

АРХІТЕКТУРА ІНТЕРНЕТУ

Статтю присвячено дослідженню внутрішньої структури Всесвітньої павутини. Метою є визначення співвідношення складових Інтернет-простору. Об'єкт дослідження складають різнорівневі елементи, які забезпечують функціонування Інтернету.

Інтернет є дуже складною структурою, для забезпечення праці якої потрібні різні види **синхронної** взаємодії великої кількості компонентів. Передача даних у мережі Інтернет проходить за моделлю **клієнт-сервер**, тобто є **серверна сторона** (один комп'ютер, або ціла локальна мережа, що працює, як один), що містить певні дані, та **клієнтська сторона**, що ці дані запитує. Для забезпечення налагодженості праці цих двох сторін, передача даних у мережі Інтернет виконується згідно з загальноприйнятими **протоколами**, кожен з яких являє собою «сукупність правил, що регламентують формат даних та процедури їх пересилання у каналі зв'язку» [1:518]. Протоколи є строго регламентованими, як правило, на законодавчому рівні, наприклад, в організації ISO (*International Organization for Standardization*). Праця Інтернету забезпечується набором протоколів, сукупність яких називається **стеком**. На даний момент безперечним стандартом є стек TCP/IP (*Transmission Control Protocol/Internet Protocol*).

Стек TCP/IP – це дуже розвинута і багаторівнева система. Існує дві класифікації протоколів, що входять до стеку: модель OSI (*Open Systems Interconnection*) та модель TCP/IP (вони дуже мало різняться між собою – в моделі OSI було додано два додаткових рівні). Ці моделі класифікують протоколи за **рівнями** (*layers*) комунікації – «підрозділами моделі OSI, які складається з підсистем одного рангу» [9:6]. «Не всі системи поєднані безпосередньо, навпрямки, тому деякі системи працюють як ретрансляційне обладнання» [9:29], тобто, коли вони приймають дані, вони відправляють їх на іншу адресу, і так дані переходять від системи до системи, поки не дістануться системи-адресата. Це означає, що всі рівні

«вкладаються» один в інший, як лялька-матрьошка, тобто для виконання однієї операції на високому рівні потрібно виконати кілька операцій рівнем нижче, а для виконання кожної з цих операцій потрібно по кілька операцій ще нижчого рівня; і все зводиться до бінарних (див. нижче) операцій на першому, найнижчому рівні – фізичному. Можливо виділити сім таких рівней:

1. **Фізичний** – матеріальна репрезентація бінарних (конвертованих у двійкову систему числення) даних у вигляді електричних чи оптичних (волоконно-оптична лінія передачі) сигналів або радіохвиль. На фізичному рівні єдиною доступною одиницею інформації є **біт**, тобто відсутність сигналу (позначається нульом) або його наявність (позначається одиницею). Для переведення комунікації між різними частинами комп'ютера (а згодом і між різними комп'ютерами) на вищій рівень, була розроблена **двійкова система числення**, яка оперує лише двома цифрами (0, 1).
2. **Канальний** – передача даних між вузлами у межах однієї локальної мережі [3:27]. Дані передаються у вигляді одиниць, які називаються **фреймами**. Кожен фрейм складається із наступних частин:
 - **Заголовка (header)** – канальна адреса адресата й адресанта, а також інша керівна інформація
 - **Наповнення (payload)** – у фреймі, як у контейнері, знаходиться пакет
 - **Кінцевого поля (trailer)** – контрольний код, який дозволяє перевірити цілісність переданого фрейма. Якщо виявляється, що фрейм дійшов до адресата у пошкодженому вигляді, то адресат запрошує повторну висилку фрейму
3. **Мережовий** – реалізує міжмережову взаємодію між комп'ютерами. На мережовому рівні дані передаються адресату, якого можуть відділяти від адресанта десятки мережових вузлів і для передачі одного пакету на цьому рівні потрібно буде здійснити десятки з'єднань на канальному рівні. Як було

зазначено раніше, для передачі на канальному рівні використовується канална, тобто фізична адреса отримувача. На мережовому ж рівні використовуються логічна адреса, індивідуальна для кожного комп'ютера. Це називається **IP-адреса**. Приклади: 192.168.0.1. Типовий пакет складається із:

- **Мережевий заголовок** – IP-адреса адресата та інша керівна інформація
- **Дані** – безпосередньо інформація, яку потрібно передати

Пакети можуть різнитися за структурою. Кінцеве поле у пакетів зустрічається рідко.

