

# SIECI KOMPUTEROWE – MYŚLENIE KOMPUTACYJNE W PRAKTYCE

Mariusz Piwiński  
Instytut Fizyki

Wydział Fizyki, Astronomii i Informatyki Stosowanej  
Uniwersytet Mikołaja Kopernika w Toruniu  
ul. Grudziądzka 5, 87-100 Toruń  
[Mariusz.Piwinski@fizyka.umk.pl](mailto:Mariusz.Piwinski@fizyka.umk.pl)

*Abstract. Computational thinking is an iterative process based on three stages: abstraction (formulation of the problem), automation (creating solution) and analyses (implementation of the solution and evaluation). Such methodology can be also applied for teaching complex issues of computer networks. This work presents simple exemplary scenarios of laboratories requiring analytical and computational thinking.*

## 1. Wstęp

Sieci komputerowe stanowią jeden z najszybciej rozwijających się działów technologii informacyjno-komunikacyjnych ICT, które w błyskawiczny sposób z rozwiązań typowo biznesowych zostały przeniesione do prawie wszystkich sfer życia codziennego. Stało się tak, między innymi, za sprawą powszechnego dostępu do Internetu, który jest wykorzystywany do realizacji różnorodnych usług. Niezależnie mamy do czynienia z coraz większą liczbą urządzeń, których typowa funkcjonalność rozszerzana jest poprzez podłączenie do sieci teleinformatycznej. Dotyczy to nie tylko urządzeń mobilnych takich jak smartfony czy tablety, ale również urządzeń codziennego użytku takich jak telewizory, lodówki, moduły oświetleniowe czy nawet zamki drzwiowe. Bezpośrednim przejawem tego typu działań jest zjawisko nazywane Internetem Rzeczy (ang. *Internet of Things*) [1]. Częściami składowymi IoT są:

- **Urządzenia**, pozwalające na gromadzenie, przesyłanie danych oraz wykonywanie pewnych działań,
- **Systemy teleinformatyczne**, zajmujące się przetwarzaniem informacji,
- **Sieci teleinformatyczne**, służące do łączenia poszczególnych urządzeń oraz systemów teleinformatycznych,

- **Procesy**, wykorzystywane do przetwarzania informacji, podejmowania działań, sterowania urządzeniami oraz prognozowania zachowań.

Podstawowym celem Internetu Rzeczy jest stworzenie inteligentnych obszarów, czyli budynków, miast, sieci energetycznych, samochodów czy autostrad, które będą wspierać ludzi zarówno w pracy jak i życiu codziennym.

Realizacja takiego podejścia wymaga budowy oraz utrzymywania wydajnych sieci teleinformatycznych. To z kolei pociąga za sobą coraz większe wymagania stawiane administratorom. Dotyczą one nie tylko wiedzy oraz umiejętności praktycznych, ale przede wszystkim logicznego i analitycznego myślenia, a także umiejętności szybkiego wyciągania prawidłowych wniosków i rozwiązywania pojawiających się problemów. Takie działania wpisują się w myślenie komputacyjne, które obecnie stosowane jest w wielu dziedzinach życia, w tym również w edukacji [8].

## 2. Fundamentem jest wiedza

W opracowaniach dotyczących metodyki nauczania można spotkać wiele rozwiązań mających ułatwić przyswajanie wiedzy. Typową propozycją jest odejście od wiedzy encyklopedycznej na rzecz podejścia praktycznego. W takim ujęciu pozornie mogłoby się wydawać, iż uczenie się sztywnych reguł czy definicji może być startą czas. Jednakże należy podkreślić, iż często są one niezbędne do precyzyjnego określenia języka komunikacji oraz zdefiniowania poszczególnych pojęć, sposobów działania funkcji czy procesów. O ile samo nauczanie się na pamięć sztywnych regulek bez ich zrozumienia nie ma sensu, to trzeba zaznaczyć, iż nie da się zrozumieć pewnych procesów bez precyzyjnego określenia ich definicji. Dlatego o myśleniu problemowym, czy też komputacyjnym można mówić wyłącznie w oparciu o posiadaną solidną wiedzę. W przeciwnym przypadku wyciągane wnioski będą błędne i nie będą prowadziły do rozwiązania napotkanych problemów.

