasta, przystąpił do realizacji programu, którego celem jest poprawa jakości usług transportu publicznego. W dużej mierze finansowanie inwestycji na te cele pochodzi ze środków Unii Europejskiej. Pierwszy etap programu transportowego został zrealizowany. Pasażerowie z pewnością zauważyli zakup nowych autobusów o podwyższonym standardzie, ale także:

- Przebudowę ważnych ulic (m.in. Lubelskiej, węzła Wyzwolenia-Warszawa czy Rejtana);
- Budowa połączenia drogowego ulic: Rzecha i Lubelskiej wraz z mostem im. Tadeusza Mazowieckiego;

- Utworzenie Inteligentnego Systemu Transportowego dla miasta włączając wyposażenie przystanków komunikacji miejskiej w biletomaty i tablice informacyjne (tablice zmiennej treści). Zadaniem Rzeszowskiego Inteligentnego Systemu Transportowego jest zwiększenie liczby pasażerów transportu publicznego, skrócenie czasu przejazdu pojazdów i usprawnieniem ruchu drogowego wszystkich uczestników dróg wraz z zapewnieniem bezpieczeństwa. W ramach Rzeszowskiego Inteligentnego Systemu Transportowego (RIST), obejmującego swym zasięgiem realizację szerszego wdrożenia: Zintegrowanego Systemu Zarządzania Ruchem i Transportem Publicznym (ZSZRiTP) integrują się działania następujących podsystemów [11]:
  - ✓ SOSRD – System Obszarowego Sterowania Ruchem Drogowym;
  - ✓ SZTP –System Zarzadzania Transportem Publicznym;
  - ✓ E-INFO – System Informacji Pasażerskiej;
  - ✓ E-BILET;
  - ✓ PLATFORMA TELEINFORMATYCZNA.
- Przebudowa niektórych skrzyżowań i zatok przystankowych;
- Wprowadzenie buspasów i zwiększenie długości ścieżek rowerowych;
- Wprowadzenie strefy płatnego parkowania w centrum miasta;
- Montaż wag samochodowych.

Te wszystkie działania to jednak sam szkielet, podstawa, bowiem istnieją duże szanse na kontynuację programu transportowego, dzięki dużemu dofinansowaniu z programu Polska Wschodnia, o które miasto będzie się starać. Wśród elementów kontynuacji programu transportowego wyróżnić można [10]:

- Zakup nowych autobusów, w tym także pojazdów o napędzie elektrycznym wraz z budową niezbędnej infrastruktury do ich obsługi (stacje ładowania, czy stacje warsztatowe);
- Dalsza przebudowa wiat przystankowych i zatok autobusowych;
- Przebudowa skrzyżowań m.in. Piłsudskiego z Cieplińskiego czy Wyzwolenia z Okulickiego;
- Zakup tablic zmiennej treści i wag samochodowych;
- Zakup oprogramowania do optymalizacji rozkładów jazdy;
- Zamontowanie systemu do monitorowania zatok autobusowych umożliwiających odczyt rejestracji pojazdów;
- Zakup systemu monitorowania buspasów i skrzyżowań;
- Wyposażenie strefy płatnego parkowania w system zliczania wolnych miejsc (dotychczasowy jest niedokładny).

Aby uzyskać dofinansowanie, miasto musi przygotować dokumentację aplikacyjną, poprzedzoną wykonaniem badań min. preferencji użytkowników środków transportu, natężenia ruchu.

5. Analiza wybranych rozwiązań na rzecz zrównoważonego transportu miejskiego, zastosowanych w Rzeszowie na tle innych miast

Wśród przykładów „dobrych praktyk" w tematyce zrównoważonego rozwoju transportu miejskiego wymienia się wprowadzanie buspasów i tworzenie infrastruktury dla rowerzystów. Dla zrównoważonego rozwoju istotne są te działania, których celem jest zmniejszenie udziału samochodów osobowych w przemieszczaniu się na korzyść komunikacji zbiorowej czy jazdy rowerem.

Zasadniczym celem wprowadzenia buspasów było usprawnienie i uwiarygodnienie komunikacji miejskiej, a także wywołanie przekonania rzeszowskich kierowców do pozostawienia samochodów osobowych i korzystania z komunikacji miejskiej. W tabeli Tabela 1 przedstawiono wybrane stolice województw, w których wyznaczono buspasy (do analizy zestawiono dane z tych miast, w których długość buspasów przekracza 5 km, ograniczenia w liczbie miast wynikają także z braku dostępu do danych). Pod względem długości buspasów pierwsze miejsce przypada Warszawie (48 km). Warszawa zdecydowanie wyprzedza pozostałe miasta objęte analizą. Drugim co do wielkości wynikiem charakteryzuje się Kraków (25,2 km) i Wrocław (21,5 km). Inaczej wyniki analizy przedstawiają się jako długość buspasów w stosunku do powierzchni administracyjnej miast. Takie odniesienie jest jednak bardziej zasadne ze względu na zróżnicowanie wielkości miast.

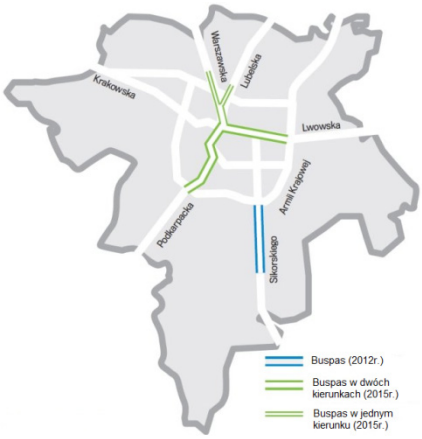
Tabela 1. Długość buspasów w stosunku do powierzchni miasta w latach 2013-2015

Miasto	Długość buspasów (w km)			Długość buspasów na 1 km² powierzchni (w km)	Zmiana procentowa (%)
	2013	2014	2015	2015	2013/2015
Białystok	11,8	12,6	12,8	0,125	8,5
Kielce	8,0	10,5	15,3	0,139	91,3
Kraków	23,7	23,7	25,2	0,077	6,3
Łódź	10,7	18,9	18,9	0,065	76,6
Olsztyn	6,8	6,8	13,3	0,151	95,6
Poznań	6,0	6,7	6,9	0,026	15,0
Rzeszów	3,3	3,3	11,4	0,097	245,5
Warszawa	48,0	48,0	48,0	0,093	0,0
Wrocław	21,5	21,5	21,5	0,073	0,0

Największą wartość wskaźnika długości buspasów na kilometr kwadratowy powierzchni miasta w 2015 roku odnotowano dla Olsztyna (0,151 km) oraz Kielc (0,139 km). Najniższa wartość wskaźnika charakteryzuje Poznań (0,026 km), Łódź (0,065 km), Wrocław (0,073 km), Kraków (0,077 km) i Warszawę (0,093 km). Rzeszów natomiast odnotował największy wzrost długości buspasów w 2015 roku: w stosunku do 2013 r aż o 245%. Na drugiej pozycji plasował się Olsztyn ze wzrostem o około 96%.

Obecnie długość buspasów na terenie Rzeszowa wynosi 11,4 kilometra, w całym systemie komunikacyjnym miasta jest to niewielki odcinek, jednak kluczową rolę

odgrywa ich zlokalizowanie w pewnych newralgicznych punktach centrum miasta. Przebieg buspasów przedstawiono na rys. 4.



Rys. 4. Przebieg buspasów w Rzeszowie [11]

Na uwagę zasługuje fakt, iż Rzeszów był pierwszym miastem w Polsce, które wprowadziło system 2+1 (dwóch pasażerów plus kierowca) w ramach poruszania się po buspasach, umożliwiający przejazd tymże pasem tych pojazdów, w których znajduje się minimum trzech pasażerów. Należy jednak podkreślić, że długość buspasów nie jest tak istotna, jeżeli ich miejsce wyznaczenia nie służy do ominięcia tych ulic czy punktów, które są źródłami kongestii.

Drugim wskaźnikiem mierzalnym przyjętym w analizie, umożliwiającym porównanie rozwoju miast, jest długość ścieżek rowerowych. Transport rowerowy jest jednym z alternatywnych rozwiązań dla transportu samochodowego.

Stworzenie warunków przyczyniających się do zwiększenia roli ruchu rowerowego w podróżach mieszkańców miast ogranicza negatywny wpływ transportu samochodowego na środowisko. W tabeli Tabela 2 przedstawione zostały zmiany w długości ścieżek rowerowych, jakie zaszły w latach 2011 i 2015 w stolicach województw.

Stan infrastruktury rowerowej z roku na rok poprawia się, co zauważyć można zarówno w dużych jak i w małych miastach. Zaprezentowane w tabeli zestawienie aktualnych długości ścieżek rowerowych wskazuje właśnie na tą poprawę. Miastem o największym wzroście długości ścieżek rowerowych, w wartości bezwzględnej do 2015 roku jest Warszawa (165,4 km w stosunku do 2011r.). Na drugim miejscu prezentuje się Gdańsk (77,1 km), na trzecim natomiast znalazł się Lublin (70,4 km). W ujęciu procentowym największy wzrost zanotowały Kielce (147%), Opole (138%) i Olsztyn (134%), zaś najmniejszy przyrost analizowanej wielkości odnotowano dla Wrocławia –jedynie 16,1%.

Tabela 2. Zmiany w długości ścieżek rowerowych w latach 2011-2015 w stolicach województw wraz z długością ścieżek rowerowych w stosunku do powierzchni miast

Miasto	Długość ścieżek rowerowych							
	Ogółem (w km)					Zmiana procentowa (%)	na 1 km² (w km)	na 10 tys. mieszk. (w km)
	2011	2012	2013	2014	2015	2011/2015	2015	2015
Białystok	66,2	75,2	91,8	106	110	66,2	1,08	3,7
Bydgoszcz	63,8	69,1	76,6	77,3	81,7	28,1	0,46	2,3
Gdańsk	90	122,5	147,8	159,8	167,1	85,7	0,64	3,6
Katowice	50,4	54,1	56	58,1	60	19,0	0,36	2,0
Kielce	16,7	18,7	18,7	28	41,3	147,3	0,38	2,1
Kraków	108	134,8	145	154	166	53,7	0,51	2,2
Lublin	59,6	65,2	79,2	109,6	130	118,1	0,88	3,8
Łódź	72,8	76,4	88	116	125	71,7	0,43	1,8
Olsztyn	31	40	42	59,3	72,5	133,9	0,82	4,2
Opole	28,1	32,9	37,4	59,2	66,8	137,7	0,69	5,6
Poznań	105	120,7	129	133,9	136	29,5	0,52	2,5
Rzeszów	65,9	82	95,5	120	120	82,1	1,03	6,5
Warszawa	291,8	318,3	359,6	412,5	457,2	56,7	0,88	2,6
Szczecin	83,6	87	98,5	110	114,7	37,2	0,38	2,8
Wrocław	197,9	204,1	210,3	215,4	229,8	16,1	0,78	3,6
Zielona Góra	38,4	39,6	b.d	b.d	62,2	62,0	0,22	4,5

W analizie, prezentującej zestawienie długości ścieżek rowerowych do powierzchni miast, największym wzrostem wykazał się Białystok (1,08 km na 1 km^2) i Rzeszów (1,03 km). Największa wartość wskaźnika ścieżek rowerowych na 10 tys. mieszkańców w 2015 roku charakteryzował Rzeszów (6,5) i Koszalin (5,9), najmniejszą wartością zaś Łódź (1,8), Katowice (2,0) i Kielce (2,1).

6. Podsumowanie

Możliwość wyboru zróżnicowanych form mobilności w obrębie miasta wpływa na odczucia atrakcyjności i funkcjonalności tego miasta dla jego mieszkańców. Wybór innego środka transportu niż indywidualny, powinno przekładać się na odczuwalny zysk, np. na zwiększenie szybkości przemieszczania się, czy wymiernie niższe koszty. Właściwe wykorzystanie szansy powstałej przez wprowadzanie warunków dla zrównoważonej mobilności, może pozwolić na zmniejszenie natężenia ruchu, a w ten sposób ograniczyć emisje hałasu czy substancji szkodliwych do środowiska, zwiększyć bezpieczeństwo na ulicach.

Zastosowane rozwiązania w Rzeszowie wskazują na właściwy kierunek podążania w celu realizacji koncepcji zrównoważonego rozwoju. Rzeszów nie odbiega od innych miast pod względem intensywności wprowadzanych działań na rzecz realizacji strategii zrównoważonego rozwoju transportu. Co więcej, jak wskazuje analiza i rankingi, znajduje się w czołówce miast intensywnie wdrażających rozwiązania, na rzecz polepszenia warunków życia mieszkańców w ramach koncepcji zrów-

noważonego rozwoju. Pamiętać jednak należy o konieczności właściwego dopasowania wprowadzanych działań, rozwiązań, ze względu na różnorodność miast. Nie wszystko co jest dobre dla jednych miast, sprawdzi się dla innych, stąd konieczność opracowywania indywidualnych planów dla miast.

Bibliografia

[1] E. Płaczek, „Zrównoważony rozwój nowym wyzwaniem dla współczesnych operatorów logistycznych," *Prace Naukowe Politechniki Warszawskiej - Transport,* nr 84, pp. 80-84, 2012.

[2] P. Brzustewicz, „Zrównoważone rozwiązania w transporcie miejskim - kierunki rozwoju," *ACTA Universities Nicolai Copernici - Zarządzanie XL,* nr 413, p. 86, 2013.

[3] Główny Urząd Statystyczny, „Wskaźniki zrównoważonego rozwoju Polski 2015," Urząd Statystyczny, Katowice, 2015.

[4] M. Antonowicz, „Czy nowe cele Zrównoważonego Rozwoju należą jeszcze do ONZ?," Globalne Południe, 25 Październik 2015. [Online]. Available: http://globalnepoludnie.pl/Czy-nowe-Cele-Zrownowazonego.

[5] *Komunikat komisji EUROPA 2020 - Strategia na rzecz inteligentnego i zrównoważonego rozwoju sprzyjającego włączeniu społecznemu,* 2010.

[6] „Europolis 2015," Polska Fundacja im. Roberta Schumana, 5 Lipiec 2015. [Online].

[7] D. Miłaszewicz i B. Ostapowicz, „Warunki zrównoważonego rozwoju transportu w świetle dokumentów UE," *Studia i Prace Wydziału Nauk Ekonomicznych i Zarządzania,* nr 24, p. 104, 2011.

[8] B. Kowalski, „RBF - Czy plany transportowe są potrzebne," Railway Business Forum, 5 Lipiec 2015. [Online]. Available: http://www.rbf.net.pl/wiecej,Czy-plany-transportowe-sa-potrzebne,736.

[9] T. Bużałek, „Łódzka inicjatywa na rzecz Przyjaznego Transportu," w *Wyzwania zrównoważonego rozwoju w Polsce*, Wrocław, Centrum Rozwiązań Systemowych, 2010, pp. 270-271.

[10] „Serwis informacyjny UM Rzeszów - Rozbudowa inteligentnego systemu transportu drogowego na terenie miasta Rzeszowa," 5 Lipiec 2015. [Online]. Available: http://www.rzeszow.pl/miasto-rzeszow/realizowane-projekty/projekty-zrealizowane/projekty-dofinansowane-ze-srodkow-unii-europejskiej-2007-2013/rozbudowa-inteligentnego-systemu-transportu-drogowego-na-terenie-miasta-rzeszowa.

[11] Gazeta Wyborcza, „Buspasy w Rzeszowie. Zasady, mapa, lokalizacje, zdjęcia," 5 Lipiec 2015. [Online]. Available: http://rzeszow.wyborcza.pl/rzeszow/56,34962,1786835

1,mapa-buspasow-w-rzeszowie,,2.html.

[12] „Rzeszowski Inteligentny System Transportowy - Budowa systemu integrującego transport publiczny miasta Rzeszowa i okolic," 5 Lipiec 2015. [Online]. Available: http://www.transport.erzeszow.pl/zakres-rzeczowy/rzeszowski-inteligentny-system-transportowy.

ROZDZIAŁ 4

Architektura informatyczna bezprzewodowych systemów monitoringu środowiskowego

Mirosław HAJDER

Wyższa Szkoła Informatyki i Zarządzania z siedzibą w Rzeszowie,
Miroslaw.Hajder@gmail.com

Streszczenie

W rozdziale, zaprezentowano architekturę informatyczną systemów monitoringu środowiskowego bazującego na zastosowaniu bezprzewodowych sieci sensorowych. Autorzy koncentrują się na zapewnieniu charakterystyk eksploatacyjnych niezbędnych w systemach infrastruktury krytycznej. W tym celu, poddają analizie przepływy informacyjne w sieci tej klasy oraz złożoności czasowe algorytmów wykonania podstawowych funkcji systemów monitoringu. Podstawowym efektem prowadzonych badań jest opracowanie i weryfikacja architektury systemu monitoringu zapewniającej efektywne śledzenie stanu środowiska. Rozdział adresowany jest do osób zajmujących się zarządzaniem kryzysowym oraz projektowaniem systemów monitoringu środowiskowego.

1. Wprowadzenie

Problem zachowania dla przyszłych pokoleń różnorodności środowiska naturalnego jest jednym z najważniejszych przed którymi stoi ludzka cywilizacja. Ochrona otaczającej nas przyrody, to wyjątkowo złożone, wieloplanowe zadanie. Zauważmy, że z ochroną przyrody należy pogodzić stabilny rozwój państwa, którego obowiązkiem jest troska o swoich obywateli. Nie powinno więc być tak, że któryś z celów stawianych przez państwo będzie dominował koniecznością zachowania otaczającej nas przyrody dla przyszłych pokoleń. Z drugiej jednak strony, ochrona przyrody nie może być celem samym w sobie i dominować nad wszystkim pozostałym.

Degradacja otaczającej nas przyrody nie zawsze jest skutkiem planowych działań człowieka. Zdarza się również, że za zagrożeniem pewnych gatunków stoi sama przyroda. Bywa również tak, że człowiek nieświadomie przyczynia się do ich zagrożenia, a nawet wyginięcia. W takich sytuacjach zbawiennym może okazać się działanie człowieka, który ingerując w ekosystem może doprowadzić go do pier-

wotnego stanu. Aby reakcja nastąpiła w odpowiednim czasie konieczna jest obserwacja przyrody, będąca jedną z najszerzej wykorzystywanych metod poznawczych, polegającą na względnie długiej, ukierunkowanej i planowej apercepcji przedmiotów i zjawisk otaczającej nas rzeczywistości [1]. Doskonałe przykłady obserwacji środowiska naturalnego pokazane zostały jeszcze w I wieku w *Historii naturalnej* Pliniusza Starszego. W 38 tomach zawierających informacje z dziedziny astronomii, fizyki, geografii, botaniki, zoologii, medycyny wykorzystywał on obserwację jako podstawową metodę naukową.

Nie tylko ochrona otaczającego nas środowiska, ale również efektywne wykorzystanie jego zasobów, wymaga posiadania informacji o właściwościach dynamicznych obiektów tworzących je, wiedzy o zmianach będących rezultatem działań antropogenicznych. Pozwala ona przewidzieć skutki ingerencji człowieka w przebieg naturalnych procesów, jest ona również niezbędna przy prognozowaniu niebezpiecznych zjawisk przyrodniczych.

Jednym ze skutków systematycznej poprawy jakości życia ludzi jest niepokojące pogorszenie stanu otaczającego nas środowiska. W wielu miejscach kuli ziemskiej, koncentracja substancji szkodliwych w powietrzu, wodzie lub glebie zagraża zdrowiu mieszkających i pracujących tam osób. Dlatego w ostatnim dziesięcioleciu wiele uwagi poświęcono przeciwdziałaniu negatywnym skutkom urbanizacji i uprzemysławiania kolejnych obszarów naszej planety. Jedną z wykorzystywanych do tego celu metod jest monitorowanie stanu środowiska, pozwalające śledzić szkodliwe zjawiska zachodzące w naszym otoczeniu i w odpowiednim czasie podejmować stosowne działania. Monitoring może być prowadzony na trzech różnych poziomach: *wpływu* (badanie silnych oddziaływań w skali lokalnej); *regionalnym* (analiza przemieszczania i transformacji zanieczyszczeń oraz ich wpływu na ekonomikę regionu); *tła* (badania na terenie rezerwatów biosfery, wyłączonych z jakiejkolwiek działalności gospodarczej) [2], [3].

Zazwyczaj, terminem *monitoring środowiska* określamy system obserwacji i kontroli stanu otoczenia, wspomagający racjonalne wykorzystanie zasobów naturalnych, ochronę przyrody i zapewnienie stabilnego funkcjonowania różnych systemów gospodarczych. Zgodnie z inną, powszechnie wykorzystywaną definicją, monitoring to system permanentnej obserwacji składowych środowiska i biosfery jako całości poprzez pomiar ich wybranych charakterystyk.

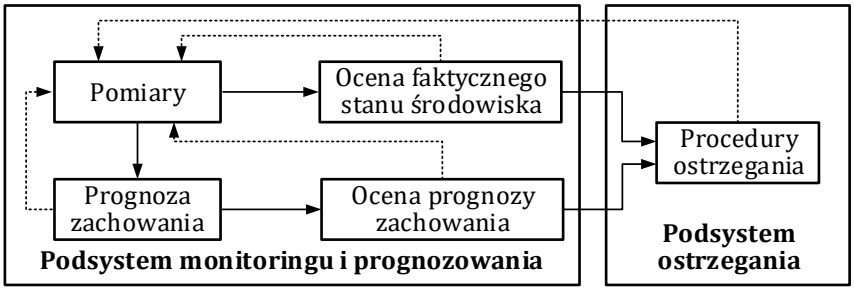
Analizą zagrożeń zajmuje się fundamentalna dyscyplina naukowa, nazywana *ogólną teorią bezpieczeństwa* (OTB) [4], [5]. Pierwszoplanowym obiektem badań OTB jest określenie fundamentalnych prawidłowości przejścia naturalnych systemów przyrodniczych, obiektów technicznych, sfery biologicznej i struktur społeczno-gospodarczych ze stanu normalnego funkcjonowania, do stanu awaryjnego lub katastroficznego, a także zasad ich wzajemnego współdziałania w procesie wspomnianej zmiany stanu. Ponadto, OTB zajmuje się budową naukowych podstaw diagnozowania, monitoringu i prognozowania zagrożeń, a także metodami

przeciwdziałania katastrofom i likwidacji ich skutków. W OTB, szczególne znaczenie zajmuje określenie sposobów szacowania i pomiarów skali zagrożeń oraz poziomu ochrony obiektów i obszarów. Na tej podstawie, definiowane są ilościowe i jakościowe parametry podejmowanych decyzji, zawartość dokumentów normatywnych oraz konkluzje komisji badających skutki awarii i katastrof. W OTB, za najbardziej uogólnione kryteria oceny, przyjęto uważać poziom ryzyka dla zdrowia i działalności człowieka, a także jakość i zagrożenia jego życia.

Systemy monitoringu środowiska mogą pracować w warunkach ekstremalnych, mogą być również celem ataków cybernetycznych. Dlatego, projektując je szczególną uwagę należy zwrócić na zapewnienie ciągłości ich funkcjonowania oraz odporność na ataki, których celem byłoby zafałszowanie opracowywanych prognoz zmian pogody czy też środowiska naturalnego jako całości. Jest to szczególnie istotne bowiem w systemach monitoringu celem poprawy trafności generowanych prognoz coraz szerzej wykorzystuje się elementy pomiarowe będące własnością osób prywatnych i jawiące się jako komponenty Internetu rzeczy.

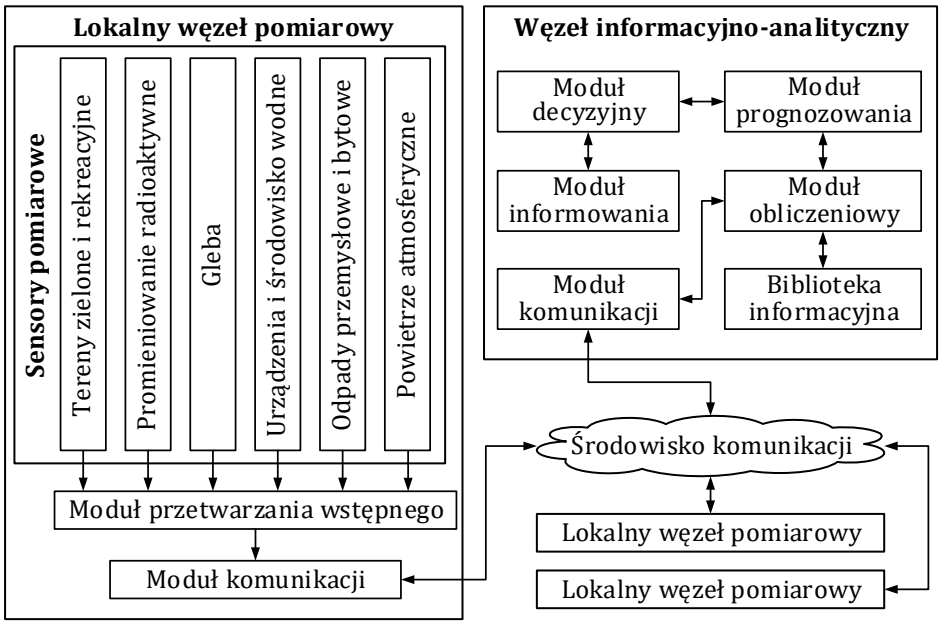
2. Architektura informacyjna systemu monitoringu

Architektura informacyjno-komunikacyjna systemu monitoringu przedstawiona została na rys. 1.



Rys. 1. Architektura funkcjonalna systemu monitoringu i ostrzegania

Cechą szczególną systemów monitoringu środowiska, wykorzystywanych w strukturach inteligentnych miast są: powiązanie z nimi systemów ostrzegania oraz szerokie wykorzystanie sprzężenia zwrotnego pozwalającego na bieżąco reagować na zachodzące zmiany środowiskowe. Architektura funkcjonalna systemów monitoringu środowiskowego została przedstawiona na rys. 2.



Rys. 2. Architektura funkcjonalna systemu monitoringu środowiskowego

3. Architektura topologiczna systemów monitoringu

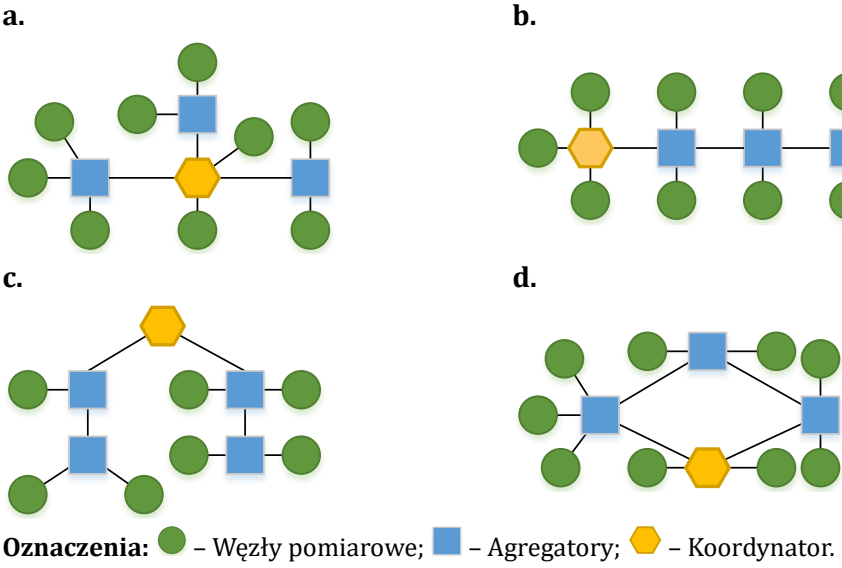
Współczesne sensorowe sieci monitoringu można budować w postaci struktur jednościeżkowych, w których istnieje wyłącznie jedna trasa łącząca węzeł pomiarowy z koordynatorem sieci oraz wielościeżkowej z trasami alternatywnymi łączącymi węzeł nadawczy i odbiorczy. Trasy te mogą być zarówno jedno- jak i wieloskokowa. W architekturze jednoskokowej, węzeł pomiarowy łączy się bezpośrednio z koordynatorem sieci, co w przypadku większych odległości pochłania znaczną energię na realizację komunikacji, zarówno od strony węzła pomiarowego jak koordynatora. Jeżeli zastosowanie ścieżek jednoskokowych jest obligatoryjne (np. kiedy struktura pomiarowa musi być płaska), węzeł pomiarowy wyposaża się w odnawialne źródło energii, ewentualnie minimalizuje wykonywane w nim operacje obliczeniowe. Z uwagi na swoje ograniczenia, architektury jednoskokowe wykorzystywane są stosunkowo rzadko.

W jednościeżkowej architekturze wieloskokowej dane od węzła pomiarowego do odbiorczego przechodzą przez wiele węzłów pośrednich, jednak trasa ich przesyłania jest absolutnie jednoznaczna i marszrutyzacja nie jest wymagana. Dzięki komunikacji wieloskokowej, odległości, na które w każdym skoku przesyłana jest informacja są ograniczone, co sprzyja minimalizacji niezbędnej mocy zasilania każdego z węzłów. Również wymagania energetyczne stawiane przed koordynatorem sieci są łatwiejsze do spełnienia: chociaż główne węzły sieci są z zasady wyposa-

żane w odnawialne źródło energii, to minimalizacja zapotrzebowania energetycznego sprzyja wydłużeniu długości eksploatacji urządzeń zasilających. Istotną wadą rozwiązań jednościeżkowych (zarówno jedno- jak i wieloskokowych) jest niska odporność na uszkodzenia: awaria chociażby jednego węzła na trasie przesyłania danych powoduje jej niespójność. Pomimo tego, sieci takie są z uwagi na niski koszt stosowane bardzo szeroko.

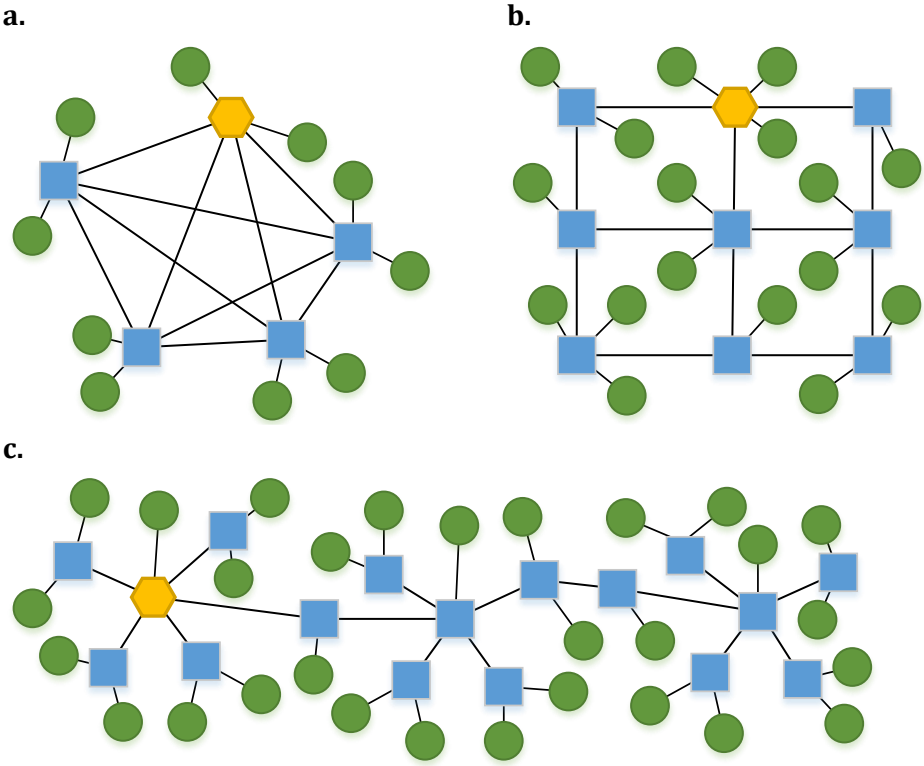
Architektura sieci wielościeżkowej zakłada, że dane pomiędzy węzłem pomiarowym a głównym agregatorem mogą być przesyłane za pomocą wielu alternatywnych ścieżek. Dlatego, każdy z węzłów pośrednich, przez które przesyłane są dane musi posiadać możliwość ich marszrutyzacji. W efekcie, węzły tę są bardziej rozbudowane, ich cena i zużywana moc jest wyższa. Jako węzły routujące wykorzystywane są przede wszystkim agregatory.

Przykłady trzech podstawowych jednościeżkowych topologii sieci sensorowych oraz jednej wielościeżkowej, zostały przedstawione na rys. 3.



Rys. 3. Proste topologie sieci sensorowych: a. Gwiazda; b. Magistrala; c. Drzewo; d. Pierścień

W topologiach z rysunków a., b. oraz c. dla dowolnej pary wierzchołków istnieje tylko jedna łącząca je ścieżka. W topologii pierścieniowej (rysunek d.) każdy z agregatorów może być dołączony do koordynatora za pomocą dwóch alternatywnych ścieżek. Topologie zaprezentowane na rys. 3 znajdują zastosowanie w niewielkich sieciach monitoringu. Jeżeli śledzony obszar oraz liczba węzłów pomiarowych są większe, najczęściej wykorzystywane są drzewo klastrowe, krata oraz sieć bazująca gęstym rdzeniu. Sieci te zostały zaprezentowane schematycznie na rys. 4. Chociaż sieci te składają się ze znacznie większej liczby węzłów i połączeń, idee ich budowy zostały zaczerpnięte z prostych topologii sieci sensorowych.



Rys. 4. Topologie rozbudowanych sieci sensorowych: a. Topologia z zupełnym rdzeniem; b. Topologia kratowa; c. Topologia drzewa klastrowego

4. Wpływ architektury na procedury projektowe

Zastosowanie bezprzewodowych sieci sensorowych (BSS) jest jedną z najbardziej aktywnie rozwijających się metod gromadzenia i przesyłania danych. Postęp w tym kierunku wynika głównie z rozszerzenia zakresu zastosowania takich sieci. Tak więc rozwiązanie zadania ujednolicenia procesu projektowania BSS staje się coraz bardziej pilne. Sieci sensorowe ciasno wypełniły swój segment rynku, duża liczba firm oferuje swoje rozwiązania techniczne. Dostępne są programowo-sprzętowe zestawy komponentów BSS, przeznaczone głównie do rozwiązywania precyzyjnie zdefiniowanych, wąskich tematycznie, zdań natury technicznej. Wysoki poziom specjalizacji utrudnia dostosowanie dostępnych zestawów do rozwiązania innych, nawet podobnych zadań. Nawet w przypadku potrzeby wykonania drobnych modyfikacji w architekturze systemu, konieczne jest ponowne wykonanie wszystkich kroków projektowania. Obecnie trudno jest nazwać chociażby jeden zestaw zunifikowanych narzędzi projektowania BSS, każdy z dostawców wykorzystuje własne zamknięte narzędzia analizy parametrów i syntezy rozwiązań technicznych. Dlatego, metody rozwiązania zadania projektowego opracowane dla

konkretnej organizacji BSS są zazwyczaj nieprzydatne, nawet w zbliżonych organizacjach i architekturach.

Bezprzewodowa sieć sensorowa jest rozproszonym, samoorganizującym się zbiorem wielu sensorów oraz urządzeń wykonawczych połączonych pomiędzy sobą za pomocą bezprzewodowego kanału radiowego. Zasięg takiej sieci może wynosić od kilku metrów do kilku kilometrów. Tak duże odległości uzyskuje się poprzez retranslację (przekazywanie) komunikatów w łańcuchu komponentów komunikacyjnych, od jednego do drugiego elementu.

Wielu badaczy i inżynierów akcentuje ogólne właściwości sieci czujników, takie jak ograniczony czas działania węzłów wynikający z zastosowania w każdym z nich autonomicznego źródła zasilania, zawodne połączenie między węzłami, będące skutkiem zastosowania kanału radiowego jako medium transmisyjnego oraz konieczność samoorganizacji funkcjonowania – automatycznej lub wymagającej minimalnego udziału człowieka.

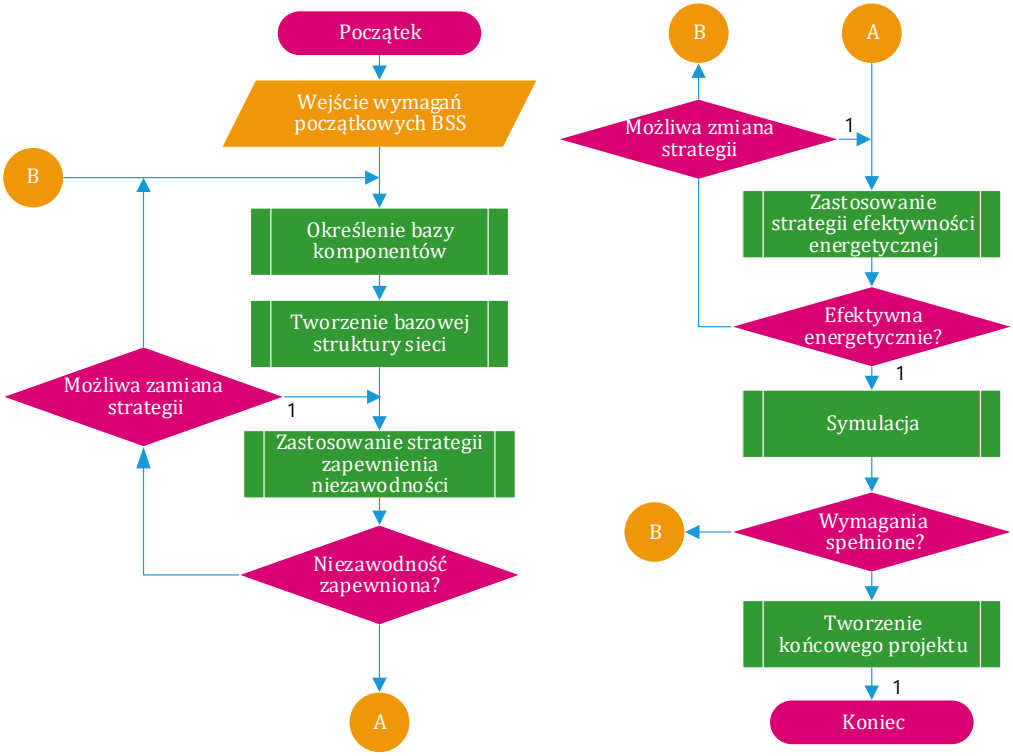
Rozważmy obecnie architekturę topologiczną sieci sensorowej. Proces projektowania topologii BSS w dużej mierze zależy od jej docelowego przeznaczenia. Jednak przygotowanie niezależnych tras dla każdego zadania jest czasochłonne i niepraktyczne – nierzadko konieczne jest połączenie tras przesyłania komunikatów pochodzących od różnych usług sieci sensorowej. Do tego celu możemy stosować różne struktury topologiczne, które mogą być kryterium klasyfikacji BSS. Z uwagi na sieć połączeniową najczęściej wyróżnia się sieci statyczne, słabo dynamiczne oraz dynamiczne.

Cechą klasyfikacyjną może być rodzaj struktury topologicznej, zgodnie z którą rozróżnia się układy statyczne, słabo dynamiczne i dynamiczne. W zadaniu Zleceniodawcy zastosowanie znajdzie statyczna topologią węzłów. Procedurę projektowania takiej sieci pokazano graficznie na rys. 5. Na podstawie schematu blokowego rysunku można wyróżnić szereg, omówionych zalecanych etapów procedury projektowania.

Rozważmy poszczególne etapy procedury projektowania.

Wejście wymagań początkowych BSS określanych na podstawie analizy zadania projektowego. Lista wymagań obejmuje takie parametry, jak: cykl życia sieci, zasięg, poziom aktualności informacji, szybkość transmisji danych oraz poziom niezawodności sieci. Definiowanie powyższych parametrów może mieć różny stopień szczegółowości określający dokładność danych wejściowych.

Działania podejmowane na etapie definiowania wymagań początkowych formalnie opisuje wyrażenie: $W = f(P)$, gdzie: $P = \{p_1, \cdots, p_n\}$ – wektor wymagań początkowych; $W = \{w_1, \cdots, w_n\}$ – wektor sformalizowanych wymagań; $f(p)$ – funkcja formalizacji; n – liczba wymagań. Lista wymagań początkowych definiowana jest w opisie przedmiotu zamówienia i składa się z kilkunastu-kilkudziesięciu wymagań początkowych. Efektem działań projektowych jest stworzenie wektora sformalizowanych wymagań o podobnym rozmiarze.



Rys. 5. Procedura projektowania bezprzewodowej sieci sensorowej ze statycznymi połączeniami

Kolejnym krokiem procedury z rys. 5 jest wybór bazy komponentów. Szybki rozwój technologii SMS wynika przede wszystkim z dynamicznego rozwoju mikroelektroniki, która zapewniła dostępność bazy komponentów, charakteryzujących się odpowiednim poziomem technicznym. Do podstawowych parametrów uwzględnianych przez konstruktorów w procesie projektowania sieci sensorowych należy zaliczyć: zakres częstotliwości; szybkość transmisji oraz jej zasięg, czułość i moc wyjściowa sygnału, napięcie zasilania oraz moc zasilania w trybach odbioru, transmisji i uśpienia. Etap wyboru bazy komponentów ma na celu zdefiniowanie podzbioru V zbioru U urządzeń technicznych wystarczających do zbudowania projektowanej sieci sensorowej. Teoretycznie najlepiej jest minimalizować liczebność podzbioru V. W praktyce – niewielki podzbiór V minimalizuje koszty utrzymania magazynu części zamiennych. Niech $U = \{u_1, \cdots, u_k\}$ – k elementowy zbiór dostępnych komponentów systemu. Wtedy funkcję definiującą bazę komponentów można zapisać za pomocą wyrażenia: $V = f(W, U)$, gdzie: gdzie: $V \subset U$ – podzbiór dostępnych komponentów tworzących SMS, wybranych przez system wspomagania projektowania lub przez projektanta na bazie analizy wektora W wymagań formalnych; $f(x)$ – funkcja wyboru podzbioru elementów.

Kolejnym etapem procedury projektowania jest określenie bazowej struktury sieci monitoringu. Sieć bazowa zawiera minimalny zestaw elementów niezbędnych do realizacji zadań stojących przed SMS. Podstawowym wymaganiem stawianym przed taką strukturą jest zapewnienie spójności projektowanej sieci. Na tym etapie projektowania SMS rozwiązuje się dwa podzadania: **a.** Segmentację sieci czujników; **b.** Budowę powiązań pomiędzy segmentami SMS a jej węzłem centralnym. Funkcję tworzenia bazowej struktury można opisać wzorem: $(D, E) = f(W, V)$, gdzie: D – wektor połączeń pomiędzy węzłami sieci; E – zbiór węzłów pośrednich (tranzytowych) włączonych do sieci w celu zapewnienia powiązania pomiędzy klastrami; $f(x)$ – funkcja tworzenia bazowej struktury sieci.

