

Laboratorium

Zadanie 1.

Cel i kontekst:

Celem ćwiczenia jest określenie czasu pobierania pliku o zadanej wielkości przy różnych przepustowościach łączy sieciowych lub wielkości pliku możliwego do ściągnięcia przy danej przepustowości. Pozwala to zrozumieć zależność pomiędzy teoretyczną przepustością kanału a faktycznym czasem transferu danych, który jest zależny od wielu czynników — w tym narzutu protokołów warstwowych modelu OSI. Przepustowość (Bandwidth) – maksymalna liczba bitów, które mogą być przesłane przez kanał transmisyjny w jednostce czasu, wyrażana w bitach na sekundę (b/s). Opóźnienie (Latency) – czas, jaki upływa między wysłaniem pierwszego bitu a jego odebraniem po stronie odbiorcy. Narzut protokołów – dodatkowa objętość danych związana z nagłówkami i stopkami ramki, pakietu lub segmentu (np. Ethernet, IP, TCP).

Treść zadania: Oblicz teoretyczny czas potrzebny do pobrania pliku lub wielkość pliku dla parametrów podanych przez prowadzącego.

Przykład:

Wielkość pliku: 1 GB

Przepustowość: 10 Mb/s

W jakim czasie pobierzemy plik.

Uwaga: wynik należy skorygować o narzut Ethernet/IP/TCP (rzędu kilku procent) i ewentualne retransmisje (przyjmując 1%)

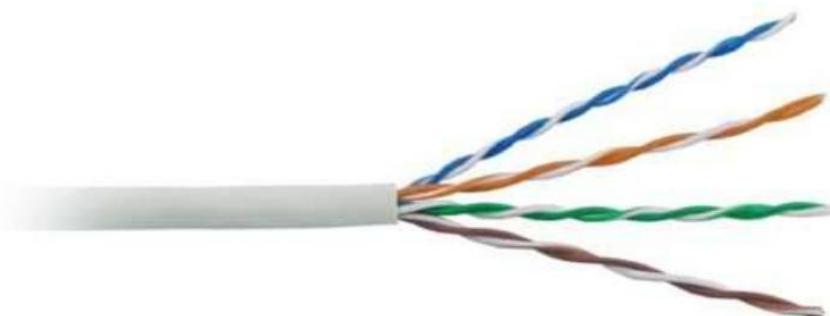
Sprawozdanie powinno zawierać:

- dane wejściowe
- sposób obliczenia
- uzyskane wyniki,
- wnioski

Zadanie 2

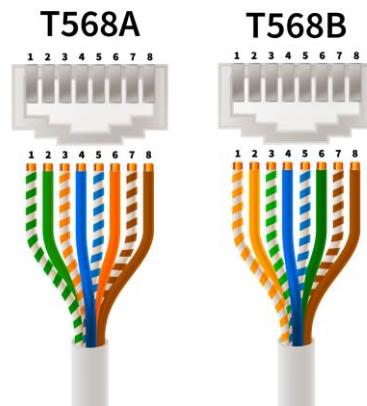
Cel i kontekst:

Budowa kabla UTP: Kabel UTP (Unshielded Twisted Pair) składa się z czterech par skręconych przewodów miedzianych, co ogranicza interferencje elektromagnetyczne (EMI) i przeszłuchy (crosstalk). Zalecane kategorie: Cat 5e (do 1 Gb/s), Cat 6 (do 10 Gb/s na krótkich dystansach). Maksymalna długość odcinka między urządzeniami: 100 m.



Rysunek 1. Skrętka UTP – przekrój i pary przewodów

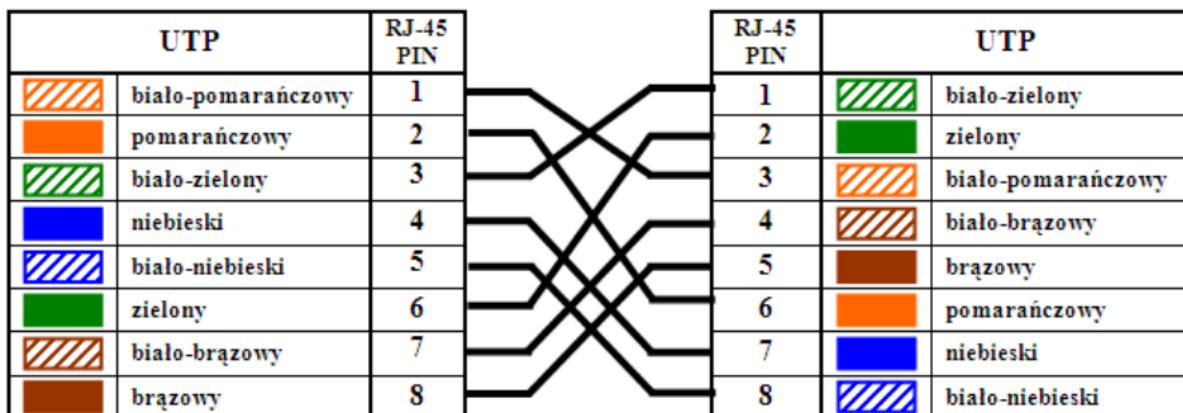
Standardy zakończenia: Normy T568A i T568B określają kolejność przewodów we wtyku RJ-45. Typy kabli: prosty (PC ↔ SWITCH, łączymy RJ-45 w takiej samej kolejności), skrosowany (PC ↔ PC / SWITCH ↔ SWITCH – bez Auto-MDI/MDIX, patrz połączenia skrosowane), rollover (połączenia konsolowe).