Dobrym przykładem może być uczenie zagadnień z zakresu sieci komputerowych. Na samym początku należy stwierdzić, iż tematyka ta jest bardzo złożona. Co więcej stosowane rozwiązania cały czas ewoluują, co czyni sytuację jeszcze bardziej skomplikowaną. W związku z tym nauczając tych zagadnień najczęściej rozpoczyna się od omówienia dwóch modeli warstwowych używanych do opisu działania sieci. Są nimi Model TCP/IP oraz Model OSI [7]. Na ich przykładzie opisyje się w sposób podstawowy elementy komunikacji, protokoły pracujące w poszczególnych warstwach oraz sam proces przetwarzania danych. Pojawiają się takie pojęcia jak opakowanie i rozpakowanie danych, komunikacja między warstwami, a przede wszystkim stos protokołów sieciowych [6]. Taka teoretyczna podbudowa ma uzmysłowić słuchaczom, iż mamy tutaj do czynienia nie z pojedynczym procesem, a z wieloma współpracującymi ze sobą technologiami i rozwiązaniami.

Modułowy charakter tego rozwiązania pozwala na łatwe dokonywanie modyfikacji poszczególnych funkcji bez potrzeby zmiany ogólnego sposobu obsługi danych.

Po takim wprowadzeniu warto zająć się aspektem praktycznym, który polega na przyporządkowaniu poszczególnym warstwom odpowiednich ról oraz określeniu urządzeń, które mogą realizować poszczególne funkcje. Dopiero dysponując taką wiedzą można rozpocząć analizę sposobu działania sieci komputerowej. Stosując dobre praktyki mówiące o tym, że najlepiej przyswajamy wiedzę podczas ćwiczeń praktycznych, można tutaj zaproponować wiele zadań pozwalających na weryfikowanie i utrwalanie zdobytej wiedzy. Do realizacji tego typu ćwiczeń wystarczy jedynie komputer podłączony do sieci, wyposażony w analizator sieciowy Wireshark [5].

### 3. Analiza danych – proste zadania problemowe

Scenariusz praktycznej analizy przesyłanych danych może polegać na wykorzystaniu komendy **ping** w celu sprawdzenia osiągalności interfejsu zdalnego hosta. Podczas badania przesyłane dane zostaną przechwycone przez program Wireshark, co pozwoli na ich analizę na tle modelu OSI. Rysunek 1 przedstawia wynik komendy **ping** wywołanej na interfejs bramy domyślnej. Przedstawione ćwiczenie pozwala na zastosowanie posiadanej wiedzy w praktyczny sposób podczas analizy i interpretacji przechwyconych danych. Podczas realizacji zadania w naturalny sposób pojawia się omówiony model warstwowy, gdyż zastosowane oprogramowanie nie tylko pozwala na przechwytywanie przesyłanych danych, ale również na ich interpretację na tle modelu OSI oraz datagramów protokołów stosu TCP/IP (Rysunek 2).

```
C:\Users\ping 158.75.5.254

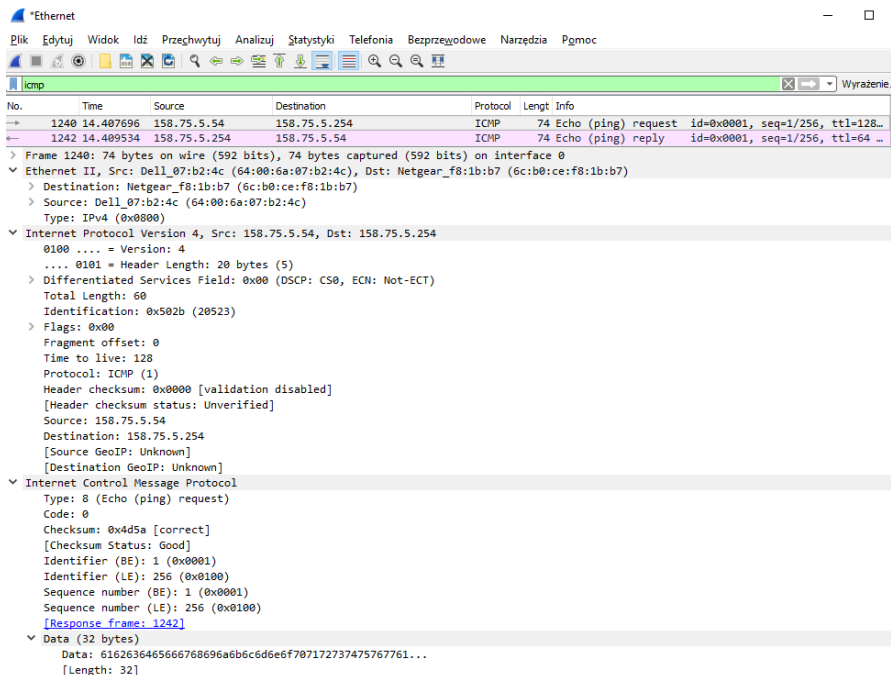
Pinging 158.75.5.254 with 32 bytes of data:
Reply from 158.75.5.254: bytes=32 time=1ms TTL=64
Reply from 158.75.5.254: bytes=32 time=1ms TTL=64
Reply from 158.75.5.254: bytes=32 time=1ms TTL=64
Reply from 158.75.5.254: bytes=32 time=1ms TTL=64

Ping statistics for 158.75.5.254:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 1ms, Average = 1ms
```