W kolejnym kroku procedury projektowej podejmowana jest próba zapewniania niezawodności sieci. Jedną z głównych właściwości SMS jest poziom niezawodności sieci. W tym kontekście terminy niezawodność i odporność na uszkodzenia są traktowane jako synonimy. Zgodnie z klasyczną definicją niezawodność określa prawdopodobieństwo awarii systemu pod wpływem czynników zewnętrznych. Parametrem wywodzącym się z teorii grafów, doskonale ilustrującym szeroko pojętą niezawodność jest spójność grafu. Grafem $G(V, E)$ nazywamy obiekt matematyczny złożony ze n elementowego zbioru V wierzchołków $V = \{v_1, \cdots, v_n\}$ oraz zbioru E krawędzi $E = V \times V$. W istocie graf to relacja binarna $E = V \times V$ na zbiorze wierzchołków V. Pojęciem pomocniczym wykorzystywanym przy definiowaniu spójności jest *ścieżka*, w celu określenia której zdefiniujemy pojęcie *trasy*. *Trasą* w grafie G nazywamy skończony ciąg krawędzi postaci: $(v_1, v_2), (v_2, v_3), \ldots, (v_{n-1}, v_n)$, w którym dwie sąsiednie krawędzie są sąsiednie albo identyczne. Wierzchołek v_0 nazywamy wierzchołkiem początkowym trasy, v_n – końcowym. Powyższy zapis pokazuje trasę od wierzchołka v_1 do v_n. *Ścieżką* w grafie G nazywamy trasę, w której wszystkie krawędzie są różne. Jeżeli w ścieżce różne są również wszystkie jej wierzchołki (wyłączając sytuację, kiedy $v_1 = v_n$) to ścieżka taka nazywana jest drogą. Droga lub ścieżka są zamknięte, jeżeli $v_1 = v_n$. *Cyklem* nazywamy ścieżkę zamkniętą zawierającą co najmniej jedną krawędź, dlatego też każda pętla lub para krawędzi wielokrotnych jest cyklem. Graf G nazywamy spójnym, jeżeli pomiędzy dowolną parą jego wierzchołków istnieje przynajmniej jedna ścieżka.

Spójność grafów można rozpatrywać również z punktu widzenia jego części składowych. Rozważmy dwa grafy: $G = \big(V(G), E(G)\big)$ oraz $H = \big(V(H), E(H)\big)$, których zbiory wierzchołków $V(G)$ i $V(H)$ są rozłączne. Sumą grafów $G \cup H$ będzie graf, którego zbiorem wierzchołków jest zbiór $V(G) \cup V(H)$, a zbiorem krawędzi $E(G) \cup E(H)$. Graf, który można przedstawić za pomocą sumy (w tym przypadku) dwóch grafów nazywamy grafem niespójnym, w przeciwnym razie graf nazywamy grafem spójnym. Naturalnie graf niespójny możemy przedstawić za pomocą sumy grafów spójnych zwanych składowymi grafu.

Podstawową metodą zmiany niezawodności sieci jest przebudowa jej struktury topologicznej. Jest oczywiste, że awarie wynikają z naruszenia ścieżek przesyłania

informacji od nadawcy do odbiorcy. Takie zakłócenia są możliwe, na przykład, w przypadku awarii jednego z węzłów tranzytowych. W związku z tym zwiększenie liczby możliwych tras transmisji informacji liniowo zwiększa niezawodność systemu jako całości. W ten sposób funkcję niezawodności można przedstawić za pomocą następujących wyrażenia $(D^*, E^*) = f(D, E, V)$, gdzie: D^* – wektor połączeń pomiędzy węzłami uzupełniony nowymi ścieżkami,E^* – zbiór węzłów tranzytowych uwzględniające dodane na danym etapie projektowania; $f(x)$ – funkcja dodawania nowych ścieżek. Wynik operacji η_r sprawdzenia niezawodności sieci można zapisać jako: $\eta_r = g(D^*, E^*, W)$, $\eta_r \in \{0,1\}$ – wynik sprawdzenia niezawodności sieci, $g(x)$ – funkcja sprawdzenia niezawodności.

Kluczowym krokiem dla sieci zasilanych z wewnętrznych źródeł energii jest zapewnienie efektywności energetycznej. Pozwala ona na maksymalizację czasu funkcjonowania sieci. Aby rozwiązać ten problem, niezbędne jest prawidłowe rozmieszczenie węzłów tranzytowych, co pozwoli równomiernie rozłożyć obciążenie w całej sieci, a tym samym zapewnić minimalną zmianę wskaźników średniego zużycia energii urządzeń. Drugim sposobem osiągnięcia optymalnego zużycia energii jest ustawienie właściwego trybu dostępu do zasobów SMS. Każdy z węzłów funkcjonalnych i tranzytowych może znajdować się w jednym z czterech trybów: transmisja, odbiór, obliczenia i uśpienie. Minimalizacja czasów pracy węzłów w trybach z największym zużyciem energii jest głównym sposobem maksymalizacji czasu funkcjonowania sieci. Funkcję zapewniania efektywności energetycznej można zapisać jako: $(D^{\blacksquare}, E^{\blacksquare}) = f(D^*, E^*, V)$, gdzie: $D^{\blacksquare}$ – wektor połączeń międzywęzłowych, uzupełniony o nowe ścieżki; $E^{\blacksquare}$ – zbiór węzłów pośrednich z uwzględnieniem zamienionych lub dodanych na danym etapie; $f(x)$ – funkcja realokacji węzłów tranzytowych w celu zapewnienia efektywności energetycznej. Wynik operacji η_e sprawdzenia efektywności energetycznej można przedstawić w postaci: $\eta_e = g(D^{\blacksquare}, E^{\blacksquare}, V)$, $\eta_e \in \{0,1\}$ – wynik sprawdzenia efektywności energetycznej sieci, $g(x)$ – funkcja sprawdzenia efektywności energetycznej.

Przedostatnim krokiem procedury projektowania połączeń są badania symulacyjne. W jego trakcie przeprowadza się modelowanie zaprojektowanej SMS. W procesie modelowania szacuje się podstawowe cechy systemu, będące podstawą do przeprowadzenia wielokryterialnej analizy SMS. Jeśli określone metodami matematycznymi parametry spełniają wymagania zdefiniowane na etapie wprowadzania i analizy założeń początkowych, inicjowane jest przejście do etapu generowania projektu końcowego. W przeciwnym przypadku do projektu zostaną wprowadzone udoskonalenia, w tym celu poszczególne kroki procedury projektowania zostaną wykonane ponownie: $C = f(D^{\blacksquare}, E^{\blacksquare})$, gdzie: C – wektor parametrów wyjściowych zaprojektowanej sieci; $f(x)$ – funkcja określania parametrów wyjściowych. Następnie otrzymujemy wynik analizy wielokryterialnej względem całego wektora W: $\eta_c = g(C, W)$, $\eta_c \in \{0,1\}$, $g(x)$ – funkcja analizy wielokryterialnej.

Procedurę projektowania zamyka tworzenie projektu końcowego, polegające na ostatecznym skonfigurowaniu modelu SMS.

5. Optymalizacja instrumentarium systemu monitoringu

W celu prezentacji rozwiązania zadania optymalizacji instrumentarium rozważmy system monitoringu środowiska (SMS). Kluczową właściwością systemu jest ciągłość jego pracy. Powinien on bezprzerwowo, po akceptowalnych kosztach dostarczać wiarygodnej informacji o stanie środowiska. Dlatego system monitoringu powinien zostać oparty o zunifikowany zestaw wielofunkcyjnych, bezobsługowych sensorów o wysokiej niezawodności, a sieć połączeniowa powinna być nadmiarowana.

Powyższe uwarunkowania zdecydowały o opracowaniu własnych metod doboru instrumentarium SMS oraz projektowania sieci połączeniowej. W celu projektowania instrumentarium zaproponowano podejście dwustopniowe: w pierwszym kroku dokonywany jest wybór asortymentu stosowanych sensorów, w drugim zaś określane są kryteria, na podstawie których przeprowadzona zostanie optymalizacja zestawu środków pomiarowych. Wstępny dobór komponentów, to typowe podzadanie projektowania systemu złożonego, polegające na określeniu zestawu konkretnych urządzeń oraz porównaniu go z rozwiązaniami alternatywnymi. Etap porównania bazuje najczęściej na klasycznych metodach kwalimetrycznych, systemach ekspertowych, a obecnie coraz częściej na sztucznej inteligencji. Wyniki kroku wykorzystywane są dalej podczas optymalizacji zestawu urządzeń, której zadaniem jest poprawa parametrów eksploatacyjnych SMS. Z punktu widzenia efektywności eksploatacji celowym jest uwzględnienie charakterystyk opisujących ilość pomiarów wykonywanych przez konkretny zestaw sensorów w jednostce czasu. Uwzględniając periodyczność zachodzących zjawisk środowiskowych wspomniana ilość powinna dotyczyć stosunkowo długiego czasu, minimum kwartału, najlepiej roku. W szczególności, można zapewniać maksymalne obciążenia sensora lub ich zestawów w trakcie wykonania zadanego programu monitoringu, bądź dążyć do minimalizacji kosztów jednostkowych pomiaru.

Danymi źródłowymi kroku doboru i optymalizacji są lista a_i określająca mierzone parametry środowiska $(i = 1, \cdots, n)$ oraz lista b_j typów sensorów wykorzystywanych do pomiaru parametrów określonych przez listę a_i $(j = 1, \cdots, m)$ mierzonych parametrów. Ponieważ wykorzystywane sensory są z założenia wielofunkcyjne, mogą mierzyć wiele różnych parametrów i $n \gg m$. Uwzględniając warunki eksploatacji sprzętu pomiarowego różne dla różnych pór roku przyjmijmy, że okresem rozliczeniowym będzie rok. Niech τ_{ij} oznacza czas niezbędny sensorowi j-tego typu do wykonania pomiaru i-tego parametru, $\tau_{ij} \in \mathbb{Q}$, a R_i to liczba pomiarów i-tego parametru w cyklu rozliczeniowym, $R_i \in \mathbb{N}$. Wtedy jako T_j oznaczymy sumaryczny czas pracy j-tego typu sensora w okresie rozliczeniowym, który jest równy: $T_j = \sum_{i=1}^{n} R_i \tau_{ij}$. Z kolei sumaryczny czas T pracy zestawu typów sensorów w ciągu roku jest równy: $T = \sum_{i=1}^{n} \sum_{j=1}^{m} R_i \tau_{ij}$, gdzie: $T_j \leq T$. W dalszych rozważaniach kluczowe znaczenie odgrywa czas τ_{ij}. Jego wartość może być otrzymana

metodami doświadczalnymi, można również oprzeć się na ocenie ekspertowej bazującej na teorii zbiorów rozmytych.

Rozważmy pierwsze z kryteriów optymalizacji systemu, którym jest *maksymalizacja obciążenia* sensorów pomiarowych. Obciążenie j-tego sensora uzyskaniem wartości i-tego parametru ekosystemu określać będziemy za pomocą czasu T_{ij} niezbędnego do wykonania koniecznych pomiarów. Czas T_{ij} jest częścią ogólnego czasu T_i. W procedurze optymalizacji rozwiązania przyjęto, że czas wykonania pomiaru i-tego parametru przez j-ty sensor jest równy liczbie pomiarów w cyklu rozliczeniowym, tj. $T_{ij} = R_i$. Wtedy funkcję Z celu możemy zapisać jako:

$$Z = \sum_{i=1}^{n} \sum_{j=1}^{m} T_{ij} / \tau_{ij} \rightarrow \max , \tag{1}$$

z uwzględnieniem następujących ograniczeń: $\sum_i^n T_{ij} \leq T_j \;\; , \;\; \forall i = 1, \cdots, n \;\; ;$ $\sum_j^m \left(T_{ij} / \tau_{ij} \right) \leq R_i, \forall j = 1, \cdots, m$. Formuła optymalizacyjna (1) gwarantuje maksymalizację ilości pomiarów wykonywanych przez konkretny sensor w okresie rozliczeniowym.

Zgodnie z drugim podejściem poszukiwać będziemy architektury SMS zapewniającej *minimalizację kosztów pomiarów*. Jest to kryterium ekonomiczno-techniczne – pozwala uzyskać minimalny koszt pomiaru w cyklu rozliczeniowym (kwartalnym, rocznym itd.). Podstawowymi nośnikami kosztów są: roczny C_j koszt odtworzenia j-tego sensora pomiarowego wraz z przypadającą na niego częścią kosztów odtworzenia autonomicznego zasilania oraz urządzeń komunikacyjnych; roczny C_{wz} koszt odtworzenia węzła zarządzającego; roczny C_p koszt dzierżawy powierzchni niezbędnych na lokalizację sensorów i węzła zarządzającego; roczne C_e koszty eksploatacji systemu włączające m. in. opłaty dzierżawne za pasma komunikacyjne oraz koszty zasilania węzła zarządzającego; koszty C_0 osobowe odzwierciedlające wynagrodzenia pracowników.

Jednostkowe koszty pomiaru możemy określać w dwojaki sposób. Po pierwsze, można rozważać średni koszt C_{avg} pomiaru, który określamy na podstawie wyrażenia:

$$C_{avg} = \frac{\sum_{j=1}^{m} C_j + C_{wz} + C_p + C_e + C_o}{\sum_{i=1}^{n} R_i} . \tag{2}$$

Jeżeli przybliżenie kosztów oferowane przez wyrażenie (2) jest niewystarczające, w procesie optymalizacji można zastosować koszt C_{ij} pomiaru i-tego parametru przez j-te urządzenie pomiarowe. Koszt ten może zostać określony na podstawie wyrażenia: $C_{ij} = \left(C_j + C_{wz}^j + C_p^j + C_e^j + C_o^j \right) / R_{ij}$, gdzie: C_{wz}^j , C_p^j , C_e^j , C_o^j – koszty określone wcześniej dla wyrażenia (2) przeliczone dla j-tego sensora pomiarowego; R_{ij} – liczba pomiarów i-tego parametru, przez j-ty sensor. Funkcja celu Z dla drugiego rozważanego podejścia ma postać:

$$Z = \sum_{i=1}^{n} \sum_{j=1}^{m} C_{ij} \rightarrow \min, \tag{3}$$

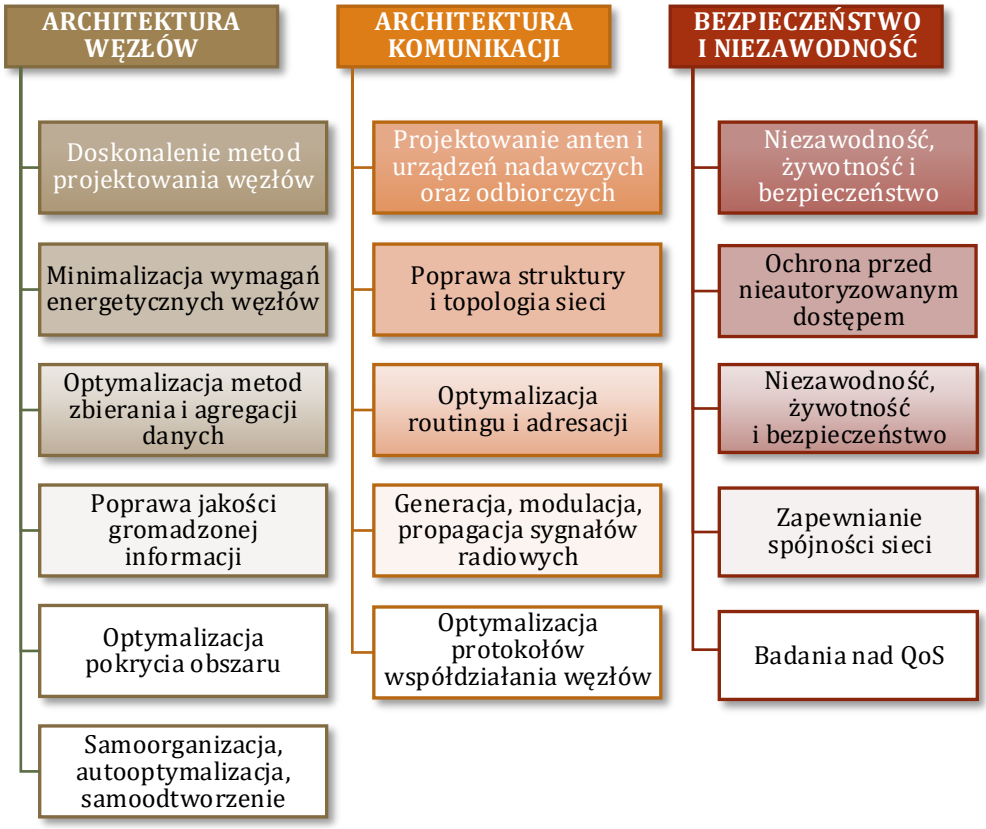
z uwzględnieniem następujących ograniczeń: $\sum_i^n R_{ij}\, \tau_{ij}, \forall i = 1, \cdots, n; \sum_j^m n_{ij} \leq R_i$, $\forall j = 1, \cdots, m$.

6. Problemy dla przyszłych badań

Pomimo tego, iż sieci sensorowe rozwijane są już blisko od dwóch dekad, stan wiedzy o ich projektowaniu jest nadal niezadowalający. Tworząc sieci tej klasy można bazować na bogatym doświadczeniu projektowania sieci teleinformatycznych. Chociaż dla sieci tych, metody formalne pozwalające jednocześnie analizować wszystkie aspekty ich funkcjonowania nie istnieją, to algorytmy cząstkowe, pozwalające rozwiązywać zadanie projektowe dla wybranego obszaru funkcjonalnego pozwalają uzyskiwać zadowalające efekty. Jednak ich wykorzystanie do projektowania sieci sensorowych jest bardzo ograniczone. Sieci sensorowe to systemami złożone, na których funkcjonowanie przejawia wpływ wiele czynników, nieobecnych lub pomijanych w sieciach teleinformatycznych. Przykładowo, w sieciach sensorowych zużycie energii przez węzły jest krytyczne i należy je minimalizować. W sieciach teleinformatycznych, energia zużywana przez węzły komunikacyjne jest znikomą częścią mocy wymaganej do zasilania węzłów obliczeniowych, w szczególności realizujących zadania wyszukiwania informacji. Dlatego, niezbędne jest opracowanie i weryfikacja nowych metod analizy i syntezy sieci tej klasy.

Na rys. 6 sklasyfikowano najważniejsze problemy badawcze pojawiające się w procesie projektowania sieci sensorowych. W dalszej części rozdziału skoncentrujemy się na kilku wybranych.

Istotne obawy rodzą również ograniczenia bezpieczeństwa w przetwarzaniu danych osobowych wynikające z rozporządzenia Rady Europy RODO. Niestety, w większości przypadków interpretacja obowiązujących przepisów jest asekurancka i nadmiarowa i znacznie przekracza niezbędny zakres. W eksploatowanym SMS mieszkańcy, którzy dokonają osobistej rejestracji będą informowani o potencjalnych zagrożeniach bezpieczeństwa. Dlatego obecnie opracowywany jest oparty na *blockchain* system ich uwierzytelniania. Inne prace, których zaawansowanie pozwala oczekiwać uzyskania interesujących wyników to m. in. wykorzystanie *blochchain*: do prognozowania zagrożeń; poprawy bezpieczeństwa w sieci bezprzewodowej; organizacji ruchu w sieci, zapewnianiu żywotności systemu monitoringu.



Rys. 6. Aktualne obszary badawcze dla sieci sensorowych

Bibliografia

[1] Jan Such and Małgorzata Szcześniak, *Filozofia nauki*. Poznań: Wydawnictwo: Naukowe UAM, 2007.

[2] G. B. Wiersma, Ed., *Environmental monitoring*. Boca Raton, USA: CRC Press, 2004.

[3] P. Patnaik, *Handbook of environmental analysis : chemical pollutants in air, water, soil, and solid wastes*, 2nd ed. Boca Raton, USA: CRC Press, 2010.

[4] L. Davis, *Natural Disasters*, II ed. New York, USA: Facts On File, 2009.

[5] A. M. Gunn, *Encyclopedia of Disasters: Environmental catastrophes and human tragedies*. Westport, USA: GREENWOOD PRESS, 2008.

ROZDZIAŁ 5

Efektywna lokalizacja węzłów pomiarowych i agregujących w sieciach monitoringu

Mirosław HAJDER, Janusz KOLBUSZ

Wyższa Szkoła Informatyki i Zarządzania z siedzibą w Rzeszowie,
Miroslaw.Hajder@gmail.com, JKolbusz@wsiz.edu.pl

Streszczenie

Prezentowany rozdział przedstawia metody i środki syntezy systemów monitoringu środowiskowego z punktu widzenia lokalizacji węzłów pomiarowych i agregujących oraz określania sposobu ich połączenia. Zgodnie z obowiązującymi dobrymi praktykami zakłada się, że system monitoringu będzie miał wielowarstwową strukturę hierarchiczną, najniższym poziomem której są sensory pomiarowe, najwyższym zaś centralny węzeł sieci. Do rozmieszczania komponentów sieci oraz ich połączenia zaproponowano metody optymalizacyjne i heurystyczne. Rozdział adresowany jest do osób zajmujących się zarządzaniem kryzysowym oraz monitoringiem środowiska.

1. Wprowadzenie

Systemy monitorujące stan środowiska (SM) są obecnie stosowane szerzej niż kiedykolwiek. O ile wcześniej stanowiły one niezależne funkcjonalnie systemy informacyjno-pomiarowe to obecnie stanowią one element inteligentnych miast – *smart cities* (SC). Choć zakres zadań realizowanych przez infrastrukturę SC jest bardzo obszerny, te związane z monitoringiem stanu środowiska należą do najistotniejszych.

W procesie projektowania, budowy i eksploatacji SC, systemy informatyczne i teleinformatyczne odgrywają szczególną rolę. W zasadzie, od efektywności oferowanych przez nie narzędzi informacyjno-analitycznych zależy skuteczność monitorowania. W systemach SC informacje dostarczane przez SM wykorzystywane są do efektywnego zarządzania przestrzenią miejską. Aby zarządzanie było skuteczne, informacje o stanie środowiska muszą być wiarygodne i pojawiać się w możliwie najkrótszym czasie. Aby wymagania stawiane przed procesem decyzyjnym SC były spełnione, konieczna jest budowa niezawodnej infrastruktury pomiarowo-informacyjno-komunikacyjnej gwarantującej dokonanie w czasie rzeczywistym obiek-

tywnej oceny stanu środowiska, wykorzystywanej przy podejmowaniu decyzji zarządczych. W praktyce projektowania, budowy i eksploatacji systemów monitoringu środowiska ścierają się dwie alternatywne koncepcje. Pierwsza z nich, zakłada niezmienność podlegających monitorowaniu warunków środowiskowych, których wpływ należy uwzględnić w procedurze projektowania sieci pomiarowej. W tym przypadku, najważniejszym kryterium rozmieszczenia węzłów jest lokalizacja obszarów, na których mogą pojawiać się zanieczyszczenia. Rozwiązania zadania projektowego poszukuje się wtedy w obszarze z maksymalną koncentracją czynników szkodliwych. Z punktu widzenia projektowania i budowy obsługującego SM podsystemu teleinformatycznego jest trudniejsze do rozwiązania, co jest konsekwencją terytorialnej niejednorodności rozmieszczenia podlegających obsłudze węzłów pomiarowych. Konieczną może okazać się budowa systemu heterogenicznego komunikacyjnie. Pomimo tych utrudnień systemy oparte na pierwszej koncepcji są stosunkowo tanie [1], [2], [3], [4].

Druga strategia zakłada równomierne rozmieszczenie węzłów pomiarowych na całym monitorowanym obszarze. Dzięki temu śledzi się nie tylko potencjalne źródła zanieczyszczeń, ale również trasy przewozu substancji szkodliwych czy skażenia będące skutkiem działań przestępczych. Z punktu widzenia projektowania systemów informatycznych zadanie to jest prostsze od poprzedniego – budowana sieć jest homogeniczna i ma postać kraty.

2. Uogólnione zadanie syntezy optymalnej

Aby określić dane wejściowe dla zadania syntezy optymalnej sieci połączeniowej przyjmiemy, że jej podstawowym przeznaczeniem jest przekazanie informacji do adresatów przy zachowaniu ustalonych wcześniej poziomów jakości wykonywanych usług komunikacyjnych. Wtedy danymi wejściowymi są:

1. Charakterystyki strumieni informacyjnych, przekazywanych pomiędzy abonentami systemu, uwzględniające zarówno strukturę ruchu, jak również jego rozmiar, dla każdego z kierunków komunikacji;
2. Dane lokalizacyjne podmiotów tworzących projektowany system;
3. Wymagane wartości wskaźników jakości opisujący przekazywanie informacji pomiędzy podmiotami;
4. Charakterystyki ilościowe i jakościowe sprzętu oraz oprogramowania niezbędnego do budowy sieci połączeniowej.

Ograniczenia dotyczące struktury topologicznej projektowanego systemu wynikające z wykorzystywanych algorytmów, klasy zastosowanego sprzętu, istniejącej infrastruktury technicznej oraz możliwości finansowych inwestora.

Dalej, będziemy rozważać system komunikacyjny N przedstawiony za pomocą grafu $G^N = (V^N, E^N)$, gdzie: V^N i E^N to odpowiednio zbiór jego wierzchołków i krawędzi. Ponieważ usługi wykonywane w systemie lub sieci komputerowej cha-

rakteryzuje różna wrażliwość na parametry transmisji, dane przekazywane podczas ich realizacji dzielimy na typy $k \in K^N$, gdzie K^N – zbiór typów danych przesyłanych w konkretnym systemie lub sieci. Do każdej z grup przypisujemy rodzaje przesyłanej informacji charakteryzujące się zbliżonymi wymaganiami jakości transferu. Ponadto, transmitowaną informację podzielimy z punktu widzenia terminowości na typy $q \in Q^N$, gdzie: Q^N– zbiór danych opisywanych zbliżonymi probabilistyczno-czasowymi parametrami przesyłania.

Dodatkowo, w projektowaniu informację klasyfikujemy na kategorie $l \in L^N$ z punktu widzenia jej długości, gdzie: L^N – zbiór grup długości, najczęściej wyrażonej liczbą znaków lub ich niezmiennych kombinacji tworzących transmitowaną informację. W ogólnym przypadku, każda porcja informacji przesyłanej w momencie t pomiędzy węzłami a_i oraz a_j, dla momentu t powinna zostać opisana sześcioelementowym zbiorem postaci (a_i, a_j, k, q, l, t), gdzie: $a_i, a_j \in V^N$, $k \in K^N$, $q \in Q^N$, $l \in L^N$, $t \in T^N$, przy czym, T^N to przedział czasowy, w którym badane jest zachowanie systemu.

Dla zapewnienia wysokiej skuteczności projektowania przeanalizowane muszą zostać również parametry strumieni informacyjnych przekazywanych pomiędzy parami węzłów. W szczególności, należy określić funkcje rozkładu odstępów czasowych pomiędzy czasami pojawienia się żądań komunikacyjnych, rozmiaru przekazywanych danych, priorytetów, prawdopodobieństwa pojawienia się informacji, której przekazanie będzie opisane konkretnych priorytetem. Chociaż poniższe założenie nie w każdym przypadku jest słuszne, najczęściej zakłada się, że rozkład odstępów czasowych pojawiania się kolejnych porcji informacji podlegających przesłaniu pomiędzy wybraną parą węzłów i oraz j sieci komunikacyjnej ma charakter wykładniczy z parametrem $\lambda_{ij}(q)$, gdzie: $q_{ij} \in Q^N$, $i, j \in V^N$. Podobnie zachowuje się rozkład rozmiaru informacji podlegającej przesłaniu, dla której parametrem $\mu_{ij}(q)$. Sumaryczna intensywności Λ_{ij} pojawiania się informacji przekazywanej pomiędzy węzłami i oraz j dla każdej klasy ważności q określa się wyrażeniem $\Lambda_{ij} = \sum_{q \in Q^N} \lambda_{ij}(q)$. Podobnie można określić sumaryczną długość M transmitowanej informacji dla różnych priorytetów: $M_{ij} = \sum_{q \in Q^N} \mu_{ij}(q)$. Zbiór parametrów Λ_{ij} określany będzie za pomocą macierzy ciążenia oznaczanej dalej jako $\|\Lambda\|$. Zbiór obiektów tworzących projektowaną sieć oznaczymy jako A, a ich względne rozmieszczenie opisuje macierzy odległości $\|L\|$ złożonej z elementów l_{ij} uwzględniających odległości pomiędzy podmiotami $a_i \in$ A oraz $a_j \in A$.

Zauważmy, że w ogólnym przypadku odległości l_{ij} nie odzwierciedlają fizycznych odległości pomiędzy węzła i oraz j. Oznacza to, że dla elementów macierzy $\|L\|$ nierówność trójkątów może być niespełniona. Najprostszym, szacunkowym sposobem określenia odległości l_{ij} dla rzeczywistych sieci jest wykorzystanie współrzędnych geograficznych węzłów i wzorów na długość ortodromy. Ponieważ sieć połączeniowa może być budowana nie tylko na bazie kanałów dedykowanych, ale również dzierżawionych założenie, że koszt budowy kanału jest liniową funkcją

jego długości podstawą w wielu przypadkach może nie być słuszne. Dlatego, akceptowalne jest zastosowanie innych kryteriów *bliskości* węzłów sieci.

Jeżeli syntezowana jest sieć połączeniowa, w której zdefiniowane są wymagania dotyczące jakości operacji komunikacyjnych, to są one definiowane w postaci zbioru charakterystyk probabilistyczno-czasowych, do których w pierwszej kolejności zaliczamy: średni czas $t_{\mathrm{avg}}(q)$ dostawy pakietu do adresata z rozbiciem względem priorytetów, dopuszczalny czas $t_{\max(q)}$ dostawy pakietu z rozbiciem na priorytety, prawdopodobieństwo $\Pr\{t > \mathrm{t}_{\max(q)}\}$ przekroczenia podczas przesyłania wartości maksymalnej, prawdopodobieństwo $P(q)$ poprawnego odebrania pakietu z rozbiciem na priorytety oraz prawdopodobieństwo $P_{utr}(q)$ utraty pakietu podczas transmisji, również z uwzględnieniem priorytetów. Spełnienie wymagań stawianych przed $T(q)$, $\Pr\{t > T(q)\}$, $P(q)$ dla niezmiennej topologii i pełnej bezawaryjności jej elementów wykonuje się poprzez wybór odpowiedniego sprzętu lokalizowanego w węzłach komutacji oraz kanałów komunikacyjnych. Do tego celu można zastosować dodatkowo algorytmiczne metody poprawy jakości komunikacji, jej kompresji, wykrywana i poprawy błędów transmisji.

Prawdopodobieństwo utraty informacji w procesie transmisji jest określone niezawodnością programowo-sprzętowych komponentów węzłów komutacji i kanałów przesyłania. W sieciach o strukturze hierarchicznej, dodatkowo, dostępne są mechanizmy nadmiarowania komponentów systemu. W sieciach z rozproszoną topologią, w których zazwyczaj istnieje wiele dróg łączących pary węzłów, spełnienie wymagań dotyczących $P_{utr}(q)$ można zapewnić wykorzystaniem dynamicznych metod zarządzania rozkładem strumieni informacyjnych w sieci połączeniowej.

Ponieważ parametry komunikacji, w głównej mierze, określane są poprzez charakterystyki wykorzystywanych urządzeń programowo-sprzętowych ich parametry można traktować jako zbiór ograniczeń, które należy uwzględnić podczas projektowania systemu komunikacji. Ponieważ podczas syntezy optymalnych sieci połączeniowych uwzględnienie różnorodności metod sterowania i ich konkretnych realizacji jest trudne na wstępnych etapach projektowania zakłada się statyczność zarządzania strumieniami informacyjnymi. Uproszczenie takie, dopuszczalne na etapie projektowania topologicznego, prowadzi do istotnego zmniejszenia złożoności zadania syntezy, pozwalając przy tym utrzymać wartości charakterystyk efektywności na akceptowalnym poziomie.

Kolejnym, często stosowanym uproszczeniem, jest pomijanie potencjalnych awarii i błędów elementów programowo-sprzętowych węzłów komutacji i kanałów komunikacyjnych. W rezultacie, określenie wskaźników niezawodności strukturalnej systemu łączności jest niemożliwe. Wieloletnie doświadczenie eksploatacji pokazuje, że niezawodność urządzeń węzłowych i kanałów komunikacyjnych jest na tyle wysokie, że jeżeli pomiędzy dowolną parą abonentów można zestawić dwie lub więcej alternatywnych tras, nawet rygorystyczne wymagania niezawodnościowe będą spełnione. Z tego powodu, w systemach płaskich w miejsce określania

akceptowalnego poziomu parametrów niezawodnościowych stawia się wymaganie dotyczące minimalnej liczby v^A rozłącznych ścieżek łączących dowolną parę węzłów. W systemach hierarchicznych problem niezawodności rozwiązuje się poprzez ograniczenie liczby pośrednich ścieżek przesyłania informacji pomiędzy dowolna parą węzłów wchodzących w jego skład. Zbyt duża liczba poziomów systemu hierarchicznego dodatkowo zwiększa opóźnienia komunikacyjne, co może być problem w przypadku zastosowania sieci rozgłoszeniowych probabilistycznych metod dostępu. Dlatego, liczba poziomów hierarchii zazwyczaj ogranicza się do 3-4.

Czynnikiem komplikującym projektowanie sieci połączeniowych jest również konieczność uwzględniania priorytetów dostawy informacji. W takim przypadku, wykorzystywane modele są rozbudowane, a ich wykorzystanie na etapie syntezy trudne. Powyższy problem rozwiązuje się poprzez wykorzystanie jednego uśrednionego priorytetu. Działanie takie nie pogarsza w zauważalny sposób dokładności procesu modelowania.

Jedną z najważniejszych decyzji podejmowanych na etapie projektowania topologii jest wybór kryterium, na którego podstawie oceniana będzie efektywność wariantów rozwiązania zadania projektowego. Najczęściej, kryterium tym są wskaźniki związane z oceną nakładów niezbędnych do realizacji funkcji celu. Z uwagi na wysoki koszt wykonania instalacji teletechnicznych oraz ograniczone budżety inwestycji pominięcie przy projektowaniu sieci połączeń kryteriów ekonomiczno-technicznych jest niedopuszczalne.

Do dalszych rozważań wprowadźmy następujące oznaczenia: X^N – zbiór danych źródłowych, E^N – zbiór akceptowalnych wartości kryteriów oceny efektywności wariantów projektowanej struktury; P^N – funkcja wyjściowa lub model sieci; Y^N – zbiór zmiennych wyjściowych; G^N – uogólniona funkcja oceny jakości wariantów topologii. Wtedy zadanie optymalizacji struktury sieci połączeniowej możemy zdefiniować jako poszukiwanie w podzbiorze dopuszczalnych rozwiązań $X^D \subseteq X^N$ rozwiązania $\hat{x} \in X^D$, dla którego, dla wszystkich $x \in X^D$ spełnione są dwa poniższe warunki:

$$g(\hat{x}) \geq g(x);$$
$$g(x) = G^N\left(x, P^N(x)\right). \tag{1}$$

i ograniczenia. Zbiór zmiennych wejściowych Y^N określa się jako kombinację wskaźników jakości procesów przesyłania oakietów w sieci połączeniowej. Projektowanie realizowane jest poprzez zmianę znaczenia parametrów opisujących strukturę projektowanej sieci. Mogą to być m. in.: organizacja sieci (ilość węzłów i ich rozmieszczenie); długość, organizacja oraz trasy przebiegu linii komunikacyjnych, konfiguracja wyposażenia węzłów sieciowych. Zmiany wprowadzane są tak, aby przy zadanych parametrach strumieni przesyłanych po sieci, trybach przełączania, ograniczeniach przepustowości linii komunikacyjnych, wnoszonych przez

nie opóźnieniach, a także zadanych algorytmach trasowania i zarządzania spełnione zostały wymagania dotyczące jakości projektowanej sieci połączeniowej w odniesieniu do jakości transmisji.

Element $\hat{x} \in X^D$, określany przez trójkę (P^N, G^M, X^N), spełniający warunek (1), nazywany jest rozwiązaniem zadania syntezy. Złożoność poszukiwania optymalnej kombinacji (P^N, G^M, X^N)implikuje trudność rozwiązania całego zadania i wynika z konieczności uwzględnienia znacznej liczby czynników wpływających na procesy funkcjonowania systemu połączeń, a także ze stochastycznym charakterem procesów zachodzących w sieci.

Rozważmy obecnie dane wejściowe analizowanego zadania. Przyjmijmy, że projektowana sieć połączeniowa składać się będzie z dwóch podstawowych warstw: abonenckiej i rdzenia. Oczywiście, dwuwarstwowość nie jest obligatoryjna, bowiem w skład sieci abonenckich mogą wchodzić sieci dostępu nazywane również rdzeniami niższego poziomu. Złożoność rozwiązywanego zadania utrudnia rozwiązanie go na bazie jednolitego modelu. Najczęściej do tego celu wykorzystuje się kombinację metod analitycznych i modelowania analitycznego, w których stosuje się szereg akceptowalnych uproszczeń.

Rzeczywiste strumienie uzależnione są od wielu parametrów. Jednak w praktyce, w celu zmniejszenia złożoności projektowania strumienie najczęściej opisujemy za pomocą rozkładów przedziałów czasowych pomiędzy momentami pojawiania się zagregowanych danych oraz ich długości. Najczęściej przyjmuje się, że rozkłady długości są identyczne i wprowadza się założenie o niezależności momentów pojawianie się danych oraz ich długości. Jeżeli szybkości przesyłania datagramów po kanałach transmisji są niezmienne, to czasy przesyłania zagregowanych danych powiązane są rozkładem wykładniczym.

Wprowadzenie powyższych ograniczeń pozwala, po pierwsze, wykorzystywać względnie proste modele matematyczne funkcjonowania sieci połączeń, po drugie, z wystarczającą dokładnością opisywać procesy pojawiające się na poziomie funkcjonowania topologii sieci.

Proponowana do wykorzystania metodyka składa się z następujących etapów:

1. Wybór lokalizacji regionalnych węzłów komutacji;
2. Syntezy topologii abonenckich sieci połączeniowych;
3. Syntezy struktury rdzenia sieci.

Jeżeli lokalizacja regionalnych węzłów komutacji jest wyznaczona odgórnie można określić dane wejściowe procesu projektowania. W tym celu poszczególnych abonentów przypisuje się do konkretnych węzłów. Jeżeli liczba regionalnych węzłów komutacji jest równa M, to wszyscy abonenci systemu komunikacji dzieleni są na M rozłącznych grup. Liczebność każdej z grup jest równa n_i , a $\sum_{i=1}^{M} n_i = N_A$, gdzie: $i = 1, \cdots, M$, N_A – sumaryczna liczba abonentów sieci do których zaliczmy również abonentów zlokalizowanych bezpośrednio w węzłach komutacji.

Chociaż z projektowego punktu widzenia sieć komunikacyjna o płaskiej strukturze wydaje się efektywna, to jej realizacja okazuje się bardzo kosztowna, porównywalna z budową rdzenia sieci. Dlatego, sieć dostępowa budowana jest najczęściej w postaci mniej kosztownej sieci hierarchicznej. Wymiana informacji pomiędzy abonentami oparta jest o regionalne węzły komutacji i rdzeń sieci. Rdzeń sieci najczęściej posiada płaską architekturę, a jego współczynnik spójności jest wyższy. W idealnym przypadku, dowolną parą węzłów łączą przynajmniej dwie rozłączne trasy.

W najprostszym przypadku, dla struktury dwupoziomowej, połączenie abonentów różnych sieci dostępowych bazować będzie na trzech odcinkach, każdy z których może posiadać różne parametry. Przyjmijmy, że średni czas przesłania informacji po każdej z dwóch sieci dostępowych będzie równy t_1, po rdzeniu sieci zaś t_2. Jeżeli wymiana komunikatów pomiędzy węzłami abonentów odbywa się za pomocą regionalnego węzła komutacji, to intensywność strumieni informacyjnych pomiędzy abonentami i węzłami regionalnymi znajdującymi się w danej sieci dostępowej oraz pomiędzy węzłami komutacji za pośrednictwem rdzenia sieci może być określona na podstawie poniższej metodyki.

Intensywność λ_k strumienia zagregowanych danych przesyłanych pomiędzy k-tym węzłem końcowym a regionalnym węzłem komutacji, do którego został on dołączony określa się wyrażeniem:

$$\lambda_k = \sum_{j=1, j\neq k}^{N_A} \lambda_{kj} + \sum_{j=1, j\neq k}^{N_A} \lambda_{jk},\ k \in n_1, k \neq i\ , \tag{2}$$

gdzie: $i = 1, \cdots, M$ – zbiór regionalnych węzłów komutacji. Zbiór wszystkich λ_k, gdzie: $k \in n_i$ oraz $i = 1, \cdots, M$ dla każdego i-tego węzła komutacji określa macierz $\|\Lambda_j\|$ intensywności strumieni zagregowanych danych pomiędzy węzłami końcowymi i regionalnymi węzłami komutacji dla i-tej sieci dostępowej. Z kolei intensywność strumienia zagregowanych danych pomiędzy i-tym i j-tym regionalnym węzłem komutacji w rdzeniu sieci można określić za pomocą wyrażenia:

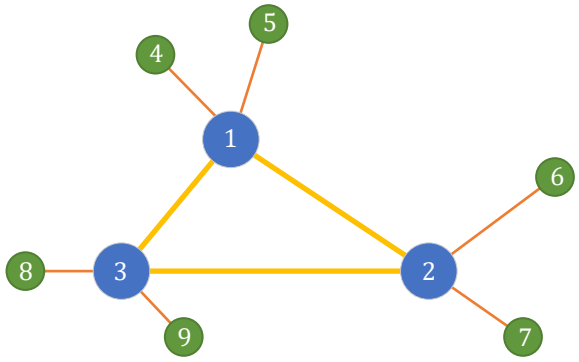
$$\lambda_{ij}^{M} = \sum_{p\in n_j} \sum_{k\in n_j} \lambda_{pk},\ \ i, j = 1, \ldots, M\,. \tag{3}$$

Zbiór elementów λ_{ij}^{M} tworzy macierz ciążenia $\|\Lambda_M\|$ odnoszącą się do rdzenia sieci. Zauważmy, że $\sum_{j=1, j\neq i}^{M} \lambda_{ij}^{M} = \sum_{j=1, j\neq k}^{M} \lambda_{ji}^{M}$.