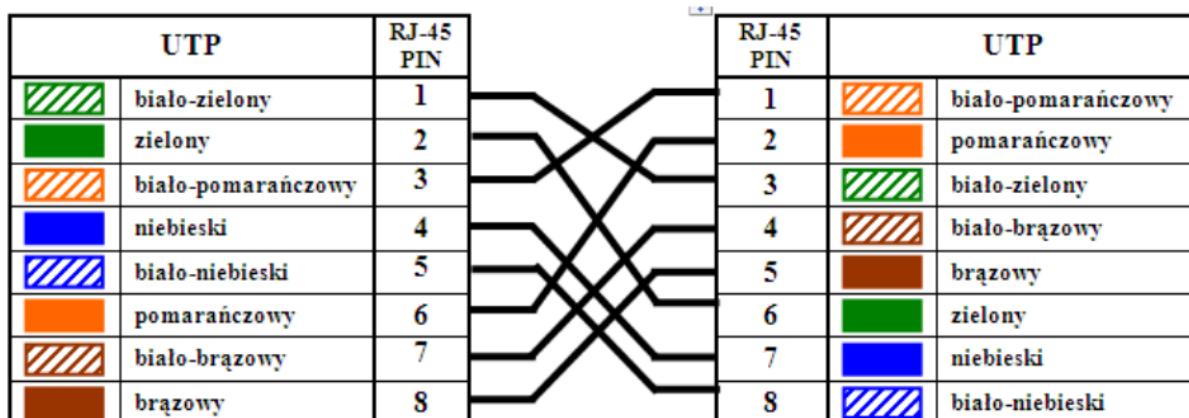
Rysunek 2. Złącze RJ-45 – ułożenie kabli standard T568A i T568B

Połączenia skrosowane:

T568B



T568A



Normy i zalecenia stosowane przy budowie okablowania strukturalnego.

Kraj	Polska	Europa	USA	Świat
Norma	PN-EN 50173	EN 50173	TIA/EIA 568A	ISO/IEC 11801

Treść zadania:

Zadanie polega na samodzielnym wykonaniu przewodu sieciowego kategorii 5e, zgodnie z obowiązującymi standardami TIA/EIA-568A/B. Poprawne wykonanie kabla jest kluczowe dla niezawodności transmisji danych w sieciach Ethernet.

1. Wyciąć ze szpuli określoną przez prowadzącego długości kabla, końcówki kabla należy przycinać tak by „płaszczyzna cięcia” była prostopadła do obrysu przewodu,
 2. Z końców kabla UTP Usunąć izolację zewnętrzną (zazwyczaj białą lub szarą) na długości około 1,5 cm, przy czym należy szczególnie uważać aby nie uszkodzić żadnego zewnętrznych (kolorowych) przewodów transmisyjnych,
 3. Rozkręcić skręcone pary przewodów transmisyjnych,
 4. Ułożyć przewody transmisyjne w odpowiedniej kolejności,
 5. Przyciąć wszystkie osiem składowych przewodów transmisyjnych aby miały równą długość – około 1 cm,
 6. Umieścić w łączniku modularnym RJ-45 osiem przewodów transmisyjnych w odpowiedniej kolejności,
 7. Umieścić tak przygotowany łącznik modularny RJ-45 z ułożonymi przewodami transmisyjnymi w zaciskarce i ścisnąć szczypce zaciskarki, co spowoduje przebiecie przewodów transmisyjnych blaszkami i zaciśnięcie ich w łączniku RJ-45,
 8. Dla drugiego końca wyciętego ze szpuli kabla UTP powtórzyć kroki od 3 do 7,
 9. Zweryfikować poprawność wykonania kabla przyłączeniowego.
- Do wykonania powyższych operacji należy wykorzystać dostępne dla każdej grupy laboratoryjnej narzędzia w postaci:
- - uniwersalnej zaciskarki wtyków,
 - - elektronicznego testera kabli sieciowych.

Uwaga 1: Każdy Student otrzymuje 2 sztuki łączników RJ45 i kabel UTP o odpowiedniej długości. Student musi przedstawić poprawnie działający kabel przyłączeniowy. Poprawność jest sprawdzana na testerze. Tester składa się z nadajnika i przystawki. Tester pozwala zidentyfikować: przerwane przewody, zamienione przewody oraz zwarcie.

W sprawozdaniu:

W przypadku błędów opis jaki to błąd. W tabeli przedstaw połączenia kabli lub błędy odczytane z testera

Sposób połączenia (np. T568A, prosty)	
RJ-45 „1”	RJ-45 „2”
1	1
2	2
3	3
....	...