**Rysunek 1** Wynik testu osiągalności interfejsu bramy domyślnej przeprowadzonego za pomocą komendy ping.

W analizowanych danych można zauważyć, iż w warstwie drugiej wykorzystywana jest technologia Ethernet, a dane zostały przesłane w postaci ramki z wypełnionymi adresami MAC oraz polem definiującym protokół warstwy 3 (IPv4 0x0800). Następnie widoczny jest opakowany datagram protokołu IPv4 z polami adresowymi (źródłowym 158.75.5.54 i docelowym 158.75.5.254), polem TTL (128) oraz typem obsługiwanego protokołu. W tym przypadku nie jest to protokół warstwy wyższej

a wartość 1 określa protokół ICMP (ang. *Internet Control Message Protocol*). W zależności od poziomu zaawansowania słuchaczy można sformułować kolejne pytania dotyczące znaczenia poszczególnych pól protokołu IPv4 takich jak identyfikacja (0x502b), flagi (0x00) czy fragment offset (0). W toku kolejnych ćwiczeń uczestnicy mogą badać, czy pola te przyjmują takie same wartości dla innych pakietów związanych z komunikacją ICMP, czy też być może ich wartości są unikalne.



**Rysunek 2** Dane przechwycone w programie Wireshark związane z testem osiągalności interfejsu bramy domyślnej

W kolejnym kroku przechodzimy do analizy datagramu ICMP, który zawiera pola typ (8) oraz kod (0). Program Wireshark analizując te wartości interpretuje je zgodnie ze specyfikacją RFC 792 [2] jako żądanie echo (echo request). W tego typu zadaniach ważna jest umiejętność powiązania wartości odpowiednich pól datagramów z informacją zwracaną użytkownikowi przez system operacyjny. Zatem jest to najlepszy moment do zadania kolejnych pytań. W wyniku wykonania testu ping, na ekranie pojawia się informacja, iż testowanie trwa krócej niż 1 ms i odbywa się przy użyciu 32 bajtów danych (Rysunek 1). Ale co to oznacza w praktyce? Z analizy prowadzonej w programie Wireshark wynika, iż czas liczony jest od wysłania zapytania (echo request) do otrzymania odpowiedzi (echo replay). Widoczne jest to w znacznikach czasowych przyporządkowanych każdej analizowanej ramce.

Badanie odpowiedzi ping zmusza osobę wykonującą zadanie do ponownej analizy opakowanych danych tym razem otrzymanych od bramy. W związku z powyższym widoczna jest zmiana relacji pomiędzy źródłowymi i docelowymi adresami MAC oraz IP, a także zmiana typu protokołu ICMP na 0 odpowiadającemu wiadomości echo replay. Podczas takich działań ponownie utrwalana jest wiedza z zakresu enkapsulacji danych w stosie protokołów TCP/IP. Kolejne pytania mogą dotyczyć widocznej na ekranie wartości TTL oraz zawartości przesyłanych danych. Aby odpowiedzieć na te pytania po raz kolejny należy w sposób świadomy przeanalizować przechwycone dane.