W podobny sposób macierz odległości $\|L\|$ dla każdej i-tej sieci dostępowej $(i = 1, \cdots, M)$ przekształcana jest w dwie macierze $\|L_i\|$ oraz $\|L_M\|$. Elementy pierwszej macierzy określają wyłącznie odległości pomiędzy węzłami i-tego regionu, a $\|L_M\|$ odległości pomiędzy węzłem regionalnym a rdzeniem sieci. Rozważmy przykład sieci złożonej z 9 węzłów. Przyjmijmy, że jej macierz $\|\Lambda\|$ ciążenia posiada poniższą postać:

$$\|\Lambda\| = \begin{bmatrix} 0 & 6 & 5 & 3 & 1 & 1 & 0 & 2 & 1 \\ 5 & 0 & 4 & 0 & 2 & 3 & 1 & 0 & 0 \\ 6 & 3 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 3 & 0 & 1 & 0 & 2 & 2 & 0 & 0 & 0 \\ 1 & 2 & 1 & 2 & 0 & 1 & 1 & 1 & 0 \\ 1 & 3 & 0 & 2 & 1 & 0 & 2 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 2 & 0 & 1 & 0 & 1 & 2 & 0 & 0 & 2 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 & 2 & 0 \end{bmatrix}.$$

Do dalszej analizy przyjmijmy, że liczba M regionalnych węzłów sieci wynosi 3. Węzły te zostaną rozmieszczone węzłach 1, 2, 3 uwzględnionych w macierzy ciążenia. Zbiory $n_i, i - 1, \cdots, N$ będą zawierać elementy: $n_1 = \{4,5\}, n_2 = \{6,7\}, n_3 = \{8,9\}$. Wtedy sieć komunikacyjna, w której rdzeń ma postać sieci zupełnej, a sieci dostępu – gwiaździstej ma postać przedstawioną na rys. 1.



Rys. 1. Przykładowa sieć dwupoziomowa

Wykorzystując wyrażenia (2) oraz (3) otrzymujemy następujące postaci macierzy ciążenia:

$$\begin{aligned} \|\Lambda_1\| &= [0\ 0\ 0\ 16\ 18\ 0\ 0\ 0\ 0\ ], \\ \|\Lambda_2\| &= [0\ 0\ 0\ 0\ 0\ 20\ 12\ 0\ 0\ ], \\ \|\Lambda_3\| &= [0\ 0\ 0\ 0\ 0\ 0\ 0\ 14\ 14], \end{aligned}$$

$$\|\Lambda_n\| = \begin{bmatrix} 0 & 13 & 11 \\ 12 & 0 & 8 \\ 12 & 7 & 0 \end{bmatrix}.$$

Źródłowa macierz odległości $\|L\| = [l_{ij}], i,j = 1,\cdots,3$ może zostać przekształcona do postaci źródłowych macierzy cząstkowych $\|L_i\|$ oraz $\|L_M\|$, które posiadają następującą postać:

$$\|L_1\| = \begin{bmatrix} 0 & l_{14} & l_{15} \\ l_{41} & 0 & l_{45} \\ l_{51} & l_{54} & 0 \end{bmatrix}, \|L_2\| = \begin{bmatrix} 0 & l_{26} & l_{27} \\ l_{62} & 0 & l_{67} \\ l_{72} & l_{76} & 0 \end{bmatrix}, \|L_3\| = \begin{bmatrix} 0 & l_{38} & l_{39} \\ l_{83} & 0 & l_{89} \\ l_{93} & l_{98} & 0 \end{bmatrix}, \|L_M\| = \begin{bmatrix} 0 & l_{12} & l_{13} \\ l_{21} & 0 & l_{23} \\ l_{31} & l_{32} & 0 \end{bmatrix}.$$

Uzyskane dane źródłowe zostały rozdzielona tak, że możliwe jest niezależne rozwiązanie podzadań projektowania sieci dostępu i rdzenia sieci. Dla każdego podzadania można wprowadzić własną numerację węzłów, co pozwala znacznie uprościć poszczególne podzadania syntezy. Dzięki wprowadzonemu podziałowi sumaryczna złożoność rozwiązywanych zadań jest mniejsza niż złożoność zadania bazowego. Wspomniana dekompozycja nie sprzyja uzyskaniu optymalnego rozwiązania całości zadania. Dlatego w takim przypadku celowe jest interakcyjna procedura rozwiązania, w której możliwe jest uzyskanie rozwiązania dla drugich wariantów rozmieszczenia regionalnych węzłów komutacji i przypisania im węzłów końcowych. Podział i przypisanie można wykonywać do momentu, kiedy uzyskane zostanie zadowalające nas rozwiązanie. Wbrew pozorom, powyższa metodyka wymaga do uzyskania akceptowalnego rozwiązania znacznie mniejszych nakładów obliczeniowych niż rozwiązanie zadania bazowej postaci.

Najczęściej, budując sieć dostępową węzeł końcowy jest połączony z regionalnym węzłem komutacja za pomocą jednego kanału komunikacyjnego. Wykorzystanie takiego rozwiązania, to konsekwencja potrzeby minimalizowania kosztów budowy i późniejsze eksploatacji sieci. W związku z tym, z punktu widzenia niezawodności strukturalnej sieć dostępowa posiada wiszące wierzchołki (liście grafu-drzewa) i do sterowania strumieniami informacyjnymi można zastosować metodę statyczną, co zapewnia akceptowalną złożoność realizacji. Sieć dostępowa zrealizowana w tej technologii posiada ograniczoną niezawodność i jest zazwyczaj budowana na podstawie 2-3 poziomów. Wybór pomiędzy dwoma lub trzema poziomami nie jest kluczowy bowiem różnica w kosztach ich realizacji sięga 10-16%. Jeżeli uwzględnimy, że istotna część kosztów budowy sieci połączeniowej to budowa sieci dostępowej, wysoka efektywność techniczno-ekonomiczna tego poziomu przekłada się na bezpośrednio na parametry całej sieci połączeniowej.

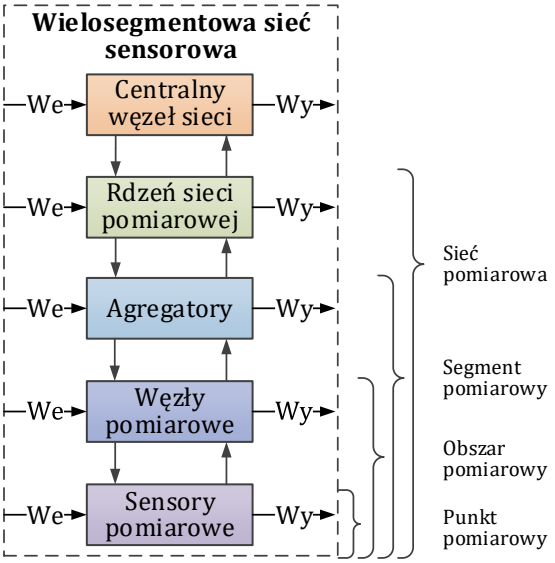
Z syntezą sieci dostępowej wiążą się dwa przeciwstawne kryteria projektowania. Z jednej strony, minimalizuję się nakłady niezbędne na budowę i eksploatację sieci. Osiąga się to głównie poprzez minimalizację połączeń nadmiarowych. Z drugiej strony, dążymy do zapewnienia sprawności systemu w przypadku uszkodzenia pewnej liczby połączeń lub węzłów. Uzyskujemy to poprzez zwiększenie wierzchołkowej lub krawędziowej spójności systemu. Tak więc, z systemu trudno jest wyeliminować pewną, chociażby minimalną, nadmiarowość. Przykładowo, celowym jest takie zaprojektowanie połączeń, by w przypadku uszkodzenia ograniczonej liczby elementów (węzłów, kanałów) zachowana została możliwość skomunikowania się, za pomocą nadmiarowych ścieżek, dowolnych regionalnych węzłów komutacji. Dla większości zastosować wystarczy, aby dowolna para węzłów była połączona za pomocą dwóch rozłącznych ścieżek. Rozłączną ścieżką komunikacyjną będziemy nazywać trasę przesyłania zagregowanej informacji pomiędzy

parą węzłów, które nie zawiera wspólnych elementów (węzłów, kanałów komunikacyjnych). Z kolei rdzeń sieci nie powinien być realizowany w postaci drzewa, tj. pozbawiony być pozbawiony wierzchołków-liści. Oczywiście, rozwiązaniem idealnym z punktu widzenia niezawodności byłby rdzeń będący grafem zupełnym. W praktyce jest to niemożliwe ze względów kosztowych. Najczęściej rdzeń zawiera 30-50% krawędzi grafu zupełnego.

3. Hierarchiczność sensorowych sieci monitoringu środowiskowego

Aby zaproponować efektywne rozwiązanie zadania topologicznego projektowania sieci sensorowych, w pierwszej kolejności rozważmy ich architekturę. Sieci te są pochodną komórkowych sieci bezprzewodowych, budowanych obecnie jako wielopoziomowe architektury hierarchiczne złożone z 3-5 poziomów komunikacyjnych [5]. Oprócz hierarchiczności, sieci te charakteryzuje heterogeniczność, w pierwszej kolejności wynikająca z różnorodności wymagań stawianych przed obsługą ruchu na każdym z poziomów hierarchii. Teoretycznie, można wyobrazić sobie płaską sieć homogeniczną, jednak rozwiązanie takie nie znajduje obecnie żadnego uzasadnienia. Heterogeniczność komunikacyjna posiada wiele zalet, m. in.: zapewnia szeroki asortyment dostępnych sposobów integracji użytkownika z siecią oraz możliwość budowy, na jej wyższych poziomach, wieloprotokołowych, multimedialnych systemów łączności. Z kolei hierarchiczność wpisuje je w główny kierunek rozwoju systemów informatycznych [6], [7], [8], [9]. Jako przyczynę szerokiego rozpowszechnienia systemów hierarchicznych najczęściej wskazuje się [10], [11], [12]: łatwiejszą analizę rozbudowanych systemów podzielonych na mniejsze części składowe; wysoką specjalizację bloków funkcjonalnych; uproszczenie eksploatacji, konserwacji i obsługi.

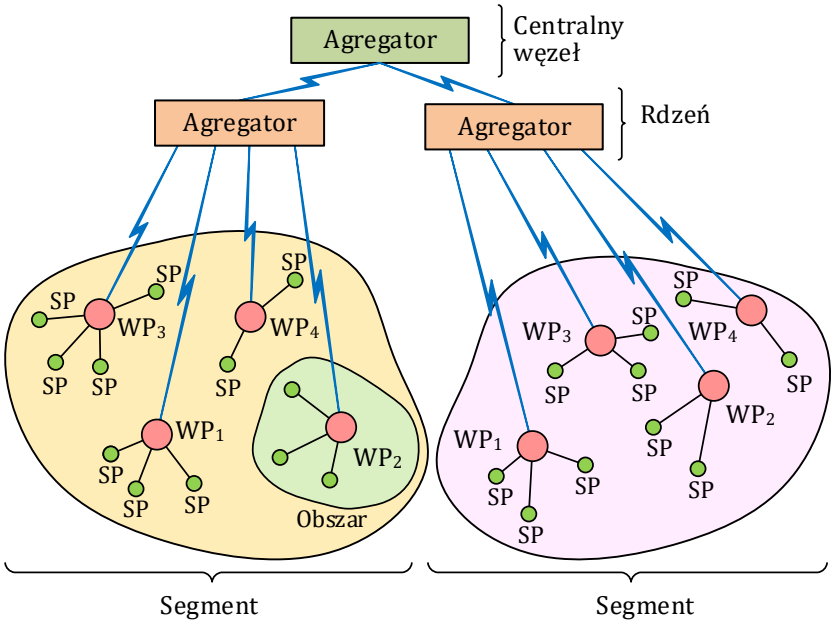
Najczęściej wymienianymi cechami systemów hierarchicznych są [11]: sekwencyjne rozmieszczanie podsystemów funkcjonalnych z zachowaniem ich podległości pionowej; nadrzędność poziomów wyższych nad niższymi; zastosowanie sprzężenia zwrotnego zapewniającego wpływ poziomów niższych na wyższe. Ogólną budowę system hierarchiczny zaprezentowano na rys. 2.



Rys. 2. Warstwy hierarchicznej sieci sensorowej

Hierarchia składa się z pięciu poziomów, z których najniższy grupuje węzły końcowe, najwyższy zaś zawiera centralny węzeł zarządzający. Elementami poszczególnych poziomów hierarchii są: węzły końcowe, komórki, obszary, rdzeń sieci oraz węzeł centralny. Węzeł końcowy, to miejsce lokalizacji użytkownika bądź ich grupy. O grupie, mówić będziemy, jeżeli tworzące ją jednostki, pomiędzy sobą, połączone są za pomocą innych środków komunikacji niż te wykorzystywane w komórce. Obszar, to grupa komórek obsługiwanych przez jeden wspólny przełącznik. Rdzeń sieci obejmuje wszystkie połączone przełączniki zintegrowane za pomocą centralnego węzła zarządzającego. Rzeczywistą realizację analizowanej hierarchii, rozważmy na przykładzie jej rozwinięcia, przedstawionego na rys. 3.

Najwyższy poziom tworzy przełącznik główny, integrujący grupę przełączników rdzenia. Urządzenia te wykorzystywane są z kolei do łączenia stacji bazowych SB, na których bazie budowane są komórki K. Grupa komórek obsługiwanych przez jeden przełącznik tworzy obszar. W każdej z komórek, rozmieszczony jest zbiór węzłów końcowych WK, z których część może zostać przemieszczona do sąsiednich komórek. To właśnie poprzez zmianę stacji bazowej obsługującej wybrane WK, wyrównywane są obciążenia wybranych komórek. Cała opisana powyżej struktura tworzy komórkową sieć bezprzewodową. Zauważmy, że nazwa (komórkowa sieć bezprzewodowa) nie do końca odzwierciedla architekturę sieci. O ile komunikacja wewnątrz komórek rzeczywiście wykorzystuje rozgłoszeniową łączność bezprzewodową, to już połączenia poza nimi opierają się łączach przewodowych lub dwupunktowych łączach radiowych. Do dalszych rozważań przyjmiemy, że wejścia każdego z przełączników obsługiwane są za pomocą przewodów optycznych, zapewniających wielokanałowość komunikacji.



Rys. 3. Architektura hierarchicznej sieci sensorowej

4. Optymalizacyjny algorytm syntezy

W procesie projektowania sieci monitoringu środowiska należy wykonać szereg powiązanych ze sobą zadań, których poprawne rozwiązanie jest warunkiem koniecznym dla zapewnienia prawidłowego funkcjonowania budowanej sieci. Jednym z podstawowych zadań tej grupy jest zadanie optymalnego rozmieszczenia koncentratorów i połączenia z nimi węzłów pomiarowych. Zadanie to można sformułować w następujący sposób: *na określonym obszarze należy rozmieścić minimalną liczbę koncentratorów sieci sensorowej zapewniająca dołączenie do sieci z określonymi parametrami minimalnymi wskazanej liczby węzłów pomiarowych*. Podobne zadanie, odniesione do bezprzewodowych sieci dostępowych było wielokrotnie analizowane w literaturze. Jednak analizowane w pracy sieci bezprzewodowe sieci sensorowe istotnie różnią się od tradycyjnych sieci bezprzewodowych. Po pierwsze, gęstość rozmieszczenia węzłów pomiarowych dołączanych następnie do koncentratorów jest znacznie mniejsza od gęstości rozmieszczenia użytkowników końcowych obsługiwanych przez stacje bazowe sieci bezprzewodowej. Po drugie, moce sygnału wykorzystywane w tradycyjnych sieciach bezprzewodowych są znacznie większe od tych, które stosujemy w sieciach sensorowych. Dlatego, powszechnie stosowane w procesie projektowania tradycyjnych sieci bezprzewodowych pomijanie strat sygnału w kanale radiowym pomiędzy koncentratorem i węzłem pomiarowym jest nieakceptowalne. Po trzecie, sieci monitoringu środowi-

skowego są zazwyczaj elementem infrastruktury krytycznej, w związku z czym, nakładane na nie wymagania niezwodnościowe są wielokrotnie wyższe niż w przypadku sieci zapewniających dostęp użytkownikom.

W niniejszej pracy rozważany jest zmodyfikowany algorytm uwzględniający wspomniane wyżej straty. W celu jego opisania rozważmy: zbiór K węzłów pomiarowych jednego typu, $K = \{1,2,\cdots,k\}$, gdzie k – moc zbioru K. Rozwiązanie zadania przedstawionego jako zadanie całkowitoliczbowego programowania dyskretnego sprowadza się do określenia wartości funkcji oraz dalszej jej minimalizacji.

$$\Phi = \sum_{k,m,n} W_{kmn} X_{mn} Y_{km} + \sum_{m,n} c_n X_{mn} + \sum_{k} \sum_{m} \frac{1}{g_{km}} Y_{km} \to \min , \tag{4}$$

gdzie:

Optymalizacja opisana wyrażeniem (4) posiada poniższe ograniczenia:

1. W każdym obszarze obsługi można zlokalizować, co najwyżej jeden koncentrator, tj.:

$$\forall m \quad \sum_{n=1}^{N} x_{mn} \le 1 ; \tag{5}$$

2. Każda stacja może być dołączona dokładnie do jednego koncentratora, co można zapisać:

$$\forall k \quad \sum_{m=1}^{M} y_{km} = 1 \tag{6}$$

3. Sumaryczny ruch, generowany przez węzły pomiarowe dołączone do koncentratora n-tego typu, zlokalizowanego m-tym obszarze obsługi nie może przekroczyć przepustowości koncentratora, tj.:

$$\forall m \quad \sum_{n=1}^{N} b_n x_{mn} \ge \sum_{k=1}^{K} b_k y_{km} \tag{7}$$

gdzie: b_n – przepustowość koncentratora n-tego typu; b_k – przepustowość niezbędna do obsługi k-tej stacji pomiarowej.

5. Podsumowanie

Zadanie przydziału węzłów pomiarowych do komutatorów sieci sensorowej jest przykładem zadania przydziału zasobów z ograniczeniami i z punktu widzenia teorii złożoności obliczeniowej należy do klasy zadań NP-trudnych. Obecnie, nie są znane algorytmy pozwalające rozwiązać takie zadanie w czasie wielomianowym. W rezultacie, zadania takie są rozwiązywane za pomocą metod przybliżonych, zapewniających uzyskanie rozwiązań quasi-optymalnych w akceptowalnym czasie. W odróżnieniu od metod dokładnych, których złożoność czasowa ma charakter wykładniczy, metody przybliżone opisuje wielomianowa zależność pomiędzy ilo-

ścią danych wejściowych a czasem rozwiązania zadania. Pomimo ograniczonej dokładności uzyskiwanych rozwiązań, metody przybliżone, których doskonałymi przykładami są bazujące na analogiach biologicznych metody genetyczne, ewolucyjne oraz immunologiczne, pozwoliły rozwiązywać zadania przydziału zasobów również w systemach o bardzo dużym rozmiarze. Zadania te będą tematem dalszych prac autorów.

Bibliografia

[1] Darren L. Spohn, *Data Network Design*. New York: McGraw-Hill, 2002.

[2] Richard V. Slyke. (2001) Network planning and design. ftp://ftp.shore.net/members/ws/Support/BDC/nd.pdf.

[3] Vivek Alwayn, *Optical Network Design and Implementation*. Indianapolis: Cisco Press, 2004.

[4] Thomas L. Saaty, *Models, Methods, Concepts and Applications of the Analytic Hierarchy Proces*. Boston: Kluwer Academic Publishers, 2000.

[5] Roger L. Freeman, *Fundamentals ofTelecommunications*, II ed. Hoboken, New Jersey: John Wiley & Sons, 2005.

[6] Dipankar Raychaudhuri and Mario Gerla, *Emerging Wireless Technologies and the Future Mobile Internet*. Cambridge: Cambridge University Press, 2011.

[7] Andrew S. Tannenbaum, *Strukturalna organizacja systemów komputerowych*, V ed. Gliwice, Polska: Helion, 2006.

[8] Eldad Perahia and Robert Stacey, *Next Generation Wireless LANs: Throughput, Robustness, and Reliability in 802.11n*. Cambridge: Cambridge University Press, 2008.

[9] Emerson May, *Wireless Communications & Networks*, 2nd ed. New Jersey: Prentice Hall, 2004.

[10] M. D. Mesarovic, D. Macko, and Y. Takahara, *Theory of hierarchical, multilevel systems*. New York: Academic Press, 1970.

[11] N. J. Smith and A. P. Sage, *An introduction to hierarchical systems theory*. Dallas, Texas: Information and Control Sciences Center, SMU Institute of Technology, 2005.

[12] Mirosław Hajder, Paweł Dymora, and Mirosław Mazurek, "Projektowanie topologii transparentnych sieci optycznych," in *Konferencja Polski Internet Optyczny: technologie, usługi i aplikacje*, Poznań, 2002, pp. 183-195.

ROZDZIAŁ 6

Dekompozycja grafu-modelu sieci pomiarów środowiskowych

Piotr HAJDER[♣], Mariusz NYCZ[♦]

[♣]*Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie,*
[♦]*Politechnika Rzeszowska im. Ignacego Łukasiewicza*

Piootr.Hajder@gmail.com, MNycz@prz.edu.pl

Streszczenie

Rozdział rozpoczyna analiza obszaru badań, którymi są rozległe systemy pomiarów środowiskowych. Omówiono celowość stosowania systemów monitoringu, ich przeznaczenie i architekturę. Dalej przedstawiono alternatywną metodę dekompozycji grafu-modelu reprezentującego system pomiarów środowiskowych. Zaproponowano efektywne rozwiązanie zadania dekompozycji grafu o słabych powiązaniach, jakimi niewątpliwie są grafy-modele, z wielomianową złożonością czasową. Dekompozycja wykonywana jest za pomocą tworzenia minimalnego podzbioru rozdzielającego przeznaczone do sieci pomiarów środowiskowych. Pokazano, że dla podgrafów o rozmiarach oraz w macierzy incydencji grafu źródłowego mogą zostać wydzielone macierze o ściśle określonej wielkości. Wykorzystując przedstawienie grafu jako zbioru wierzchołków i incydentnych im krawędzi oraz zadanej na nim funkcji adekwatności, zaproponowano kombinatoryczną metodę dekompozycji struktury grafu.

1. Definicja zadań i obszaru badań

Obecnie, jednym z największych zagrożeń bezpieczeństwa osób i mienia są zjawiska przyrodnicze, charakteryzujące się krótkim czasem trwania, lokalnym zasięgiem oraz wyjątkową intensywnością przebiegu. Ich liczba oraz skala wzrastają rokrocznie. Zjawiska te są skutkiem postępujących zmian klimatycznych, pojawiających się m. in. w wyniku niedopuszczalnej intensywności wykorzystania środowiska naturalnego. Wysokim stratom pojawiającym się w ich rezultacie, sprzyja zły stan instalacji ochronnych, a także nieprzygotowanie społeczeństwa do takich zagrożeń [1], [2], [3], [4].

Niestety, przyczyny powyższych zjawisk nie znikną w najbliższych latach czy dziesięcioleciach – zjawiska takie są nieuchronne. Dlatego, oprócz prac ukierunkowanych na likwidację potencjalnych przyczyn ich pojawiania, należy podejmować

działania minimalizujące ich skutki, poprzez odpowiednio wczesne wykrywanie potencjalnych zagrożeń i informowanie o nich.

Współczesna meteorologia, hydrologia i geologia oferują szeroki asortyment metod i środków prognozowania zjawisk przyrodniczych. Bazują one na pozyskiwaniu w czasie rzeczywistym fizycznych parametrów środowiska, których określone wartości, bądź ich zmiany poprzedzają pojawienie się wspomnianych zjawisk. Dane te wykorzystuje się do opracowania krótko- lub długoterminowych prognoz występowania zagrożeń oraz ich przebiegu. Niestety, dla zjawisk lokalnych z wysoką intensywnością oraz krótkim czasem trwania, alarmujące sygnały pojawiają się bezpośrednio przed wystąpieniem niebezpieczeństwa, co utrudnia prognozowanie oraz przeciwdziałanie ich skutkom. Współczesne systemy monitoringu środowiskowego są mało skuteczne w walce z zagrożeniami tego typu. Konieczne jest opracowanie nowych rozwiązań, w szczególności charakteryzujących się: wysoką odpornością na uszkodzenia infrastruktury pomiarowo-komunikacyjnej, minimalnym czasem reakcji na zachodzące zjawiska oraz elastycznością częstotliwości i asortymentu wykonywanych pomiarów [5], [6], [7].

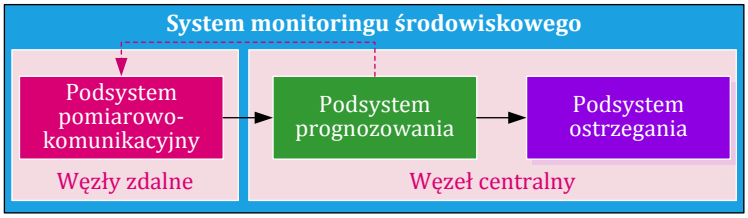
Projektowanie systemów monitoringu środowiskowego jest złożonym wieloetapowym procesem. Podstawowe trudności pojawiające się w jego trakcie, wynikają z konieczności pracy systemu w czasie rzeczywistym: wyniki pomiarów powinny być niezwłocznie przetwarzane, tak aby informacje o możliwych zagrożeniach natychmiast docierały do zainteresowanych osób [5]. Jednym z najważniejszych etapów projektowania jest projektowane topologiczne, którego celem jest określenie rozmieszczenia komponentów systemu pomiarowego oraz ich powiązań. Do rozwiązania powyższych zadań szerokie zastosowanie znajdują metody i środki bazujące na teorii grafów [8], [9], [10]. Wszelkie prace projektowe obejmujące organizację systemu pomiarowego wykonywane są na grafie-modelu, którego wierzchołki odpowiadają węzłom pomiarowym, a krawędzie kanałom komunikacyjnym. Ponieważ monitorowaniu podlegają znaczne obszary, na których poziom zagrożeń jest różny, graf-model jest niejednorodny – gęstość lokalizacji węzłów pomiarowych jest różna. Ponadto, graf ten jest rzadki, połączeniu podlegają wyłącznie wybrane pary wierzchołków, ich stopnień jest z zasady niewysoki [11], [12], [13].

Jednym z najważniejszych zadań stojących przed projektantem jest efektywny podział grafu-modelu na podgrafy obsługiwane przez węzły agregujące. Ich zadaniem jest zbieranie danych ze zlokalizowanych w sąsiedztwie węzłów pomiarowych, ich agregowane oraz dalsze przesłanie do węzła centralnego lub pośredniczącego w tym innego węzła. Z matematycznego punktu widzenia, podział ten jest dekompozycją grafu. Powinna ona zagwarantować niski koszt realizacji połączeń zarówno w podgrafie, jak również pomiędzy podgrafami, wysoką dostępność wszystkich komponentów systemu pomiarowego, a także akceptowalną złożoność czasową rozwiązania zadania, co dla dużych systemów może być trudne do zrealizowania [9], [11], [12], [16].

Zastosowanie dekompozycji grafu jest bardzo szerokie: od projektowania systemów technicznych, przez badania w ekonomii i socjologii, po przetwarzanie równoległe i rozproszone w informatyce. W wybranych przypadkach, uzyskanie rozwiązania dodatkowo utrudnia potrzeba późniejszej rekonfiguracji otrzymanego podziału, w szczególności poprzez przenoszenie pomiędzy obszarami skrajnych węzłów podsieci [10], [13].

2. Architektura systemu monitoringu środowiskowego

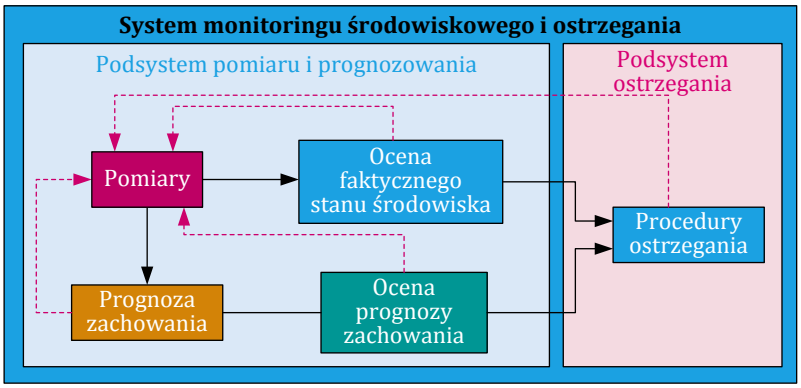
Klasyczny system monitoringu środowiskowego składa się z trzech funkcjonalnie rozdzielnych podsystemów: podsystemu pomiarowo-komunikacyjnego; prognozowania oraz ostrzegania, których wzajemne powiązania zostały zaprezentowane na rys. 1.



Rys. 1. Komponenty klasycznego systemu monitoringu środowiskowego

W klasycznym rozumieniu, system monitoringu przeznaczony jest do zbierania i przekazywania danych o stanie środowiska do centralnego węzła systemu. Nie przewiduje on automatycznej reakcji na wyniki pomiarów, które wykonywane są stosunkowo rzadko, w ściśle określonych godzinach. Pomiędzy systemem pomiarowym a prognostyczno-ostrzegawczym nie istnieje sprzężenie zwrotne, pozwalające np., na automatyczne zwiększenie częstości bądź dokładności pomiarów. W przypadku obiektów względnie wolnozmiennych (np. dużych rzek) rozwiązanie takie jest akceptowalne, ponieważ sytuacje kryzysowe nie pojawiają się natychmiast, a narastają przez stosunkowo długi czas.

Szeroki asortyment środków informatyki i telekomunikacji, ich stosunkowo niska cena i dostępność spowodowały modyfikację architektury systemu monitoringu i ostrzegania do postaci zaprezentowanej na rys. 1.



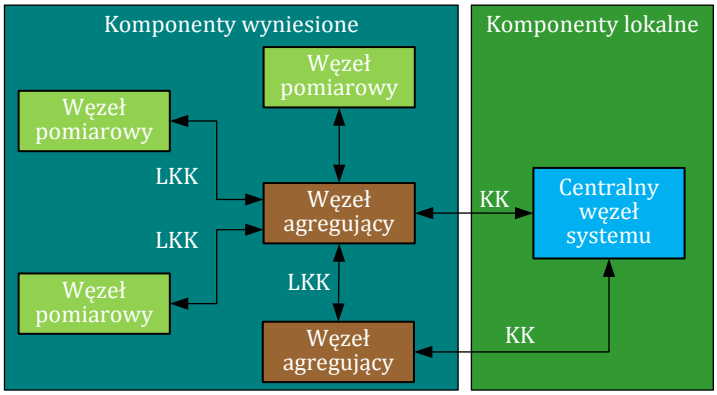
Rys. 2. Komponenty systemu monitoringu środowiskowego i ich wzajemne powiązania

z funkcjonalnego punktu widzenia analizowany system składa się z dwóch podsystemów: monitoringu i prognozowania środowiskowego oraz ostrzegania. Architektura takiego rozwiązania została przedstawiona na rys. 1.

W odróżnieniu od dotychczasowych rozwiązań w tym obszarze, analizowany system adaptuje swoje parametry do aktualnego stanu środowiska. Przykładowo, jeżeli prawdopodobieństwo wystąpienia zagrożenia zwiększa się, to pomiary mogą być wykonywane i ocenianej częściej. W tym celu, oprócz, przedstawionych na rysunku liniami ciągłymi, bezpośrednich relacji informacyjnych, w analizowanych systemach występują sprzężenia zwrotne (zaznaczone liniami przerywanymi), pozwalające dostosowywać charakterystyki ich do aktualnego stanu środowiska.

Architektura systemu pomiarów została przedstawiona na rys. 2 i ma organizację hierarchiczną złożoną z trzech typów węzłów:

1. Centralnego węzła systemu, przechowującego i przetwarzającego dane pomiarowe;
2. Węzłów agregujących, pośredniczących w gromadzeniu danych pomiarowych;
3. Węzłów pomiarowych, bezpośrednio wykonujących pomiary fizyko-chemicznych parametrów środowiska.



Rys. 3. Architektura systemu pomiarów środowiskowych

Węzły agregujące łączone są z węzłem centralnym za pomocą kanałów komunikacyjnych KK charakterystycznych dla sieci rozległych, a wybór konkretnego rozwiązania jest uzależniony od ich lokalnej dostępności.

Węzły agregujące łączone są z CWS, ale można je dołączyć do innych WA.

W celu zapewnienia efektywnego funkcjonowania systemu pomiarowego niezbędne jest jego podzielenie na mniejsze fragmenty obsługiwane za pomocą konkretnych węzłów agregujących. Podział ten jest wykonywany za pomocą dekompozycji grafu modelu systemu. Efektywność podziału odgrywa decydującą rolę w funkcjonowaniu systemów monitoringu.

3. Dekompozycja grafu modelu

Znane z teorii grafów metody poszukiwania minimalnych przekrojów i przecięć teoretycznie rozwiązują problem podziału sieci komputerowej na podsieci. Algorytmy te, zostały przedstawione m. in. w pracach [7], [8], [9], [10], [11]. Jeżeli sieć komputerową opisano w postaci zbioru elementów w przestrzeni metrycznej, to rozwiązanie zadania dekompozycji odbywa się w klasie ekstremalnych zadań ciągłego podziału zbiorów. Do tego celu wykorzystuje się głównie metodę Lagrange'a. Jednak większość znanych metod dekompozycji zorientowano na struktury jednorodne, co eliminuje je z projektowania sieci korporacyjnych i implikuje konieczność prowadzenia nowych badań.

Rozważmy zadanie dekompozycji, jako zadanie określenia minimalnego podzbioru rozdzielającego grafu. W teorii grafów pod pojęciem podzbioru rozdzielającego spójnego, nieskierowanego grafu $G = (V, E)$, rozumiemy niepusty podzbiór

B, dla którego graf $G' = (V \backslash B, E')$ utworzony przez operację usunięcia wierzchołków zawartych w zbiorze B ze zbioru A, jest niespójny. Jeżeli $A = \{a_i\}$ to zbiór wierzchołków grafu G, $i = 1, \ldots, N$, gdzie: N – liczba wierzchołków grafu; $E = \{e_{i,j} = (e_i, e_j)\}$, gdzie: $i, j = 1, \ldots, N$ i $i \neq j$. Dla grafu niespójnego tworzenie obszarów realizowane jest proste i sprowadza się do rozwiązania kombinatorycznego zadania podziału grafu na minimalne, niepokrywające się podgrafy [7]. Jednak dla grafów spójnych, z uwagi na złożoność przeszukiwania większość tych algorytmów okazuje się nieefektywna.

4. Spójność grafu, a macierz incydencji

Rozpatrzymy problem analizy spójności grafu na bazie jego macierzy incydencji, zdefiniujemy i udowodnimy konieczny i wystarczający warunek spójności grafu. W tym celu rozważymy nieskierowany, niespójny graf $G = (V, E)$, nazywany również grafem źródłowym, z n wierzchołkami, któremu odpowiada macierz incydencji $[B]_{n \times n}$. Załóżmy, że graf G składa się z dwóch podgrafów $G_1 = (V_1, E_1)_m$ oraz $G_2 = (V_2, E_2)_k$, z odpowiadającymi im macierzami incydencji $[B_{11}]_{m \times m}$ oraz $[B_{22}]_{k \times k}$, gdzie: $m + k = n$. Dalej przejmiemy, że wierzchołki od 1 do m to wierzchołki pierwszego podgrafu, a wierzchołki od $(m + 1)$ do n to wierzchołki drugiego podgrafu. Wtedy, macierz incydencji B grafu G może zostać przedstawiona w postaci macierzy blokowo-diagonalnej:

$$B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}. \tag{1}$$

Dla macierzy B incydencji grafu $G = (V, E)_n$, spełniony jest warunek dotyczący tworzących ją podmacierzy $(B_{12})^T_{m \times k} = [B_{21}]_{k \times m}$. Dzięki temu, przy analizie spójności grafu można rozpatrywać wyłącznie jedną z powyższych podmacierzy, na przykład $[B_{12}]_{m \times k}$. Dowolny element $b_{i,j}$ macierzy określa połączenie pomiędzy wierzchołkami v_i oraz v_j, przy czym $i \in \{1,2, \ldots, m\}$, a $j \in \{m + 1, m + 2, \ldots, n\}$. Wynika stąd, że v_i jest wierzchołkiem podgrafu $G_1 = (V_1, E_1)_m$, a v_j, to wierzchołek podgrafu $G_2 = (V_2, E_2)_k$. Zgodnie ze wcześniejszymi ustaleniami oba powyższe podgrafy G_1 i G_2 są rozłączne, tj. dla macierzy $[B]_{m \times m}$ element $b_{i,j} = 0$. W ten sposób, można założyć, że warunkiem spójności grafu G jest niezerowa macierz incydencji $[B_{12}]_{m \times k}$. Można udowodnić, że jednoczesne przestawienie odpowiednich wierszy i kolumn macierzy incydencji nie prowadzi do zmiany połączeń grafu. Ponadto, odpowiednie przestawienie wierszy i kolumn prowadzi do wydzielenia podmacierzy zerowych. Dzięki temu możliwe jest dokonanie oceny spójności grafu. Jeśli postać macierzowa analizowanego grafu jest blokowo-diagonalna z podmacierzami zerowymi, to mamy do czynienia z grafem niespójnym. Powyższe właściwości zostały wykorzystane do badania spójności grafu.

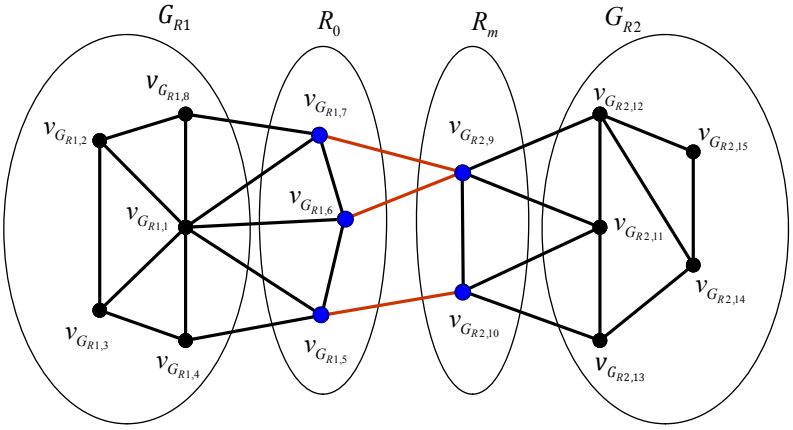
W ramach kombinatorycznego rozwiązania zadania poszukiwania minimalnego podzbioru rozdzielającego wykorzystywane są również operacje nad macierzami

binarnymi, w szczególności operacje określenia pełnej podmacierzy, tj. nie zawierającej się w żadnej innej macierzy [7], [12].

5. Podzbiory pokrywające i rozdzielające

W większości znanych algorytmów poszukiwania podstawowej podmacierzy macierzy binarnej zakłada się, że poszukiwanie podzbioru pokrywającego pełnych podmacierzy realizowane jest wierszami, tj. jako źródłowa podmacierz wybierany jest określony wiersz. Następnie, za pomocą sekwencyjnego przeszukiwania parami realizowane jest połączenie pełnych podmacierzy. Proces łączenia wykonywany jest do momentu otrzymania minimalnego pokrycia macierzy wejściowej jej podstawowymi podmacierzami. Niestety, przy zwiększeniu rozmiarów macierzy B znacznie zwiększają się nakłady na przeszukiwanie wszystkich wariantów rozwiązania. Najczęściej algorytmy kombinatoryczne wymagają $n(n+1)/2$ operacji porównania oraz $n(n-1)^2$ operacji przestawienia elementów macierzy.

W odróżnieniu od znanych metod, w których analizowane są podmacierze jednostkowe, w proponowanej metodzie, zadanie tworzenia pełnej podmacierzy rozwiązuje się względem *pełnych* zerowych podmacierzy, co pozwala zrezygnować z wykorzystania dodatkowych macierzy. Do dalszej analizy wykorzystamy graf przedstawiony na rys. 3.



Rys. 4. Analizowany graf ze strukturą regionalną

W ogólnym przypadku macierz incydencji przykładowego grafu źródłowego może być przedstawiona w postaci macierzy blokowo-diagonalnej zaprezentowanej za pomocą poniższego wyrażenia:

$$\begin{bmatrix} \boxed{B_{11}} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & \boxed{B_{33}} \end{bmatrix} \tag{2}$$

Podmacierze $[B_{11}]_{m\times m}$ oraz $[B_{33}]_{k\times k}$ opisują odpowiednio podgrafy: $G_1 = (V_1, E_1)_m$ z wierzchołkami $v_{G_{R1,1}}, v_{G_{R1,2}}, \ldots, v_{G_{R1,8}}$ oraz $G_2 = (V_2, E_2)_k$ z wierzchołkami $v_{G_{R2,11}}, v_{G_{R2,12}}, \ldots, v_{G_{R2,15}}$, a podmacierz $[B_{22}]_{l\times l}$ odpowiada zbiorowi rozdzielającemu grafu źródłowego z wierzchołkami $v_{R_{2,9}}, v_{R_{2,10}}$. Rozmiar l macierzy B_{22} można wyliczyć ze wzoru: $l = n - (m + k)$. Podmacierz $[B_{12}]_{m\times l}$ określa krawędzie pomiędzy podgrafem $G_1 = (V_1, E_1)_m$ i zbiorem rozspajającym, zbiór krawędzi tworzy odpowiedni przekrój grafu. Analogicznie, podmacierz $[B_{23}]_{l\times k}$ określa połączenia i przekrój pomiędzy zbiorem rozpajającym i podgrafem $G_2 = (V_2, E_2)_k$. Podmacierz $[B_{13}]_{m\times k}$ określa połączenia pomiędzy podgrafami $G_1 = (V_1, E_1)_m$ oraz $G_2 = (V_2, E_2)_k$, które nie weszły w zbiór rozspajający, a ponieważ zgodnie z warunkiem one nie występują, to analizowana macierz jest zerowa.