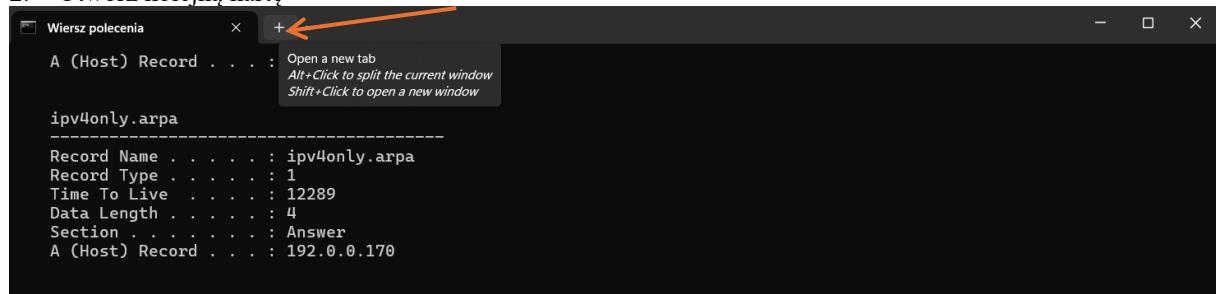
Zadanie 3

Cel i kontekst:

Celem ćwiczenia jest poznanie działania systemu DNS (Domain Name System) w praktyce oraz zrozumienie, że komputer zapamiętuje odwiedzone adresy stron w lokalnej pamięci podręcznej (cache DNS).

Treść zadania:

1. Otworzyć Okno Wiersza Poleceń (kliknąć przycisk Start, wskazać polecenie Uruchom..., w oknie Otwórz: wpisać „cmd”, a następnie kliknąć OK).
2. Otwórz kolejną kartę



3. W wierszu poleceń wpisz: ipconfig /displaydns
4. Zwróć uwagę na kluczowe elementy:
 - Record Name – nazwa domeny (np. www.google.com)
 - Record Type – typ rekordu DNS (np. A – adres IPv4)
 - Time To Live (TTL) – czas przechowywania wpisu w pamięci podręcznej
5. Otwórz przeglądarkę i odwiedź stronę, której wcześniej nie odwiedzałeś (np. https://www.wikipedia.org). Wróć do wiersza poleceń i ponownie wpisz ipconfig /displaydns. Sprawdź, czy w wynikach pojawiła się domena www.wikipedia.org.

W sprawozdaniu:

Odpowiedz na pytania:

1. Czym jest pamięć podręczna DNS?
2. Dlaczego komputer przechowuje odwiedzone adresy?

Załącz zrzut ekranu uzyskanych wyników (wystarczy 1).

Zadanie 4

Cel i kontekst:

Cel: Zapoznanie z zasadami definiowania i walidacji dokumentów XML. Należy utworzyć własny dokument XML, a następnie opisać jego strukturę za pomocą DTD i XSD.

XML (eXtensible Markup Language) to język znaczników do reprezentacji danych w sposób czytelny dla człowieka i maszyny. Dokument XML jest hierarchiczny (drzewo), posiada pojedynczy element główny, poprawne zagnieżdżenia i zamknięcia znaczników; atrybuty ujęte są w cudzysłowy.

Zasady poprawnego XML-a:

1. Każdy element musi mieć znacznik otwierający i zamykający.
2. Elementy muszą być prawidłowo zagnieżdżone.
3. Dokument XML ma jeden główny (root) element.
4. XML jest wrażliwy na wielkość liter.
5. Atrybuty muszą być ujęte w cudzysłowy.

Przykładowy dokument XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<biblioteka>
    <ksiazka id="1">
        <tytul>Wiedźmin: Miecz przeznaczenia</tytul>
        <autor>Andrzej Sapkowski</autor>
        <rok_wydania>1992</rok_wydania>
        <gatunek>fantasy</gatunek>
        <ocena>9</ocena>
    </ksiazka>
    <ksiazka id="2">
        ...
    </ksiazka>
</biblioteka>
```

DTD (Document Type Definition) opisuje strukturę XML-a, określając jakie elementy i atrybuty mogą wystąpić w dokumencie.

Definicja DTD:

```
<!ELEMENT biblioteka (ksiazka+)>
<!ELEMENT ksiazka (tytul, autor, rok_wydania, gatunek, ocena)>
<!ATTLIST ksiazka id ID #REQUIRED>
<!ELEMENT tytul (#PCDATA)>
<!ELEMENT autor (#PCDATA)>
<!ELEMENT rok_wydania (#PCDATA)>
<!ELEMENT gatunek (#PCDATA)>
<!ELEMENT ocena (#PCDATA)>
```

(ksiazka+) – oznacza, że w bibliotece musi być co najmniej jedna książka.

#PCDATA – oznacza tekst (dane znakowe).

ID – atrybut, który musi być unikalny w całym dokumencie

Walidacja XML za pomocą XSD (schematu)

XSD (XML Schema Definition) to sposób opisu struktury XML, który umożliwia:

- określanie typów danych (np. string, integer, date)
- definiowanie ograniczeń (np. minimalna/maksymalna długość)
- walidację bardziej złożonych struktur

Schemat XSD:

```
<?xml version="1.0" encoding="UTF-8"?>
<xss:schema xmlns:xss="http://www.w3.org/2001/XMLSchema">
    <xss:element name="biblioteka">
        <xss:complexType>
            <xss:sequence>
                <xss:element name="ksiazka" maxOccurs="unbounded">
                    <xss:complexType>
                        <xss:sequence>
                            <xss:element name="tytul" type="xss:string"/>
                            <xss:element name="autor" type="xss:string"/>
                            <xss:element name="rok_wydania" type="xss:integer"/>
                            <xss:element name="gatunek" type="xss:string"/>
                            <xss:element name="ocena" type="xss:integer"/>
                        </xss:sequence>
                        <xss:attribute name="id" type="xss:integer" use="required"/>
                    </xss:complexType>
                </xss:element>
            </xss:sequence>
        </xss:complexType>
    </xss:element>
</xss:schema>
```

```
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

Treść zadania:

Napisz plik XLM umożliwiający przesyłanie danych o rejestrujących się osobach, w szczególności: imię, nazwisko, płeć, wiek. Dla przygotowanego pliku XML opracuj odpowiednie pliki DTD i XSD. Wykorzystując stronę np. FreeFormatter XML Validator (zakładka „XML Validator – XSD”) przeprowadź analizę poprawności stworzonych plików.