Opisane ćwiczenia można rozszerzyć o elementy pracy grupowej. W tym przypadku uczestnicy dokonują weryfikacji dostępności interfejsów sieciowych swoich komputerów wymieniając się uzyskanymi wynikami np. dotyczącymi adresów MAC kart sieciowych.

```
C:\Users\tracert 158.75.1.4

Tracing route to koala.uci.umk.pl [158.75.1.4]
over a maximum of 30 hops:

  1    1 ms    1 ms    4 ms    158.75.5.254
  2    3 ms    <1 ms  <1 ms    158.75.64.105
  3    <1 ms   <1 ms   <1 ms    ucirt.man.torun.pl [158.75.33.33]
  4    1 ms    <1 ms   <1 ms    158.75.64.22
  5    <1 ms   <1 ms   <1 ms    koala.uci.umk.pl [158.75.1.4]
```

**Rysunek 3** Wynik śledzenia ścieżki do zdalnego hosta za pomocą komendy tracert

Tego typu zadania najczęściej rozbudowywane są przy użyciu komendy **tracert** umożliwiającej śledzenie ścieżki, którą pokonują pakiety (Rysunek 3). Dzięki analityce sieciowej Wireshark uczestnicy mogą samodzielnie wykonać różne testy, które pozwolą im na sformułowanie wniosków dotyczących praktycznego działania protokołu ICMP. Jednym z nich może być stwierdzenie, iż śledzenie takie jest realizowane w oparciu o wyzerowanie wartości pola TTL poprzez kolejny router, co tym samym powoduje odrzucenie pakietu. Do sformułowania takiego stwierdzenia niezbędna jest wiedza z zakresu sposobu obsługi protokołu IP przez routery, co potwierdza jedynie fakt, iż myślenie problemowe musi być oparte na solidnej wiedzy. Dodatkowo z analizowanych danych wynika, iż router odrzucając pakiet informuje o tym fakcie nadawcę poprzez przesłanie do niego datagramu ICMP, w którym jako treść wiadomości pojawia się nagłówek (minimum warstwa 3) odrzuconej wiadomości (Rysunek 4).

Dodatkowo zadania te można wzbogacić o aspekty praktyczne wykorzystując urządzenia sieciowe, takie jak koncentratory i przełączniki sieciowe. W takim przypadku uczestnicy samodzielnie budują sieć z istniejących urządzeń, konfiguruje interfejsy sieciowe, a następnie przeprowadzają opisywane testy. W takim podejściu mogą pojawić się błędy uniemożliwiające prawidłowe działanie sieci. Zatem

w sposób naturalny zmusza to uczestników zajęć do podejmowania różnych prób, formułowania hipotez oraz ich praktycznej realizacji w celu rozwiązania problemów. Ponadto tego typu zadania pozwalają na rozwijanie umiejętności pracy grupowej.

The screenshot shows the Wireshark interface with the following data:

No.	Time	Source	Destination	Protocol	Length	Info
125	3.323230	158.75.5.54	158.75.1.4	ICMP	106	Echo (ping) request id=0x0001, seq=39/9984, ttl=1...
126	3.328418	158.75.5.254	158.75.5.54	ICMP	70	Time-to-live exceeded (Time to live exceeded in tr...
498	8.841212	158.75.5.54	158.75.1.4	ICMP	106	Echo (ping) request id=0x0001, seq=42/10752, ttl=...
499	8.841632	158.75.64.105	158.75.5.54	ICMP	70	Time-to-live exceeded (Time to live exceeded in tr...
502	8.843941	158.75.5.54	158.75.1.4	ICMP	106	Echo (ping) request id=0x0001, seq=44/11264, ttl=...