W ten sposób można stwierdzić, że dla podgrafu o rozmiarze m i podgrafu o rozmiarze k w macierzy incydencji grafu źródłowego może być wydzielona zerowa podmacierz o rozmiarze $m \times k$ i dwie podmacierze rozspajające o rozmiarach odpowiednio $(m \times l)$ i $(l \times k)$, gdzie: $l = n - (m + k)$. Dla $l = 0$ graf G dzielony jest na dwa niezależne podgrafy. W ten sposób, jako optymalny podział grafu G na dwa podgrafy G_1 oraz G_2, będziemy uważać taki podział, przy którym zbiór rozspajający X jest minimalny dla k dążącego do m oraz $m + k$ dążącego do n. Dalej, operacje przestawiania wierszy i kolumn można wykluczyć dzięki reprezentacji wierszy macierzy w postaci zbiorów i wykonywania operacji nad nimi.

W powyższym przypadku, zadanie określenia maksymalnego podzbioru rozdzielającego można zdefiniować w następujący sposób: dla dowolnych wierzchołków v_i oraz v_j grafu $G = (V, E)$, takich że: $v_i \in V_1$ oraz $v_j \in V_2$, należy określić podzbiór X wierzchołków grafu, spełniających następujące warunki:

1. $V_1 \cap X = \varnothing$;
2. $V_2 \cap X = \varnothing$;
3. $V_1 \cap V_2 = \varnothing$;
4. $V_1 \cup V_2 \cup X = V$;
5. $X \to \min$;
6. $D_1 < D_{\max}$;
7. $D_2 < D_{\max}$.

Relacja pomiędzy rozpatrywanymi podzbiorami określana jest wyrażeniem: $X = V \setminus (V_1 \cup V_2)$. Oczywiście, podzbiór X będzie minimalnym przy maksymalnych rozmiarach podzbiorów V_1 oraz V_2. W ten sposób, zadanie określenia minimalnego podzbioru X może być sprowadzone do określenia maksymalnych podzbiorów V_1 i V_2. Zatem, formowanie pełnej zerowej podmacierzy może zostać zamienione na formowanie dwóch rozłącznych zbiorów, odpowiadających macierzy blokowej $[B_{1,3}]$ źródłowej macierzy incydencji. Podstawową operacją proponowanego algorytmu jest sekwencyjne połączenie pomiędzy sobą zbiorów V_i, gdzie $i = 1, \ldots, n$.

Jako zbiór źródłowy V_0 wybiera się zbiór V_i o maksymalnej mocy (największej liczbie elementów). Następnie, spośród pozostałych zbiorów wybiera się zbiór V_i, utworzony poprzez połączenie z V_0 zbioru V_M=($V_0 \cap V_i$) o maksymalnej mocy. Jednocześnie określa się wartość zmiennej $r_1 = k + m$, gdzie: k – liczba połączonych zbiorów, m – moc zbioru V_M. Po wykonaniu pierwszego kroku wielkość ta jest równa: $r = 2 + m$.

W drugim i kolejnych krokach wykonywane jest analogiczne przecięcie zbioru V_m z pozostałymi zbiorami: $V_M = (V_0 \cap V_i)$. Na każdym i-tym kroku obliczana jest wartość zmiennej r_i. Proces kontynuowany jest do momentu, aż $(k + m) = n$ lub $i = n$. W takim przypadku graf źródłowy przedstawia sobą dwa podgrafy, pierwszy z których zawiera k, drugi zaś m wierzchołków. Wierzchołki drugiego grafu określane są za pomocą zbioru V_M, a pierwszego zbiorem $V \backslash V_M$. Zauważmy, że po spełnieniu warunku: $(k + m) = n$, graf z definicji jest niespójny i proces obliczeń się zakańcza. W szczególnym przypadku, dla grafu słabo-spójnego proces kończy się $i = n$.

Dla grafów o wysokiej spójności wykorzystanie zaproponowanego algorytmu może okazać się nieefektywne. Wynika to z faktu, że w grafach tych jeden jedyny maksymalny przekrój w istocie nie istnieje. W takim przypadku rozbicie na podsieci realizowane jest na podstawie szeregu kryteriów, wśród których najważniejszym jest stosunek intensywności wewnętrznych i zewnętrznych strumieni informacyjnych. Wtedy, jako dane źródłowe wykorzystywać będziemy wartości strumieni informacyjnych pomiędzy węzłami sieci, określane za pomocą macierzy sąsiedztwa $\Lambda = \left|\lambda_{i,j}\right|_m$. W macierzy tej, każdy element $\lambda_{i,j}$ określa intensywność strumienia informacyjnego pomiędzy wierzchołkami v_i oraz v_j. Z uwagi na wysokie koszty realizacji połączeń, analizowane grafy-modele systemów monitoringu środowiskowego są z definicji słabo spójne.

6. Typy wierzchołków i przekroje sieci

Rozróżniamy wierzchołki: wewnętrzne i graniczne. Wierzchołkami wewnętrznymi $S_{i,j}^{+}$, nazywamy wierzchołki pozwalające utworzyć wyłącznie krawędzie na bazie podzbioru wierzchołków jednego obszaru, tj. $i, j \in \{r\}$, gdzie: r – zbiór indeksów wierzchołków pewnego zbioru S_Z. Odpowiednio, granicznym wierzchołkiem $S_i^{-} \in S_Z$ będziemy nazywać wierzchołek S_j, dla którego $j \notin \{r\}$.

Przekrojem sieci nazywamy zbiór A_p, składający się z v krawędzi, przy usunięciu których źródłowy graf jest rozdzielany na dwa niespójne podgrafy. Dla dowolnego grafu może istnieć wiele przekroi, jednak z zasady istnieje tylko jeden minimalny, składający się z minimalnej możliwej liczby krawędzi. Znane są heurystyczne metody określania minimalnego przekroju bazujące na sortowaniu wszystkich krawędzi grafu $e_{i,j}$ źródłowego w kolejności zmniejszania się współczynnika $k_{i,j}$ wykorzystania kanałów transmisji danych [7], [12].

W pracy [7] opisano algorytm znajdowania wszystkich podstawowych podmacierzy zadanej macierzy binarnej, bazujący na sekwencyjnym przeszukiwaniu i łączeniu parami pełnych macierzy. Oczywiście, przy zwiększaniu rozmiaru macierzy źródłowej znacznie wzrastają nakłady na przeszukiwanie wszystkich wariantów, dlatego zaproponowano następujący algorytm rozwiązania danego zadania, bazujący na tworzeniu pełnej podmacierzy, w której $l = min$. Przy realizacji danego podejścia wykorzystuje się zasadę ekwiwalentnego przestawienia wierszy i kolumn macierzy incydencji przy którym nie jest naruszana topologia grafu.

Przedstawimy proponowany algorytm. Początkowo wybierany jest i-ty wiersz spełniający warunek $max \sum_{j=1}^{n} a_{i,j}$ i przestawiany jest z pierwszym wierszem, odpowiednio przestawiane są również odpowiednie kolumny. Następnie, zgodnie z warunkiem: $max \sum_{j=1}^{n}(a_{1,j} \cap a_{k,j})$, wybierany jest k-ty wiersz, który z kolei jest zamieniany z drugim wierszem. Jeżeli występuje kilka wierszy z identyczną sumą, wybierany jest wiersz z maksymalną wartością $a_{k,j}$.

W rezultacie wykonania przekształceń macierzowych przewidzianych przez algorytm, źródłowa macierz sąsiedztwa sprowadzana jest do postaci zbliżonej do blokowej. Analiza otrzymanej macierzy pozwala określić obszary lokalne i wiążące ich krawędzie. Jednakże, nawet dla grafu o niskiej spójności algorytm ten nie gwarantuje otrzymania jednoznacznego rozwiązania. W przypadku grafów o wysokiej spójności i równomiernym rozkładzie strumieni informacyjnych, algorytm ten nie daje rozwiązania, którego znalezienie wymaga zmiany strumieni oraz usunięcia wybranych krawędzi grafu. W takim przypadku wśród wybranych zbiorów rozdzielających, określa się zbiór z minimalną wartością v oraz spełniający poniższy warunek:

$$\min \sum_{i=1}^{v} I_{i,j}^{-} \sum_{j=1}^{v} I_{i,j}^{-}.$$

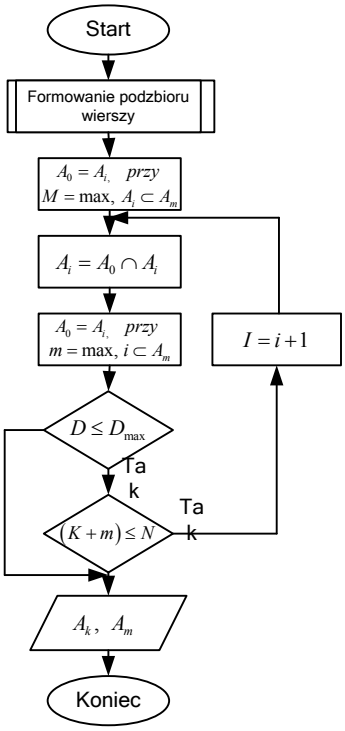
Dalszy proces optymalizacji polega na usunięciu jednej lub kilku krawędzi $I_{i,j}^{-}$ oraz zmianie rozkładu obciążenia dopóty, dopóki spełniony jest warunek:

$$\sum_{i=1}^{v} t_{i,j}^{-} \geq \sum_{i=1}^{v} I_{i,j}^{-}.$$

7. Kombinatoryczny algorytm dekompozycji grafu

Zgodnie z powyższymi rozważaniami wyłączenie operacji przestawiania wierszy i kolumn może zostać osiągnięte dzięki przedstawieniu wierszy macierzy w postaci zbiorów i wykonaniu operacji nad zbiorami. Rzeczywiście graf można przedstawić jako parę $G = (A, \Gamma)$, gdzie: $A = \{a_i | i = 1,2, \dots, n\}$ – zbiór wierzchołków grafu, Γ – związek pomiędzy podzbiorami zbioru A, który z uwzględnieniem tego, że operacje wykonywane są nad zerowymi macierzami określimy w następujący sposób: $\Gamma(a_i) = \left(j | \forall S_{i,j} = 0, j \neq i, j = 1,2, \dots, n\right)$, dla wszystkich $i = 1,2, \dots, n$.

Dalej jako $\Gamma(a_i)$ oznaczymy podzbiór A_i, który określa istnienie jedynki w j-ej kolumnie, i-tym wierszu macierzy incydencji grafu źródłowego. W tym przypadku proces formowania pełnej zerowej podmacierzy może zostać zamieniony na tworzenie rozłącznych podzbiorów odpowiadających blokowej macierzy $M_{1,3}$ źródłowej macierzy incydencji. Algorytm formowania podgrafów został przedstawiony w postaci graficznej na rys. 4.



Rys. 5. Algorytm tworzenia podgrafów

Bazową operacją prezentowanego algorytmu jest sekwencyjne połączenie zbiorów A_i. Jako zbiór źródłowy A_0 wybierany jest zbiór A_i z maksymalną liczbą elementów. Następnie, z pozostałych zbiorów wybiera się zbiór A_i, utworzony poprzez połączenie ze zbiorem A_0 maksymalnego zbioru $A_M = (A_0 \cap A_i)$. Po każdej operacji połączenia określamy liczbę $r_1 = k + m$, gdzie: k – liczba połączonych zbiorów, m – moc zbioru A_M. Po pierwszym kroku, wartość $r = 2 + m$. Na drugim i kolejnych krokach połączenie A_M z pozostałymi podzbiorami $A_M = (A_0 \cap A_i)$ realizowane jest w identyczny sposób. Przy tym, dodatkowo dla podzbiorów A_i $(i = 1,2, \dots, n)$ określimy pomocnicze podmacierze oznaczone jako $\left\|C_1^i\right\|_{2\times d}$, dla których pierwszy z indeksów wskazuje numer kolumny macierzy $|\Lambda|_{n\times n}$ na przecięciu z którą w i-tym wierszu macierzy źródłowej $|\Lambda|_{n\times n}$ znajduje się wartość $\lambda_{i,j} \neq 0$, a jej pierwszy wiersz określa samo znaczenie tej wielkości. Tutaj wielkość r określa sumaryczną ilość elementów w i-tym wierszu macierzy źródłowej

$|\Lambda|_{n\times n}$. Zgodnie z proponowanym algorytmem na każdym kroku formowane są dwa bezpośrednio niepowiązane ze sobą podgrafy, numery wierzchołków których będziemy odpowiednio wnosić do zbiorów A_0 oraz A_M. Numery wierzchołków, tworzących podzbiory powiązania danych podgrafów będziemy wnosić do wektora M_c.

Po stworzeniu podzbiorów A_i $(i = 1,2, \dots, n)$ oraz dodatkowych macierzy $\left|C^i\right|_{2\times d}$ wykonywane jest sekwencyjne łączenie zbiorów A_i pomiędzy sobą. Jako źródłowy zbiór A_0 wybierany jest zbiór A_i z maksymalną liczbą elementów. Następnie z pozostałych zbiorów wybiera się zbiór A_i, tworzony poprzez połączenie z A_0 maksymalnego zbioru $A_M = (A_0 \cap A_i)$. Określana jest liczba $r = k + m$, gdzie: k – liczba połączonych zbiorów, m – moc zbioru A_M. Po pierwszym kroku algorytmu wartość $r = 2 + m$.

W celu określenia wewnętrznych strumieni otrzymanych grafów tworzymy podzbiory powiązania M_c. Następnie, na podstawie zbiorów A_0, A_M, wektora M_c i macierzy $\left|C^i\right|_{2\times m}$ określamy intensywność strumieni względem podgrafów M_1 oraz M_2.

Na drugim i kolejnych etapach w identyczny sposób realizowane jest połączenie zbioru A_M z pozostałymi $A_M = A_0 \cap A_i$. Na każdym i-tym kroku wyliczana jest odpowiednia wartość. Proces łączenia kończony jest dla $k + m = n$ lub $i = n$. W pierwszym przypadku źródłowy graf jest niespójny, w drugim podział grafu źródłowego określany jest relacją podzbiorów A_0, A_M, M_c.

W celu określenia czasowej oceny formowania podgrafów, określimy liczbę operacji proponowanego algorytmu. Uwzględniając, że na każdym kroku obliczeń tworzone są jednocześnie dwa podgrafy, to wystarcza $n/2$ kroków, gdzie: n – liczba wierzchołków grafu. Dalej, ponieważ formowanie podzbioru A_M rozpoczyna się z wierzchołka posiadającego minimalną liczbę krawędzi, to liczba operacji przecięć zbioru na każdym kroku obliczeń nie przewyższa średniej liczby r krawędzi grafu. W takim przypadku można uważać, że sumaryczna liczba operacji tego typu nie przekroczy wielkości $nr/2$, co jest mniejsze od innych znanych algorytmów.

8. Podsumowanie

Prowadzone prace pozwoliły stworzyć, przebadać i potwierdzić efektywność oryginalnej metody dekompozycji grafów. W szczególności, dzięki przeprowadzonym pracom:

1. Uzasadniono celowość rozwiązania zadania dekompozycji struktury grafu o słabych powiązaniach jako zadania tworzenia minimalnego podzbioru rozdzielającego grafu, a także określono konieczne i wystarczające warunki istnienia takiego zbioru.

2. Udowodniono, że dla podgrafu o rozmiarze m i podgrafu o rozmiarze k , w macierzy incydencji grafu źródłowego może zostać wydzielona zerowa podmacierz o rozmiarze $m \times k$ i dwie macierze rozdzielające, o rozmiarze odpowiednio $m \times 1$ oraz $1 \times k$, gdzie: $n - (m + k) = 1$.
3. Wykorzystując reprezentację grafu jako zbioru wierzchołków i zadanej na nim funkcji adekwatności między nimi, opracowano kombinatoryczną metodę de-kompozycji struktury grafu, odróżniającą się od znanych wysoką zbieżnością.

Bibliografia

[1] M. Hajder i M. Nycz, „Co należy wiedzieć na temat zagrożeń naturalnych?," w *Innowacyjna Gmina*, Leżajsk, Urząd Gminy Leżajsk, 2011, pp. 49-67.

[2] M. Hajder i M. Nycz, „Analiza systemu hydrologicznego i przeciwdziałanie zagrożeniom powodziowym Tarlaki," w *Innowacyjna Gmina*, Leżajsk, Urząd Gminy Leżajsk, 2011, pp. 123-137.

[3] M. Hajder i M. Nycz, „Architektura systemu wykrywania zagrożeń powodziowych o zasięgu lokalnym," w *Innowacyjna Gmina*, Leżajsk, Urząd Gminy Leżajsk, 2011, pp. 83-95.

[4] M. Hajder i M. Nycz, „Koncepcja systemu prognozowania lokalnych zagrożeń środowiskowych," w *Innowacyjna Gmina*, Leżajsk, Urząd Gminy Leżajsk, 2011, pp. 69-82.

[5] M. Hajder, H. M. Loutskii i W. Stręciwilk, Informatyka. Wirtualna podróż w świat systemów i sieci komputerowych., I red., M. Hajder, Red., Rzeszów: Wydawnictwo Wyższej Szkoły Informatyki i Zarządzania, 2002.

[6] R. Diestel, Graph Theory, 4 red., New York: Springer, 2010.

[7] M. J. Atallah, Red., Algorithms and Theory of Computation Handbook, Boca Raton: CRC Press, 2002.

[8] L. A. Sanchis, "Multiple-Way Network Partitioning with Different Cost Functions," *IEEE. Trans. on Computers,* vol. 42, pp. 221-232, 12 1993.

[9] S. Tragoudas, „Min-Cut Partitioning on Underlying Tree and Graph Structures," Southern Illinois Univ., Carbondale, IL., 2007.

[10] G. Vijayan, "Generalization of Min-Cut Partitioning to Tree Structures and Its Applications," *IEEE. Trans. on Computers,* vol. 40, pp. 167-180, 3 1991.

[11] R. L. Graham, M. Grotschel i L. Lovasz, Handbook of Combinatorics, tom 1, Amsterdam: Elsevier, 1995.

[12] R. L. Graham, M. Grotschel i L. Lovasz, Handbook of Combinatorics, tom 2, Amsterdam: Elsevier, 1995.

[13] S. Narasimhan, „Allocation of Concentrators in Computer Network with Multiple Cover-age of Terminals," *IEEE Computer Communications,* tom 26, nr 2, pp. 245-256, 2--3.

[14] M. Hajder, L. Hajder i T. Bartczak, „Projektowanie niezawodnej łączności wielokanałowej dla sieci sensorowej," w *Innowacyjna Gmina*, Leżajsk, Urząd Gminy Leżajsk, 2011, pp. 169-182.

ROZDZIAŁ 7

Ataki typu odmowa usługi na serwery webowe: podatność, wykrywanie, przeciwdziałanie

Mariusz NYCZ, Michał Wroński

Politechnika Rzeszowska

MNycz@prz.edu.pl, MWronski@prz.edu.pl

Streszczenie

W rozdziale poddano analizie ataki typu DoS na serwery webowe utrzymujące strony internetowe. W szczególności przedstawiono obiekty ataków, analizę podatności, sposoby ich wykrywania oraz przeciwdziałania. Rozdział prezentuje wyniki badań własnych autorów i porównuje je materiałami źródłowymi. Jest on adresowany do informatyków zajmujących się bezpieczeństwem serwisów www, a także osób i podmiotów, którzy na bazie platform Arduino oraz Raspberry Pi chciałyby zbudować własny detektor ataków.

1. Wprowadzenie

W ostatnim dziesięcioleciu strony internetowe organów administracji publicznej stały się najważniejszym źródłem informacji dla obywateli. Zaznajamiają one z dotychczasowymi osiągnięciami władzy, prezentują jej plany, pozwalają załatwiać zdalnie coraz większą liczbą spraw urzędowych. Mogą również brać udział w promocji turystycznej i biznesowej, stanowią podstawowe źródło, pozyskiwania nowych inwestorów zapewniających mieszkańcom miejsca pracy. Informatywna, czytelna i prosta w obsłudze strona internetowa jest obecnie podstawowym elementem budowy wizerunku jednostki administracji publicznej. Jest absolutnie pewne, że w najbliższych latach liczba i rozmiar stron będą się zwiększać, a ich funkcjonalność będzie obejmować coraz to nowe usługi.

Oprócz niewątpliwych zalet, wykorzystanie stron Internetowych w bieżącej działalności urzędu, posiada również wiele wad. Zdalny dostęp do usług oraz ich nieszablonowy charakter wymagają od aplikacji obsługującej stronę większej odporności na błędy użytkownika, a także zachowania szczególnej ostrożności przy gromadzeniu i przetwarzaniu danych, co wynika z konieczności zachowania poufności tych ostatnich. Nie bez znaczenia dla wiarygodności urzędu i satysfakcji użytkowników pozostają również dostępność strony oraz szybkość jej funkcjonowania –

każda z osób korzystających z witryny oczekuje, że będzie ona dostępna permanentnie, a dostęp do zasobów będzie realizowany niemal natychmiast [1].

Zagwarantowanie wysokiej jakości usług bazujących na stronie Internetowej, wymaga dysponowania odpowiednim sprzętem komputerowym i wystarczająco szybkim łączem, które określamy mianem *infrastruktury świadczenia usług*. Infrastruktura ta może być eksploatowana samodzielnie przez jednostkę administracji, może być również dzierżawiona w postaci usługi hostingowej, świadczonej przez podmiot zewnętrzny. Powyższe rozwiązania można traktować jako alternatywne, każde z nich posiada swoje wady i zalety. W Polsce, w jednostkach administracji publicznej, dominuje pierwsze z nich, co jest konsekwencją nie do końca słusznego przekonania, że zlecanie usług hostingowych na zewnątrz pogarsza poziom bezpieczeństwa. W rzeczywistości, problem ten jest nieco bardziej skomplikowany: istnieją usługi, najczęściej ogólnokrajowe, które z mocy prawa muszą być obsługiwane w oparciu o własne serwery. Strona sprzętowa pozostałych usług może być z powodzeniem realizowana w postaci zlecenia zewnętrznego. Bez względu na to, który ze schematów obsługi zostanie wybrany, jednostka udostępniająca stronę Internetową musi posiadać odpowiednie zaplecze techniczno-administracyjne. Jeżeli część sprzętowa obsługi będzie realizowana samodzielnie, liczebność personelu będzie zdecydowanie większa, a jego poziom merytoryczny będzie najczęściej gorszy niż w przypadku zlecania usługi podmiotom komercyjnym.

Bezpieczeństwo strony Internetowej wymaga nie tylko modyfikacji podejścia organizacyjno-technicznego, ale również mentalnego użytkowników. Zmiana ludzkich przyzwyczajeń, ciągnących się nierzadko przez wiele dziesięcioleci jest zadaniem bardzo trudnym i często kończy się niepowodzeniem. Jedynym sposobem ich przeobrażenia są systematyczne szkolenia, które np. zgodnie z wytycznymi standardu ISO 27001 powinny odbywać się nie rzadziej niż raz w roku.

W każdym przypadku, poziom ochrony należy dostosować do pojawiających się zagrożeń i ich skutków. Nieodpowiedni dobór może, z jednej strony (niedostateczny poziom), skutkować naruszeniem bezpieczeństwa systemów i samej informacji, poprzez ingerencję w jej poufność, integralność oraz dostępność, z drugiej zaś (nieadekwatnie wysoki poziom) utrudniać użytkownikom korzystanie z zasobów i generować niepotrzebne koszty. Realizacja ataków naruszających poufność i integralności jest działaniem wymagającym posiadania przez intruza znacznych zasobów technicznych i intelektualnych. Z kolei ataki na dostępność, uważaną za podstawowy parametr bezpieczeństwa, mogą bazować na względnie skromnych zasobach sprzętowych i ograniczonej wiedzy. Dlatego, ochrona przez atakami tego typu jest szczególnie trudna i większość stron może się stać celem ataku. Powszechnie uważa się, że szeroki asortyment rodzajów ataków, prostota ich realizacji oraz wysoka złożoność walki z nimi powoduje, że ataki typu *Denial of Service* (DoS) są jednymi z najbardziej skutecznych i najczęściej wykorzystywanych metod zablokowania dostępu do serwisu.

Niniejszy rozdział prezentuje wyniki badań autorów w tym zakresie.

2. Architektura ataków

2.1. Tło prowadzonych badań

Ataki DoS realizowane są w środowisku sieci komputerowej, a ich wykrywanie najczęściej opiera się na wyszukiwaniu anomalii ruchu [2]. W danym przypadku, pojęciem *sieci* określamy grupę komputerów lub systemów komputerowych, powiązanych w celu wspólnego korzystania z zasobów: sprzętowych, programowych i informacyjnych. Typowa sieć składa się ze zbioru węzłów zintegrowanych ze sobą za pomocą urządzeń sieciowych. Najliczniejszą grupę węzłów tworzą tzw. *węzły obliczeniowe*, wśród których rozróżniamy: *serwery* – główne jednostki usługodawcze w sieci, korzystające z modelu klient-serwer, wyposażone w rozbudowane oprogramowanie sieciowe i znaczne zasoby pamięciowe, obliczeniowe i wejścia-wyjścia. Korzystając z serwera zazwyczaj zakładamy, że interfejs użytkownika będzie realizowany bezpośrednio na jednostce końcowej; *stacje robocze* – jednostki komputerowe wyposażone w jednostanowiskowy system operacyjny z podziałem czasu i przeznaczone do zapewnienia współdziałania użytkownika z zasobami sieci [3]. Tak zorganizowane struktury dołączane są do Internetu – sieci globalnej. Z tego faktu, wynika skala problemu – w Internecie ataki na interesującą nas jednostkę mogą być realizowane praktycznie z każdego urządzenia dołączonego do niego.

Bazując na danych statystycznych wynika, że w jednostkę komputerową dołączoną do Internetu wyposażona jest większość gospodarstw domowych w krajach rozwiniętych i tylko 10% w krajach rozwijających się. Chociaż średnia światowa sięgająca 30% nie jest imponująca, daje to liczbę około 1 mld gospodarstw domowych korzystających z zasobów Internetu. Rozwijając wcześniejszą myśl, należy przyjąć iż ataki typu DoS mogą być wykonywane, często bez wiedzy właścicieli, ze znacznej części wspomnianej liczby jednostek (komputerów, tabletów, smartfonów, ale również Internetu rzeczy).

Chociaż ataki typu *odmowa usługi* należą do najstarszych sposobów ingerencji w funkcjonowanie systemów informacyjnych, nadal zalicza się je do najbardziej skutecznych. Z czasem okazało się, że ataki DoS, w których atak na jednostkę użytkownika odbywa się z jednego komputera są mało skuteczne, a walka z nimi stosunkowo prosta. Dlatego, ataki tej klasy, końcem XX wieku ewoluowały do wersji rozproszonej, określanej terminem *Distributed Denial of Service* (DDoS). Atak realizowany jest za pośrednictwem *bootnetów* – grup zainfekowanych komputerów. Pierwsza próba realizacji została zanotowana w 1999 roku przez Uniwersytet stanu Minnesota, obecnie w grupie *odmowa usługi*, ataki DDoS są jedynymi realizowanymi.

Sposób realizacji ataku jest uzależniony od warstwy modelu, na którym jest on wykonywany. Jeżeli do komunikacji pomiędzy komponentami sieci używany będzie model TCP/IP można wykorzystać klasyfikację zaprezentowaną na rys. 1.

Klasyfikacja ataków DDoS			
Warstwa aplikacji	Warstwa transportowa	Warstwa Internetu	Warstwa dostępu do sieci
HTTP/HTTPS Flooding, FTP Flooding, SSL/TLS, Telnet DDoS, SIP (VoIP) Flooding, SMTP, IRC, Mail Bombs, SQL Slammer, Distributed Reflector/ DNS Flood, DNS Cache poisoning.	SYN Attack, UDP Flooding, ACK Flooding, PUSH, FIN, RST, URG Flood, NetBIOS Flood, TCP NULL Flood, TCP ECE Flood, TCP Xmas, UDP port 0, Land attack.	Frag, Opentear, Nestea, TearDrop, Jolt, Smurf, Fraggle, Papasmurf, Smack, IP address, IP Packet Option, IP non-existing protocol, Forged Source IP, IP spoofed, ICMP/IGMP Flooding.	VLAN Hopping, MAC Flood, DHCP Attacks, ARP Attacks.

Rys. 1. Taksonomia ataków DDoS w oparciu o warstwy modelu TCP/IP [4]

Uproszczoną (pominięto część warstw modelu) klasyfikację ataków DDoS na sieci bezprzewodowe pokazano na rys. 2.

Klasyfikacja ataków DDoS na sieci bezprzewodowe				
Warstwa fizyczna	Warstwa łącza danych	Warstwa sieci	Warstwy wyższe modelu ISO	
Zakłócenia radiowe	Imitacja połączeń	Określenie lokalizacji	Protokoły autentykacji	Protokoły i programy
	Detekcja nośnej	Wykorzystanie zasobów	4-stronna synchronizacja	Modyfikacja MAC protokołu
	Zachowanie mocy	Zmiana trasy	Autentykacja EAP/802.1X	Nieaktualne ramki
		Wykorzystanie tunelowania		

Rys. 2. Klasyfikacja ataków DDoS na sieci bezprzewodowe

W celu sklasyfikowania ataków zastosowano kryterium mieszane uwzględniające warstwę modeli oraz kluczowe usługi świadczone w sieci. Najczęściej cytowaną publikacją naukową dotycząca taksonomii ataków DDoS jest praca J. Mirkowic i P. Reihera [5]. Zaproponowana klasyfikacja bazuje na odpowiedzi na kilka istotnych pytań: **a.** Jakie techniki realizacji ataku są wykorzystywane? **b.** Dzięki jakim rozwiązaniom udaje się pokonać mechanizmy ochronne? **c.** Które z ataków mogą zostać efektywnie przetworzone za pomocą dostępnych systemów ochronnych? **d.** Które spośród ataków nie zostały dotąd dokładnie zbadanie i dlaczego? **e.** Dostępne są dwa mechanizmy ochronne A oraz B. Jak wyglądałoby ich działanie, w przypadku ataku typu C? Jakie są ich słabe punkty, czy mogą się one wzajemnie uzupełniać i w jaki sposób? Czy istnieją sytuacje, kiedy zastosowanie mechanizmu A jest preferowane względem B i odwrotnie? **f.** Jaki wkład własny można wprowadzić do obszaru ataków DDoS? Zaproponowaną w pracy [5] klasyfikację również dzisiaj można uważać za kompletną. W zasadzie, obejmuje ona wszystkie znane

typy ataków, w tym również te, które się jeszcze nie pojawiły, lecz ich pojawienie jest realne. Klasyfikacja części ochronnej zawiera nie tylko znane podejścia, ale również możliwość stworzenia których jest dostatecznie udokumentowana i mogą być one poddane stosownej analizie. Tematu klasyfikacji ataków DDoS dotyczy również szereg innych prac [6], [2], [7], [8], [9], [10].

Z informatycznego punktu widzenia, najciekawsze są ataki na warstwę sieci. Ataki tej klasy zostały dopracowane i zbadane najlepiej spośród wszystkich pozostałych. Mogą one sparaliżować ruch w sieci, przenikać połączenia pomiędzy nadawcą i odbiorcą informacji, przekierowywać pakiety, nierzadko na nieistniejące ścieżki, tworzyć cykle w ścieżkach, przeciążając w ten sposób wybrane obszary sieci. Ataki realizowane na warstwę sieci, nieco bardziej szczegółowo, można podzielić na następujące rodzaje: **a.** Ataki określenia lokalizacji – napastnik gromadzi informacje dotyczące lokalizacji węzłów sieci, organizacji sieci i tablic i protokołów trasowania, a następnie na ich podstawie planuje atak; **b.** Ataki na przeciążenie tablic trasowania – atakujący doprowadza do wykorzystania ścieżek bazujących na nieistniejących węzłach, co jest dopuszczalne jeżeli algorytm trasowania periodycznie aktualizuje informacje o trasowaniu. Zadanie intruza polega na utworzeniu ścieżek utrudniających funkcjonowanie innych tras przesyłania informacji; **c.** Ataki na wybrane protokoły trasowania mające na celu zmianę trasy przesyłania pakietów; **d.** Atak typu *zużycie zasobów* – atakujący węzeł zmniejsza czas życia wybranych komponentów systemu, np. pamięci SSD poprzez realizacją częstych odwołań do nich; **e.** Atak typu *zatrucie cache trasowania* – do tablicy trasowania wprowadzane są błędne dane przekierowujące ruch interesujący atakującego; **f.** Ataki *fazy przesyłania danych* – wbrew danym tablicy trasowania węzły podmiotu atakującego biorą udział w transmisji danych. W trakcie przemieszczania się pakietu przez węzeł, jego zawartość może być zmodyfikowana lub zniszczona; **g.** Ataki *fazy poszukiwania trasy* – atakujący stara się zestawić trasę przesyłania informacji, niezgodną z protokołem trasowania; **h.** Atak *fazy obsługi trasowania* – intruz generuje nieprawdziwe informacje o stanie kanału komunikacyjnego; **i.** Atak wykorzystujący *tunelowanie* – para intruzów, korzystając z tunelowania i translacji pakietów w tunelu dokonuje ataku na środki komunikacji systemu. Wszystkie powyższe ataki, w mniejszym lub większym stopniu prowadzą do utrudnień w świadczeniu usług przez system informacyjny, w tym serwer webowy.

Skuteczne ataki mogą być realizowane również na innych poziomach modelu ISO, w tym na poziomach transportowym i aplikacyjnym. Zadaniem protokołów warstwy transportowej jest nawiązywanie stałych, gwarantowanych połączeń pomiędzy nadawcą i odbiorcą informacji, wraz z zarządzaniem strumieniami i przeciążeniami. Najczęstsze ataki na tym poziomie wiążą się działaniami mającymi naruszyć synchronizację komunikacji. Z racji niedoskonałości kanału komunikacyjnego, na ataki tej klasy szczególnie podatne są sieci bezprzewodowe. Innym rodzajem ataków na poziomie warstwy transportowej jest przechwycenie sesji. Intruz ukrywa się pod adresem IP ofiar korzystając z sesji z interesującym go węzłem.

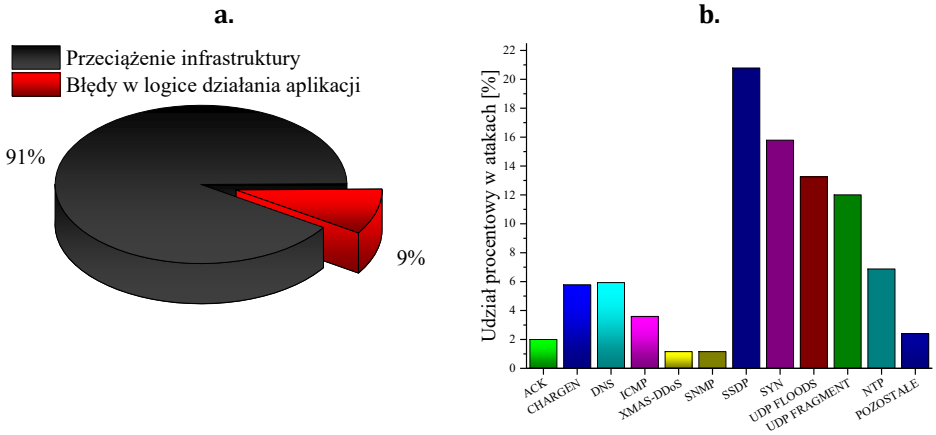
Ataki na dostępność zasobów systemu operacyjnego mogą być przeprowadzane również w oparciu o warstwę aplikacji modeli ISO. Zauważmy, że na poziomie tym zaimplementowane są protokoły charakteryzujące się wysoką podatnością na ataki, takie jak SMTP i HTTP oraz rzadziej używane TELNET i FTP. Ataki mogą być realizowane za pomocą programów złośliwych, jak również poprzez przeciążanie wybranych fragmentów infrastruktury obliczeniowej lub komunikacyjnej. Profesjonalne ataki na dostępność mają zazwyczaj charakter wielopoziomowy i mogą składać się zarówno z jednoczesnych jak i rozłożonych w czasie działań. Od kilkunastu lat stosowane są rozproszone ataki ze wzmocnieniem DRDoS (ang. *Distributed Reflected Denial of Service*), które dzięki zastosowaniu komunikacji rozgłoszeniowej potęgują siłę ataku.

Do najbardziej znanych zrealizowanych ataków DDoS możemy zaliczyć przeprowadzone w 2000 roku uderzenie w komercyjne portale Yahoo, AoL i CNN, które całkowicie zablokowało dostęp do obsługujących je serwerów. Mimo upływu lat, na dzień dzisiejszy nie udało się opracować skutecznych zabezpieczeń przed atakami tej klasy. W znacznej mierze wynika to z faktu, że do przeprowadzenia tego typu ataków nie jest wymagany specjalistyczny sprzęt ani obszerna wiedza, a ich realizacja wymaga znikomych zasobów programowo-sprzętowych i kapitału.

2.2. Statystyki ataków przeciążeniowych

Częstotliwość ataków odmowy usługi z każdym rokiem wzrasta. Według raportu *Prolexic DDoS Attack Raport* za pierwszy kwartał 2015 roku, zanotowano wzrost ataków DDoS o 35 procent, w porównaniu z czwartym kwartałem 2014 roku [11]. Średnia długość ataku DDoS wynosiła 24.82 godzin, co stanowi 15,37 procentowy spadek w porównaniu z czasem trwania incydentu w czwartym kwartale 2014 (29.33 godzin) i wzrost 42,8 procent w porównaniu z czasem trwania ataku w pierwszym kwartale 2014 r. Pomimo wzrostu możliwości infrastruktury potencjalnych ofiar, dominującym celem ataków w badanym okresie była infrastruktura co pokazano na poniższym wykresie.

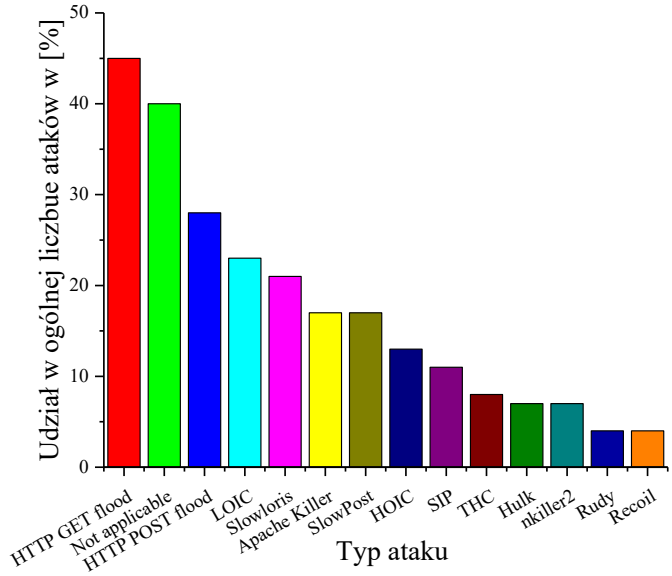
Według raportu [11] najczęściej atakowanym protokołem był protokół SSDP co stanowi wzrost o 15% w stosunku z okresem poprzedni. W głównej mierze atakowany protokół dotyczył urządzeń użytkowników domowych. Na poniższym wykresie zaprezentowano częstotliwości występowania ataków DDoS w warstwie infrastruktury.



Rys. 3. Statystyki ataków typu DDoS w I kw. 2015 roku: a. Procentowy udział poszczególnych celów; b. Udział procentowy różnych protokołów

Zdecydowana większość przeprowadzanych ataków była oparta na bazie ataków infrastruktury natomiast ataki warstwy aplikacji stanowiły mniej niż 10%. Trend ukierunkowania ataków w stronę infrastruktury trwa dłużej niż jeden rok i spowodowany jest wykorzystaniem mniejszych zasobów potrzebnych do przeprowadzenia takiego ataku niżeli jest to w przypadku ataków na aplikacje. W przypadku ataków na warstwę aplikacji obserwuje się tendencje wykorzystania otwartych serwerów proxy w Internecie. Spowodowane jest to nadużywaniem i niepoprawną konfiguracją systemów webowych opartych o WordPress i Joomla [11]. Najczęstszym atakiem na warstwę aplikacji nadal pozostaje HTTP GET floods pomimo, że część respondentów odnotowała spadek o 20% incydentów tego typu w porównaniu z zeszłym rokiem [12]. Na rys. 4 zaprezentowano rozkład ataków na warstwę aplikacji.

Prognozy ograniczenia zjawiska na najbliższe lata nie są optymistyczne. Przewiduje się, że znaczenie incydentów mających na celu paraliż infrastruktury oraz aplikacji będzie niezmiennie wzrastało. Głównych przyczyn tego trendu należy się dopatrywać zarówno w prostocie przeprowadzenia samego ataku jak i jego skuteczności przy jednoczesnym skomplikowanym i kosztownym procesie obrony. Dlatego należy dobierać mechanizmy obrony jak i wybierać systemy wykazujące mniejszą podatności na ataki tego typu.

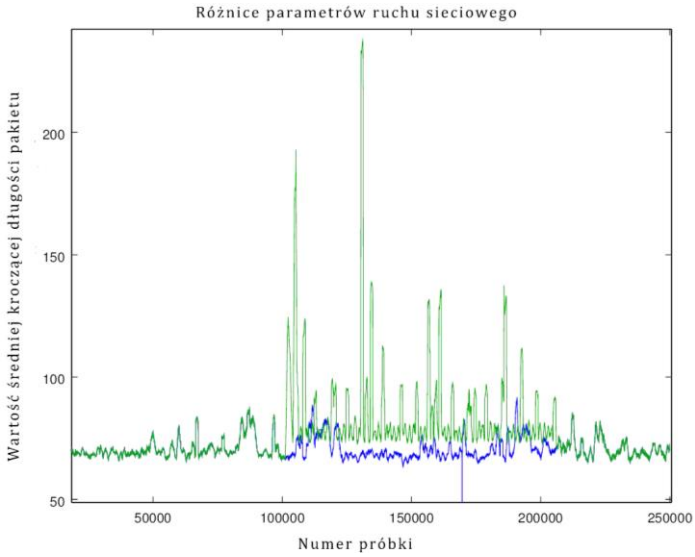


Rys. 4. Procentowy rozkład zidentyfikowanych zagrożeń na warstwę aplikacji w 2014 [12]

3. Wykrywanie ataków *odmowy usługi* w ruchu sieciowym

Oczywiście pomimo trudności w ograniczeniu zjawiska należy podjąć kroki, które mogą do niego mimo wszystko doprowadzić. Jedną z obiecujących strategii jest statystyczna analiza ruchu sieciowego na różnych poziomach infrastruktury. Strategia ta ma rację bytu w związku z widocznymi zmianami wybranych parametrów statystycznych ruchu już krótką chwilę po rozpoczęciu ataku.