W sprawozdaniu:

W odpowiednim folderze na github-ie załącz pliki XLM, DTD, XSD oraz zapisz efekty walidacji.

Zadanie 5.

Cel i kontekst:

Celem niniejszego zadania jest zrozumienie i praktyczne zastosowanie zasady oddzielenia struktury dokumentu HTML od warstwy prezentacyjnej, definiowanej w CSS. W ramach ćwiczenia student zapoznaje się z budową poprawnej strony internetowej, w której zawartość, struktura oraz wygląd są od siebie niezależne. Podejście to jest zgodne z filozofią tworzenia stron zgodnych ze standardami W3C i zasadami tzw. czystego kodu (Clean Code).

HTML (HyperText Markup Language) jest językiem opisującym strukturę dokumentu w sieci. Każdy dokument HTML ma hierarchiczną strukturę opartą na elementach, które pełnią funkcję semantyczną – określają znaczenie poszczególnych fragmentów treści. CSS (Cascading Style Sheets) natomiast służy do definiowania sposobu prezentacji treści, czyli kolorów, czcionek, układu, rozmiarów i innych właściwości wizualnych. Rozdzielenie tych dwóch warstw umożliwia łatwiejsze zarządzanie projektem oraz jego późniejszą rozbudowę.

W nowoczesnych stronach internetowych stosuje się elementy semantyczne HTML5, takie jak <header>, <nav>, <main>, <section>, <article>, <aside> czy <footer>. Elementy te zastępują dawne znaczniki <div> używane bez kontekstu. Zastosowanie ich zwiększa dostępność stron, poprawia optymalizację SEO (Search Engine Optimization) oraz umożliwia przeglądarkom i programom czytającym lepsze rozumienie struktury treści.

Arkusze stylów CSS pozwalają na pełną kontrolę nad wyglądem strony. Stosuje się je w trzech podstawowych formach:

- Wewnętrzne – za pomocą znacznika <style> w sekcji <head>;
- Zewnętrzne – poprzez plik .css podłączony przez <link rel="stylesheet">;
- Wbudowane (inline) – za pomocą atrybutu style w pojedynczym znaczniku HTML.

W praktyce zaleca się stosowanie wyłącznie arkuszy zewnętrznych, co umożliwia centralne zarządzanie wyglądem strony.

Struktura plików projektu

W ramach ćwiczenia student powinien przygotować minimalny zestaw plików projektu:

- index.html – dokument główny zawierający strukturę strony;
- style.css – arkusz stylów odpowiedzialny za wygląd;
- folder /img – zasoby graficzne (jeśli występują).

Zaleca się stosowanie logicznego podziału plików i nazw (np. /css, /js, /img), aby ułatwić dalszą rozbudowę projektu. W pliku HTML w sekcji <head> należy umieścić odwołanie do pliku CSS za pomocą elementu: <link rel="stylesheet" href="style.css">

Przykładowa struktura dokumentu HTML

```
<!DOCTYPE html>
<html lang="pl">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```

<title>Strona przykładowa</title>
<link rel="stylesheet" href="style.css">
</head>
<body>
<header>
    <h1>Moja pierwsza strona</h1>
</header>

<nav>
    <ul>
        <li><a href="index.html">Start</a></li>
        <li><a href="galeria.html">Galeria</a></li>
        <li><a href="kontakt.html">Kontakt</a></li>
    </ul>
</nav>

<main>
    <section>
        <h2>O projekcie</h2>
        <p>Strona wykonana w ramach ćwiczeń z technologii internetowych.</p>
    </section>
</main>

<footer>
    <p>© 2025 Autor: Jan Kowalski</p>
</footer>
</body>
</html>

```

Fragment zewnętrznego arkusza stylów CSS

```

/* style.css */
body {
    font-family: Arial, sans-serif;
    background-color: #f8f8f8;
    color: #222;
    margin: 0;
    padding: 0;
}

```

```

header, footer {
    background-color: #007acc;
    color: white;
    text-align: center;
    padding: 1em;
}

```

```

nav ul {
    list-style: none;
    background: #e0e0e0;
    margin: 0;
    padding: 0.5em;
    display: flex;
    justify-content: center;
    gap: 1em;
}

```

```

nav a {
    text-decoration: none;
    color: #007acc;
    font-weight: bold;
}

```

```
}
```

```
main {
```

```
    max-width: 900px;
```

```
    margin: auto;
```

```
    background: white;
```

```
    padding: 2em;
```

```
    border-radius: 8px;
```

```
    box-shadow: 0 0 10px rgba(0,0,0,0.1);
```

```
}
```

Treść zadania:

Do strony utworzonej na poprzednich zajęciach dodaj plik style.css. Plik ten ma odpowiadać za wygląd strony, usuń niepotrzebne element z pliku XXX.html.

W sprawozdaniu:

Rezultaty pracy zapisz (github).