The detailed view of packet 502 shows the following structure:

- Frame 499: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface 0
- Ethernet II, Src: Netgear\_f8:1b:b7 (6c:b0:ce:f8:1b:b7), Dst: Dell\_07:b2:4c (64:00:6a:07:b2:4c)
- Internet Protocol Version 4, Src: 158.75.64.105, Dst: 158.75.5.54
- Internet Control Message Protocol
  - Type: 11 (Time-to-live exceeded)
  - Code: 0 (Time to live exceeded in transit)
  - Checksum: 0xf4ff [correct]
  - [Checksum Status: Good]
- Internet Protocol Version 4, Src: 158.75.5.54, Dst: 158.75.1.4
  - 0100 .... = Version: 4
  - ... 0101 = Header Length: 20 bytes (5)
  - Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  - Total Length: 92
  - Identification: 0x6318 (25368)
  - Flags: 0x00
  - Fragment offset: 0
  - Time to live: 1
  - Protocol: ICMP (1)
  - Header checksum: 0x13b9 [validation disabled]
  - [Header checksum status: Unverified]
  - Source: 158.75.5.54
  - Destination: 158.75.1.4
  - [Source GeoIP: Unknown]
  - [Destination GeoIP: Unknown]
- Internet Control Message Protocol
  - Type: 8 (Echo (ping) request)
  - Code: 0
  - Checksum: 0xf7d4 [unverified] [in ICMP error packet]
  - [Checksum Status: Unverified]
  - Identifier (BE): 1 (0x0001)
  - Identifier (LE): 256 (0x0100)
  - Sequence number (BE): 42 (0x002a)
  - Sequence number (LE): 10752 (0x2a00)

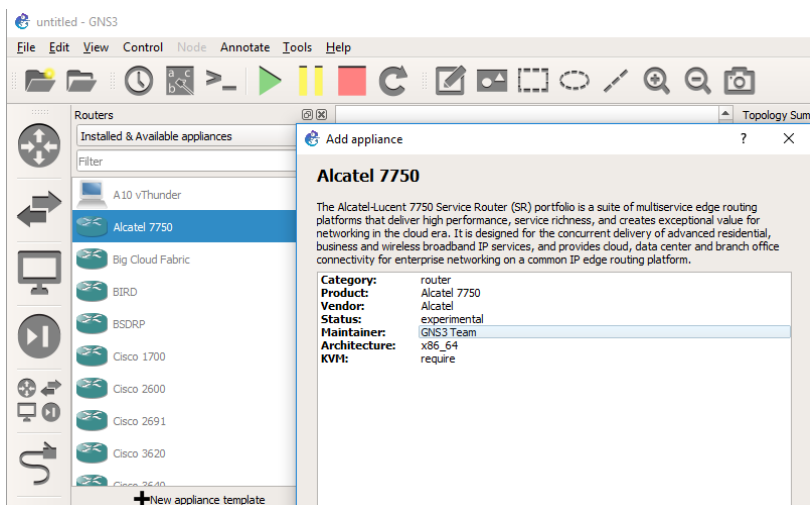
**Rysunek 4** Dane przechwycone w programie Wireshark związane z testem śledzenia ścieżki do zdalnego hosta za pomocą komendy tracert

## 4. Praca grupowa – sieci złożone

Sieci komputerowe pozwalają na realizację wielu scenariuszy pracy grupowej. Jednym z przykładów może być zadanie polegające na budowie dużej sieci komputerowej, w której za każdą część (np. router wraz z przełącznikami sieciowymi) odpowiada inna grupa. Zadanie to pozwala nie tylko na realizację swoich indywidualnych celów (uruchomienie sieci lokalnej), ale również współpracę z innymi grupami (uruchomienie routingu między sieciami). Korzystając z tak zbudowanej topologii można przeprowadzać opisane już testy, a także zastanawiać się nad możliwą optymalizacją sieci np. pod kątem uruchomionych protokołów routingu. Jako przykładowy scenariusz można zaproponować konfigurację routerów w oparciu routing statyczny, a następnie dynamiczny. Wykorzystując protokół routingu RIP oraz OSPF w kolejnym kroku można przeprowadzić badanie sposobu ich działania,

a następnie wykonać testy dotyczące szybkości zbieżności zastosowanych rozwiązań np. po pojawieniu się dodatkowego połączenia.