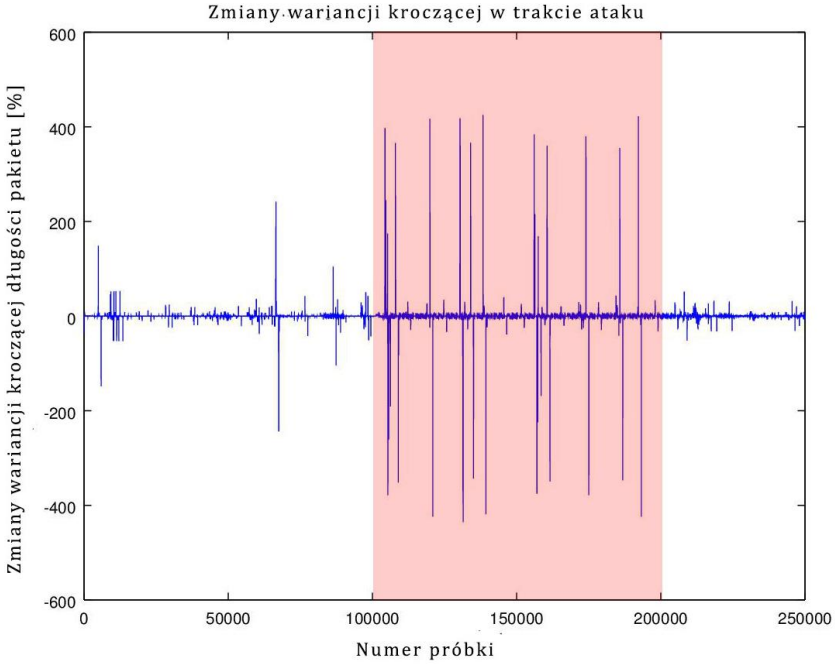
Użycie metod statystycznych wydaje się tutaj kluczowe – ponieważ pozwala na wykrycie potencjalnego zagrożenia w sposób automatyczny, bez potrzeby czasochłonnej analizy każdego pakietu docierającego do urządzenia. Również ze względów związanych z ochroną prywatności internautów metody tego rodzaju wydają się przyszłością, jako że nie dokonują żadnego sprawdzenia zawartości pakietu, bazują natomiast na parametrach takich jak np. jego długość czy rodzaj.



Rys. 5. Różnice pomiędzy typowym ruchem sieciowym (linia niebieska) a trwającym atakiem typu slowloris (linia zielona). Atak zaczyna się przy próbce numer 100000.

Aby przybliżyć podstawy metod tej klasy, na kolejnych stronach omówionych zostało kilka wniosków płynących z badań nad ich skutecznością. I tak rys. 6 przedstawia porównanie średniej kroczącej długości pakietu dla danych spotykanych typowo w infrastrukturze (linia niebieska) oraz tych samych danych, do których wstrzyknięto pakiety pochodzące z ataku typu slowloris (linia zielona). Obliczenie średniej kroczącej, przy zastosowaniu odpowiedniego algorytmu, ma liniową złożoność obliczeniową, a więc nie wprowadza zbyt wiele dodatkowych opóźnień. Co więcej, przesuwanie okna czasowego, w którym została ona obliczona, jest jeszcze mniej kosztowne – gdyż przy każdej takiej operacji wystarczy niewielka korekta poprzedniego wyniku.

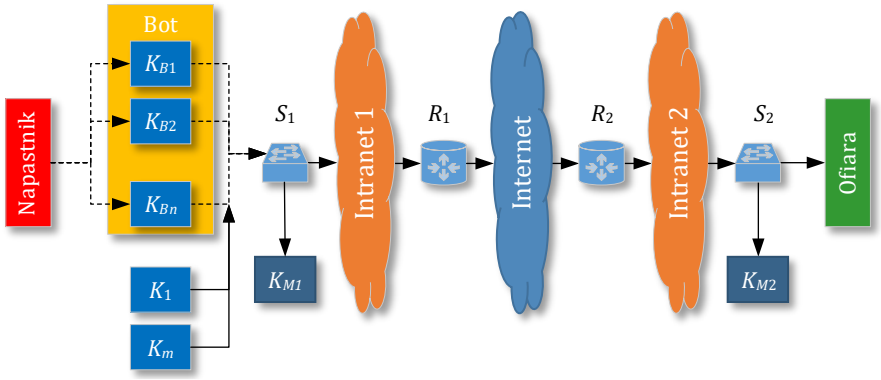
Porównując dwa wykresy z rys. 5 możemy łatwo wyciągnąć wnioski co do różnicy obu zestawów danych. Co jednak, jeśli nie mamy osobnego zbioru odniesienia? Okazuje się, że również tutaj metody statystyczne radzą sobie całkiem dobrze. Wystarczy analizować procentowe zmiany parametrów pomiędzy kolejnymi oknami czasowymi i różnice pomiędzy typowym a zaburzonym ruchem stają się jeszcze bardziej oczywiste, co obrazuje rys. 6 [13].



Rys. 6. Procentowe zmiany wariancji kroczącej pomiędzy kolejnymi krokami analizy w skali procentowej. Na czerwono zaznaczono czas występowania ataku

Przedstawione powyżej dane to procentowe zmiany wariancji długości pakietów, trafiających na urządzenie sieciowe. Mimo że gwałtowne zmiany zdarzają się również w przypadku typowego ruchu, nie trzeba wprawnego oka doświadczonego administratora, żeby zauważyć, że zaznaczony czerwonym kolorem okres, w którym trwał atak wyróżnia się ewidentnie regularnością i skalą zmian. A skoro do wyciągnięcia takiego wniosku nie trzeba wiedzy i doświadczenia, można tą czynność zostawić oprogramowaniu, które o zdarzeniu poinformuje obsługę urządzenia.

W celu oceny efektywności zaproponowanych poniżej metryk służących wykrywaniu ataków, zbudowano system pomiarowy, którego struktura została przedstawiona na rys. 7. Podstawą przeprowadzenia ataku jest bot o zmiennej liczbie komponentów. Z doświadczenia autorów wynika, że skuteczny atak może być przeprowadzony już za pomocą 6-8 jednostek.



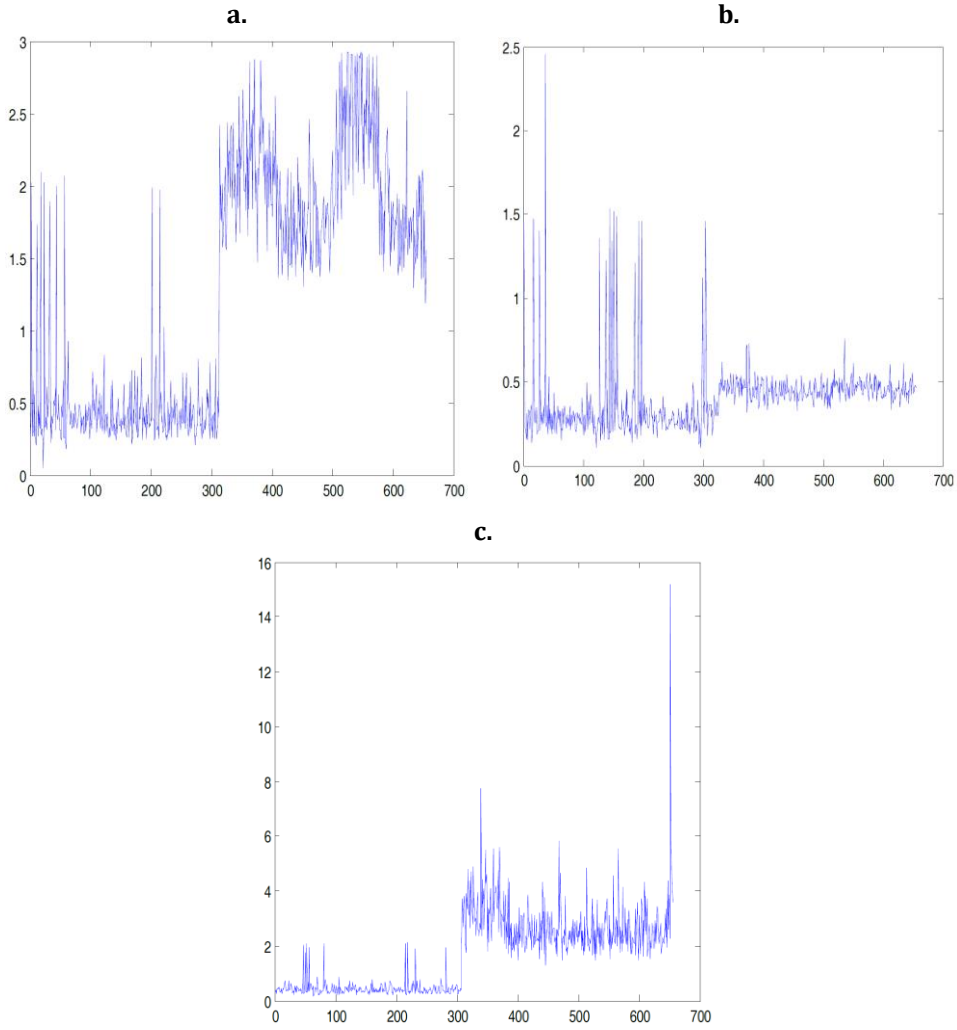
Rys. 7. Konfiguracja eksperymentalnej sieci badawczej

W calu przenalizowania metody zbudowano eksperymentalną sieć badawczą, która składa się z dwóch segmentów intranetowych, połączonych ze sobą za pośrednictwem sieci Internet. Dołączenie Intranetów do Internetu wykonywane jest za pomocą pary routerów R_1 oraz R_2. Z kolei, dołączenie do Intranetów użytkowników końcowych lub ich grup, bazuje na przełącznikach S_1 i S_2. Fragmenty sieci z napastnikiem i ofiarą są śledzone za pomocą komputerów monitorujących, odpowiednio: K_{M1} i K_{M2}, na których zainstalowano oprogramowanie Wireshark. Jednostki komputerowe $K_1, \cdots, K_m$ komunikują się z ofiarą, ale łączność ta nie nosi cech ataku. Atakujący bot tworzą jednostki komputerowe $K_{B1}, \cdots, K_{Bn}$, w eksperymentach wykorzystywano boty o rozmiarze od kilku do kilkudziesięciu komputerów. W zależności od rodzaju eksperymentu, *Ofiarą* jest stacja robocza, serwer aplikacyjny bądź serwer webowy.

Wśród części administratorów bezpieczeństwa panuje przekonanie, że istotna zmiana wielkości ruchu sieciowego z dużym prawdopodobieństwem świadczy o ataku na dostępność. Z teorii i praktyki sieci komputerowych wiemy, że stosunek pomiędzy ruchem przychodzącym i wychodzącym, nawet w okresie krótkoterminowym, jest wielkością względnie stałą, zależną poziomu hierarchii sieci Internet, na której zlokalizowany jest analizowany element. Dla poziomu liści grafu-modelu (użytkownicy końcowi) stosunek ten znajduje się w granicach 20:1, dla serwerów sieci lokalnych 6:1, a komputerów rdzenia sieci regionalnych i korporacyjnych 3:1 [3], [14], [15], [16]. Wiedzą o charakterze powszechnym jest również istnienie w sieci ruchu o charakterze wybuchowym, którego współczynnik wybuchowości przekracza kilkanaście, czasami kilkadziesiąt. Tak więc, sam fakt wzrostu komunikacji na łączu zewnętrznym jednostki komputerowej nie świadczy o ataku. Zauważmy, że od pewnego momentu, atak to wyłącznie ruch przychodzący, nie będący odpowiedzią na jakiekolwiek zapytanie wychodzące z atakowanego elementu. Moment ten pojawia się w sytuacji, kiedy serwer lub stacja robocza nie jest w stanie odpowiadać na zapytania tj., kiedy cel ataku został osiągnięty.

W świetle, powyższej argumentacji, znacznie lepszym wskaźnikiem pojawienia się ataku DDoS jest wartość ilorazu $k_{i/o}$ ruchu wejściowego λ_i i wyjściowego λ_o

$\left(k_{i/o} = \lambda_i/\lambda_o\right)$. Jeżeli atak jest prawdopodobny, znaczenie $k_{i/o}$ wzrasta, bowiem wzrostowi ruchu wejściowego nie towarzyszy odpowiedni wzrost ruchy wyjściowego. Zmiany wartości parametru $k_{i/o}$ w czasie dla ataku na serwer webowy pokazano na rys. 8.



Rys. 8. Zmiana wartości parametru $k_{i/o}$ dla różnych typów ataku: a. SYN-flood; b. HTTP-flood; c. UDP-flood

Innym skutecznym wskaźnikiem ilorazowym jest stosunek $k_{UDP/TCP}$ sumarycznej objętości dwu klas ruchu wchodzącego: Λ_{UDP} dla ruchu UDP i odpowiednio Λ_{TCP} dla TCP.

Oczywiście opisane metody to nowość i w tej chwili nie są jeszcze implementowane, ale doskonale ukazują trend w jakim zmierza nauka w związku z konieczno-

ścią obrony przed atakami „odmowy usługi". Ponadto wiedza o znacznych różnicach w statystycznych parametrach ruchu typowego i zaburzonego może dać świadomemu administratorowi szansę na wychwycenie anomalii na podstawie statystyk obecnie generowanych na urządzeniach, do których może mieć dostęp. Oczywiście pod warunkiem, że będą traktowane bardziej jako użyteczne narzędzie diagnostyczne a mniej jako ciekawostka.

4. Podatność serwerów webowych na ataki DDoS

Korzystając z eksperymentalnej sieci badawczej zaprezentowanej na rys. 7 porównano podatność serwerów webowych na ataki typu DDoS. Masowe wykorzystanie usług informacyjnych bazujących na serwerach webowych spowodowało, że zapewnienie ich ciągłej dostępności staje się kluczowym zadaniem projektowania systemów tego typu. Przedsiębiorca, mając do wyboru prywatny serwer rozlokowany we własnej infrastrukturze bądź usługi hostingowe, decydując się na wybór jednego z rozwiązań powinien uwzględniać również parametry opisujące dostępność. Dla większości badaczy, dostępność to procentowy udział czasu poprawnego funkcjonowania systemu, w ogólnym czasie jego działania [1], [3], [16]. Zazwyczaj, charakterystyki dostępności usług hostingowych są wyższe od tych które można osiągnąć samodzielnie eksploatując sprzęt komputerowy [15], chociaż zdarzają się również wyjątki od tej zasady. Ataki typu DDoS są szczególnym rodzajem ingerencji w systemy informacyjne, a skuteczność ochrony przed nimi jest w znacznej mierze uzależniona od doświadczenia personelu i posiadanych urządzeń diagnostycznych. W przypadku usług hostingowych obie powyższe charakterystyki prezentują się znacznie korzystniej, z drugiej jednak strony, prawdopodobieństwo ataku na strony usługodawcy będzie wyższe [9].

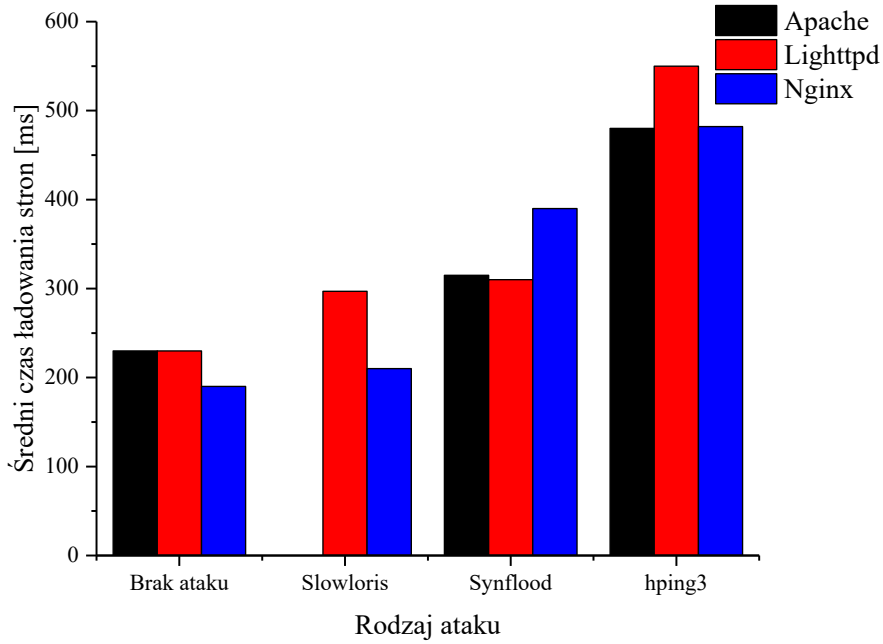
4.1. Analiza podatności serwerów web na przykładowe ataki

W przedstawionej analizie podatności wykorzystano trzy rodzaje ataków tj. Slowloris, Synflood oraz hping3 flood. Poniżej przedstawiono ich charakterystykę:

- Slowloris - atak działa na bazie wykorzystania niedoskonałości protokołu HTTP jaki i sposobu obsługi żądań. Podczas ataku wysyłane są żądania HTTP GET, które potrafią całkowicie sparaliżować dostęp do serwisu.
- Synflood – atak realizowany jest poprzez zainicjowanie dużej ilości połączeń TCP z nieprawidłowym adresem źródłowym dla pakietu SYN.
- hping3 flood – jest darmowym generatorem pakietów i analizatorem protokołu TCP/IP.

Analizowane ataki wyróżniały się różną skutecznością. Najskuteczniejszym atakiem w odniesieniu do serwera Apache był atak *Slowloris*, który całkowicie sparaliżował dostęp do zasobów serwera. Natomiast pozostałe serwery wykazywały większą odporność na atak *Slowloris* gdzie wpływ na ich wydajność był znikomy. Pozostałe zaproponowane ataki odznaczały się większą uniwersalnością. Poniższy

wykres przedstawia średnie czasy ładowania stron dla zaatakowanych trzech serwerów.



Rys. 9. Porownanie czasów ładowania stron w trakcie atakow

Podczas realizacji ataku Synflood średnie czasy ładowania stron wzrosły dla każdego z serwerów (najbardziej dla Nginx), jednak najwyższy wzrost odnotowano w czasie ataku flood wykonanego za pomocą hipng3. W sytuacji nawet nieznacznego zwielokrotnienia ataków mogłyby one całkowicie sparaliżować działanie serwera.

5. Podsumowanie

Intensywny rozwój technik ICT oraz działania podjęte przez samorządy w celu eliminacji wykluczenia cyfrowego spowodowały wzrost oczekiwań mieszkańców odnośnie zwiększenia oferty usług urzędniczych załatwianych za pośrednictwem Internetu. Wiąże się to z powstawaniem nowych usług, aplikacji, a w konsekwencji może spowodować zwiększenie podatności projektowanych systemów na zagrożenia realizowane przez ataki odmowy usługi. Dlatego tak istotnym jest opracowanie metod służącym zarówno do analizowania zagrożeń jak i umożliwionych testowanie systemów na etapie produkcyjnym. Zaproponowana przez autorów metoda statystycznej analizy ruchu sieciowego pozwala spełnić powyższe cele.

Przedstawione wyniki sugerują, że przy niewielkim nakładzie obliczeniowym można w prosty sposób wykrywać ataki typu slowloris w rzeczywistych danych sieciowych. Wybrane algorytmy mają bardzo niewielką złożoność obliczeniową, co

sprawia, że wykrywanie nie powinno wnosić dużego overheadu do funkcjonowania urządzenia. Dodatkowym atutem zaprezentowanych metod jest fakt, że implementacja tego rodzaju operacji nie wymaga specjalizowanych bibliotek matematycznych, a jedynie podstawowych narzędzi programowych. Jako główny parametr analizy brana jest długość pakietu w bajtach, co sprawia, że do sprawnego funkcjonowania zaproponowanej metodologii wystarczy szybka analiza nadchodzących danych bez potrzeby czytania zawartości pakietu. Fakt, że w podsieci o 30 klientach udało się wykryć pojedyncze źródło ataku jest dobrym argumentem przemawiającym za praktyczną przydatnością opracowanej metody.

Bibliografia

[1] D. L. Spohn, *Data Network Design*. New York: McGraw-Hill, 2002.

[2] R. J. Barnett and B. Irwin, "Towards a taxonomy of network scanning techniques," in *Proceedings of the 2008 annual research conference of the South African Institute of Computer Scientists and Information Technologists on IT research in developing countries: riding the wave of technology*, Wilderness, South Africa, 2008, pp. 1-7.

[3] M. Hajder, H. Loutskii, and W. Stręciwilk, *Informatyka. Wirtualna podróż w świat systemów i sieci komputerowych.*, I ed., Mirosław Hajder, Ed. Rzeszów, Polska: Wydawnictwo Wyższej Szkoły Informatyki i Zarządzania, 2002.

[4] Khan Zeb, Baig Owais, and Kamran Asif Muhammad, "2nd World Symposium on Web Applications and Networking," in *DDoS attacks and countermeasures in cyberspace*, Dubai, 2015.

[5] J. Mirkowvic and P. Reiher, "A taxonomy of DDoS attack and DDoS defense mechanisms," *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 2, pp. 39-53, Apr. 2004.

[6] R. A. Rodríguez-Gómez, G. Maciá-Fernández, and P. García-Teodoro, "Survey and taxonomy of botnet research through life-cycle," *ACM Comput. Surv.*, vol. 45, no. 4, pp. 1-33, Apr. 2013.

[7] A. Keshariya and N. Foukia, "DDoS defense mechanisms: a new taxonomy," in *Proceedings of the 4th international workshop, and Second international conference on Data Privacy Management and Autonomous Spontaneous Security*, St. Malo, France, 2010, pp. 222-236.

[8] A. V Uzunov and E. B. Fernandez, "An extensible pattern-based library and taxonomy of security threats for distributed systems," *Comput. Stand. Interfaces*, vol. 36, no. 4, pp. 734-747, Apr. 2014.

[9] G. Mantas, N. Stakhanova, H. Gonzalez, H. H. Jazi, and A.A. Ghorbani, "Application-layer denial of service attacks: taxonomy and survey," *Int. J. Inf. Comput. Secur.*, vol. 7, no. 2,3,4, pp. 216-239, Apr. 2015.

[10] A. Shameli-Sendi, M. Fekih-Ahmed, and M. Cheriet, "Taxonomy of Distributed Denial of Service mitigation approaches for cloud computing," *J. Netw. Comput. Appl.*, vol. C, no. 58, pp. 165-179, 2015.

[11] Akamai, "Q1 2015 State of the Internet – Security Report," 2015.

[12] Arbor Networks, "Worldwide Infrastructure Security Report," 2014.

[13] Michał Wroński, Mariusz Nycz, and Janusz Kolbusz, "Computer Science for the Information Society," in *Characteristic of Linux package dependencies as complex network-an empirical study*, Severodonetsk, 2014.

[14] M. Hajder, "Architektura wielopoziomowej sieci komunikacyjnej," in *Studium rozwoju sieci szerokopasmowej dla województwa Podkarpackiego na lata 2007-2013*, M. Stroński, Ed. Poznań: Poznańskie Centrum Superkomputerowo-Sieciowe, 2008, ch. II, pp. 11-50.

[15] M. Hajder and M. Nycz, *Sieci regionalne i korporacyjne. Metody projektowania i eksploatacji*. Rzeszów: Wyższa Szkoła Informatyki i Zarządzania z siedzibą w Rzeszowie, 2015.

[16] M. Hajder, J. Kolbusz, and M. Nycz, *Algorytmika projektowania systemów informacyjnych*. Rzeszów: Wyższa Szkołą Informatyki i Zarządzania z siedzibą w Rzeszowie, 2014.

[17] (2016, Oct.) Systemy operacyjne. [Online]. http://ranking.pl/pl/rankings/operating-systems.html

[18] M. Hajder, "Rekonfiguracja i rozbudowa sieci szerokopasmowej," in *Studium rozwoju sieci szerokopasmowej dla województwa Podkarpackiego na lata 2007-2013*, M. Stroiński, Ed. Poznań: Poznańskie Centrum Superkomputerowo-Sieciowe, 2008, ch. VII, pp. 203-221.

[19] William Stallings, *Kryptografia i bezpieczeństwo sieci komputerowych. Koncepcje i metody bezpiecznej komunikacji*. Gliwice: Helion, 2012.

[20] Leon Shklar and Richard Rosen, *Web Application Architecture*. West Sussex: John Wiley & Sons, 2003.

[21] Clément Nedelcu, *Nginx HTTP Server*. Birmingham: Packt Publishing Ltd., 2013.

[22] Khan Zeb, Owais Baig, and Muhammad Kamran Asif, "2015 2nd World Symposium on Web Applications and Networking," in *DDoS attacks and countermeasures in cyberspace*, Sousse, 2015.

[23] ITU - Committed to connecting the world. [Online]. http://www.itu.int/en/ITU-D/Statistics/Pages/stat/default.aspx

[24] (2015, Nov.) Netcraft. [Online]. http://news.netcraft.com/archives/2015/11/16/november-2015-web-server-survey.html

ROZDZIAŁ 8

Etap rekonfiguracji w cyklu życia systemów informatycznych

Mirosław HAJDER, Piotr HAJDER

Wyższa Szkoła Informatyki i Zarządzania z siedzibą w Rzeszowie
Akademia Górniczo-Hutnicza im Stanisława Staszica w Krakowie
Miroslaw.Hajder@gmail.com, Piootr.Hajder@gmail.com

Streszczenie

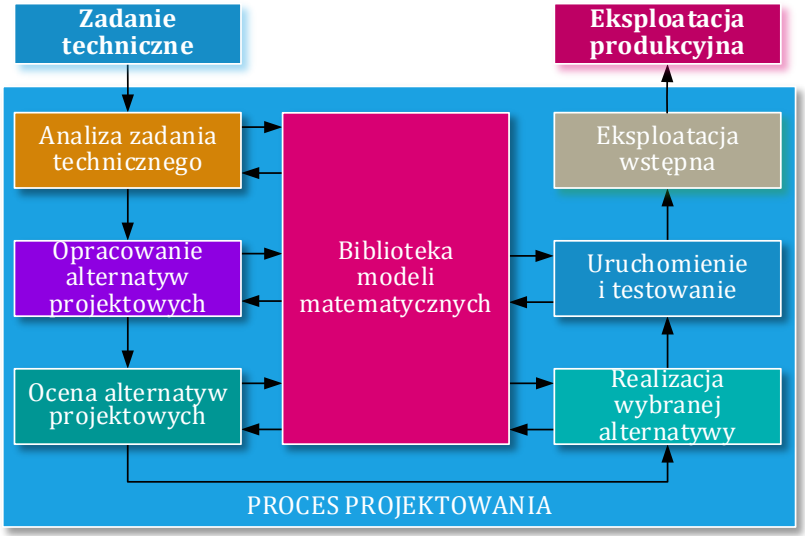
Rozdział jest analizą zagadnień pośrednio związanych z projektowaniem obiektów i systemów technicznych. Rozpoczyna się on od określenia projektowania jako procesu oraz porównania różnych jego definicji. Dalej, projektowanie powiązano z cyklem życia obiektu lub systemu technicznego, wskazując udział projektowania w zapewnieniu efektywności przebiegu cyklu życia. Następnie, omówiony składniki procesu projektowania i ich wzajemne relacje oraz powiązania z etapami cyklu życia obiektu lub systemu. Rozdział adresowany jest do osób zajmujących się zarządzaniem procedurami projektowania obiektów i systemów technicznych.

1. Etap projektowania cyklu życia

Każdy obiekt techniczny, w tym również system informatyczny lub sieć komputerowa mogą znajdować się w różnych fazach swojego istnienia. Ponieważ jednym z podstawowych celów każdego projektowania jest maksymalizacja czasu efektywnej eksploatacji zrealizowanego projektu, zagadnienie to doczekało się dogłębnej analizy matematycznej. Terminem, który został do tego celu stworzony jest cykl życia projektu. W ogólnym przypadku cykl życia przedstawia wszystkie fazy istnienia obiektu i określany jest za pomocą modelu, a opisywany w postaci metodologii. Model lub paradygmat cyklu życia określa koncepcyjne spojrzenie na organizację cyklu, podstawowe jego fazy, a także zasady przejścia pomiędzy nimi. Cykl życia włącza wszystkie fazy, od momentu podjęcia decyzji o realizacji obiektu do momentu zakończenia jego istnienia (tj. jego utylizacji). Przejścia od jednego etapu do innego etapu cyklu życia rzadko kiedy są wyraźnie określone. Czytelne rozgraniczenie etapów ma miejsce wyłącznie wtedy, kiedy etap kończony jest przyjęciem pewnej propozycji ekonomiczno-technicznej lub podjęciem decyzji o kontynuacji cyklu życia w następnym jego etapie.

Cykl życia oprogramowania określa norma ISO/IEC 12207 oraz stworzony na jej podstawie standard IEEE 12207. Dokumenty te opisują model cyklu życia jako strukturę składającą się z procesów, prac i zadań włączających opracowanie, eksploatację i obsługę oprogramowania. W chwili obecnej brak aktualnego dokumentu opisującego model cyklu życia architektury sprzętowej systemu informatycznego. Dostępne obecnie dokumenty z tego tematu datowane są na początek lat 90-tych. Dalej jako przykład rozważać będziemy regionalne sieci komputerowe, które z punktu widzenia modernizacji są obiektami szczególnie kłopotliwymi. Z racji swojego rozmiaru rozmieszczane są one na obszarach niebędących własnością ich operatora. Z drugiej strony ich regionalny charakter ogranicza przychody operatora, który zazwyczaj jest małym lub średnim przedsiębiorstwem.

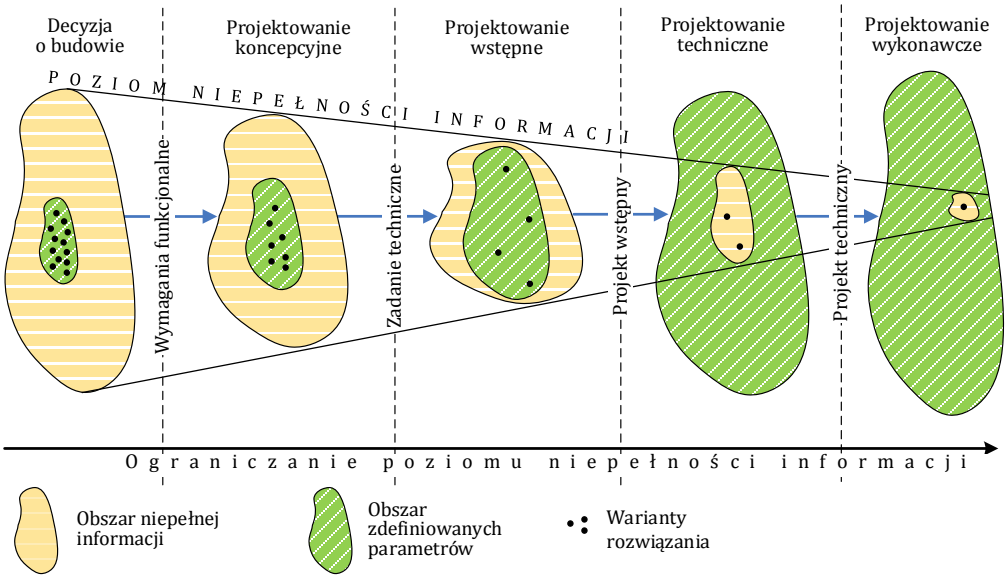
Na rys. 1 przedstawiono komponenty funkcjonalne fazy projektowej cyklu życia obiektu lub systemu.



Rys. 1. Komponenty funkcjonalne fazy projektowej cyklu życia obiektu lub systemu

Rozpoczyna się on od zdefiniowania zadania technicznego, które precyzyjnie określa oczekiwania użytkownika. Najważniejszym elementem procesu projektowania jest uniwersalna Biblioteka modeli matematycznych zawierająca elementy o różnym stopniu abstrakcji – od najprostszych do najbardziej złożonych. Kompleksowy system komputerowy wspomaga pracę projektanta, począwszy od krytycznej analizy zadania technicznego po uruchomienie i testowanie projektowanego obiektu lub systemu. Chociaż systemy o podobnej funkcjonalności istnieją, mają one bardzo ograniczony zakres działania tj. dotyczą projektowania wyłącznie w obszarze konkretnej branży.

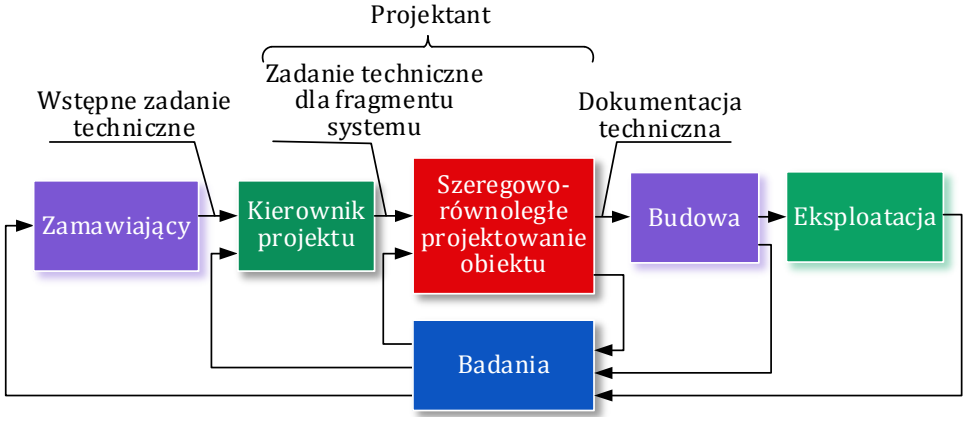
Proces projektowania można traktować jako sukcesywne zmniejszanie poziomu niewiedzy o projektowanym obiekcie bądź systemie. Idea takiego podejścia została pokazana na rys. 2.



Rys. 2. Projektowanie jako proces zmniejszania niewiedzy o obiekcie lub systemie

Wraz z przechodzeniem kolejnych kroków procedury (decyzja o budowie, projektowanie koncepcyjne, wstępne, techniczne i wykonawcze) zmniejsza się obszar niewiedzy o obiekcie, a tym samym i ilość alternatyw projektowych. Z informatycznego punktu widzenia mówimy o zawężaniu się obszaru poszukiwania rozwiązania zadania projektowego. Zakończenie procesu projektowania nie oznacza, że wiedza o obiekcie jest doskonała, a raczej ograniczenie niewiedzy do akceptowalnego poziomu.

Interesującym jest spojrzenie na cykl życia z punktu widzenia modelu kompetencyjnego, co pokazano na rys. 3.



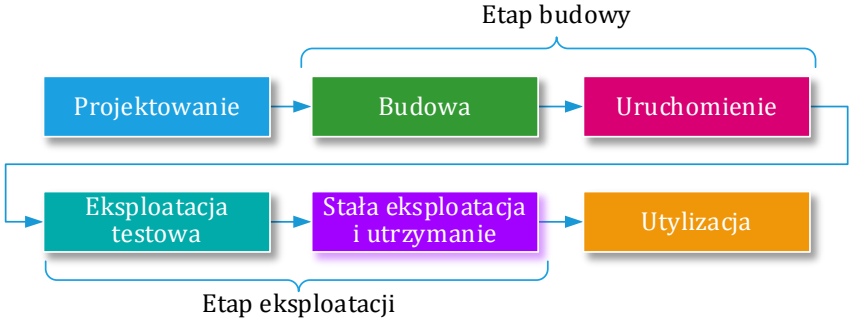
Rys. 3. Kompetencyjny model cyklu życia obiektu

Model ten zakłada, że na każdym z etapów cyklu życia prowadzone są badania, których podstawowym celem jest dostarczenie projektantowi i zamawiającemu

wiedzy o efektach prowadzonych działań. Niestety, propozycje zawarte na obu rysunkach dotyczące wykorzystania technicznych metod projektowania są w rzeczywistości wyłącznie hipotetyczne. Zazwyczaj projektowanie opiera się na intuicji projektanta lub co gorsza – zamawiającego.

2. Tradycyjne podejście do cyklu życia

Rozważmy tradycyjne podejście do cyklu życia sieci regionalnej. Rozpoczyna się ono od podjęcia decyzji o jej budowie. Decyzja taka powinna być poprzedzona analizą aktualnej dostępności sieci komputerowych oraz potrzeb w tym zakresie. Budowa jest uzasadniona tylko w tym przypadku, kiedy istniejące lub planowane do realizacji inwestycje nie spełniają rosnących potrzeb użytkowników. Pierwszym realnym etapem cyklu życia jest projektowanie sieci składające się z określenia zadania technicznego, projektowania wstępnego, technicznego oraz wykonawczego. W ich rezultacie powstaje gotowy do realizacji projekt, spełniający aktualne wymagania użytkowników. Następnym etapem cyklu jest budowa sieci włączająca dodatkowo podetap testowania zbudowanej struktury. Dzięki wprowadzeniu do etapu budowy podetapu testowania użytkownik otrzymuje funkcjonujący obiekt będący odzwierciedleniem jego potrzeb. Kolejny etap to eksploatacja obejmująca eksploatację wstępną i stałą z uwzględnieniem utrzymania sieci. Cykl życia sieci kończy utylizacja. Etapy cyklu życia tradycyjnej sieci przedstawiono na rys. 4.



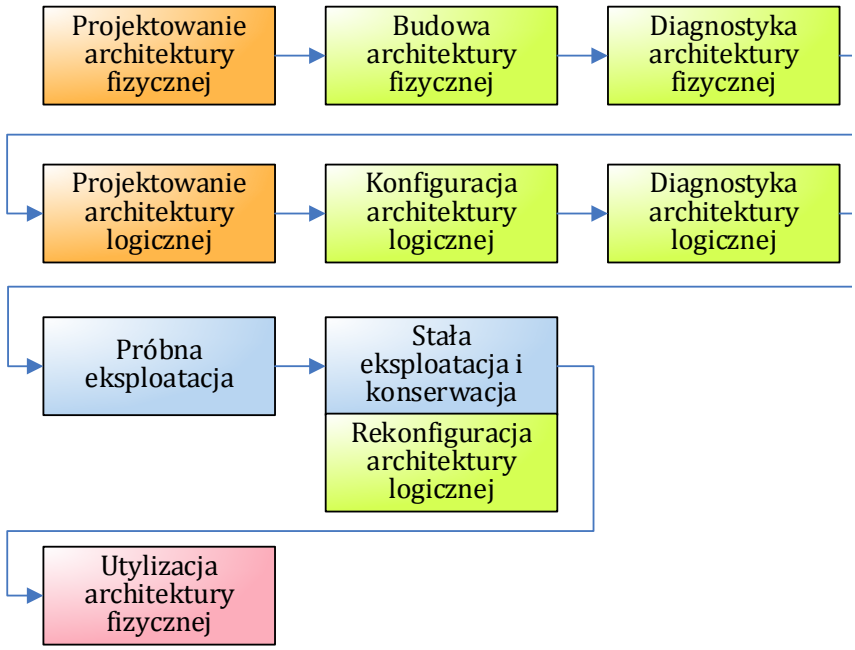
Rys. 4. Etapy cyklu życie sieci regionalnej

Zastosowanie schematu cyklu życia przedstawionego na rys. 4 do celów syntezy i analizy nowoczesnych sieci bazujących na wielokanałowości i hierarchiach topologii, przeznaczonych m. in. do obsługi Internetu rzeczy jest nieefektywne. Jest to konsekwencją następujących podstawowych uwarunkowań:

1. W nowoczesnych sieciach projektowane i eksploatowane są dwie niezależne sieci: fizyczna i logiczna, które na etapie budowy nie są jawnie rozróżniane;
2. W celu poprawy jakości funkcjonowania sieci szerokopasmowej na etapie jej eksploatacji pojawia się konieczność rekonfiguracji połączeń logicznych, wykonywanej przy niezmiennych połączeniach fizycznych sieci.

Dlatego schemat cyklu życia przedstawiony na rys. 4 powinien zostać przynajmniej zmodyfikowany do postaci zaprezentowanej na rys. 5.

Zauważmy, że stworzenie architektury logicznej nie jest związane z wykonaniem jakichkolwiek działań ingerujących w architekturę fizyczną sieci, jej tworzenie wymaga wyłącznie spełnienia ograniczeń narzucanych przez tę ostatnią. Dlatego, przy realizacji sieci logicznej zamiast terminu *budowa* wykorzystano termin *konfiguracja*. Etapy tworzenia sieci fizycznej i logicznej składają się z takich samych podetapów odnoszących się jednak do jednej z konkretnych architektur. Na etapie stałej eksploatacji i utrzymania sieci wykonywana jest jej rekonfiguracja, będąca elementem danego etapu cyklu życia. Rekonfiguracja odnosi się wyłącznie do zasobów logicznych systemu, tj. sieci połączeń logicznych. Rekonfiguracji, z definicji, nie podlegają zasoby fizyczne, które mogą być unowocześniane lub przeprojektowywane. Utylizacji, z oczywistych względów, podlegają wyłącznie fizyczne komponenty systemu.



Rys. 5. Strukturalizowany schemat cyklu życia szerokopasmowej sieci komputerowej z połączeniami logicznymi. Etapy: a. tworzenie sieci fizycznej; b. tworzenie sieci logicznej; c. eksploatacja sieci szerokopasmowej; d. utylizacja sieci fizycznej.

3. Czas życia komponentów sieci

Czas życia sieci limitowany jest trwałością jej komponentów sprzętowych. Zgodnie z wprowadzoną wcześniej klasyfikacją, elementy składowe sieci regionalnej możemy podzielić na wyposażenie węzłów komunikacyjnych oraz przewodowe

środowisko komunikacyjne. Czas życia powyższych komponentów jest bardzo różny i konieczność ich wymiany nie zawsze wynika z pogorszenia się ich parametrów technicznych. W skład wyposażenia węzłów sieci komputerowej wchodzą takie elementy jak: przełączniki, routery, wzmacniacze, regeneratory, bezprzewodowe stacje bazowe i in. Bardzo szybki postęp technologii elektronicznej powoduje, że urządzenia te starzeją się bardzo szybko. Oznacza to, że po upływie kilku lat ich parametry są znacznie gorsze od parametrów urządzeń aktualnie dostępnych na rynku. Jednak w przypadku większości urządzeń tego typu nie obserwujemy wyraźnego pogorszenia ich parametrów w stosunku do wartości nominalnych. Ich parametry są przez cały okres eksploatacji względnie stabilne. Jednak dalsze wykorzystanie przestarzałych technologicznie urządzeń wiąże się z gorszym wykorzystaniem możliwości kanałów komunikacyjnych w odniesieniu do takich parametrów jak przepustowość, stopa błędów, odporność na zakłócenia itp. Nie jest to jednak konsekwencja pogorszenia się parametrów urządzeń, a wyłącznie postępu technologii.