Zadanie 6

Cel i zakres zadania

Celem zadania jest poznanie zasad działania protokołu HTTP oraz praktyczna analiza komunikacji między przeglądarką a serwerem z wykorzystaniem narzędzi deweloperskich (DevTools). Ćwiczenie ma na celu uświadomienie studentowi, w jaki sposób dane przesyłane są w sieci, jak wyglądają żądania HTTP/HTTPS, jakie nagłówki i kody statusu są zwracane oraz jak różnią się metody GET i POST. Umiejętność analizy ruchu HTTP jest podstawowa w pracy programisty webowego, administratora sieci i testera bezpieczeństwa aplikacji. HTTP (Hypertext Transfer Protocol) jest protokołem warstwy aplikacji modelu OSI. Służy do wymiany danych między klientem (np. przeglądarką) a serwerem WWW. Jest protokołem bezstanowym – każde żądanie jest niezależne, a kontekst sesji utrzymywany jest za pomocą plików cookie lub tokenów.

Podstawowy przepływ komunikacji HTTP wygląda następująco:

1. Klient (np. przeglądarka) wysyła żądanie HTTP do serwera.
2. Serwer przetwarza żądanie i odsyła odpowiedź HTTP zawierającą kod statusu i treść.
3. Klient interpretuje odpowiedź i wyświetla wynik użytkownikowi.

Najczęściej stosowane metody HTTP:

- GET – pobranie danych (parametry w adresie URL),
- POST – wysłanie danych (np. formularzy) w treści żądania,
- PUT – aktualizacja danych,
- DELETE – usunięcie zasobu,
- HEAD – pobranie nagłówków bez treści.

Analiza żądań w narzędziach deweloperskich (DevTools)

W przeglądarce (np. Chrome, Edge, Firefox) można otworzyć narzędzia deweloperskie skrótem F12. Zakładka Network pozwala obserwować ruch sieciowy. Po wysłaniu formularza widzimy w niej nowe żądanie POST/GET. Klikając w nie, można sprawdzić szczegółowe dane żądania: nagłówki, parametry, czasy odpowiedzi oraz ciało żądania i odpowiedzi. Dzięki temu można analizować błędy komunikacji, niepoprawne parametry czy problemy z kodowaniem znaków.

Kody statusu HTTP

Kody statusu określają rezultat przetwarzania żądania przez serwer:

- 1xx – informacje wstępne (rzadko stosowane),
- 2xx – sukces (np. 200 OK, 201 Created),
- 3xx – przekierowanie (np. 301 Moved Permanently),
- 4xx – błąd po stronie klienta (np. 404 Not Found, 403 Forbidden),
- 5xx – błąd po stronie serwera (np. 500 Internal Server Error).

Przykładowe żądanie HTTP – metoda POST

```
POST /post HTTP/1.1
Host: httpbin.org
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
Content-Type: application/x-www-form-urlencoded
Content-Length: 27
Accept: /*
Connection: keep-alive
```

login=student&haslo=tajne123

Powyższy przykład przedstawia pełne żądanie POST wysyłane z formularza HTML. W ciele żądania (body) znajdują się dane przesyłane w formacie application/x-www-form-urlencoded. Nagłówki opisują sposób kodowania danych, typ przeglądarki, hosta docelowego oraz inne informacje o sesji.

Przykład możliwej odpowiedzi serwera HTTP

```
HTTP/1.1 200 OK
Date: Fri, 24 Oct 2025 12:00:00 GMT
Content-Type: application/json
Content-Length: 200
```

```
{
  "form": {
    "login": "student",
    "haslo": "tajne123"
  },
  "headers": {
    "Content-Type": "application/x-www-form-urlencoded"
  }
}
```

Odpowiedź serwera zawiera kod statusu 200 (OK), który oznacza prawidłowe przetworzenie żądania. W sekcji JSON widoczne są dane przesłane przez użytkownika. W narzędziach DevTools można obserwować zarówno wysłane nagłówki (Request Headers), jak i nagłówki odpowiedzi (Response Headers).

Przykładowy formularz HTML wysyłający dane metodą POST

```
<!DOCTYPE html>
<html lang="pl">
<head>
  <meta charset="UTF-8">
  <title>Test POST</title>
</head>
<body>
  <form method="POST" action="https://httpbin.org/post">
    <label>Login: <input type="text" name="login"></label><br>
    <label>Hasło: <input type="password" name="haslo"></label><br>
    <button type="submit">Wyślij</button>
  </form>
</body>
</html>
```

Treść zadania:

1. Otwórz przeglądarkę (np. Firefox, Chrome lub Edge).
2. Naciśnij klawisz F12, aby otworzyć narzędzia deweloperskie.
3. Przejdź do zakładki „Network” (lub „Sieć”).
4. W pasku adresu wpisz: <https://example.com> i naciśnij Enter.

5. W zakładce Network zobaczysz jedno żądanie HTTP.
6. Kliknij to żądanie, aby zobaczyć szczegóły. Zwróć uwagę na:
 - Request Method: GET
 - Status Code: 200 OK
 - Request URL: <https://example.com>
 - Response: treść strony
7. W pasku adresu wpisz: <https://httpbin.org/get?imie=Jan&miasto=Krakow>
8. W zakładce Network kliknij żądanie GET.
9. Sprawdź w zakładce Headers, że dane (parametry) są częścią adresu URL.
10. W zakładce Response zobaczysz odpowiedź w formacie JSON – serwer zwraca dane, które wysłałeś.
11. Korzystając z przykładowego formularza „Przykładowy formularz HTML wysyłający dane metodą POST” wyślij wiadomość i zobacz rezultaty.