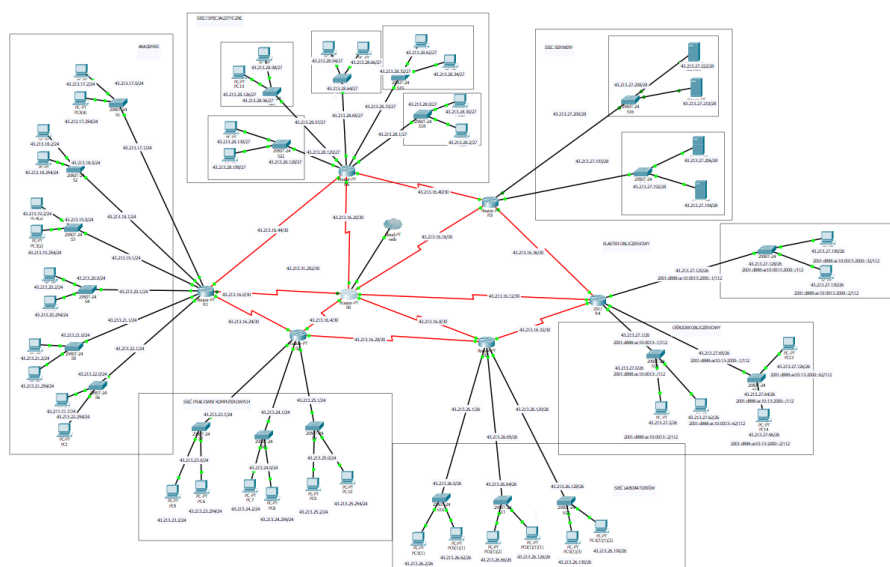
Tego typu zadania można realizować również przy wykorzystaniu symulatorów sieciowych, które pozwalają na budowę sieci, konfigurację urządzeń oraz weryfikację ich działania. Przykładem takiego oprogramowania może być Cisco Packet Tracer [4] lub GNS3 [3]. Co więcej tego typu oprogramowanie najczęściej wyposażone jest w moduł współpracy grupowej umożliwiający łączenie indywidualnych projektów w jedną dużą sieć z wykorzystaniem realnej sieci komputerowej.



**Rysunek 5** Okno dialogowe umożliwiające dodanie routera do topologii sieciowej w symulatorze sieciowym GNS3

## 5. Projekty czyli myślenie komputacyjne

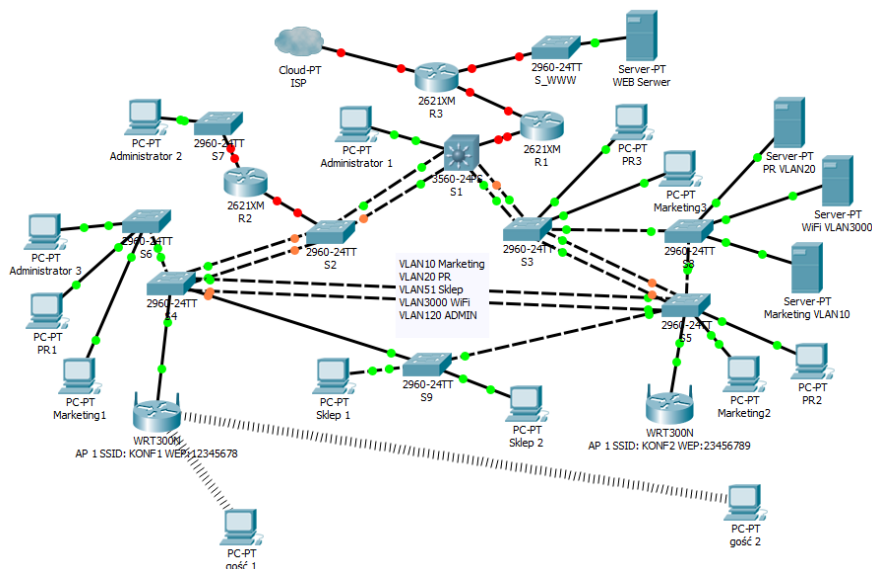
Stanowczo dużym wyzwaniem jest realizacja projektów, które mogą być wykonywane indywidualnie lub w grupie roboczej. W tym przypadku wykorzystanie symulatorów pozwala na ich realizację bez potrzeby posiadania rzeczywistych urządzeń sieciowych. W tego typu zadaniach ważne jest zdefiniowanie celów, które mają zostać osiągnięte, natomiast sposób realizacji zależy od indywidualnego podejścia uczestnika. W tym przypadku osoba wykonująca projekt musi wykorzystać swoją wiedzę do zaproponowania konkretnego rozwiązania, a następnie zrealizować je praktycznie. Podczas tego typu zadań uczestnicy napotykają szereg problemów, których rozwiązywanie przyczyni się do ugruntowania ich wiedzy i uzyskania umiejętności praktycznych. Przykładem wyniku realizacji takiego projektu jest topologia przedstawiona na Rysunku 6.