Wymiana urządzeń nie jest zabiegiem pracochłonnym i może być wykonana przez personel techniczny średniego poziomu, nie dysponujący specjalistycznym przygotowaniem. Koszt urządzeń, w stosunku do kosztów okablowania, czy też komputerów z oprogramowaniem jest stosunkowo niewielki i nie przekracza 20-30% wartości inwestycji. Z tego powodu zaleca się, aby urządzenia węzłowe zmieniane były w czasookresie od 2 do 4 lat. Zupełnie nieuzasadnionym jest powszechnie wykorzystywane w Polsce podejście, zgodnie z którym urządzenia komputerowe eksploatowane są często do ich pełnego zużycia.

W sieci regionalnych masowo wykorzystuje się okablowanie światłowodowe, dla którego charakterystycznym jest proces starzenia się włókien optycznych. W rezultacie procesów starzenia wzrasta tłumienność światłowodu, która z czasem przyjmuje niedopuszczalne wartości, uniemożliwiające dalszą eksploatację. Szacuje się, że żywotność współczesnych kabli optycznych to 25-30 lat (istnieją również badania według których czas ten jest nieosiągalny). Po tym okresie światłowody powinny zostać wymienione. Jeżeli okablowanie zostało ułożone w kanalizacji teletechnicznej proces jego wymiany nie jest szczególnie kłopotliwy. Okablowanie instalowane jest ponownie w tej samej kanalizacji, pod warunkiem, że jest ona drożna. Z kolei czas życia kanalizacji teletechnicznej szacuje się na około 50 lat.

Wszystkie powyższe działania występujące w ramach cyklu życia systemu mają na celu przedłużenie czasu życia inwestycji i pośrednio związane są z procesem jej projektowania. Wiążą się one z unowocześnieniem istniejącej struktury sieci. Jednak oprócz tego w trakcie eksploatacji sieci szerokopasmowej można również spotkać się z koniecznością rozbudowy sieci, wymagającą jej przeprojektowania. Jeżeli sieć została wykonana na bazie najnowocześniejszych technologii komunikacyjnych niezbędne zmiany podstawowych charakterystyk sieci mogą być uzyskane w

procesie jej rekonfiguracji niezwiązanej z żadnymi dodatkowymi kosztami. Rekonfiguracja może być reakcją projektanta na zmianę poziomu wykorzystania przez użytkowników logicznych kanałów komunikacyjnych, pojawienie się nowych użytkowników lub usług sieciowych. Zmiana taka może być również konsekwencją rezygnacji części użytkowników z korzystania z sieci. Poniżej opisujemy procesy projektowania, unowocześnienia i rekonfiguracji sieci realizowane w trakcie cyklu życia projektu.

4. Projektowanie, unowocześnienie i rekonfiguracja sieci

Unowocześnienie i rekonfiguracja sieci należy do najczęściej spotykanych zadań pojawiających się w trakcie jej eksploatacji. Projektowanie dowolnej sieci komputerowej opiera się na 4 podstawowych założeniach:

1. Całościowe projektowanie sieci jako jednolity, niepodzielny proces jest zbyt skomplikowane w realizacji i nie może być stosowane;
2. W przypadku wykorzystania do projektowania optymalizacji wielokryterialnej w warunkach niekompletnej wiedzy o projektowanym obiekcie optymalne rozwiązanie zadania projektowego nie istnieje;
3. Projektowanie sieci to proces iteracyjny wykonywany pod ścisłym nadzorem użytkownika;
4. Funkcjonujące sieci muszą być na bieżąco dostosowywane do aktualnych potrzeb użytkowników.

Z powyższych uwarunkowań wynika, że zaprojektowanie sieci, która efektywnie spełniałaby wymagania użytkownika przez cały okres eksploatacji jest mało prawdopodobne, a nawet niemożliwe. Jeżeli projektowana sieć byłaby niezmienna przez cały okres jej eksploatacji, na etapie tworzenia projektu sieci definiowane jest zadanie statycznego projektowania sieci.

Pomimo oczywistych niedoskonałości metoda taka jest wykorzystywana w procesie projektowania sieci telekomunikacyjnych. W tym przypadku na podstawie długotrwałej obserwacji przepływów, tworzone są sieci odpowiadające określonym schematom ruchu. W rezultacie powstają tzw. tygodniowe lub godzinne projekty sieci, które odpowiadają potrzebom użytkowników funkcjonujących w konkretnych przedziałach czasowych. Z punktu widzenia projektowania większości sieci zastosowanie tej klasy projektowania nie jest zalecane. Sieć ta nie będzie produktem finalnym i będzie dynamicznie rozwijać się, poszerzając zakres swojego funkcjonowania w planie usług i dostępności.

Drugim rodzajem projektowania jest tzw. projektowanie przyrostowe uwzględniające ulepszanie i rekonfigurację sieci. Projektowanie tego typu pozwala m. in. uzyskać odpowiedź na następujące pytania:

1. Kiedy należy podjąć działania mające na celu zmianę architektury sieci;
2. Jakiego typu działania należy podjąć:

a. Rekonfigurację polegającą na zmianie konfiguracji wybranej usługi sieciowej;
b. Unowocześnienie polegające na dodaniu lub usunięciu konkretnej usługi sieciowej przy obowiązkowym zachowaniu wykorzystywanej dotąd architektury sieci;
c. Przeprojektowanie polegające na rozwiązaniu zadania statycznego projektowania sieci, które definiujemy dla zmienionych danych wejściowych oraz ograniczeń projektowania.

Przyrostowe projektowanie sieci jest procesem silnie powiązanym z czasem jej eksploatacji. Jednym z najważniejszych problemów rozwiązywanych w trakcie tego typu projektowania jest określenie zdarzenia wyzwalającego stosowne akcje [54]. Jako przykłady zdarzeń wyzwalających mogą posłużyć:

1. Regularne odcinki czasu;
2. Naruszenie określonego parametru opisującego jakość świadczonych usług, np. opóźnienia transmisji, poziom zajętości kanału komunikacyjnego itp.;
3. Dodanie lub usunięcie nowych użytkowników;
4. Wprowadzenie do sieci nowych usług;
5. Niedostępność usług wykorzystywanych dotąd w sieci.

Przykładami zdarzeń wyzwalających są również procesy starzenia się sieci. Można także przyjąć, że zdarzeniem wyzwalającym modyfikację sieci będzie jednoczesne wystąpienie kilku z powyższych zdarzeń.

Po wystąpieniu zdarzenia wyzwalającego modyfikację architektury sieci może pojawić się jedna z dwóch poniższych sytuacji:

1. W eksploatowanej sieci istnieją *wąskie gardła* ograniczające jakość usług sieciowych udostępnianych użytkownikom. Wspomniane *wąskie gardła* są najczęściej konsekwencją nadmiarowego obciążenia kanałów komunikacyjnych wprowadzanego przez niektóre usługi dostępne użytkownikom. W tym samym czasie inne usługi, z uwagi na deficyt pasma, nie mogą być wykorzystywane na poziomie wymaganym przez użytkowników. Sytuacje takie występują najczęściej w momencie pojawienia się w sieci nowych usług lub wykorzystania już dostępnych przez dodatkowych użytkowników końcowych. Powyższe problemy można rozwiązać poprzez poprawę przepustowości pojedynczych kanałów komunikacyjnych lub rekonfigurację sieci polegającą na przesunięciu nadmiarowego ruchu na mniej obciążone kanały.
2. W sieci wystąpił wzrost ruchu ograniczający przepustowość dostępną każdemu z użytkowników. W tym przypadku potrzebna jest zmiana architektury sieci związana z koniecznością jej unowocześnienia lub przeprojektowania. W tym przypadku rekonfiguracja nie rozwiązuje problemów efektywności funkcjonowania sieci. Projektant musi podjąć decyzję czy wystar-

czającym będzie unowocześnienie polegające na zamianie poziomu dostępnych usług komunikacyjnych, czy też niezbędne będzie przeprojektowanie sieci.

Można założyć, że wykonywane stopniowo rekonfiguracje z czasem zrodzą potrzebę unowocześnienia lub przeprojektowania sieci. W praktyce najczęściej wykorzystuje się strategię, zgodnie z którą wykonywane są drobne, niekosztowne rekonfiguracje sieci, aż do momentu, kiedy niezbędne stanie się jej unowocześnienie lub przeprojektowanie. Zadanie przyrostowego projektowania sieci szerokopasmowej opiera się na wykorzystaniu takich danych wejściowych jak: zbiór usług dostępnych użytkownikom; zbiory węzłów sieci wszystkich poziomów hierarchii; macierz ruchu międzywęzłowego; istniejący i podlegający unowocześnieniu projekt sieci; struktura kosztów unowocześnienia; zbiór ograniczeń projektowania. Celem projektowania przyrostowego jest znalezienie takiej nowej konfiguracji sieci, która po minimalnych kosztach spełniać będzie aktualne wymagania użytkowników. Można wykazać, że od strony kosztowej unowocześnianie projektu jest najtańsze, jeżeli realizuje się je w ostatnim możliwym momencie. Projektowanie przyrostowe ma charakter krótkoterminowy albo reaktywny, co oznacza, że projektant reaguje na każde zdarzenie wyzwalające, wybierając na podstawie kryterium ekonomicznego pomiędzy rekonfiguracją, unowocześnieniem, a przeprojektowaniem sieci.

Ostatnim podejściem do zadania projektowania sieci regionalnej jest projektowanie ewolucyjne zaliczane do strategii długoterminowych. W projektowaniu przyrostowym każde unowocześnienie niesie ze sobą dodatkowe koszty, których można byłoby uniknąć, jeżeli unowocześnienia można byłoby przewidzieć. Projektowanie ewolucyjne polega więc na próbie przewidzenia sekwencji zmian w wymaganiach użytkowników a dotyczących parametrów eksploatowanej sieci i uwzględniania ich w projektowaniu.

Najczęściej projektanci wykorzystują podejście będące specyficznym połączeniem projektowania statycznego, przyrostowego i ewolucyjnego. Zastosowanie teoretycznie najbardziej efektywnej strategii ewolucyjnej jest bardzo trudne do realizacji, nawet jeżeli cały przyszły rozwój sieci jest znany dzięki czemu projektant posiada komplet wiedzy niezbędnej w procesie projektowania. Istota wykorzystanej metody polega na zastosowaniu małych zmian w konfiguracji lub kumulowaniu ich w jedną dużą zmianę konfiguracji systemu.

5. Zadanie rekonfiguracji

Połączenia fizyczne sieci należy zaprojektować tak, aby w długofalowym horyzoncie istniała możliwość ich rozbudowy lub rekonfiguracji. Jednak podstawowe działania rekonfiguracyjne będą wykonywane na poziomie sieci logicznej. Zakłada się, że działania te będą wykonywane stosunkowo często, ponieważ dynamicznymi

będą również zmiany w strukturze ruchu projektowanej sieci. Ich celem będzie poprawa efektywności wykorzystania dostępnych kanałów komunikacyjnych oraz jakości obsługi użytkowników końcowych. Podstawowym problemem, na który będzie musiał odpowiedzieć projektant jest określenie czy koszty podejmowanych działań są adekwatne do uzyskiwanych rezultatów.

Rekonfigurację sieci logicznej możemy rozpatrywać jako problem scentralizowanej optymalizacji, wymagającej globalnego spojrzenia na funkcjonowanie szerokopasmowej sieci komputerowej. Należy zauważyć, że potrzeba rekonfiguracji to konsekwencja zmian ruchu w sieci i nie powinna być ona postrzegana jako wada sieci i nie należy jej w żadnym przypadku wiązać z koniecznością zmiany architektury sieci fizycznej [54], [72]. Często problem rekonfiguracji jest błędnie utożsamiany w zadaniem projektowania topologii logicznej. Wnoszenie zmian w strukturę topologii logicznej jest trudno akceptowalne ze względu na konieczność modyfikacji połączeń pomiędzy użytkownikami i płynące z tego potencjalne konsekwencje (zakłócenia ruchu). Zauważmy, że problem ten jest problemem wielokryterialnym i nie należy rozważać go wyłącznie jako zadania poprawy wydajności sieci, ale również jako zadanie, którego celem jest minimalizacja liczby zmian wprowadzanych do topologii logicznej. Podstawowymi danymi wejściowymi rekonfiguracji są dotychczasowa sieć logiczna oraz aktualna macierz ruchu międzywęzłowego. Najczęściej zadanie to jest definiowane jako zadanie programowania liniowego, w którym blokowane jest jedno z kryteriów optymalizacji. W ten sposób utrzymując na niezmiennym poziomie wartość jednego z kryteriów, znaczenie drugiego jest minimalizowane. W celu rekonfiguracji sieci regionalnej można wykorzystać metodę, w której optymalizacji w sensie Pareto podlegać będą jednocześnie oba kryteria. Dzięki temu w rezultacie procesu optymalizacji utworzony może być zbiór wielu rozwiązań, w przeciwieństwie do metod opartych na programowaniu liniowym, które oferują wyłącznie jedno jedyne rozwiązanie.

Dalej zakładać będziemy, że znana jest aktualna topologia logiczna, w której określono strukturę ścieżek logicznych, marszrutyzację strumieni informacyjnych oraz przydział kanałów logicznych. Można przyjąć, że rekonfiguracja jest problemem podobnym do projektowania sieci logicznej. W tym przypadku jednak nie maksymalizuje (lub minimalizuje) się funkcji celu, a wyłącznie liczbę zmian wykonywanych w topologii logicznej dokonywanych w procesie rekonfiguracji. Zazwyczaj zmiany te są kosztowne ze względu na zakłócenia ruchu i konieczność modyfikacji rozkładu kanałów falowych. Charakterystyczną cechą rekonfiguracji jest korelacja pomiędzy funkcjami celu projektowania topologii logicznej i liczbą zmian niezbędnych w procesie rekonfiguracji. Na przykład, jeżeli funkcją celu projektowania topologii logicznej jest minimalizacja średniej liczby skoków, to im mniejsza będzie wartość tego parametru, tym większa będzie liczba zmian wymaganych w procesie rekonfiguracji. Wynika z tego, że problem projektowania topologii logicznej należy rozpatrywać również z punktu widzenia późniejszej rekonfiguracji systemu połączeń.

Rozwiązanie zadania rekonfiguracji topologii logicznej realizowane będzie w dwóch etapach. W pierwszym z nich jedno z kryteriów będzie niezmienne, drugie zaś będzie minimalizowane. W drugim kroku kryteria te zostaną zamienione miejscami. Załóżmy, że kryteriami wykorzystywanymi w procesie rekonfiguracji będą minimalizacja średniej odległości w topologii logicznej oraz minimalizacja zmian wykonywanych w topologii logicznej. Procedura taka wymaga, jak to było zdefiniowane wcześniej, realizacji dwóch kroków. Pierwszy z nich to wybór jednej funkcji celu przy niezmienności pozostałych, drugi zaś to minimalizacja wybranej funkcji celu. Zaproponowana metoda gwarantuje uzyskanie wyłącznie jednego niedominującego rozwiązania na froncie Pareto. Przy tym, uzyskiwane rozwiązanie w znacznym stopniu zależy od lokalizacji początkowego rozwiązania, a wcześniejsze określenie położenia rezultatu na foncie Pareto nie jest możliwe. Dlatego przyjęto, że określenie formy frontu Patero jest zadaniem złożonym obliczeniowo. Poniżej prezentujemy obie metody określenia rekonfiguracji połączeń.

Niezmienna średnia odległość w sieci logicznej i minimalizacja liczby zmian. Dalej średnią odległość w sieci logicznej określaną za pomocą skoków, odniesioną do istniejących strumieni informacyjnych będziemy nazywać średnią odległością. Procedura rekonfiguracji rozpoczyna się od określenia średniej odległości dla nowego strumienia informacyjnego, bez uwzględnienia liczby zmian w topologii logicznej. Dalej za pomocą zadania programowania całkowitoliczbowego, funkcja celu określana jest jako minimalizacja liczby zmian w ścieżkach logicznych, co można zapisać w następujący sposób:

$$\min \sum_{(i,j)} \sum_{[\![m,n]\!]} \sum_{k} \sum_{w,v} \left| b'^{l_k}_{(i,j),[\![m,n]\!],a_{[w,v]}} - b^{l_k}_{(i,j),[\![m,n]\!],a_{[w,v]}} \right|, \tag{1}$$

gdzie: $b'^{l_k}_{(i,j),[\![m,n]\!],a_{[w,v]}}$, $b^{l_k}_{(i,j),[\![m,n]\!],a_{[w,v]}}$ – zmienne binarne określające relacje pomiędzy fizycznymi i logicznymi kanałami sieci odpowiednio dla modyfikowanej i podstawowej części systemu. Wyrażenie (1) ma postać liniową, ponieważ zmienne występujące w nim są binarne. Absolutną wartość wyrażenia (1) można przedstawić w następujący sposób:

$$\left| b'^{l_k}_{(i,j),[\![m,n]\!],a_{[w,v]}} - b^{l_k}_{(i,j),[\![m,n]\!],a_{[w,v]}} \right| = \begin{cases} 1 - b^{l_k}_{(i,j),[\![m,n]\!],a_{[w,v]}} & \text{jeżeli } b'^{l_k}_{(i,j),[\![m,n]\!],a_{[w,v]}} = 1 \\ b^{l_k}_{(i,j),[\![m,n]\!],a_{[w,v]}} & \text{jeżeli } b'^{l_k}_{(i,j),[\![m,n]\!],a_{[w,v]}} = 0. \end{cases} \tag{2}$$

Na koniec określimy nowe ograniczenie dla programowania całkowitoliczbowego:

$$\frac{1}{\sum_{sd,x} \Lambda^{x}_{sd}} \sum_{(i,j)} \sum_{sd,x} \left(x \lambda^{x}_{sd,(i,j)} \right) \le d \,. \tag{3}$$

gdzie: d – analizowana odległość.

Stała średnia odległość i minimalizacja średniej odległości. Załóżmy, że liczba zmian w topologii logicznej wnoszonych w procesie rekonfiguracji jest

równa $\aleph$. Wtedy pierwszy krok procedury rekonfiguracji polega na zamianie funkcji celu programowania całkowitoliczbowego na minimalizację wartości średniej odległości, tj.:

$$\min \frac{1}{\sum_{sd,x}\left(x\Lambda_{sd}^{x}\right)} \sum_{(i,j)} \sum_{sd,x} \left(x\lambda_{sd,(i,j)}^{x}\right). \tag{4}$$

Następnie definiujemy nowe ograniczenie dla zadania całkowitoliczbowego programowania liniowego:

$$\sum_{(i,j)} \sum_{[\![m,n]\!]} \sum_{k} \sum_{w,v} \left| b'^{l_k}_{(i,j),[\![m,n]\!],a_{[w,v]}} - b^{l_k}_{(i,j),[\![m,n]\!],a_{[w,v]}} \right| \leq \aleph. \tag{5}$$

Generując zmiany w topologii logicznej należy pamiętać, że ich sekwencja wpływa na wydajność sieci. Zdefiniowane powyżej zadania rekonfiguracji sieci logicznej mogą być rozwiązane za pomocą następujących metod:

1. Optymalizacji wielokryterialnej opartych na poszukiwaniu zbioru optimum Pareto;
2. Bazujących na wielokryterialnym podejściu ewolucyjnym wykorzystującym różnorodne techniki przeszukiwania losowego.

Metody te są szeroko opisane w dostępnej literaturze.

6. Uwzględnienie przyszłych wymagań użytkownika

W celu maksymalizacji czasu eksploatacji sieci szerokopasmowej konieczne jest uwzględnienie w procesie projektowania przyszłych wymagań użytkowników. Uwzględnienie takie jest realizowane na bazie ewolucyjnego podejścia do projektowania. Jakość projektowania ewolucyjnego w istotny sposób zależy od trafności przyszłych wymagań użytkowników dotyczących rozmiaru i typu generowanego przez nich ruchu. Jeżeli dla projektowanej sieci nie są znane perspektywy jej eksploatacji, to przewidywania dotyczące generowanego w niej ruchu są mniej precyzyjne. Dlatego, aby osiągnąć satysfakcjonujące rezultaty projektowania należy wykorzystywać krótsze okresy planowania.

Prognozowanie w sieciach regionalnych obejmuje najczęściej:

1. Wymagań użytkowników i aplikacji w odniesieniu do generowanego ruchu;
2. Liczby użytkowników oraz wykorzystywanych przez nich usług dla każdego węzła końcowego;
3. Ilości oraz sposobu rozmieszczania węzłów końcowych;
4. Rozmiaru ruchu zewnętrznego w punktach integrujących projektowaną sieć z sieciami innych operatorów;
5. Wymagań aplikacji rozproszonych.

Jeżeli w analizowanej sieci obserwujemy znaczne zmiany ruchu, czasokres jej badania może być krótszy, ponieważ prawdopodobieństwo popełnienia znacznych

błędów przy szacowaniu jest niewielkie. W ostatnich latach największy wzrost ruchu obserwujemy w sieciach IP. Nie oznacza to jednak, że tworzone prognozy pozbawione są błędów.

Większość wykonywanych obecnie projektów sieci zakłada, że dane wejściowe procesu projektowania są w pełni zdefiniowane przed jego rozpoczęciem. Jednak projektowanie zakładające jednoznaczność danych wejściowych prowadzi do zbyt daleko idącego uproszczenia, a tym samym nie może zagwarantować efektywnego wykorzystania projektowanej sieci w dłuższych przedziałach czasowych. Brak możliwości identyfikacji oraz dokładnej specyfikacji wszystkich elementów bezpośrednio wpływających na projektowany obiekt, w szczególności w dłuższych przedziałach czasowych, prowadzi do niejednoznaczności poszukiwanego rozwiązania, a w szerszym kontekście do jego nieokreśloności. W odniesieniu do procesu projektowania można zdefiniować trzy typy nieokreśloności: danych wejściowych, kryteriów oraz modelu.

Proces projektowania to w istocie ograniczanie nieokreśloności informacyjnej o projektowanym obiekcie. Największy poziom nieokreśloności występuje na etapie podejmowania decyzji o budowie sieci, następnie jest on zmniejszany i osiąga minimalną wartość w projekcie technicznym. Zmniejszanie poziomu nieokreśloności szczególną rolę odgrywa w trakcie definiowania zadania technicznego oraz projektowania wstępnego. Na tych etapach podejmowane są decyzje dotyczące celów i metod projektowania, określane są optymalne proporcje pomiędzy potrzebami użytkowników, a możliwościami technicznymi i finansowymi będącymi w dyspozycji inwestora. Właściwe podejście to tego zagadnienia zdecydowanie zwiększa efektywność rozwiązania zadania projektowego.

W ten sposób koniecznym stało się poszerzenie klasycznego pojęcia optymalizacji rozwiązań poprzez wprowadzenie oprócz zbioru dopuszczalnych wariantów rozwiązania Y, zbioru nieokreśloności X, odzwierciedlającego nieokreśloność kryteriów, danych źródłowych i modeli. Wtedy kryterium optymalności $f(x,y)$ będzie opisane na iloczynie zbiorów XY, co uniemożliwia na bazie optymalizacji jednoznacznie określić najlepsze rozwiązanie zadania projektowego. Wynika to z faktu, że każdej wartości x ze zbioru X nie odpowiada własny wariant najlepszego rozwiązania otrzymywany poprzez optymalizację $f(x,y)$ na zbiorze Y dla określonego x.

W pracach projektowych można wykorzystać efektywny, a zarazem prosty, sposób uwzględnienia i ograniczenia zjawiska nieokreśloności w procesie projektowym szerokopasmowych sieci komputerowych. Problem uwzględnienia nieokreśloności przy określaniu poziomu efektywności oraz wybór rozwiązania sprowadzają się do stworzenia zbioru nieokreśloności A i całkowaniu lub maksymalizacji iloczynu współczynników wagowych oraz cząstkowych wskaźników efektywności.

Wykorzystanie szerszego zbioru kryteriów pozwala uzyskać wyższą efektywność rozwiązania w kontekście technicznym i ekonomicznym. Liczba i typ kryte-

riów uzależnione są od projektanta oraz tworzonego obiektu. Innymi słowy dopuszcza się zastosowanie różnych zestawów kryteriów przy projektowaniu rdzenia sieci oraz sieci dostępowych. Dzięki temu, otrzymane rozwiązanie lepiej wpisuje się w rzeczywiste warunki funkcjonowania, a zarazem potrzeb i możliwości przyszłych jego użytkowników.

Oferowana metoda pozwala sprowadzić wybór rozwiązania zadania projektowego w warunkach wielokryterialności oceny efektywności, do oceny na zbiorze wartości skalarnych, przy uwzględnieniu niejednoznaczności oceny i wyboru rozwiązania. Jest to możliwe dzięki wprowadzeniu zasad zwijania, które przekształcają wektor efektywności w skalarny funkcjonał, a proces wyboru rozwiązania w warunkach wielokrotności wskaźników jakości sprowadzają do formalnej procedury optymalizacji. W procesie rozwiązania zadania projektowego istnieje możliwość wskazania żądanych zakresów wartości wskaźników efektywności, co można zapisać jako:

$$f_i \in [f_{i\min}, f_{i\max}],\ i = 1,2,\ldots,n \tag{6}$$

Na tej podstawie można również określić zakres wartości współczynników wagowych, tj.

$$a_i \in [a_{i\min}, a_{i\max}],\ i = 1,2,\ldots,n, \tag{7}$$

Formalny sposób opisu zbioru współczynników wagowych A z uwzględnieniem uogólnionych właściwości wielościanów wypukłych zaprezentowano w [78], [80]. Z algorytmicznego punktu widzenia sprowadza się to do określenia współrzędnych wierzchołków elementarnych czworościanów wpisanych w wielościan A. Dzięki temu, że $1 \geq a_i \geq 0$, $i = 1,\ldots,n$ oraz ponieważ $\sum_{i=1}^{n} a_i = 1$ możliwym staje się poglądowe, geometryczne przedstawienie zbioru A w przestrzeni n-wymiarowej.

7. Podsumowanie

Powyżej zaprezentowano zasady rozbudowy i rekonfiguracji sieci regionalnej. W szczególności przedstawiono w nim:

1. Organizacyjne i techniczne zasady zmiany konfiguracji sieci celem dostosowania jej architektury do bieżących potrzeb użytkowników. Rekonfiguracja opiera się przede wszystkim na zmianie struktury połączeń logicznych;
2. Metodę tworzenia nowych sieci połączeń logicznych;
3. Metodę uwzględniania w procesie projektowania przyszłych wymagań użytkowników.

Zdefiniowano następujące podstawowe wnioski:

1. Projektowanie regionalnej sieci komputerowej powinno składać się z następujących po sobie etapów projektowania sieci fizycznej i logicznej;

2. Najszybszemu procesowi starzenia w sieciach podlegają urządzenia komunikacyjne będące wyposażeniem węzłów sieci;
3. Cykl życia sieci najlepiej opisuje model spiralny;
4. Projektowanie sieci powinno opierać się na modelu przyrostowym lub ewolucyjnym;
5. Ponieważ zmiana konfiguracji sieci logicznej wywołuje niepożądane zakłócenia ruchu należy dążyć, aby liczba wprowadzanych zmian była minimalna przy zachowaniu parametrów sieci na niezmiennym poziomie;
6. Z punktu widzenia rekonfiguracji sieci połączeń logicznych do oceny jakości rozwiązań celowym jest wykorzystanie średniej odległości w skokach w sieci;
7. Projektując topologię logiczną należy pamiętać o silnej korelacji parametrów topologii i złożoności procesu rekonfiguracji;
8. Uwzględnienie w procesie projektowania przyszłych wymagań użytkowników należy oprzeć o różne rodzaje nieokreśloności.

Wnioski i spostrzeżenia autorów w pełni potwierdza dostępna literatura [1]-[44].

Bibliografia

[1] M. D. Mesarovic, D. Macko, and Y. Takahara, *Theory of hierarchical, multilevel systems*. New York: Academic Press, 1970.

[2] N. J. Smith and A. P. Sage, *An introduction to hierarchical systems theory*. Dallas, Texas: Information and Control Sciences Center, SMU Institute of Technology, 2005.

[3] Thomas L. Saaty, *Fundamentals of Decision Making with the Analytic Hierarchy Process*. Pittsburgh: RWS Publications, 2000.

[4] Ch. S. Wasson, *System Analysis, Design, and Development: Concepts, Principles, and Practices*. New Jersey: Wiley-Interscience, 2005.

[5] Anany Levitin, *Introduction to the Design and Analysis of Algorithms*. Phenix: Addison Wesley, 2006.

[6] Jennifer Mendez-Rangel and Carlos Lozano-Garzon, "A network design methodology proposal for E-health in rural areas of developing countries," in *6th Euro American Conference on Telematics and Information Systems*, Valencia, 2012, pp. 1-7.

[7] Richard V. Slyke. (2001) Network planning and design. ftp://ftp.shore.net/members/ws/Support/BDC/nd.pdf.

[8] Darren L. Spohn, *Data Network Design*. New York: McGraw-Hill, 2002.

[9] I. N. Bronsztejn, K. A. Siemiendiajew, G. Musiol, and H. Muhlig, *Nowoczesne kompendium matematyki*. Warszawa: Wydawnictwa Naukowe PWN, 2004.

[10] Roger L. Freeman, *Fundamentals ofTelecommunications*, II ed. Hoboken, New Jersey: John Wiley & Sons, 2005.

[11] Dipankar Raychaudhuri and Mario Gerla, *Emerging Wireless Technologies and the Future Mobile Internet*. Cambridge: Cambridge University Press, 2011.

[12] Andrew S. Tannenbaum, *Strukturalna organizacja systemów komputerowych*, V ed. Gliwice, Polska: Helion, 2006.

[13] John L. Hennessy and David A. Patterson, *Computer Architecture a Quantitative Approach*, I ed. San Francisco, California: Morgan Kaufmann, 2002.

[14] Eldad Perahia and Robert Stacey, *Next Generation Wireless LANs: Throughput, Robustness, and Reliability in 802.11n*. Cambridge: Cambridge University Press, 2008.

[15] Emerson May, *Wireless Communications & Networks*, 2nd ed. New Jersey: Prentice Hall, 2004.

[16] Mirosław Hajder, Paweł Dymora, and Mirosław Mazurek, "Projektowanie topologii transparentnych sieci optycznych," in *Konferencja Polski Internet Optyczny: technologie, usługi i aplikacje*, Poznań, 2002, pp. 183-195.

[17] Thomas L. Saaty, *Models, Methods, Concepts and Applications of the Analytic Hierarchy Proces*. Boston: Kluwer Academic Publishers, 2000.

[18] Sivarama Dandamudi, *Hierarchical Scheduling in Parallel and Cluster Systems*. New York: Springer, 2003.

[19] Taha A. Hamdy, *Operations Research: An Introduction*. New Jersey: Prentice Hall, 2005.

[20] S. M. Sinha, *Mathematical Programming: Theory and Methods*. Delhi: Elsevier Science, 2006.

[21] Guoqiang Mao and Barış Fidan, *Localization Algorithms and Strategies for Wireless Sensor Networks*. Hershey: Information Science Publishing, 2009.

[22] Jonathan L. Gross and Jay Yellen, *Handbook of graph theory*. London: CRC Press, 2004.

[23] Paolo Santi, *Topology Control in Wireless Ad Hoc and Sensor Networks*. Chichester: John Wiley & Sons, 2005.

[24] Anil K. Jain, Narasimha Murty, and Patrick J. Flynn, "Data Clustering: A Review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264-323, 1999.

[25] Nicholas Jardine, *Mathematical Taxonomy*. London: John Wiley & Sons, 1971.

[26] Mark S. Levin, "Towards hierarchical clustering," in *Computer Science - Theory and Applications: Second International Symposium on Computer Science in Russia*, 2007, pp. 205-215.

[27] Miguel F. Acevedo, *Real-Time Environmental Monitoring. Sensor and systems*. Boca Raton: CRC Press, 2015.

[28] Anna Hać, *Wireless sensor network designs*, 2nd ed. Hoboken: John Wiley & Sons Inc, 2013.

[29] Bang Wang, *Coverage Control in Sensor Networks*. London: Springer-Verlag, 2010.

[30] Mohammad Illyas and Imad Mahgoub, Eds., *Handbook of sensor networks: compact wireless and wired sensing systems*, 2nd ed. Boca Raton: CRC Press, 2012.

[31] Pavel Ripka and Alois Tipek, Eds., *Modern Sensors Handbook*. Chippenham: Antony Rowe Ltd, 2007.

[32] Martin P. Clark, *Networks and Telecommunications: Design and Operation*. Chicheste: John Wiley & Sons, 1991.

[33] Sharon Evans, *Telecommunications Network Modeling, Planning and Design*. London: The Institution of Electrical Engineers, 2003.

[34] Vivek Alwayn, *Optical Network Design and Implementation*. Indianapolis: Cisco Press, 2004.

[35] Nino Boccara, *Modeling Complex Systems*, 2nd ed. New York: Springer, 2010.

[36] T. Trzaskalik, *Modelowanie optymalizacyjne*. Łódź: Przedsiębiorstwo Specjalistyczne Absolwent, 2001.

[37] M. Stasiak, M. Głąbowski, and P. Zwierzykowski, *Modelowanie i wymiarowanie ruchomych sieci bezprzewodowych*. WKiŁ: Warszawa, 2009.

[38] L. Davis, *Natural Disasters*, II ed. New York, USA: Facts On File, 2009.

[39] J. Ingleton, Ed., *Natural Disaster Management*. Leicester, England: Tudor Rose, 1999.

[40] Ch. S. Desai and M. Zaman, *Advanced Geotechnical Engineering. Soil - Structure Interaction Using Computer and Material Models*. Boca Raton, USA: CRC Press, 2014.

[41] Khairy A.H. Kobbacy and Prabhakar D.N. Murthy, Eds., *Complex System Maintenance Handbook*. London: Springer-Verlag London Limited, 2008.

[42] James D. McCabe, *Network Analysis, Architecture and Design*, 3rd ed. New York: Morgan Kaufmann, 2007.

[43] Charles S. Wasson, *System Analysis, Design, and Development: Concepts, Principles, and Practices*. New Jersey: Wiley-Interscience, 2005.

[44] Robert B. Northrop, *Introduction to Complexity and Complex Systems*. New York: CRC Press, 2010.

ROZDZIAŁ 7

Struktury połączeniowe w dobie sieci obliczeń brzegowych

♦Mirosław HAJDER, ♠Piotr HAJDER

♦*Wyższa Szkoła Informatyki i Zarządzania z siedzibą w Rzeszowie,*

♠*Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie*

Miroslaw.Hajder@gmail.com, Piootr.Hajder@gmail.com

Streszczenie

W rozdziale autorzy analizują procedurę projektowania połączeń pomiędzy komponentami systemu informacyjnego. Zakłada się, że budowane systemy będą wielkoskalowe i zastosowanie do ich połączenia tradycyjnych rozwiązań nie będzie skuteczne. Podstawą nowej architektury połączeń są: wielokanałowość, hierarchizm oraz osadzanie. Rozdział określa metody wykorzystane do rozwiązania postawionego zadania, w większości są to oryginalne metody opracowane w trakcie badań. Wyniki badań powinny znaleźć zastosowanie przy realizacji nowych typów sieci, w tym Internetu rzeczy itp. tj. w rozwiązaniach, gdzie integracji podlegają wielkie ilości węzłów ze stosunkowo niewielką mocą obliczeniową. Rozdział jest krótkim raportem z badań prowadzonych przez autorów na zlecenie niezależnego sponsora.

1. Wstęp

Podmiotem badań autorów są narzędzia syntezy i analizy heterogenicznych systemów rozproszonych z hierarchiczną organizacją komunikacji międzywęzłowej, realizowaną na bazie wielokanałowości fizyczno-protokolarnej. Badania, w zbliżonym stopniu, dotyczą każdego z typów systemów rozproszonych (obliczeniowych, informacyjnych i semantycznych). *Przedmiotem* badań są metody poprawy efektywności komunikacyjnej systemów rozproszonych, bazujące na wielokanałowych wirtualnych połączeniach hierarchicznych, od strony metodologicznej wykorzystujące teorie: grafów, mnogości, podziałów, planowania eksperymentu naukowego; optymalizację kombinatoryczną i in. *Celem* badań jest poszerzenie teorii projektowania systemów rozproszonych o opracowane i zweryfikowane metodyki, metody, algorytmy i sposoby poprawy efektywności ich funkcjonowania, oparte na wielokanałowych wirtualnych połączeniach hierarchicznych.

Wśród najważniejszych przesłanek implikujących *aktualność* tematu badań można wymienić: **a.** *Integrację sieci komputerowych i telekomunikacyjnych*, w

pierwszej kolejności, wyrażającą się postępującym przejmowaniem od telekomunikacji usług głosowych i obrazowych, co utrudnia efektywne wykorzystanie zasobów komunikacyjnych sieci komputerowych, dedykowanych do transmisji danych; **b.** *Poszerzenie asortymentu jednocześnie oferowanych usług*, *których kompatybilność komunikacyjna jest ograniczona*. Wymaga to nowego podejścia do zarządzania jakością QoS (ang. *Quality of Service*) lub dynamicznego wydzielania niezależnych kanałów komunikacyjnych dla wybranych użytkowników (w szczególności, korzystających z infrastruktury krytycznej CIS (ang. *Critical Infrastructure Systems*)) lub usług, w tym usług multimedialnych, świadczonych w czasie rzeczywistym; **c.** *Wysoką dynamikę zmian parametrów i charakterystyk funkcjonowania systemu komunikacyjnego*. Wzorzec obciążenia sieci nie ma obecnie charakteru statycznego, a dynamicznie się zmienia. Dlatego, nie jest możliwe zaprojektowanie i wykonanie sieci spełniającej wymagania użytkowników, nawet w skali średnioterminowej. Ponadto, ruch sieciowy ma często charakter wybuchowy, którego współczynnik wybuchowości przekracza granice akceptowalne przez technologie. Eliminacja jego szkodliwego wpływu wymaga izolacji komunikacji, najlepiej poprzez elastyczną rekonfigurację połączeń na poziomie logicznym, bez konieczności modyfikacji architektury sprzętowej; **d.** *Rozwój technologii obliczeniowych i komunikacyjnych na tle niezmienności metod projektowania*. Pomimo pojawienia się nowych rozwiązań technicznych, typów sieci oraz usług, do ich projektowania nadal wykorzystuje się narzędzia, których podstawy naukowe opracowano początkiem lat 70-tych (L. Kleinrock, M. Gerla). Do projektowania stacjonarnych i mobilnych sieci obliczeń brzegowych (ang. *edge computing*), w tym Internetu rzeczy (IoT), sieci obliczeń mgłowych (FC) i rosowych (DC), sieci rozszerzonej rzeczywistości (AR), zazwyczaj wykorzystuje się znane od lat metody i środki projektowania.

Współczesne systemy rozproszone syntezowane z zastosowaniem dotychczasowych metod, charakteryzuje nieefektywność wykorzystania mocy obliczeniowych, przepustowości kanałów komunikacyjnych, pamięci masowych oraz zasobów informacyjnych. Badania w zakresie poprawy ich efektywności koncentrują się na komunikacji między komponentowej, przy czym, niewiele z nich ukierunkowanych jest na formalną syntezę komponentów i architektury sieci komunikacyjnej, analizę procesów transmisji, badanie ekonomicznych uwarunkowań konkretnych inwestycji. W rezultacie, sieci świadczące nowe usługi, oparte na współczesnych technologiach tworzone są na bazie dotychczasowych metod i środków projektowania. Zastosowanie szeroko rozumianych współczesnych rozwiązań technicznych wymaga zmiany: ilościowej i jakościowej podstawy badań, nowego zbioru modeli matematycznych, nowych metod oceny efektywności, planowania i optymalizacji sieci połączeniowych. Ich brak skutkuje nieproduktywnym wykorzystaniem zasobów, w szczególności pasma komunikacyjnego sieci rdzeniowych. Dlatego, podstawowym zadaniem służącym poprawie efektywności wykorzystania fizycznych zasobów komunikacyjnych jest opracowanie algorytmicznych i technologicznych podstaw projektowania i rekonfiguracji połączeń sieci. Analizując aktualność tematu badań należy również odnieść się do chmur różnych typów, w których

zdolność efektywnego zarządzania wydajnością i automatyzacja skalowalności są jednymi z najważniejszych wymagań [1], [2], [3], [4], [5], [6], [7].

2. Systemy rozproszone– organizacja i architektura

Naszą uwagę skoncentrujemy na analizie problemów eksploatacji systemów rozproszonych (SR), poszukiwaniu kierunku badań oraz określeniu skuteczności aktualnych metod ich projektowania. Systemy rozproszone to dzisiaj również sieci obliczeń brzegowych, włączające szerokie spektrum organizacji i architektur, m. in.: sieci: sensorowe, mobilnego gromadzenia danych, mobilnej analizy sygnatur; współdzielone sieci *peer-to-peer* i *ad hoc*; lokalne chmury, mgły i rosy obliczeniowe; siatki i klastry obliczeniowe; rozproszone pamięci masowe z odtworzeniem; zdalne serwisy chmurowe i in. Systemy rozproszone w klasycznym rozumieniu, zdefiniowanym przez A. Tanenbauma, do budowy których można również zastosować nowoczesne technologie, wymagają uwspółcześnienia podejścia do projektowania. Analizie architektury różnorodnych systemów rozproszonych poświęcono publikacje [4], [5], [8], [9].

Na początku badań, zdefiniowano pojęcie współczesnych SR, przedstawiając m.in. ich organizację i architekturę oraz klasyczne parametry i charakterystyki [1], [5], [10]. Aby udowodnić, iż prowadzone badania wpisują się w szeroki kierunek prac, których celem jest poprawa efektywności SR, można zaprezentować wyniki analiz, dotyczących kierunków rozwoju systemów rozproszonych [7], [11]. Przeprowadzone maszynowe badania literaturowe polegały na analizie ilościowej dostępnych publikacji i ich cytowań, identyfikowanych zestawem prostych lub złożonych słów kluczowych. Badaniu podlegały publikacje dostępne w bazach *CiteSeerX* oraz *GoogleScholar*. Zastosowana metodyka została oparta na zmodyfikowanej idei zaprezentowanej w 2007 roku przez B. Kitchenham'a i S. Charters'a, którzy prowadzili badania w obszarze produkcji i modernizacji oprogramowania. Podobnie jak w ich przypadku, strategia wyszukiwania była tworzona dynamicznie, w trakcie poszukiwania i przeszukiwania źródeł. Do obróbki zgromadzonych danych wykorzystano zmodyfikowaną analizę klastrową, wykorzystującą metodę analogii ze środkiem ciężkości.