W sprawozdaniu:

Odpowiedz na pytania:

1. Gdzie znajdują się parametry w metodzie GET?
2. Czy są one szyfrowane?
3. Jaka jest różnica między GET a POST w miejscu przesyłania danych?

Wykonaj zrzuty ekranu.

Zadanie 7

Cel zadania

Celem ćwiczenia jest:

- dodanie prostego formularza HTML, który pozwala użytkownikowi wprowadzić dane osobowe,
- stworzenie pliku XML, który będzie mógł przechowywać te dane,
- przygotowanie pliku XSD, który sprawdzi poprawność (walidację) danych zapisanych w XML

1. Rozbudowa strony HTML

Otwórz swój poprzedni projekt (HTML + CSS) i zapisz go jako nowy projekt.

W sekcji treści strony utwórz prosty formularz zawierający:

- pole tekstowe (textbox) dla imienia,
- pole tekstowe dla nazwiska,
- pole wyboru płci (radio buttons) – np. „Kobieta” / „Mężczyzna”,
- pole tekstowe wiek

Na tym etapie formularz nie musi wysyłać danych — chodzi o zrozumienie struktury i powiązania z XML.

2. Utworzenie pliku XML

Stwórz plik o nazwie `dane_osobowe.xml`, który będzie przechowywać dane wprowadzone w formularzu. Plik powinien zawierać podstawowe informacje o jednej osobie.

3. Walidacja danych (XSD)

Stwórz plik `dane_osobowe.xsd`, który będzie sprawdzał poprawność danych z XML. Zdefiniuj w nim typy danych oraz ograniczenia.

Zadanie 8 (by Cisco Public)

Kroki w instalacji Cisco Packet Tracer:

1. **Przejdz na stronę Cisco Networking Academy:** Wejdź na stronę [Cisco Networking Academy](https://www.cisco.com/cisco-networking-academy).
2. **Zaloguj się lub załóż konto:** Musisz mieć konto, aby pobrać i używać Packet Tracer. Jeśli go nie masz, utwórz darmowe konto.

3. **Pobierz instalator:** Znajdź opcję pobrania aplikacji i wybierz instalator odpowiedni dla swojego systemu operacyjnego (Windows, Linux lub macOS).
4. **Zainstaluj program:** Uruchom pobrany plik instalacyjny i postępuj zgodnie z instrukcjami wyświetlonymi w kreatorze instalacji.

Cele:

Część 1: Badanie ruchu internetowego http.

Część 2: Wyświetlenie elementów zestawu protokołów TCP/IP.

Wprowadzenie

Prezentowana symulacja ma za zadanie szczegółowo przedstawić zasadę działania zestawu protokołów TCP/IP w relacji do modelu OSI. Praca w trybie symulacji pozwala przeglądać zawartość danych przesyłanych w sieci w każdej warstwie. Kiedy dane przesyłane są przez sieć, to dzielone są na mniejsze części i oznaczane w sposób, który pozwoli na ich ponowne złożenie kiedy dotrą do celu. Każda część ma przyporządkowaną określoną nazwę (jednostka danych protokołu [PDU]) która jest związana z określoną warstwą modeli TCP/IP i OSI. Przedstawiona symulacja programu Packet Tracer umożliwia obserwację poszczególnych warstw i związanych z nimi jednostek PDU. Poniższe kroki prowadzą użytkownika przez proces żądania wyświetlenia strony z serwera WWW za pomocą przeglądarki internetowej dostępnej na komputerze klienta. Mimo wyświetlenia dużej ilości informacji, która zostanie omówiona bardziej szczegółowo w dalszej części kursu, jest to okazja aby poznać funkcjonowanie programu Packet Tracer oraz możliwość wizualizacji procesu enkapsulacji.

Część 1: Badanie ruchu internetowego http

W części 1 tego ćwiczenia będziesz używał Packet Tracer (PT) w trybie symulacji do generowania ruchu w sieci i badania ruchu HTTP.

Krok 1: Przełączanie się z trybu Realtime do trybu Simulation.

W prawym dolnym rogu interfejsu Packet Tracer znajdują się zakładki, które pozwalają przełączać się pomiędzy trybami czasu rzeczywistego (Realtime mode) i symulacji (Simulation mode). Packet Tracer zawsze uruchamia się w trybie Realtime, w którym protokoły sieciowe pracują w realnym czasie z realną prędkością. Jednakże możliwości Packet Tracer pozwalają użytkownikowi "zatrzymanie czasu" poprzez przełączenie się w tryb symulacji. W trybie symulacji pakiety są wyświetlane jako animowane koperty, czas jest zorientowany zdarzeniowo, a użytkownik może obserwować zdarzenia w sieci krok po kroku.

- a. Kliknij ikonę trybu Simulation, aby przełączyć się z trybu Realtime do trybu Simulation.
- b. Wybierz HTTP z filtrów listy zdarzeń (Event List Filters).