**Rysunek 6** Projekt sieci wykonany w ramach zadania problemowego

Kolejny etap tego typu zadań polega na optymalizacji proponowanych rozwiązań, badaniu ich słabych i mocnych stron. Bardzo często w takich przypadkach wykorzystywane są gotowe projekty celowo zawierające błędy. Zadaniem osoby realizującej takie zadanie jest zidentyfikowanie i wyeliminowanie problemów, a następnie uruchomienie brakujących funkcjonalności. Tego typu podejście wymaga posiadania dogłębnej wiedzy, ale również umiejętności praktycznych umożliwiających zaproponowanie różnego rodzaju testów. Prawidłowa analiza otrzymanych danych pozwoli na stawianie hipotez, a następnie określenie algorytmu ich weryfikacji. W przypadku braku prawidłowego rozwiązania następuje powrót do fazy testów, analizy danych i stawiania kolejnych hipotez, aż do osiągnięcia założonych celów. Przykład takiego problemowego zadania z zakresu technologii VLAN przedstawia Rysunek 7. Zadanie wymaga znalezienia błędów, które zostały popełnione na etapie tworzenia topologii oraz konfiguracji urządzeń. Następnie musi zostać uruchomiona komunikacja pomiędzy urządzeniami przydzielonymi do różnych sieci VLAN. Ponadto wymagane jest również zadbanie o prawidłowe działanie protokołu STP, który ma w różny sposób rozpinać swoje drzewo dla różnych wirtualnych sieci. Problemy dotyczą również działania sieci bezprzewodowej, w wyniku których tylko jeden punkt dostępowy jest w stanie obsługiwać mobilne urządzenia gości. Ostatecznym wynikiem zrealizowanego zadania ma być poprawnie działająca sieć komputerowa zgodnie ze zdefiniowanymi wytycznymi.





Rysunek 7 Topologia sieciowa zadania problemowego dotyczącego technologii VLAN

## 6. Podsumowanie

Niniejsze opracowanie miało na celu zaprezentowanie kilku scenariuszy realizacji zadań z zakresu sieci komputerowych. Przedstawiono ćwiczenia pokazujące w jaki sposób prosta analiza danych, praktyczna realizacja sieci oraz rozwiązywanie problemów może przyczynić się do lepszego zrozumienia poruszanych zagadnień. Tego typu działania rozwijają krytyczny sposób myślenia, bardzo mocno ceniony na współczesnym rynku pracy.

## Literatura

1. Ashton K, That 'Internet of Things' Thing, <http://www.rfidjournal.com/articles/view?4986>
2. <https://tools.ietf.org/html/rfc792>
3. <https://www.gns3.com/>
4. <https://www.netacad.com/courses/packet-tracer>
5. <https://www.wireshark.org/>
6. Piwiński M., Praktyczna analiza sieci komputerowych z wykorzystaniem programu Wireshark, Informatyka w Edukacji V, A.B. Kwiatkowska, M.M. Sysło,

Wydawnictwo Naukowe UMK, 277-285, Toruń, 2008,  
<http://repozytorium.umk.pl/handle/item/1686>

7. Piwiński M., „Uczniowie i komputery w sieci...”, „Komputer w Szkole”, nr 5, (2003), s 38, <https://repozytorium.umk.pl/handle/item/1667>
8. Wing J. Computational Thinking Benefits Society". 40th Anniversary Blog of Social Issues in Computing, (2014)  
<http://socialissues.cs.toronto.edu/index.html%3Fp=279.html>.