4. **Транспортний** – забезпечує надійність доставки пакетів [5:205], тобто відповідає за виявлення помилок та виправлення їх. Цей рівень допомагає встановити, чи співпадають прийняті адресатом дані з тими даними, що відправив адресант. У разі виявлення помилок дані відкидаються і у адресанта запрошується повторна висилка спотвореного сегмента даних. Приклади протоколів, що працюють на цьому рівні: TCP (*Transmission Control Protocol*), UDP (*User Datagram Protocol*).
5. **Сеансовий** – «встановлює та розриває зв'язок між комп'ютерами, керує діалогом між ними, а також надає засоби синхронізації» [5:206]. Засоби синхронізації дозволяють відновляти передачу даних у разі обриву зв'язку з момента зупинки. Основною метою цього рівня є здійснення **сеанса** зв'язку – логічного з'єднання між комп'ютерами, яке містить три фази:
 - Встановлення зв'язку і передача інформації про майбутній сеанс
 - Передача інформації
 - Розрив зв'язку
6. **Представницький** – «змінює форму інформації, що передається, але не змінює її змісту» [5:206]. Цей рівень може бути використаним для підвищення безпечності, наприклад, якщо інформація буде зашифрована по протоколу SSL (*Secure Socket Layer*).

7. **Прикладний** – це «набір різноманітних протоколів, завдяки яким користувач отримує доступ до ресурсів, що спільно використовуються» [5:206]. Приклади протоколів: HTTP (*HyperText Transfer Protocol*), FTP (*File Transfer Protocol*), POP (*Post Office Protocol*), SMTP (*Simple Mail Transfer Protocol*), SMB (*Server Message Block*) та багато інших.

Схематично ці рівні можна представити у вигляді наступної таблиці [7]:

Таблиця 1

Рівні мережової моделі OSI

Тип даних	Рівень	Функції
Дані	7: Прикладний	Взаємодія програми з Інтернетом
	6: Представницький	Зміна форми передаваної інформації без зміни змісту
	5: Сеансовий	Встановлення сеансу зв'язку між двома машинами
Сегменти	4: Транспортний	Виключає помилки при трансфері даних
Пакети	3: Мережовий	Передача даних між двома вузлами незалежно від мереж
Фрейми	2: Канальний	Передача даних між двома вузлами у межах однієї локальної мережі
Біти	1: Фізичний	Матеріальна репрезентація даних

Розвинута система протоколів виконує величезну працю задля передачі даних. Цими даними можуть бути **веб-сторінки, файли** або **службова інформація** програм (наприклад, текст повідомлень у програмах для обміну миттєвими повідомленнями). Всесвітню павутину формують веб-сторінки, тому нижче буде описано саме їх.

Для перегляду веб-сторінок використовуються спеціальні програми – **браузери** (найпопулярніші браузери: *Internet Explorer, Mozilla Firefox, Opera, Google Chrome, Apple Safari*), головною функцією яких є відображення сторінки у тому вигляді, як планував автор сторінки. Це досягається через коректну інтерпретацію браузером мови програмування (див. нижче), якою написана сторінка і, як наслідок, заплановане автором розташування та вигляд тексту (пасуючий кегль) і графічних елементів (малюнки та елементи дизайну, наприклад, анімовані кнопки) [8]. Взагалі, **мова програмування** – це штучна мова, яка була створена для запису чіткої послідовності дій (**алгоритм**) для виконання поставленої проблеми, текст написаний якою (текст програми ще називається **сирцевим кодом** (*source code*) чи **лістингом**) є зрозумілим і для людини і для комп'ютера. Мови Веб-програмування можна розділити на:

- **мови розмітки** (HTML, XML)
- **мови сценаріїв** (JavaScript, VBScript)
- **серверні мови** (PHP, Perl).

В основі кожного сайту лежить сукупність (у деяких випадках – один) **гіпертекстових документів** (загальноприйнята назва – **веб-сторінка**) – «текстовий документ, який містить посилання на інші частини даного документа, на інші документи та об'єкти нетекстового формату (звук, графіка, відео)» [1:647], для оформлення яких використовується **мова гіпертекстової розмітки** (HTML – *Hypertext Markup Language*). Вона дає дуже обмежені, лише базові можливості при створенні сторінки: користувач має можливість лише переглядати статичну сторінку без можливості інтерактивної взаємодії із сервером, що її містить. Єдина можлива взаємодія – це запросити у сервера наступну сторінку при натисненні на посилання. Певні елементи інтерактивності привносять **сценарії** (безперечним стандартом серед мов для написання сценаріїв є JavaScript) – код, написаний іншою мовою програмування, який вбудован в основний код сторінки (написаний мовою HTML)

та містить набір інструкцій, що виконуються на клієнтській машині. Але це є лише часткова інтерактивність, оскільки сценарій виконується на клієнтській машині, а зв'язка із сервером всеодно немає. До того ж, сценарій здатен виконувати лише ті функції, які автор прорахував і включив до можливих варіантів виконання сценарію. Якщо на певному етапі користувач зробив щось, чого автор не передбачив, трапляється **виняток** – непередбачувана подія, що негативно впливає на виконання сценарію і призводить або до неправильних результатів, або до повної неможливості його виконання.