- 1) HTTP może już być jedynym widocznym zdarzeniem. Kliknij Edit Filters, aby wyświetlić dostępne widoczne zdarzenia. Zaznacz pole przycisku wyboru Show All/None i zauważ, jak pola wyboru przełączają się na odznaczone lub zaznaczone, w zależności od ich aktualnego stanu.
- 2) Kliknij pole wyboru Show All/None dopóki wszystkie pola zostaną odznaczone, a następnie wybierz HTTP. Kliknij w dowolnym miejscu poza oknem Edit Filters, aby go ukryć. Pozycja Visible Events powinna pokazywać teraz tylko protokół HTTP.

Krok 2: Wygenerowanie ruchu HTTP.

Obecnie panel Simulation jest pusty. W górnej części Event List panelu Simulation znajduje się sześć kolumn. Gdy ruch jest generowany i przemieszcza się krok po kroku, na liście zaczynają wyświetlać się zdarzenia. Kolumna Info używana jest do sprawdzenia zawartości określonego zdarzenia.

Uwaga: serwer WWW (Web Server) i klient WWW (Web Client) wyświetlane są w lewym okienku. Panele mogą być dostosowane do odpowiedniego rozmiaru poprzez przesuwanie linii oddzielającej, która znajduje się obok paska przewijania, w lewo lub w prawo (gdy pojawi się strzałka z dwoma grotami). Schemat powinien wyglądać tak jak na rysunku 1



Rys. 1. Schemat połączeniowy

- a. Kliknij Web Client, który znajduje się w lewym okienku.
- b. Następnie kliknij kolejno w zakładkę Desktop i ikonę Web Browser, aby ją otworzyć.
- c. W polu URL wpisz www.osi.local i kliknij Go. Ze względu na brak skonfigurowania serwera DNS nie otrzymasz poprawnej odpowiedzi.
- d. Na Server-PT wybierz:
 - Desktop → IP Configuration
 - IP Address: np. 192.168.1.2
 - Subnet Mask: 255.255.255.0
 - Default Gateway: 192.168.1.1
- e. Na Server-PT:
 - Wejdź w zakładkę Services → DNS
 - Upewnij się, że przycisk DNS Service = ON
 - Dodaj rekord:
 - Name: www.osi.local
 - Address: 192.168.1.2
 - Kliknij Add
- f. Na PC-PT
 - Wejdź w Desktop → IP Configuration
 - Wpisz:
 - IP Address: 192.168.1.10
 - Subnet Mask: 255.255.255.0
 - Default Gateway: 192.168.1.1
 - DNS Server: 192.168.1.2 ← adres Twojego serwera!
 - Zamknij okno
- g. Sprawdź czy na Serwerze protokół HTTP jest włączony:
 - Services → HTTP
 - Włącz: HTTP = ON
- h. Na PC-PT → Desktop → Web Browser:
 1. Wpisz: <http://www.osi.local>
 2. Kliknij GO
 3. Kliknij kilka zakładek aby spowodować dodatkowy ruch.
- i. Zatrzymaj symulacje

Krok 3: Zbadanie zawartości pakietu HTTP.

a. Kliknij pierwszy kolorowy kwadrat w kolumnie Info na liście Event List. Może okazać się konieczne, aby rozwinąć panel symulacji lub użyć paska przewijania poniżej listy zdarzeń .

Na ekranie pojawi się okno PDU Information at Device: Web Client. Ponieważ jest to początek transmisji, w oknie tym znajdują się tylko dwie zakładki (Model OSI i Outbound PDU Details). Gdy rozpatrywanych będzie więcej zdarzeń, to wyświetlane będą trzy zakładki - dojdzie dodatkowo zakładka Inbound PDU Details. Kiedy zdarzenie jest ostatnim zdarzeniem w strumieniu ruchu, to wyświetlane są tylko karty OSI Model i Inbound PDU Details.

b. Upewnij się, że wybrana jest zakładka OSI Model. Upewnij się, że w kolumnie Out Layers podświetlone jest pole Layer 7.

Jaki tekst wyświetlany jest obok etykiety Layer 7? _____

Jakie informacje wyświetlane są w ponumerowanych krokach bezpośrednio poniżej pól In Layers i Out Layers? _____

c. Kliknij Next Layer. Powinna być podświetlona warstwa 4. Jaka jest wartość portu docelowego (Dest Port)?

d. Kliknij Next Layer. Powinna być podświetlona warstwa 3. Jaka jest wartość docelowego adresu IP (Dst IP)? _____

e. Kliknij Next Layer. Jakie informacje wyświetlane są na tej warstwie? _____

f. Kliknij zakładkę Outbound PDU Details .

Informacje wyświetlane pod PDU Details odzwierciedlają warstwy modelu TCP/IP.

Uwaga: Informacje podane w sekcji Ethernet II dostarczają jeszcze bardziej szczegółowych informacji niż te, które wyświetlane są poniżej warstwy 2 na zakładce OSI Model. Zakładka Outbound PDU Details zawiera informacje bardziej opisowe i szczegółowe. Wartości pod DEST MAC i SRC MAC w ramach szczegółów PDU (PDU Details) sekcji Ethernet II wyświetlane są w zakładce OSI Model pod warstwą 2,

ale nie są oznaczone jako takie.

Jakie są wspólne informacje wymienione w sekcji IP szczegółów PDU (PDU Details) w porównaniu do informacji wymienionych w zakładce OSI Model? Z którą warstwą jest to związane?

Jakie są wspólne informacje wymienione w sekcji TCP szczegółów PDU (PDU Details) w porównaniu do informacji wymienionych w zakładce OSI Model? Z którą warstwą jest to związane?