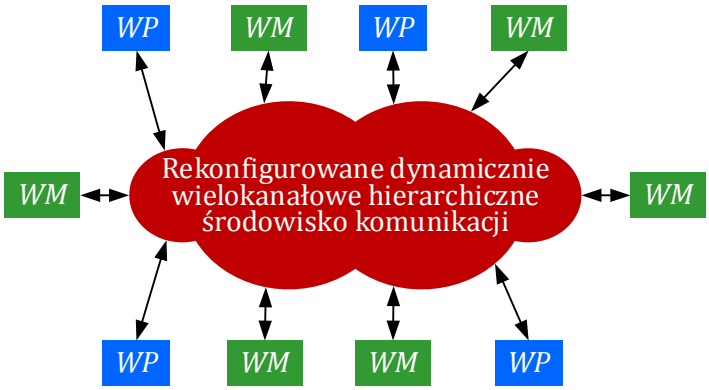
Z badań wynika, że kierując się kryterium ilościowym (liczba publikacji lub cytowań) najczęściej pojawiało się 6 celów prac w obszarze poprawy efektywności, które wyszczególniono na rys. 1.

Rys. 1. Wyniki badań nad celami prac w obszarze poprawy efektywności SR

W latach 80-tych XX wieku K. Hwang sformułował hipotezę, zgodnie z którą najpoważniejsze ograniczenia funkcjonowania systemów równoległych i rozproszonych wynikają z komunikacji międzywęzłowej. Podobna teza przewija się także w publikacjach takich autorów jak: I.D. Scherson, A.S. Yousef, B. Narahari i in. Autor szerzej odniósł się do hipotezy Hwanga w [12], gdzie ją zmodyfikował i uwspółcześnił. Odnosząc wspomnianą hipotezę do postępów technologii oraz oczekiwań użytkowników, dochodzimy do wniosku, że rozwiązaniem części zdefiniowanych wyżej problemów jest budowa architektury obliczeniowej na bazie dynamicznie rekonfigurowanych połączeń, przy czym każdy z węzłów powinien być obsługiwany przez zbiór logicznie niezależnych kanałów. Idea takiego rozwiązania, jego właściwości oraz zastosowania zostały w [5], [13], [14], [15], [16]. Badania analityczne pokazały, że zastosowanie rekonfigurowanych dynamicznie połączeń jest szczególnie efektywne w przypadku obszarów oznaczonych na rys. 1 kolorem czerwonym [17] [18].

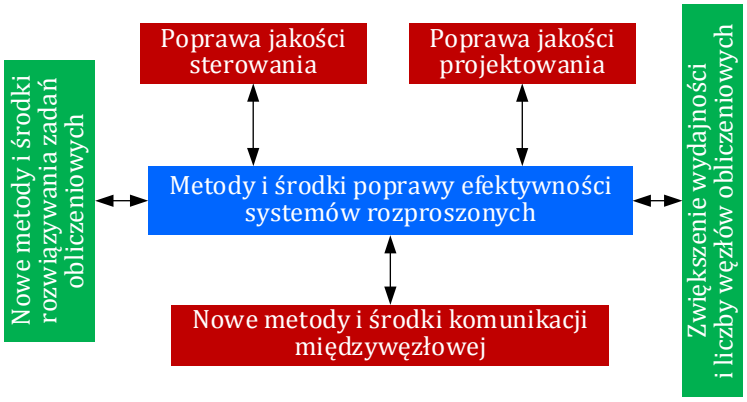
Uproszczona idea architektury komputerowej opartej na rekonfigurowanym dynamicznie środowisku połączeniowym została pokazana na rys. 2. W poniższym systemie, każdy element przetwarzający może uzyskać status brzegowego, dzięki czemu może być ona wykorzystywana zarówno dla klasycznych SR, jak również sieci obliczeń brzegowych.

Architektura składa się z trzech wzajemnie niezależnych komponentów: WP – węzłów przetwarzających, opartych na modelu COW lub podobnym; WM – węzłów pamięci masowej korzystających z technologii gwarantującej dostęp do równoległego odczytu/zapisu informacji oraz najważniejszego elementu – rekonfigurowanego dynamicznie wielokanałowego hierarchicznego środowiska komunikacji. Może ono funkcjonować na bazie optycznych technologii przewodowych (pełny zakres funkcjonalny), bezprzewodowych (ograniczony zakres) lub ich połączenia. Środowisko zapewnia zestawienie połączeń pomiędzy dowolną parą węzłów.



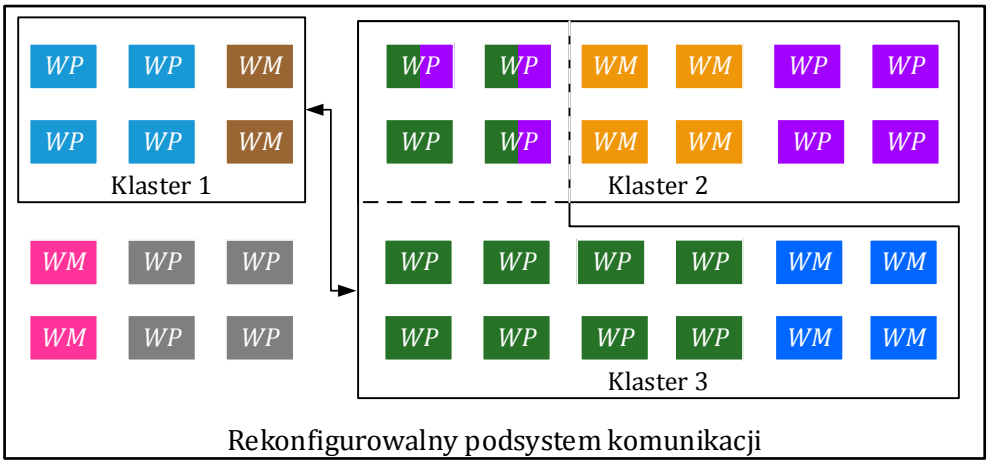
Rys. 2. Rekonfigurowalny system rozproszony ze środowiskiem komunikacji nowego typu. Oznaczenia: WP – węzeł przetwarzający; WM – węzeł magazynujący

Korzystając z wyszczególnionych wcześniej badań literaturowych, zdefiniowano podstawowe sposoby poprawy efektywności współczesnych systemów rozproszonych, których wyniki zaprezentowano na rys. 3 [5], [11]. Na czerwono oznaczono te spośród sposobów, które mogą efektywnie wykorzystać zalety nowego środowiska komunikacyjnego.



Rys. 3. Podstawowe metody i środki poprawy efektywności SR

Idea rekonfigurowalnego środowiska obliczeniowego została pokazana na rys. 4. Wydzielone środowiska obliczeniowe nazywane klastrami mogą być względem siebie autonomiczne lub połączone, jak ma to miejsce w przypadku klastrów 1 oraz 3. Połączenia pomiędzy klastrami realizowane są za pomocą tych samych środków co połączenia wewnętrzne i mogą być wykonane jako połączenia dwupunktowe, grupowe lub rozgłoszeniowe. Poszczególne węzły RSK (WP – przetwarzające, WM – magazynujące) mogą być wykorzystywane wyłącznie w jednym klastrze bądź współdzielone pomiędzy nimi. Umożliwia to ich wykorzystanie do wymiany informacji pomiędzy klastrami za pomocą komunikacji międzyprocesowej lub wspólnych obszarów pamięci operacyjnej lub masowej.



Rys. 4. Rekonfigurowalne środowisko obliczeniowe

Architektura wykorzystująca wielokanałowe wirtualne połączenia hierarchiczne jest wręcz idealnym środowiskiem do budowy systemów obliczeń brzegowych (ang. *edge computing* (EC)), w tym sieci sensorowych i Internetu rzeczy IoT. Dzięki elastyczności systemu połączeń, każdy z elementów (obliczeniowych, magazynujących) może być jednocześnie elementem brzegowym, jak również rdzeniowym.

Mając na uwadze projektowanie systemów EC założono, że zadanie syntezy będzie procesem trójetapowym złożonym z kroków: doboru komponentów dla fragmentów systemu, ich rozmieszczanie oraz projektowania połączeń, w szczególności hierarchicznych, pomiędzy komponentami. Podstawą rozważań była budowa hierarchii systemu, w szczególności na bazie klasteryzacji hierarchicznej HNDP (ang. *Hierarchical Network Design Problem*). Tematem tym zajmowali się m. in. T. Thomadsen, J.R. Current, C.S. ReVelle, J.L. Cohon, H. Pirkul, D. Saha, A. Mukherjee. Do tego celu najczęściej wykorzystuje się algorytm analizy skupień (ang. *Agglomerative Hierarchical Clustering*) oraz tzw. metodę hierarchiczną z przecięciem klasterów. Znane są również metody oparte na rozwiązaniu zadania NHNDP (ang. Non-Hierarchical Network Design Problem). Metody teorii systemów hierarchicznych można wykorzystać również do rozwiązania zadań selekcji komponentów i ich rozmieszczenia. W obu przypadkach zastosowanie znajduje hierarchiczne projektowanie morfologiczne na bazie kliki morfologicznej. Na bazie analizy dokonano oceny efektywności poszczególnych metod projektowania.

W ogólnym przypadku, rozwiązanie zadania kosztów cyklu życia można sprowadzić do znalezienia minimum funkcjonału C kosztów projektowania, budowy, eksploatacji oraz utylizacji sieci:

$$C(U,\Omega,Y) \rightarrow \min , \tag{1}$$

uwzględniając ograniczenia V_i na probabilistyczno-czasowe i strukturalne charakterystyki sieci:

$$V_i(U,\Omega,Y) \le V_{i0} \tag{2}$$

oraz wymagania przynależności zbioru wariantów projektowanej sieci $C(U, \Omega, Y)$, spełniających ograniczenia (2) do obszaru Q_0 rozwiązań realizowalnych technicznie:

$$Q(U,\Omega,Y) \in Q_0, \tag{3}$$

gdzie: U – wektor parametrów obciążenia sieci, włączający intensywność strumieni informacyjnych pomiędzy każdą parą węzłów przełączających, rozkład długości przesyłanych komunikatów, priorytety strumieni itd.; Ω– wektor parametrów technicznych elementów składowych sieci, takich jak: przepustowość węzłów i sprzętu tworzącego kanały, ich niezawodność itd.; Y– wektor parametrów sieci logicznej; V_{i0} – zbiór ograniczeń dla obszaru rozwiązań realizowalnych technicznie.

3. Fazy, stadia i etapy życia obiektu lub systemu

Pojęciem najczęściej wykorzystywanym w badaniach jest *hierarchia*. Dlatego, na jego wstępie przeanalizowano kilka jej definicji (M.D. Mesarovic, N.J. Smith, T.L.Saaty). Do dalszych rozważań zastosowano jednak własne określenie teoriografowe, zgodnie z którym dowolna struktura hierarchiczna może zostać przedstawiona w postaci skierowanego grafu acyklicznego, zapewniające przypisywanie poszczególnym elementom hierarchii zarówno parametrów ilościowych lub jakościowych.

Niech $A = \{a_1, a_2, \cdots, a_n\}$ oznacza zbiór elementów. Każdy element $a \in A$ posiada typ $\tau(a)$. Liczba typów elementów jest skończona $\tau_1, \cdots, \tau_m$, $m \le n$ i typy te są uporządkowane relacją porządku częściowego, oznaczoną symbolem $\le$, tzn. $\tau_1 \le \cdots \le \tau_m$. Typy elementów pozwalają wprowadzić relację równoważności R na zbiorze A ($R \subseteq A^2$): $aRa \Leftrightarrow \tau(a) = \tau(b)$, $a, b \in A$. Z kolei relacja porządku częściowego na typach pozwala uporządkować zbiór ilorazowy:

$$A / R = \{A_1, A_2, \ldots, A_m\} : A_i < A_j \Leftrightarrow \forall a \in A_i, b \in A_j, \tau(a) < \tau(b).$$

Załóżmy, że jeżeli $\tau(a) < \tau(b)$, wtedy element a podlega w hierarchii elementowi b. W celu zapewnienia możliwości wielokrotnego osadzenia obiektu niższego poziomu na zasobach poziomu wyższego, wprowadzimy opartą na relacji podlegania funkcję gęstości f: $f: A \to N$, gdzie $f(a) = k$ oznacza, że elementowi a podlega k elementów niższych typów. Funkcja gęstości określa relację równoważności R na zbiorze A: $a\tilde{R}b \Leftrightarrow a\tilde{R}b \vee f(a) = f(b)$, $a, b \in A$. W rezultacie, otrzymujemy zbiór ilorazowy posiadający następującą postać:

$$A / \tilde{R} = \left\{ A_{i_1}^1, \ldots, A_{i_{k_1}}^{k_1}, A_{i_2}^1, \ldots, A_{i_2}^{k_2}, \ldots, A_{i_m}^1, \ldots, A_{i_m}^{k_m} \right\},$$

gdzie: $A_{ij} = \bigcup_{l=1}^{k_j} A_{ij}^l$, $j = 1, \cdots, m$, $k_j = k_1, \cdots, k_m$. W ten sposób, uzyskujemy graf acykliczny reprezentujący zbiory A oraz $A/\check{R}$. W przypadku, kiedy typy są liniowo uporządkowane, powyższe zależności są identyczne i zbiór ilorazowy będzie dokładnie taki sam jak wyżej. W tym względzie, czynnikiem decydującym jest funkcja

gęstości *f*. Jeżeli *f* jest jednorodna, tj. $\forall a \in A, f(a) = const$, to funkcję *f* definiujemy na zbiorze $A' \in A$, gdzie: A' zawiera elementy ze zbioru A, posiadające typy uporządkowane relacją: $\tau_2 < \tau_3 < \cdots < \tau_m$. Funkcja gęstości dla elementów najniższego typu jest równa 0. Wynika stąd, że struktura zbioru ilorazowego posiada postać drzewa. Tak więc, dowolna struktura hierarchiczna może zostać przedstawiona w postaci skierowanego grafu acyklicznego. Zaproponowana definicja doskonale pasuje do opisu hierarchii tworzonej za pomocą wielokanałowości oraz do osadzania. Idea zapisu została po raz pierwszy przedstawiona w [6], a następnie rozwinięta w [4] i [19].

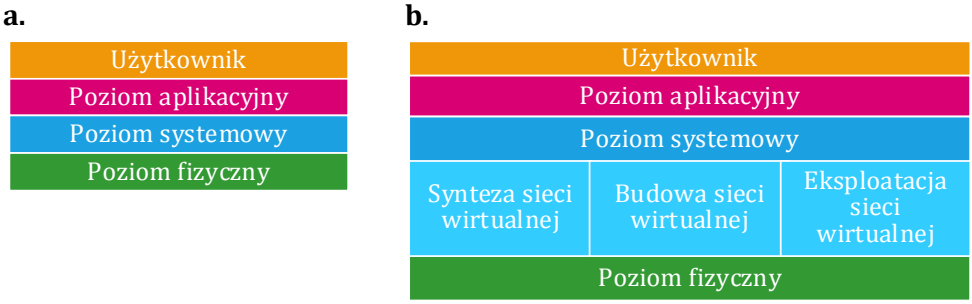
Jako dalsze rozwinięcie wykorzystania grafów acyklicznych do reprezentacji systemów hierarchicznych zaproponowano, poparte stosownymi dowodami zastosowanie lewych jednoznacznych grafów skierowanych (zgodnie z terminologią Weissteina) [15], [20], [21]. Dodatkowo, w celu matematycznego opisu hierarchii postuluje się zastosowanie wielowrotników η_i^n opisanych podstawowym wyrażeniem:

$$\eta_i^n = \left\{ m_{1i}^n, m_{2i}^n, \cdots, m_{(i-1)i}^n, m_{ii}^n, m_{(i+1)i}^n, \cdots, m_{I_n i}^n \right\},$$

gdzie: i – numer poziomu lokalizacji wielowrotnika, $I = 1, \cdots, I_N$, I_N – liczba warstw wykorzystywanej hierarchii; n – numer wielowrotnika w ramach danej warstwy. Wielowrotniki wyposażone są w trzy rodzaje wejść i wyjść, które oznaczamy odpowiednio jako: m_{ki}^n, m_{ii}^n, m_{ji}^n. Wejścia i wyjścia, oznaczone jako m_{ki}^n wiążą *n*-ty wielowrotnik *i*-tej warstwy z *k*-tym wyższym poziomem. Oznaczenie m_{ii}^n opisuje interfejsy w ramach tej samej, *i*-tej warstwy hierarchii. Z kolei, symbol m_{ji}^n dotyczy powiązań *n*-tego wielowrotnika *i*-tego poziomu hierarchii z *j*-tą wyższą warstwą.

Dalsze badania zostały poświęcone analizie wielokanałowości, a w szczególności jej relacjom z hierarchicznością połączeń. Idea zastosowania wielokanałowości do poprawy efektywności komunikacyjnej systemów rozproszonych pojawiła się jeszcze w latach 80-tych XX wieku. Proponowane wtedy rozwiązania były mało atrakcyjne, co wynikało ze stosunkowo niewielkiej przepustowości, a także wysokich kosztów realizacji kanałów. Realna zmiana nastąpiła w latach 90-tych ubiegłego wieku i była związana z wdrożeniem technologii WDM (ang. *Wavelength Division Multiplexing*). Pionierami w tym zakresie byli R. Dutta, L. Fratta, A.E. Kamal, B. Mukherjee, O.G. Morozov, V.A. Burdin i in. opisujący wykorzystanie WDM do budowy podsystemów komunikacji SR. W pracach takich autorów jak: R. Dutta, L. Fratta, A.E. Kamal, B. Mukherjee analizując sieci wielokanałowe ograniczano się do rozwiązań bazujących technologii WDM. Metody opisu komponentów zostały przejęte bezpośrednio z teorii grafów, przy czym, wykorzystywane elementy (węzły, kanały oraz ścieżki) podzielono na fizyczne i logiczne (wirtualne). Zapożyczono z niej również ilościowe i jakościowe charakterystyki sieci (stopnie wierzchołków i topologii, wagi krawędzi, średnicę grafu, długości ścieżek, regularność, symetryczność i in.) oraz sposoby reprezentacji. Wprowadzono także nowe klasy

obiektów, na przykład, ścieżki logiczne sklasyfikowano na: jedno- lub wieloskokowe oraz hybrydowe. W pracach tych rozważane jest wyłącznie nałożenie topologii logicznej na zasoby sieci fizycznej. Analizowane organizacje mają, co prawda, strukturę hierarchiczną, ale z punktu widzenia ich syntezy i analizy, hierarchiczność ma znaczenie trzeciorzędne. Istotę rozwiązań tego typu najlepiej ilustruje zaproponowany przez R. Dutta i zmodyfikowany w [6], model konceptualny klasycznej sieci wielokanałowej (patrz rys. 5).



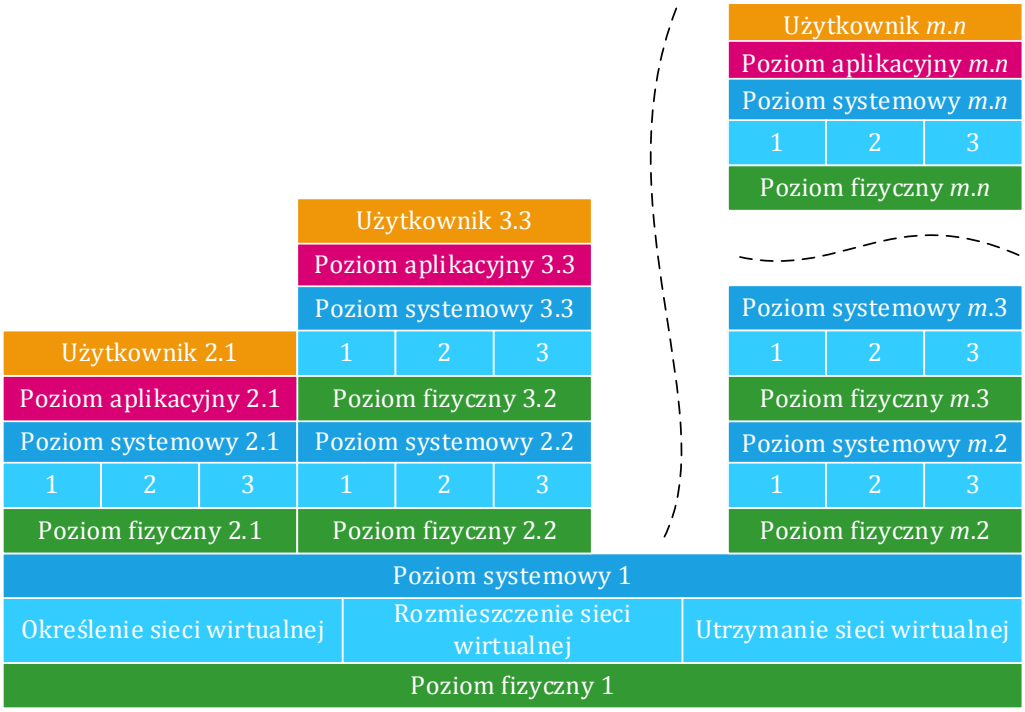
Rys. 5. Modele konceptualne klasycznej stratyfikowanej sieci wielokanałowej: **a.** Ogólna postać modelu; **b.** Model rozwinięty o tworzenie i utrzymanie połączeń wirtualnych

Z punktu widzenia teorii systemów hierarchicznych, model z rys. 5 jest przykładem hierarchii wielostratowej. Podejście takie trywializuje organizację systemu połączeń i ma kilka istotnych wad, z których najważniejsze to: skoncentrowanie się na ścieżkach logicznych, jako podstawowych elementach sieci połączeń; ignorowanie hierarchiczności funkcjonalnych powiązań komponentów architektury komunikacyjnej; ograniczenie do dwóch liczby analizowanych poziomów, uniemożliwiające badanie realnych organizacji; brak parametrów określających korelację fizycznych i logicznych charakterystyk systemu połączeń; zastosowanie klasycznych reprezentacji grafowych lub macierzowych do morfologicznego opisu sieci, ograniczające możliwości jej analizy i syntezy [10], [19], [22]. W tym obszarze, prace koncentrowały się na poprawie skuteczności projektowania poprzez łączenie podetapów projektowania, minimalizacji złożoności czasowej i pamięciowej projektowania, a także tworzeniu takich sieci w warunkach niekompletności informacji [2], [5], [10], [23], [24], [25], [26], [27].

Do eliminacji powyższych wad zaproponowano metody oferowane przez teorie: grafów i wielopoziomowych systemów hierarchicznych. W szczególności, opis architektury połączeń można sprowadzić do postaci wielopoziomowej struktury hierarchicznej, a następnie do jej syntezy i analizy zastosować metody teorii wielopoziomowych systemów hierarchicznych, wykorzystujące optymalizację kombinatoryczną.

W badaniach rozwinięto osadzanie i hierarchizację w odniesieniu do wielokanałowych systemów połączeniowych, przypisując im nowe, rozszerzone znaczenie. Przyjęto, że hierarchizacja wykonywana jest na bazie osadzania, którego narzędziem realizacji jest wielokanałowość. W proponowanym podejściu hierarchizacji

podlegają nie tylko węzły lub ich skupiska, ale przede wszystkim komponenty systemu komunikacji. Zagadnienia zwielokrotniania kanałów komunikacyjnych nie są analizowane, zakłada się dostępność grupy technologii pozwalających utworzyć w kanale wyższego poziomu zbiór niezależnych kanałów logicznych. Nowa struktura ma organizację stratyfikowanej hierarchii wieloeszelonowej i została przedstawiona na rys. 6.



Rys. 6. Model konceptualny stratyfikowanej sieci wieloeszelonowej. Oznaczenia: 1. Określenie sieci wirtualnej; 2. Rozmieszczenie sieci wirtualnej; 3. Utrzymanie sieci wirtualnej

Zaproponowane struktury wielopoziomowe odnoszą się do hierarchii komunikacyjnej i posiadają zmienną liczbę warstw zależną od konkretnej organizacji przesyłu informacji. Nie odzwierciedlają one jednak bezpośrednio funkcjonalnych zależności operacji komunikacyjnych realizowanych przez system rozproszony. Nowy model konceptualny, przedstawiony na rys. 6, uwzględnia wirtualizację komunikacji. W dotychczasowych pracach wielokanałowy system połączeniowy przedstawiano za pomocą struktury trójpoziomowej, której warstwami są poziomy: fizyczny, systemowy i aplikacyjny (patrz rys. 5). Szczególną rolę w modelu tym odgrywa poziom systemowy, na którym wykonywane są wszystkie działania związane z budową i eksploatacją sieci logicznych. Ze stworzonych w ten sposób zasobów korzysta warstwa aplikacji, wyłącznie do której ma dostęp użytkownik. W rzeczywistości poziom aplikacji może pojawiać się na wielu różnych warstwach.

W związku z tym, tradycyjna struktura wielostratowa w systemach z hierarchią topologii wirtualnych nie występuje, a wykorzystywana struktura najbardziej przypomina hierarchię wieloeszelonową. Analiza powyższych zagadnień jest szczególnie istotna przy analizie odporności na uszkodzenia. W klasycznych pracach wskazuje się, że odporność ta jest zapewniana na poziomie fizycznym, a poziom systemowy przeznaczony jest wyłącznie do realizacji rozłącznych ścieżek nadmiarowych. W sieciach w hierarchią topologii wirtualnych zadanie to rozwiązywane jest na każdym z wykorzystywanych poziomów i z punktu widzenia spójności analizowana musi być każda sieć wirtualna [28], [29].

W badaniach rozwinięto wspomniane wcześniej pojęcie *osadzania*. W dotychczasowych badaniach (A.N. Melichow, C. Fernandez, K. Efe, H. Lee) osadzanie (ang. *embedding*) polega wyłącznie na umieszczaniu grafu-gościa w węzłach grafu-gospodarza, do czego wykorzystywana jest funkcja odwzorowania. Rozważmy dwa skończone nieskierowane grafy $G_G = (V_G, E_G)$ i $G_H = (V_H, E_H)$, gdzie: V_G,V_H – zbiory ich wierzchołków; E_G,E_H – zbiory ich krawędzi. Operacja b osadzania grafu G_G w grafie G_H, to bezpośrednie odwzorowanie węzłów V_G w węzły V_H, co można zapisać jako $b: V_G \to V_H$. Graf G_G określany jest nazwą grafu gościa, a graf G_H grafu gospodarza. Interpretację powyższej definicji, w szczególności w odniesieniu do wzajemnych relacji komponentów grafu gościa i gospodarza, a także parametrów sieci wynikowej przedstawiono w [1], [6], [30].

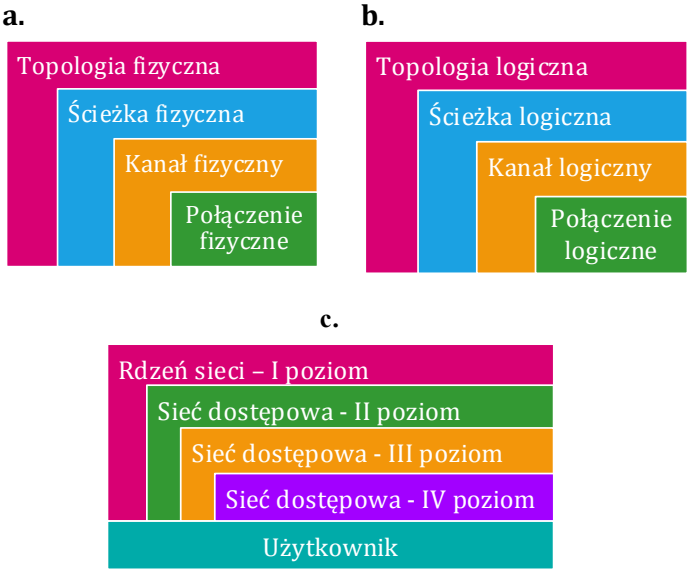
W dotychczasowych pracach (Melichow, P.M. Fernandez, K. Efe) operacja osadzania realizowana była z wykorzystaniem boolowskich operacji na grafach, w szczególności iloczynu kartezjańskiego zapewniającego addytywne i multiplikatywne dziedziczenie parametrów. Koncentrowano się na zdefiniowaniu i analizie dziedziczności przez graf wynikowy takich parametrów grafów gościa i gospodarza jak: liczba węzłów i krawędzi, średnica, stopień wierzchołków, spójność itp. W badaniach, skoncentrowano się na właściwościach komunikacyjnych grafów, powstałych w wyniku osadzania: wprowadzając dla każdego węzła sieci pojęcia ruchu wejściowego i wyjściowego, a dla wybranych organizacji zdefiniowano i udowodniono dziedziczność charakterystyk komunikacyjnych grafów gościa i gospodarza.

Z punktu widzenia badań, ten rodzaj osadzania ma charakter teoriografowy i został nazwany *osadzaniem topologii*. Dotychczasowe badania dotyczące osadzania topologii koncentrują się na jego wykorzystaniu do budowy skalowalnych organizacji topologicznych, sieci sortujących oraz odpornych na uszkodzenia sieci połączeniowych. Jako bazowe wykorzystuje się klasyczne topologie, takie jak: pierścień, drzewo, hipersześcian, toroid itd. Osadzanie topologii jako operacja, zostało rozwinięte w szeregu publikacji [6], [31], [32], [33], w których rozważa się algebrę spójnych skończonych grafów nieskierowanych, a także skalowanie sieci połączeniowych z wykorzystaniem operacji ograniczonego lub nadmiarowego quasi-mnożenia macierzowych reprezentacji grafów.

Z punktu widzenia rozpatrywanych zagadnień, osadzanie topologii ma znaczenie drugoplanowe. Jednak nie należy go lekceważyć, bowiem znalazło ono zastosowanie do: **a.** efektywnego chronienia w pamięci reprezentacji topologii połączeniowych [6], [19]; **b.** projektowania korporacyjnych i regionalnych systemów obliczeniowych z organizacją klastrową i prosto definiowalną strukturą oraz parametrami, określanymi z wykorzystaniem operacji dekompozycji grafu topologii [4]; **c.** efektywnej adresacji węzłów systemu, definiowanej na etapie tworzenia sieci [19], [34]; **d.** minimalizacji złożoności czasowej procedury projektowej [5], [19].

Kolejny rodzaj osadzania, to *osadzanie komponentów komunikacji*. Pojęciem komponentu określono rzeczywiste lub wirtualne środowisko transmisji łączące parę węzłów (logicznych lub fizycznych) tego samego poziomu hierarchii technologii. Przykładami komponentów komunikacji są zarówno kanały jak i ścieżki logiczne oraz fizyczne. Przyjęto, że wielokanałowość dostępna jest we wszystkich wykorzystywanych technologiach i zastosowano ją do realizacji osadzania tego typu. Tak więc, w kanale k-tego poziomu może funkcjonować wiele podkanałów niższego $(k+1)$-ego poziomu, zbiór kanałów k-tego poziomu może zostać osadzony w kanale logicznym poziomu $(k-1)$ lub bezpośrednio w kanale fizycznym (poziom 1) (przykład osadzania pionowego). Ścieżka logiczna k-tego poziomu może zostać osadzona na zbiorze kanałów logicznych tego poziomu, funkcjonujących w różnych kanałach poziomu $(k-1)$ (przykład osadzania poziomego).

Osadzanie komponentów transmisyjnych opisuje hierarchia odzwierciedlająca wzajemne relacje pomiędzy środkami komunikacji międzywęzłowej, nazywana *hierarchią komponentów komunikacji*. Przykład hierarchii przedstawiono na rys. 7.a (komponenty fizyczne); rys. 7.b (komponenty logiczne) oraz rys. 7.c (sieci połączeniowe).

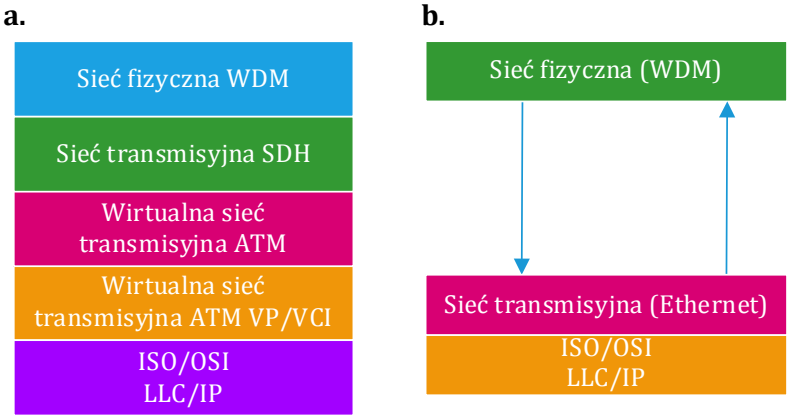


Rys. 7. Przykłady hierarchii komponentów komunikacji

Wszystkie hierarchie komponentów komunikacji mogą być przedstawione za pomocą zaproponowanych wcześniej metod opisu. Osadzanie komponentów może być realizowane tymi samymi metodami co osadzanie topologii, jest jednak wykonywane na niższym poziomie hierarchii.

Trzeci, wyróżniony typ osadzania, to *osadzanie sieci połączeniowych*. W odróżnieniu od osadzania topologii, sieć połączeniowa niższego poziomu jest wpisywana w całą sieć poziomu wyższego, a nie wyłącznie w jej węzeł lub kanał komunikacyjny. Jest to realizowane poprzez osadzanie w sieci $(k-1)$-go poziomu ścieżek logicznych k-tego poziomu, przy zachowaniu relacji ekwiwalentności węzłów różnych poziomów hierarchii. Rezultatem takiego osadzania nie są nowe sieci, a wyłącznie implementacja sieci niższego poziomu na zasobach sieci rozmieszczonej wyżej w hierarchii sieci połączeniowych. Właściwości teorio-grafowe osadzonej sieci są określane wyłącznie przez nią samą, a cechy transportowe przekazywane są przez poziomy wyższe od tego, na którym utworzona została nowa sieć. Osadzanie to można rozpatrywać jako funkcjonalne rozwinięcie osadzania komponentów komunikacji. Przykładem może posłużyć wykorzystywane przy rozwiązaniu problemu RWA (ang. *Routing and Wavelength Assignment*) wpisywanie w sieć fizyczną zbioru ścieżek tworzących sieć logiczną. Znane z literatury (m. in.: R. Dutta, A.E. Kamal, B. Mukherjee) osadzanie sieci ma charakter dwupoziomowy (sieć logiczna osadzana jest na zasobach sieci fizycznej). Do jego opisu można wykorzystać dwupoziomowe struktury rozwarstwione (ang. 2-*fibered structure*), opisane m. in. w [13]. W proponowanym podejściu nie ogranicza się liczby wykorzystywanych poziomów osadzania, tak więc osadzanie tworzy wielopoziomową *hierarchię sieci połączeniowych* (patrz rys. 7.c) [15], [35], [19].

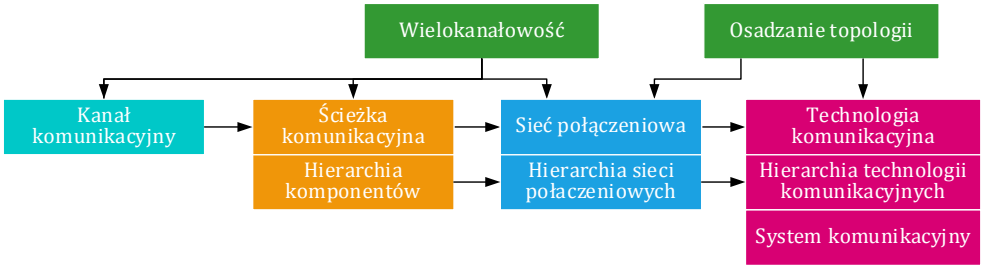
Czwarty i ostatni rodzaj osadzania to *osadzanie technologii*, polegające na umieszczeniu w zasobach jednej technologii komunikacyjnej zasobów innej, wykorzystującej zasoby tej pierwszej. Rozwiązanie takie jest znane i opisywane w literaturze (m. in. przez R. Freemana, D.L Spohna, F. Halsalla, W.M. Wishnevskyego). Przykładem osadzania technologii jest szeroko stosowane alokowanie sieci transmisyjnej Ethernet na zasobach sieci fizycznej WDM. Również w tym przypadku, na podstawie osadzania, tworzona jest *hierarchia technologii*, której warstwy wiążą się z konkretnym poziomem komunikacji, nie zaś z określoną technologią. Przykład hierarchii technologii zaprezentowano na rys. 8.



Rys. 8. Hierarchia technologii komunikacyjnych: a. rozwiązanie historyczne; b. rozwiązanie współczesne

Zaprezentowaną ideę hierarchizacji można zastosować również do sieci fizycznych, w których wyróżniamy rdzeń oraz zbiór sieci dostępowych. W tym przypadku osadzanie nie jest wykorzystywane, a budowana hierarchia ma wyłącznie znaczenie funkcjonalne. Nie zmniejsza to jednak celowości formalnego podejścia do wyboru organizacji tej struktury. Przykład hierarchii tego typu został przedstawiony na rys. 7.c.

Na rys. 9 przedstawiono funkcjonalne powiązania wielokanałowości, osadzania i hierarchii. Można rozważyć połączenie wybranych typów hierarchii, co jednak nie musi wiązać się z poprawą jakości syntezy hierarchii wynikowej. Globalna struktura hierarchiczna zbudowana zostałaby przez osadzenie hierarchii komponentów w hierarchii sieci, a uzyskanego w ten sposób rezultatu, w hierarchii technologii komunikacyjnych [15], [19].



Rys. 9. Powiązanie funkcjonalne wielokanałowości, osadzania i hierarchii

Dalej skoncentrowano się na syntezie i analizie systemów przedstawionych w postaci wielopoziomowych organizacji ze strukturą hierarchiczną. Jest to jeden ze znanych kierunków badania systemów o dużym rozmiarze. Ogólnymi zagadnieniami w tym zakresie zajmowali się m. in. M.D. Mesarovic, T.L. Saaty, L.P. Jennergren, F. Murtagh, P. Willett, Y.B. Germeyer, G.P. Zakharov. Dla wszystkich rozważa-

nych powyżej struktur hierarchicznych (hierarchii: komponentów, sieci, technologii) celowym jest formalne określenie liczby poziomów oraz dobór elementów poszczególnych warstw i sposobów ich powiązania tak, aby powstałą strukturę charakteryzowały minimalne koszty budowy i eksploatacji oraz maksymalna efektywność.

Chociaż zadanie to jest znane od lat, w tym celu, najczęściej bazując na zwartym opisie, definiuje się zbiór dopuszczalnych struktur i kryteriów ich oceny. Rezultatem takich działań są zazwyczaj zalecenia dotyczące wyboru tej lub innej organizacji przeznaczonej dla konkretnych zastosowań. Podejście takie, co prawda, pozwala rozwiązać stawiane zadanie, ale rozwiązanie ogranicza się do obszaru wskazania konkretnego zwartego opisu hierarchii. Ponadto, opisywane zadanie syntezy wykonywane jest wyłącznie na poziomie jakościowym, a modele ilościowe, nie są w ogóle rozpatrywane bądź noszą wyłącznie szczególny charakter. Przed przystąpieniem do syntezy systemów hierarchicznych poczynimy pewne założenia wstępne. Wykorzystując terminologię systemów hierarchicznych, przyjmijmy że węzły straty I_N są zbiorami systemów hierarchicznych, spełniającymi następujące warunki:

1. Połączenie obiektów jednego i tego samego poziomu, dwóch lub więcej systemów w ramach węzła straty I_N, może być zrealizowane na bazie obiektów sąsiednich poziomów, za wyłączeniem fizycznego;
2. Obiekty systemów hierarchicznych, rozmieszczone w różnych węzłach straty I_N łączą się pomiędzy sobą za pośrednictwem poziomów fizycznych tychże systemów.

Oczywiście, powyższe ograniczenia są adresowane do syntezy samej hierarchii, a nie systemu informacyjnego jako całości, nawet jeżeli posiada on organizację hierarchiczną. Do rozwiązania zadania poszukiwania optymalnej hierarchii, podobnie jak wielu innych badaczy, wykorzystano szeroko znane z literatury metody programowania całkowitoliczbowego. W tym celu, rozważmy zbiór grafów dopuszczalnych hierarchii $\{G_i = (V_i, E_i)\}$, gdzie: $I = 1, \cdots, n$. Załóżmy, że poszukujemy hierarchii opisanej grafem $G^* \in \{G_i\}$, charakteryzującej się maksymalną wartością oceny wybranej właściwości lub ich zestawu. W tym celu, analizować będziemy wektor Ψ cząstkowych wskaźników efektywności grafu $G_{i_1} \in \{G_i\}$ określony jako:

$$\Psi\left(G_{i_1}\right) = \left(\psi_1\left(G_{i_1}\right), \cdots, \psi_k\left(G_{i_k}\right), \cdots, \psi_m\left(G_{i_m}\right)\right).$$

Wtedy, funkcja celu będzie miała postać:

$$\max_{G^* \in \{G_i\}} \psi_k\left(G^*\right), \ \forall k = 1, \ldots, m.$$

Można także poszukiwać hierarchii najbardziej zbliżonej do określonej wcześniej struktury wzorcowej lub ich zbioru. Jako $G_c = (V_c, E_c)$ oznaczmy graf docelowej (wzorcowej) hierarchii, gdzie $G_c \in \{G_i\}$. W celu określenia odległości wybranego grafu od grafu docelowej hierarchii, wprowadzono funkcję bliskości ρ. Dla dwóch dowolnych grafów G_{i_1} i G_{i_2} należących do zbioru dopuszczalnych hierarchii (tj.

$G_{i_1}, G_{i_2} \in \{G_i\}$), funkcja $\rho\left(G_{i_1}, G_{i_2}\right)$ określa bliskość pomiędzy nimi. Tak więc, funkcja celu przyjmuje postać:

$$\min_{G^* \in \{G_i\}} \rho\left(G^*, G_c\right).$$

Wykorzystując zintegrowaną definicję obu zadań, możemy zapisać nową funkcję celu:

$$\min_{G^* \in \{G_i\}} \rho\left(G^*, G_c\right), \text{ dla } \psi_k\left(G^*\right) \geq r_k, \ \forall k = 1, \ldots, m,$$

gdzie: r_k – ograniczenia wartości ocenianych właściwości.