Наступний рівень розвитку Інтернету – це мови програмування, за допомогою яких є можливою генерація сторінок на сервері. Найпоширенішою мовою такого типу є PHP. Коли користувач запрошує певну сторінку, сервер генерує її за певним сценарієм, «заточуючи» під конкретного користувача, відсилає користувачу цю сторінку і витирає черговий результат виконання сценарія із своєї оперативної пам'яті. Кожен наступний користувач, який запросить цю сторінку, отримає іншу її копію, виготовлену спеціально для нього, зазвичай із зовсім різним змістом. У парі із PHP майже завжди використовується **база даних** – «організована у згідності з установленими правилами і дотримувана в пам'яті комп'ютера сукупність даних, що характеризує актуальний стан деякої предметної царини і використовується для задоволення інформаційних потреб користувачів» [4]. Найпопулярнішим типом баз даних є **реляційні бази даних** (назва походить від математичного поняття «відношення» (англ. *relation*), неофіційним синонімом якого часто виступає термін «таблиця» [6]), які складаються із таблиць. Таблиця містить колонки, які виступають як **атрибути** даних і строки, які містять певну кількість **полей** (так називається комірка таблиці) даних, що містять безпосередньо інформацію. Наприклад, так може виглядати база даних, що містить інформацію про студентів:

Таблиця 2

Приклад бази даних

Первинний ключ	Ім'я студента	Курс	Група	Бал
1	Алібабаєв Ахмед	4	42-н	83
2	Петрова Анастасія	2	26-а	65
3	Мазайло Грицько	3	30-ф	70

Характерною ознакою бази даних є наявність первинного ключа – першого стовбця, який служить ідентифікатором строки даних. Якщо відоме значення первинного ключа, то становиться можливим вибірковий **запрос** (звідси і назва мови програмування, яка використовується для роботи із базами даних – SQL, від *Structured Query Language*) саме тих даних, що потрібні, а не пошук по всій таблиці [2:885]. Наприклад, при користуванні наведеною вище базою даних потрібно знати лише номер первинного ключа для отримання всієї наявної інформації про потрібного студента.

Зв'язка PHP+MySQL започаткувала нову еру в розвитку сайтів. Стало можливим пристосувати до Веб-сайтів концепцію персонального **облікового запису** – строки у базі даних сайту, що містить сукупність інформації про обраного користувача, його прав стосовно інших користувачів у багатокористувацької системи і інформацію стосовно переваг щодо власного перебування на сайті, тобто так звані **налаштування** користувача, згідно з якими мають генеруватись унікальні сторінки саме для нього. Таким чином, кожен користувач тепер має можливість завести власний «кабінет» на сайті, до якого інші користувачі не мали доступу (перед початком роботи користувач має ввести власні **нікнейм** та **пароль**: ідентифікатор та кодову фразу, яка (в ідеалі) відома лише цьому користувачу).

Найновішою розробкою є технологія AJAX (*Asynchronous JavaScript And XML*), яка дозволяє повну інтерактивність, що досягається за допомогою постійного оновлення змісту сторінки. Найновіші портали, наприклад, *Facebook* та *ВКонтакте*

використовують цю технологію для підвищення власної привабливості для користувачів.

Інтернет не є гомогенною структурою, тому що різні його елементи виконують різні функції. Трансформація Всесвітньої павутини в атрибут сучасного життя сильно змінила конфігурацію інформаційного середовища, що оточує нас. Саме тому різноманітні дослідження Інтернету видаються перспективними і плідними для подальших розвідок вчених.

ЛИТЕРАТУРА

1. Бройдо В.Л., Ильина О.П. Архитектура ЭВМ и систем. – СПб: Питер, 2009. – 2-е изд. – 720 с.
2. Дейтел Х.М., Дейтел П.Дж., Нието Т.Р. Как программировать для Internet & WWW. – М.: Бином, 2002. – 1184 с.
3. Досталек Л., Кабелова А. TCP/IP и DNS в теории и на практике. – СПб: Наука и Техника, 2006. – 608 с.
4. Когаловский М.Р. Энциклопедия технологий баз данных. – М.: Финансы и статистика, 2002. – 800 с.
5. Колисниченко Д.Н., Аллен П.В. Linux: Полное руководство. – СПб: Наука и Техника, 2007. – 2-е изд. – 784 с.
6. Реляционная модель данных // http://ru.wikipedia.org/wiki/Реляционная_модель_данных
7. Сетевая модель OSI // http://ru.wikipedia.org/wiki/Сетевая_модель_OSI
8. How the Web Works // <http://public.web.cern.ch/public/en/About/WebWork-en.html>
9. Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model. – ITU, 1994. – 64 p.