Jaka jest wartość pola Host wymienionego w sekcji HTTP szczegółów PDU (PDU Details)? Z którą warstwą z zakładki OSI Model będzie ta informacja związana?

g. Kliknij następny kolorowy kwadrat w kolumnie Info na liście Event List. Tylko warstwa 1 jest aktywna (nie wyszarzona). Urządzenie przenosi ramkę z bufora i umieszcza ją w sieci.

h. Przejdź do następnego pola HTTP Info wewnątrz listy Event List i kliknij pole kolorowego kwadratu. Okno to zawiera zarówno warstwy In Layers i Out Layers. Zwróć uwagę na kierunek strzałki bezpośrednio pod kolumną In Layers; jest skierowana ku górze, wskazując kierunek, w którym informacja podróżuje.

Przejrzyj te warstwy sporządzając notatki z przeglądanych pozycji. Na szczytce kolumny strzałka wskazuje w prawo. Oznacza to, że serwer wysyła właśnie informacje z powrotem do klienta.

Porównując informacje wyświetlane w kolumnie In Layers z tymi w kolumnie Out Layers, jakie są między nimi główne różnice?

i. Kliknij zakładkę Outbound PDU Details . Przewiń w dół do sekcji HTTP.

Jaka jest pierwsza linia wyświetlana w wiadomości HTTP?

j. Kliknij ostatni kolorowy kwadrat w kolumnie Info. Ile zakładek zostało wyświetlonych i dlaczego?

Część 2: Wyświetlenie elementów zestawu protokołów TCP/IP

W części 2 tego ćwiczenia używany będzie tryb symulacji Packet Tracer po to, aby zobaczyć i zbadać kilka innych protokołów zawartych w zestawie TCP/IP.

Krok 1: Obserwacja dodatkowych zdarzeń

a. Zamknij wszystkie otwarte okna z informacją PDU.

b. W sekcji Event List Filters > Visible Events, kliknij przycisk Show All.

Jakie dodatkowe typy zdarzeń (Event Types) są wyświetlane?

Te dodatkowe typy zdarzeń (protokoły) odgrywają różne role w ramach zestawu TCP/IP. Jeśli wymieniony jest protokół odwzorowania adresów (ARP), to wyszukuje on adresy MAC. DNS odpowiedzialny jest za konwersję nazwy (na przykład www.osi.local) na adres IP. Dodatkowe zdarzenia TCP odpowiedzialne są za połączenie, uzgadnianie parametrów komunikacji oraz za rozłączenie sesji komunikacji pomiędzy urządzeniami. O protokołach tych zostało wspomniane już wcześniej i będą one nadal omawiane w dalszej części kursu. Obecnie w ramach Packet Tracera istnieje ponad 35 możliwych protokołów (typów zdarzeń) dostępnych do przechytywania.

c. Kliknij pierwsze zdarzenie DNS w kolumnie Info. Zapoznaj się z zakładkami OSI Model i PDU Detail, a następnie zwróć uwagę na proces enkapsulacji. Jak spojrzyś na zakładkę OSI Model z podświetloną Layer 7, to co się tam dzieje, wypisane jest bezpośrednio poniżej w In Layers i Out Layers ("1. The DNS client sends a DNS query to the DNS server."). Jest to bardzo przydatna informacja, która pomoże zrozumieć, co dzieje się podczas procesu komunikacji.

d. Kliknij zakładkę Outbound PDU Details. Jakie informacje podane są w sekcji NAME: DNS QUERY?

e. Kliknij ostatni kolorowy kwadrat DNS Info na liście zdarzeń. Które urządzenie jest wyświetlane?

Jaka jest wartość wyświetlona obok ADDRESS: w sekcji DNS ANSWER zakładki (Inbound PDU Details)?

f. Znajdź pierwsze zdarzenie HTTP na liście i kliknij kolorowe pole kwadratu zdarzenia TCP bezpośrednio po

tym zdarzeniu. Zaznacz Layer 4 w zakładce OSI Model. Na podstawie numerowanej listy bezpośrednio poniżej obszarów In Layers i Out Layers napisz jakie informacje wyświetcone są w punkcie 4 i 5?

TCP zarządza łączeniem i rozłączaniem kanału komunikacyjnego wraz z innymi obowiązkami. To określone zdarzenie pokazuje, że kanał komunikacyjny został ustanowiony (ESTABLISHED).
g. Kliknij ostatnie zdarzenie TCP. Zaznacz Layer 4 w zakładce OSI Model. Przeanalizuj kroki opisane bezpośrednio pod obszarami In Layers i Out Layers. W oparciu o informacje zawarte w ostatniej pozycji na liście (powinna być pozycja 4) napisz jakie jest przeznaczenie tego zdarzenia?

Symulacja stanowi przykład sesji internetowej pomiędzy klientem a serwerem w sieci lokalnej (LAN). Klient generuje żądania do określonych usług działających na serwerze. Serwer musi być skonfigurowany do nasłuchiwanego na określonych portach żądań klienta. (Podpowiedź: Spójrz na warstwę 4 zakładki OSI Model żeby zobaczyć informacje o porcie.)

Na podstawie informacji, która została sprawdzona podczas przechwytywania w Packet Tracer, napisz jaki numer portu ma, nasłuchujący żądań stron WWW serwer (Web Server)?