Powyższe zadania optymalizacyjne odpowiadają złożonym modelom programowania całkowitoliczbowego lub mieszanego całkowitoliczbowego, dla których można wykorzystać różnorodne metody rozwiązania: od prostego przeszukiwania począwszy, na metodach sztucznej inteligencji skończywszy.

Projektowanie hierarchii zaproponowano w nich sprowadzić do określenia liczby i typów poziomów hierarchii do zadania wielopoziomowego rozmieszczania. Rozwiązanie uzyskiwano za pomocą metody bazującej na liniowej relaksacji oraz rozwiązaniu zadania dualnego z lokalną poprawą jakości. Jeżeli na strukturę projektowanej sieci nakładane są dodatkowe ograniczenia, zweryfikowano wykorzystanie zaimplementowanych metod: poszukiwania drzewa rozpinającego z ograniczonym promieniem; budowy optymalnego drzewa Steinera z limitowanymi długościami ścieżek i ograniczoną ilością punktów Steinera oraz poszukiwania prostokątnego drzewa Steinera z ścieżkami jednakowej długości. Z prac innych autorów wiadomo, że jeżeli niezbędne jest dokładne rozwiązanie zadania, można zastosować metodę podziału i ograniczeń, której praktyczną implementację (do rozwiązania innego zadania) przedstawiono w [36]. Niestety wszystkie powyższe metody należą do grupy algorytmów NP-zupełnych i są mało atrakcyjne dla rozwiązania postawionego zadania. Dlatego, w rozdziale dodatkowo zaproponowano i zweryfikowano algorytm projektowania hierarchii będący modyfikacją algorytmów aglomeracyjnych.

4. Wielokanałowe systemy magistralowe

Początkowo, skoncentrowano się na matematycznej definicji topologii magistralowej opisywanej m. in. w [5], [26], [37]. Klasyczny system obliczeniowy z komunikacją magistralową jest połączeniem dwóch typów równoprawnych obiektów: węzłów obliczeniowych N i magistral B. W takiej architekturze węzeł może być incydentny z dowolną liczbą magistral. Sieć magistralową złożoną z n węzłów i m magistral będziemy oznaczać jako $[n,m]$. Można ją opisać za pomocą macierzy incydencji $A = \{a_{ij}\}$ o rozmiarze $n \times m$. Element $a_{ij} \in A$ jest równy 1, jeżeli z węzłem o numerze $i = 1, \ldots, n$ jest incydentna magistrala o numerze $j = 1, \ldots, m$, w przeciw-

nym przypadku $a_{ij} = 0$. Z definicji macierzy incydencji wynika, że sieci magistralowe nie dopuszczają pętli zarówno dla magistral, jak i dla węzłów. Jeżeli liczbę magistral incydentnych z węzłem i (stopień węzła) oznaczymy jako s_i^w, $i = 1,\ldots,n$, a liczbę węzłów incydentnych z magistralą j (stopień magistrali) jako s_j^m, to dla dowolnej sieci magistralowej $[n,m]$ ma miejsce zależność, opisująca powiązanie pomiędzy sumarycznym stopniem węzłów i magistral:

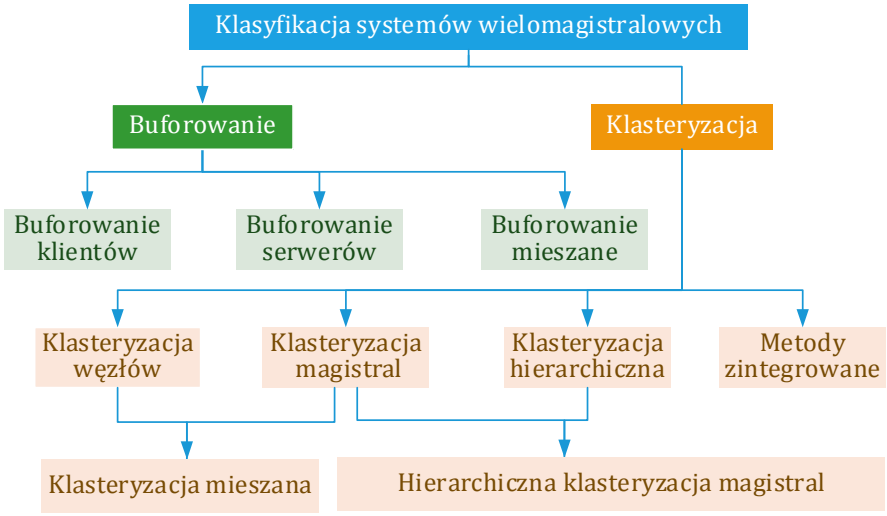
$$\sum_{i=1}^{n} s_i^w = \sum_{j=0}^{m} s_j^m = s\,. \tag{4}$$

W odróżnieniu od sieci z połączeniami bezpośrednimi, dla sieci magistralowej $[n,m]$ z macierzą incydencji A, zawsze istnieje sieć magistralowa z transponowaną macierzą incydencji A^T.

Sieci magistralowe są strukturalnie ekwiwalentne hipergrafom. Do ich analizy można wykorzystać narzędzia teorii grafów, przedstawiając hipergrafy jako grafy dwudzielne. Liczba węzłów w częściach grafu dwudzielnego X_0 i X_1 jest równa odpowiednio n oraz m. Jego krawędziami są połączenia lokalne l, których zadaniem jest powiązanie węzła obliczeniowego z magistralą. Dla hierarchicznych systemów magistralowych zaproponowano grafy wielodzielne, wraz z metodami wykonania podstawowych operacji (w szczególności, doskonałego skojarzenia) [3], [5].

Następnie, zaproponowano architekturę rekonfigurowalnego systemu komputerowego z komunikacją rozgłoszeniową i rozproszoną komutacją kanałów. Idea rozproszonej komutacji na przeniesieniu jej do węzłów końcowych. W odróżnieniu od dotychczasowych prac, w których do opisu systemów magistralowych wykorzystuje się macierzowy zapis grafów dwudzielnych lub trójdzielnych, w rozdziale zastosowano przedstawioną wcześniej algebrę spójnych nieskierowanych grafów, którą można dodatkowo wykorzystać do analizy wybranych architektonicznych parametrów systemu. Do poprawy efektywności funkcjonowania tradycyjnych architektur magistralowych zaproponowano hierarchizację oraz klasteryzację węzłów obliczeniowych i magistral komunikacyjnych. Dla każdego z zaproponowanych rozwiązań opracowano modele pozwalające określić charakterystyki wydajności i złożoności sprzętowej. Do tego celu wykorzystano analizę kombinatoryczną, analizę probabilistyczną z założeniem niezależności oraz analizę probabilistyczną opartą na prawdopodobieństwie warunkowym. Do syntezy oryginalnych struktur magistralowych zaproponowano wykorzystanie grafów na alfabetach, a do ich analizy – grafy wierzchołkowo-krawędziowe [5].

Zaproponowaną klasyfikację metod poprawy charakterystyk komunikacyjnych systemów wielomagistralowych w uproszczonej postaci przedstawiono na rys. 10.



Rys. 10. Klasyfikacja organizacji systemów wielomagistralowych

Metody bazujące na buforowaniu węzłów (klientów i/lub serwerów) mogą być realizowane programowo lub sprzętowo. Ich istota polega na czasowym przechowywaniu żądań obsługi i odpowiedzi na nie, dzięki czemu liczba kolizji przy dostępie do magistral komunikacyjnych jest ograniczana. Niestety, w wybranych przypadkach liczba kolizji przy dostępie do serwerów wzrasta, a tym samym efektywność wykorzystania zasobów obliczeniowych zmniejsza się.

Poprawę wydajności komunikacyjnej należy oprzeć na grupowaniu elementów składowych systemu (węzłów obliczeniowych lub magistral). W pierwszej grupie metod, węzły obliczeniowe (klienci lub serwery), na podstawie kryterium komunikacyjnego łączone są w niezależne grupy, komunikujące się za pomocą wydzielonego zestawu magistral. Oprócz wzorca ruchu, generowanego przez węzły tworzące klaster, kryteriami podziału mogą być również: klasy zadań rozwiązywanych w ramach danej grupy, wymagania dotyczące opóźnień komunikacji, stopy błędów i in. Drugi rodzaj klasteryzacji łączy dostępne w systemie magistrale. W szczególności, są one dzielone na mniejsze, niezależne fragmenty, które mogą być łączone w klastry, obsługujące wybranych użytkowników. Przykładowo, oddzielną grupę mogą tworzyć użytkownicy wykorzystujący usługi niewrażliwe na opóźnienia komunikacyjne, inną – generujący niewielki ruch, zaś jeszcze inną, charakteryzujący się wysoką wybuchowością generowanego ruchu. Trzecia z zaproponowanych metod, jest modyfikacją poprzednich i polega na hierarchizacji logicznych kanałów połączeniowych. Kanały łączone są w grupy obsługujące różne zestawy klientów i serwerów, którzy również mogą być poddawani grupowaniu. Dzięki różnorodności zaproponowanych metod, architekturę połączeń można dopasowywać do występujących w sieci wzorców ruchu i wymagań komunikacyjnych klientów. Ponieważ w systemach optycznych zmiana długości fali, wykorzystywanej przez element na-

dawczo-odbiorczy, trwa milisekundy, dostosowanie architektury połączeń do bieżących wymagań użytkowników może mieć charakter dynamiczny i być realizowana w czasie rzeczywistym. Dla każdej z organizacji z rys. 10 analitycznie określono najważniejsze parametry funkcjonalne (złożoność sprzętową, opóźnienia komunikacji, prawdopodobieństwo blokowania, niezawodność, żywotność i in.) oraz zaproponowano kombinatoryczno-probabilistyczną metodę ich projektowania.

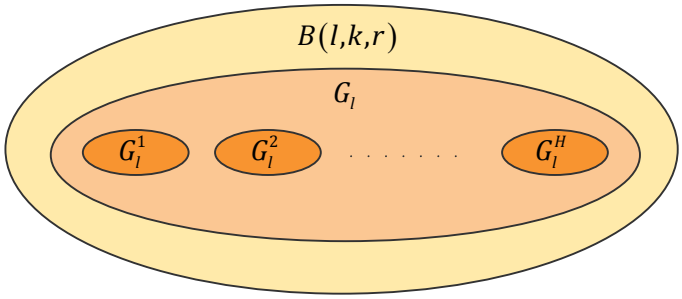
Rozważanym w trakcie badań zagadnieniem było projektowanie zbioru sieci logicznych osadzonych na zasobach jednej sieci fizycznej. Znane metody i metodyki koncentrują się na niezależnej budowie każdej z topologii i przedstawianiu ich w postaci sumarycznej macierzy sąsiedztwa będącej sumą macierzy każdej z topologii logicznych. Zakłada się, że projektowane będzie T topologii logicznych opisanych zbiorem grafów $G_L = \{G_L^t = (V_L^t, E_L^t)\}$, gdzie: V_L^t, E_L^t – odpowiednio zbiory wierzchołków i łuków t-tej topologii logicznej, $t = 1,\ldots,T$. W celu opisu zbioru grafów zaproponowano budowę spójnego, izomorficznego nadgrafu będącego podgrafem grafu na alfabecie $G_{A(l,k,r)}$, gdzie: l – długość słowa kodowego, k – podstawa kodu, r – moc pokrycia słów kodowych. Przedstawione w postaci blokowo-diagonalnej macierze sąsiedztwa każdej z topologii są modyfikowane, następnie przekształcane do postaci połączonej macierzy pokryć wierzchołkowych. Procedura kończy się określeniem rozmiaru grafu $G_{A(l,k,r)}$ i kodowaniem numerów wierzchołków każdej z sieci logicznych w wersji niezależnej lub skorelowanej. Zaproponowany sposób chronienia grafu charakteryzuje niska złożoność pamięciowa, prostota zmiany poszczególnych topologii i możliwość prostego określenia innych charakterystyk sieci (np. węzłów integrujących sieci logiczne).

W celu określenia formalnej definicji zadania oraz sposobów jego rozwiązania rozważmy spójny nieskierowany skończony graf $G_f = (V_f, E_f)$ sieci fizycznej, gdzie: V_f – zbiór węzłów fizycznych (wierzchołków grafu), $|V_f| = n_f$; E_f – zbiór fizycznych kanałów komunikacyjnych (krawędzi grafu), $|E_f| = m_f$. Załóżmy, że na bazie sieci fizycznej G_f zbudowanych zostało H różnych sieci logicznych opisanych za pomocą grafów skierowanych postaci $G_l^h = (V_l^h, E_l^h)$, gdzie: V_l^h – zbiór węzłów h-tej topologii logicznej (wierzchołków grafu), $|V_l^h| = n_l^h$; E_l^h – zbiór skierowanych kanałów komunikacyjnych (łuków grafu) o mocy m_l^h; $h = 1,\ldots,H$. Wykorzystując powyższe oznaczenia, można założyć, że budowa spójnego skierowanego izomorficznego nadgrafu G_l obejmującego zbiór sieci logicznych $\{G_l^h\}$, $h = 1,\cdots,H$, polega na stworzeniu grafu, którego zbiór V_l wierzchołków opisuje wyrażenie: $V_l = \bigcup_{h=1}^{H} V_l^h$, a zbiór łuków: $E_l \supseteq \bigcup_{h=1}^{H} E_l^h$. Z kolei nadgraf G_l ma być podgrafem

grafu de Bruijna $B(l,k,r)$. Oznacza to, że: $\bigcup_{h=1}^{H} G_l^h\left(V_l^h,E_l^h\right)\cong G_l(V_l,E_l)\subseteq B(l,k,r)$. Zgodnie z ustaleniami sieci logiczne tworzą węzły i kanały logiczne, które wykorzystują zasoby sieci fizycznej. Węzły logiczne tworzone są na bazie zasobów węzłów fizycznych, kanały logiczne – kanałów fizycznych. W praktyce może zdarzyć się sytuacja, kiedy sieci logiczne będą rozłączne względem swoich zasobów fizycznych, tj. $V_f^h\cap V_f^i=\varnothing$, gdzie: V_f^h, V_f^i – zbiór węzłów fizycznych, odpowiednio h-tej oraz i-tej sieci logicznej; $h,i=1,\ldots,H$, $h\neq i$. Oznacza to, że w celu ich połączenia i umożliwienia komunikacji pomiędzy nimi konieczne będzie określenie lokalizacji mostów. Dalej mostem będziemy nazywać krawędź grafu (w danym przypadku skierowany kanał komunikacyjny), którego usunięcie zwiększa liczbę komponentów spójności grafu (sieci). Niech M_l oznacza zbiór łuków logicznych łączących wierzchołki logiczne rozłącznych sieci logicznych i $M_l=E_l\setminus\bigcup_{h=1}^{H}E_l^h$. Wtedy, M_l jest uzupełnieniem zbioru E_l łuków nadgrafu sieci logicznych, definicję zadania można przedstawić w postaci: $\bigcup_{h=1}^{H} G_l^h\left(V_l^h,E_l^h\right)\bigcup M_l\cong .\cong G_l(V_l,E_l)\subseteq B(l,k,r)$.

Rozwiązanie zdefiniowanego powyżej zadania składa się z sekwencji następujących kroków: a. budowa i diagonalizacja macierzy sąsiedztwa każdej topologii logicznej; b. Utworzenia macierzy pokrycia wierzchołków dla każdej z topologii logicznych, a następnie budowa macierzy połączonej; c. Analiza powiązań sieci logicznych i wybór sposobu kodowania wierzchołków; d. Określenie rozmiaru nadgrafu obejmującego topologie logiczne będącego podgrafem grafu de Bruijna i rozmiaru tego ostatniego; e. Kodowanie wierzchołków.

Istota metody został zilustrowana graficznie na rys. 11.



Rys. 11. Idea przechowywania zbioru sieci logicznych

W celu określenia efektywności zaproponowanego rozwiązania zastosowano współczynnik zapełnienia pamięci operacyjnej. Dla umożliwienia praktycznego zastosowania zaproponowanej metody zmodyfikowano wybrane algorytmy projektowania sieci logicznych, w szczególności: HLTDA (*Heuristic Logical Topology Design Algorithm*), GLTDA (*Greedy Logical Topology Design Algorithm*) oraz I-MLTDA

(*Increasing Multi-hop Logical Topology Design Algorithm*). Wskazano również sposoby przystosowania zaproponowanej metody do wykorzystania w projektowaniu dokładnym, w szczególności do rozwiązań typu MILP (*Mixed-Integer Linear Programming*).

W celu zdefiniowana zadania projektowego, uwzględniającego jednoczesne tworzenie wielu topologii logicznych zaproponowano i przeanalizowano zestaw ograniczeń oraz funkcji celu. Wykazano, że z uwagi na korelację parametrów transmisji najbardziej efektywnym będzie zastosowanie zlinearyzowanej funkcji minimalizacji opóźnień komunikacyjnych. Z kolei w przypadku optymalizacji wielokryterialnej celowym jest zastosowanie minimalizacji średniej odległości w sieci wraz z minimalizacją maksymalnego obciążenia ścieżek logicznych. Rozwiązanie zadania optymalizacji wielokryterialnej jest realizowane poprzez oryginalne sprowadzenie go do postaci optymalizacji jednokryterialnej. Ponieważ złożoność czasowa algorytmów optymalizacyjnych jest wysoka, nie mogą być one wykorzystywane w przypadku dynamicznego określania nowych topologii sieci, w tym również przy rekonfiguracji wykonywanej bezpośrednio przez urządzenia sieciowe. Dlatego zaproponowano, nowy algorytm heurystyczny, w którym oprócz wymagań komunikacyjnych uwzględniane są wymagania żywotnościowe dla ścieżek logicznych, z założeniem minimalizacji nakładów na realizację sieci połączeń. W odróżnieniu od znanych algorytmów bazuje na losowym poszukiwaniu ścieżek logicznych, wielokrokowym tworzeniu efektywnego rozwiązania oraz gwarantuje pomijanie lokalnych minimów.

Następnie skoncentrowano się na budowie sieci fizycznych wykorzystujących architekturę zagiętych magistral i grafy na alfabecie. Taki wybór topologii był podyktowanych udokumentowanymi względami ekonomiczno-technicznymi. Na wstępie zaproponowano zestaw pasywnych elementów połączeniowych, sposób ich formalnego opisu oraz syntezy, w szczególności na bazie algebry grafów i przekształceń cylindrycznych. Na ich bazie przedstawiono wybrane architektury z pasywnymi połączeniami. Dalej zaproponowano, sklasyfikowano i przeanalizowano metody poprawy efektywności sieci z organizacją magistralową. Dla przedstawionych architektur określono graniczne wartości wybranych parametrów. Z uwagi na wyższy koszt realizacji, omawiając sieci wykorzystujące grafy na alfabecie skoncentrowano się na zapewnieniu podwyższonej dostępności systemu obliczeniowego. W szczególności zdefiniowano i udowodniono zależności spójności krawędziowej, średnicy sieci, liczby wierzchołków od parametrów grafu. Do przechowywania macierzy sąsiedztwa grafów w pamięci, zaproponowano wykorzystanie macierzy.

Trzecia część badań dotyczy rekonfiguracji połączeń optymalnych. Rekonfigurację sieci logicznej możemy rozpatrywać jako problem scentralizowanej optymalizacji, wymagającej globalnego spojrzenia na funkcjonowanie szerokopasmowej sieci komputerowej. Należy zauważyć, że potrzeba rekonfiguracji to konsekwencja zmian ruchu w sieci i nie powinna być ona postrzegana jako wada sieci i nie należy

jej w żadnym przypadku wiązać z koniecznością zmiany architektury sieci fizycznej. Często problem rekonfiguracji jest błędnie utożsamiany w zadaniem projektowania topologii logicznej. Wnoszenie zmian w strukturę topologii logicznej jest trudno akceptowalne ze względu na konieczność modyfikacji połączeń pomiędzy użytkownikami i płynące z tego potencjalne konsekwencje (zakłócenia ruchu). Zauważmy, że problem ten jest problemem wielokryterialnym i nie należy rozważać go wyłącznie jako zadania poprawy wydajności sieci, ale również jako zadanie, którego celem jest minimalizacja liczby zmian wprowadzanych do topologii logicznej. Podstawowymi danymi wejściowymi rekonfiguracji są dotychczasowa sieć logiczna oraz aktualna macierz ruchu międzywęzłowego. Najczęściej zadanie to jest definiowane jako zadanie programowania liniowego, w którym blokowane jest jedno z kryteriów optymalizacji. W ten sposób utrzymując na niezmiennym poziomie wartość jednego z kryteriów, znaczenie drugiego jest minimalizowane. W celu rekonfiguracji zaproponowano metodę, w której optymalizacji w sensie Pareto podlegać będą jednocześnie oba kryteria. Dzięki temu utworzony może być zbiór wielu rozwiązań, w przeciwieństwie do metod opartych na programowaniu liniowym, które oferują wyłącznie jedno jedyne rozwiązanie.

Zaproponowano i zweryfikowano kilka strategii rekonfiguracji, przykładowo zapewnienie niezmiennej średniej odległości w sieci logicznej i minimalizację liczby wnoszonych zmian. Dalej średnią odległość w sieci logicznej określaną za pomocą skoków, odniesioną do istniejących strumieni informacyjnych będziemy nazywać średnią odległością. Procedura rekonfiguracji rozpoczyna się od określenia średniej odległości dla nowego strumienia informacyjnego, bez uwzględnienia liczby zmian w topologii logicznej. Dalej za pomocą zadania programowania całkowitoliczbowego funkcja celu określana jest jako minimalizacja liczby zmian w ścieżkach logicznych, co można zapisać w następujący sposób:

$$\min \sum_{(i,j)} \sum_{[\![m,n]\!]} \sum_{k} \sum_{w,v} \left| b^{'l_k}_{(i,j),[\![m,n]\!],a_{[w,v]}} - b^{l_k}_{(i,j),[\![m,n]\!],a_{[w,v]}} \right|, \tag{5}$$

gdzie: $b^{'l_k}_{(i,j),[\![m,n]\!],a_{[w,v]}}$, $b^{l_k}_{(i,j),[\![m,n]\!],a_{[w,v]}}$ – zmienne binarne określone za pomocą przedstawionego w rozdziale 3 wyrażenia, odpowiednio dla modyfikowanej i podstawowej części systemu. Wyrażenie (5) ma postać liniową, ponieważ zmienne występujące w nim są binarne. Absolutną wartość wyrażenia (5) można przedstawić w następujący sposób:

$$\left| b^{'l_k}_{(i,j),[\![m,n]\!],a_{[w,v]}} - b^{l_k}_{(i,j),[\![m,n]\!],a_{[w,v]}} \right| = \begin{cases} 1 - b^{l_k}_{(i,j),[\![m,n]\!],a_{[w,v]}} & \text{jeżeli } b^{'l_k}_{(i,j),[\![m,n]\!],a_{[w,v]}} = 1 \\ b^{l_k}_{(i,j),[\![m,n]\!],a_{[w,v]}} & \text{jeżeli } b^{'l_k}_{(i,j),[\![m,n]\!],a_{[w,v]}} = 0. \end{cases}$$

Nowe ograniczenie dla programowania całkowitoliczbowego przyjmuje postać:

$$\frac{1}{\sum_{sd,x} \Lambda^{x}_{sd}} \sum_{(i,j)} \sum_{sd,x} \left(x \lambda^{x}_{sd,(i,j)} \right) \le d \,,$$

gdzie: d – analizowana odległość.

5. Podsumowanie

Badania prowadzone w ramach projektu pokazały skuteczność wykorzystania hierachizmu i wielokanałowości do budowy sieci połączeniowych nowej generacji. Projektowanie takich sieci będzie tematem dalszych badań autorów.

Bibliografia

[1] M. Hajder and J. Kolbusz, *Formalizacja projektowania architektury systemów informacyjnych*. Rzeszów: Wyższa Szkoła Informatyki i Zarządzania z siedzibą w Rzeszowie, 2015.

[2] M. Hajder, *Algebraiczne metody rozwiązania zadań projektowych w warunkach nieokreśloności*. Rzeszów: Wyższa Szkoła Informatyki i Zarządzania z siedzibą w Rzeszowie, 2013.

[3] M. Hajder, J. Kolbusz, and M. Nycz, *Algorytmika projektowania systemów informacyjnych*. Rzeszów: Wyższa Szkołą Informatyki i Zarządzania z siedzibą w Rzeszowie, 2014.

[4] M. Hajder and M. Nycz, *Sieci regionalne i korporacyjne. Metody projektowania i eksploatacji*. Rzeszów: Wyższa Szkoła Informatyki i Zarządzania z siedzibą w Rzeszowie, 2015.

[5] M. Hajder, *Architektura rekonfigurowalnych systemów magistralowych. Synteza i analiza rozwiązań*. Rzeszów: Wyższa Szkoła Informatyki i Zarządzania z siedzibą w Rzeszowie, 2015.

[6] M. Hajder, *Methods and facilities increasing effectiveness of designing distributed systems based on multichannels. Dissertation for the degree Doctor of Technical Science*. Kyiv: National Technical University of Ukraine “Kyiv Polytechnic Institute”, 2006.

[7] M. Hajder, "Studium rozwoju systemów informacyjnych: wirtualizacja, chmura, klaster, grid," in *Innowacyjna gmina. Produkcja informacyjna i edukacja*, M. Hajder, Ed. Rzeszów: Wyższa Szkoła Informatyki i Zarządzania z siedzibą w Rzeszowie, 2016, pp. 157-169.

[8] M. Hajder and P. Hajder, "Stacjonarne i mobilne sieci obliczeń brzegowych - nowe wyzwania," in *Innowacyjna gmina. Produkcja informacyjna i edukacja*, M. Hajder, Ed. Rzeszów: Wyższa Szkoła Informatyki i Zarządzania z siedzibą w Rzeszowie, 2016, pp. 5-16.

[9] M. Hajder, P. Dymora, and M. Mazurek, "Improvement methods of multichanel bus topologies parameters," in *Sixth International Conference Soft Computing and distributed processing*, Rzeszów, 2002, pp. 172-175.

[10] M. Hajder and M. Bolanowski, "Reconfigurable multichannel computational systems," in *Proceedings of the 3rd International Conference ACSN-2007*, Lviv, 2007.

[11] M. Hajder and M. Nycz, "Tendencje rozwojowe systemów i sieci komputerowych w latach 1990-2012," in *Innowacyjna gmina. Wydajność i efektywność*, M. Hajder, Ed. Rzeszów: Wyższa Szkoła Informatyki i Zarządzania z siedzibą w Rzeszowie, 2012, pp. 7-24.

[12] M. Hajder, H. Loutskii, and W. Stręciwilk, *Informatyka. Wirtualna podróż w świat systemów*, M. Hajder, Ed. Rzeszów: Wydawnictwo Wyższej Szkoły Informatyki i Zarządzania z siedzibą w Rzeszowie, 2002.

[13] M. Hajder and T. Bartczak, "Teoriografowy opis wielokanałowych wirtualnych połączeń hierarchicznych," in *Innowacyjna gmina. Wydajność i efektywność*, M. Hajder, Ed. Rzeszów: Wyższa Szkoła Informatyki i Zarządzania z siedzibą w Rzeszowie, 2012, pp. 37-57.

[14] M. Hajder and P. Hajder, "Architektura magistralowych systemów połączeniowych. Metody i środki analizy i syntezy," in *Innowacyjna gmina. Produkcja informacyjna i edukacja*. Rzeszów: Wyższa Szkoła Informatyki i Zarządzania z siedzibą w Rzeszowie, 2016, pp. 77-95.

[15] M. Hajder, M. Nycz, and P. Różycki, "Analiza funkcjonalno-eksploatacyjnych charakterystyk hierarchicznych sieci komunikacyjnych," in *Innowacyjna gmina. Informatyka w jednostkach samorządu terytorialnego*, M. Hajder, Ed. Rzeszów: Wyższa Szkoła Informatyki i Zarządzania z siedzibą w Rzeszowie, 2014, pp. 189-210.

[16] M. Hajder, M. Nycz, and J. Kolbusz, "Grafowe reprezentacje obiektów technicznych," in *Innwacyjna gmina. Informatyka w jednostkach samorządu terytorialnego*, M. Hajder, Ed. Rzeszów: Wyższa Szkoła Informatyki i Zarządzania z siedzibą w Rzeszowie, 2014, pp. 163-172.

[17] F. Larumbe and B. Sansò, "Optimal Location of Data Centers and Software Components in Cloud Computing Network Design," in *Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, Washington, DC, 2012, pp. 841-844.

[18] D. L. Spohn, *Data Network Design*. New York: McGraw-Hill, 2002.

[19] M. Hajder, *Metody teoriografowego opisu architektury systemów technicznych*. Rzeszów: Wyższa Szkoła Informatyki i Zarządzania z siedzibą w Rzeszowie, 2015.

[20] Douglas B. West, *Introduction to Graph Theory*, 2nd ed. New Jersey: Prentice Hall, 2000.

[21] R. Diestel, *Graph Theory*, 4th ed. New York, USA: Springer, 2010.

[22] M. Hajder, M. Bolanowski, and P. Dymora, "Models of multichannel interconnection systems," *Annales Universitatis Mariae Curie-Skłodowska - Informatica*, vol. 3, no. 1, pp. 293-304, 2005.

[23] M. Hajder, "Rekonfigurowalne środowiska obliczeniowe na bazie wielokanałowych wirtualnych połączeń hierarchicznych," in *Innowacyjna gmina. Wydajność i efektywność*, M. Hajder, Ed. Rzeszów: Wyższa Szkołą Informatyki i Zarządzania z siedzibą w Rzeszowie, 2012, pp. 25-36.

[24] M. Hajder and M. Bolanowski, "Organizacja komunikacji w wielomagistralowych systemach połączeniowych z pasywną komutacją," in *XV Konferencja Sieci i Systemy Informatyczne - SIS 2007*, Łódź, 2007.

[25] M. Hajder, M. Bolanowski, and A. Paszkiewicz, "Projektowanie topologiczne sieci WDM," in *X Konferencja Sieci i Systemy Informatyczne - Materiały konferencji*, Łódź, 2002, pp. 77-89.

[26] M. Hajder and M. Bolanowski , "Connectivity Analysis in the Computational Systems with Distributed Communications in the Multichanel Environment," *POLISH JOURNAL OF ENVIRONMENTAL STUDIES*, vol. 17, no. 2A, pp. 14-18, 2008.

[27] M. Hajder, M. Bolanowski, and A. Hanus, "The topological design of optical interconnection systems - selected problems," in *Scalable Systems and Computer Networks SCALNET-04*, Kremenchuk - Ukraine, 2004, pp. 16-22.

[28] M. Hajder and P. Dymora, "Algorithmical and topological methods of fault tolerance assurance," *Annales Universitatis Mariae Curie-Skłodowska - Informatica*, no. 2, pp. 143-151, 2004.

[29] M. Hajder, P. Dymora, and M. Bolanowski, "Fault tolerant in multichannel scalability systems," in *International Conference of Science and Technology SCALNET-04*, Kremenchuk - Ukraine, 2004, pp. 52-57.

[30] M. Hajder and M. Mazurek, "Grafy regularne w projektowaniu wielokanałowych optycznych systemów komunikacji," *Zeszyty naukowe Politechniki Rzeszowskiej - seria Informatyka*, vol. 205, no. 1, pp. 5-20, 2003.

[31] M. Hajder, P. Dymora , and M. Bolanowski , "Models of multichannel interconnection systems," *Annales Universitatis Mariae Curie-Skłodowska – Informatica*, vol. 3, pp. 293-304, 2005.

[32] M. Hajder and M. Mazurek, "Embending as a method to improve multichanel bus topology parameters," *Annales Universitatis Mariae Curie-Skłodowska, Sectio Ai Informatica*, 2004.

[33] S. Krivoi, M. Hajder, and M. Mazurek, "The computer network topologies designs and the coherent graphs algebra," *Annales Universitatis Mariae Curie-Skłodowska - Informatica*, vol. 5, pp. 379-393, 2006.

[34] M. Hajder and J. Kolbusz, "Minimalizacja złożoności pamięciowej grafowych modeli obiektów technicznych," in *Innowacyjna gmina. Informatyka w jednostkach samorządu terytorialnego*, M. Hajder, Ed. Rzeszów: Wyższa Szkoła Informatyki i Zarządzania z siedzibą w Rzeszowie, 2014, pp. 211-226.

[35] M. Hajder, M. Bolanowski, and P. Dymora, "Models of designing interconnection systems problem," in *XII Konferencja: Systemy i Sieci Informatyczne*, vol. 1, Łódź, 2004, pp. 77-85.

[36] M. Stroiński, Ed., *Studium rozwoju sieci szerokopasmowej województwa Podkarpackiego*. Poznań: Poznańskie Centrum Superkomputerowo-Sieciowe, 2008.

[37] M. Hajder and P. Dymora , "A novel approach to fault tolerant multichannel networks designing problems," *ANNALES UNIVERSITATIS MARIAE CURIE-SKŁODOWSKA, SECTIO AI: INFORMATICA* , no. 1, pp. 66-77, 2011.

[38] M. D. Mesarovic, D. Macko, and Y. Takahara, *Theory of hierarchical, multilevel systems*. New York: Academic Press, 1970.

[39] N. J. Smith and A. P. Sage, *An introduction to hierarchical systems theory*. Dallas, Texas: Information and Control Sciences Center, SMU Institute of Technology, 2005.

[40] Ch. S. Wasson, *System Analysis, Design, and Development: Concepts, Principles, and Practices*. New Jersey: Wiley-Interscience, 2005.

[41] T. Trzaskalik, *Modelowanie optymalizacyjne*. Łódź: Przedsiębiorstwo Specjalistyczne Absolwent, 2001.

[42] M. Stasiak, M. Głąbowski, and P. Zwierzykowski, *Modelowanie i wymiarowanie ruchomych sieci bezprzewodowych*. WKiŁ: Warszawa, 2009.

[43] R. B. Northrop, *Introduction to Complexity and Complex Systems*. New York: CRC Press, 2010.

[44] W. L. Butler, "Hints for computer system design," *SIGOPS Oper. Syst. Rev.*, vol. 17, no. 5, pp. 33-48, May 1983.

[45] N. Boccara, *Modeling Complex Systems*, 2nd ed. New York: Springer, 2010.

[46] M. P. Clark, *Networks and Telecommunications: Design and Operation*. Chicheste: John Wiley & Sons, 1991.

[47] S. Evans, *Telecommunications Network Modeling, Planning and Design*. London: The Institution of Electrical Engineers, 2003.

[48] A. Levitin, *Introduction to the Design and Analysis of Algorithms*. Phenix: Addison Wesley, 2006.

[49] J. D. McCabe, *Network Analysis, Architecture and Design*, 3rd ed. New York: Morgan Kaufmann, 2007.

[50] J. Mendez-Rangel and C. Lozano-Garzon, "A network design methodology proposal for E-health in rural areas of developing countries," in *6th Euro American Conference on Telematics and Information Systems*, Valencia, 2012, pp. 1-7.

[51] T. L. Saaty, *Fundamentals of Decision Making with the Analytic Hierarchy Process*. Pittsburgh: RWS Publications, 2000.

[52] R. V. Slyke. (2001) Network planning and design. ftp://ftp.shore.net/members/ws/Support/BDC/nd.pdf.

[53] L. Kleinrock, "Analytic and simulation methods in computer network design," in *Proceedings of the May 5-7, 1970, spring joint computer conference*, Atlantic City, New Jersey, 1970, pp. 569-579.

[54] V. Alwayn, *Optical Network Design and Implementation*. Indianapolis: Cisco Press, 2004.

[55] R. Nance, R. Moose, and R. Foutz, "A statistical technique for comparing heuristics: an example from capacity assignment strategies in computer network design," *J Commun. ACM*, vol. 30, no. 5, pp. 430-442, 1987.

[56] J. M. Simmons, *Optical Network Design and Planning*. New York: Springer, 2010.

[57] X. Shuizhen, "Planning, designing and building large-scale network at campus," in *3rd International Conference on Computer Research and Development*, Shanghai, 2011, pp. 495-498.

[58] P. Oppenheimer, *Top-Down Network Design Second Edition*. Indianapolis: Cisco Press, 2004.

[59] M. Pioro and D. Medhi, *Routing, Flow, and Capacity Design in Communication and Computer Networks*. San Francisco: Morgan Kaufmann, 2004.

[60] E. Lawler, *Combinatorial optimization: networks and matroids*. New York: Dover Publications, 2011.

[61] Ch. S. Wasson, *System Analysis, Design, and Development: Concepts, Principles, and Practices*. New Jersey: Wiley-Interscience, 2005.

[62] B. Korte and J. Vygen, *Combinatorial Optimization. Theory and Algorithms*, III ed. Berlin: Springer-Verlag, 2006.

[63] E. L. Lawler, *Combinatorial Optimization: Networks and Matroids*. New York: Holt, Rinehart and Winston, 1976.

[64] M. Grotschel , L. Lovasz , and A. Schrijver , *Geometric Algorithms and Combinatorial Optimization*. Berlin: Springer Verlag, 1988.

[65] K. Lange, *Optimization*. New York: Springer Verlag, 2004.

[66] М. Хайдер, "Реконфигурируемые вычислительные системы с многоканальными магистральными связями," *Управляющие системы и машины*, no. 1, pp. 30-38, 2006.

[67] М. Хайдер, "Эффективный метод проектирования и реконфигурации вычислительной системы с многоканальными связями," *Проблемы программирования*, no. 2-3, pp. 701-707, 2006.

[68] M. Hajder, P. Dymora, and M. Mazurek, "Projektowanie topologii transparentnych sieci optycznych," in *Konferencja Polski Internet Optyczny: technologie, usługi i aplikacje*, Poznań, 2002, pp. 183-195.

[69] M. Hajder, M. Mazurek, and P. Dymora, "Topologie wirtualne wielowęzłowych sieci rozległych," in *Konferencja Polski Internet Optyczny: technologie, usługi i aplikacje*, Poznań, 2002, pp. 197-211.

[70] M. Hajder, A. Paszkiewicz, and M. Bolanowski, "Koszty budowy i eksploatacji systemów komunikacji optycznej," *Zeszyty Naukowe Politechniki Rzeszowskiej - seria Informatyka*, vol. 205, no. 1, pp. 21-30, 2003.

[71] M. Hajder and A. Paszkiewicz, "Efektywna metoda wielokryterialnego projektowania sieci komputerowych," in *Wysokowydajne sieci komputerowe. Nowe technologie*. Warszawa: Wydawnictwa Komunikacji i Łączności, 2005, pp. 221-232.

[72] M. Hajder, M. Bolanowski, and A. Paszkiewicz, "Design of improvement efficiency of multichannel optical communication system criteria optimization," *Annales Universitatis Mariae Curie-Skłodowska - Informatica*, vol. 1, pp. 271-276, 2003.

[73] M. Hajder, A. Paszkiewicz, and M. Bolanowski, "Planowanie rozmieszczenia zasobów w rozproszonym systemie obliczeniowym," *Zeszyty Naukowe Politechniki Rzeszowskiej - seria Informatyka*, no. 2, pp. 33-42, 2003.

[74] M. Hajder and M. Nycz, "Synteza i analiza hierarchii systemów informacyjnych," in *Innowacyjna gmina. Wydajność i efektywność*, M. Hajder, Ed. Rzeszów: Wyższa Szkoła Informatyki i Zarządzania z siedzibą w Rzeszowie, 2012.

[75] M. Hajder and M. Mazurek, *Wybrane aspekty syntezy i analizy systemów komunikacji optycznej. Współczesne problemy sieci komputerowych - nowe technologie.*: WNT, 2004.

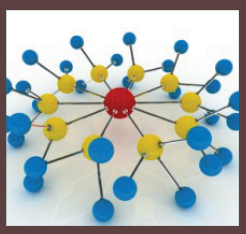
[76] M. Hajder and P. Hajder, *Internet dla szkół. Analiza projektu Ogólnopolskiej Sieci Edukacyjnej*. Warszawa: Naukowa i Akademicka Sieć Komputerowa, 2016.

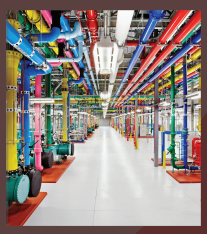
[77] М. Хайдер, "Вопросы повышения эффективности компьютерных систем и сетей на основе оптических каналов передачи данных," *Вісник Національного технічного університету України "Київський політехнічний інститут". Інформатика, управління та обчислювальна техніка*, vol. 39, pp. 79-92, 2002.

[78] Jorgen Bang-Jensen and Gregory Z. Gutin, *Digraphs: Theory, Algorithms and Applications*. London: Springer, 2010.

[79] Dorothea Blostein and Young-Bin Kwon, *Graphics Recognition. Algorithms and Applications: 4th International Workshop*. New York: Springer, 2002.

[80] B. Bollobas, *Modern Graph Theory*. New York: Springer, 1998.





W ręce czytelników oddajemy kolejną monografię Innowacyjna gmina, poświęconą problemom formalizacji poszukiwania efektywnych rozwiązań. Podobnie jak poprzednio, staraliśmy się pogodzić wysoki poziom zawartych w monografii rozdziałów, z przystępnością niezbędną czytelnikom wywodzącym się z organów administracji samorządowej.

Projektowanie – pierwszy z wiodących tematów – jest działaniem niemal stworzonym do formalizacji. Konstruowanie większości obiektów technicznych może być w różnym stopniu wspomagane przez programy komputerowe oparte o różnorodne, mniej lub bardziej zaawansowane, algorytmy. Postępy w tym obszarze poczynione w ostatnim dziesięcioleciu są na tyle znaczące, że oprogramowanie wspomaga człowieka również przy doborze strony estetycznej projektów. Prace zawarte w monografii dotyczą, przede wszystkim, projektowania sensorowych systemów monitoringu środowiskowego.

W potocznym rozumieniu, optymalizacja jest poszukiwaniem rozwiązania najlepszego z punktu widzenia zdefiniowanych kryteriów. Dzięki istnieniu oprogramowania komputerowego zastosowanie metod optymalizacji jest stosunkowo proste, a efektów jego wykorzystania trudno przecenić. Rzadkie ich zastosowanie wynika zazwyczaj z braku doświadczenia, wielu urzędnikom, w szczególności samorządowym, brakuje umiejętności praktycznego korzystania z wiedzy teoretycznej.

WYŻSZA SZKOŁA
INFORMATYKI i ZARZĄDZANIA
z siedzibą w Rzeszowie