

ЛЕКЦІЯ 1. ОСНОВИ КОМУНІКАЦІЇ В ІНТЕРНЕТ

Для розуміння того як працює веб-сторінка, можна спрощено переглянути те, що відбувається під час перегляду веб-сторінки у веб-браузері на комп'ютері або в телефоні.

1.1 Клієнти та сервери

Комп'ютери, підключені до мережі, називаються клієнтами і серверами. Спрощена схема взаємодії між пристроєм клієнта та сервером подана на рисунку 1.1.

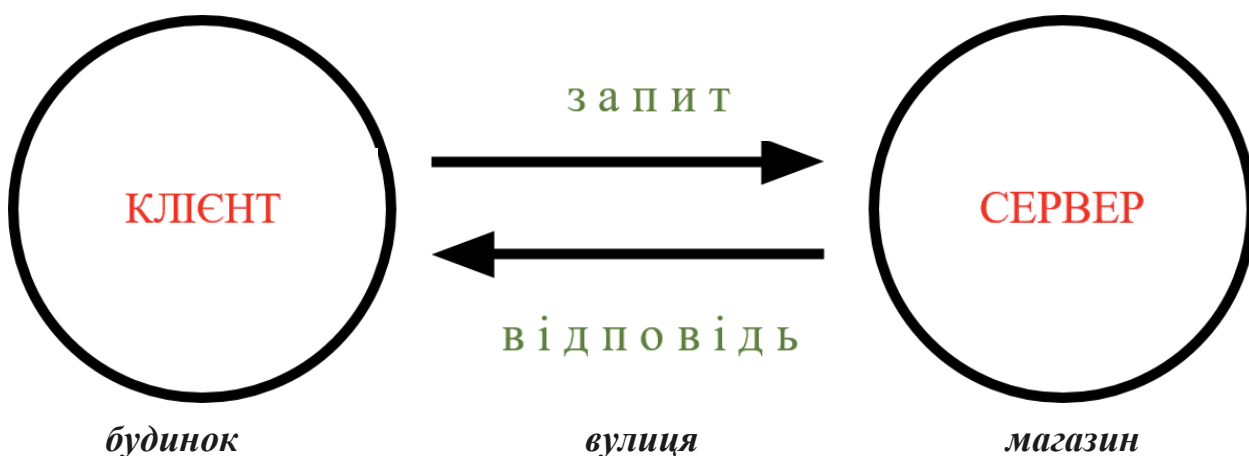


Рисунок 1.1 – Схема з'єднання в інтернет

Клієнти – це типові, підключені до Інтернету, пристрої веб-користувача (наприклад, комп'ютер, підключений до Wi-Fi, або телефон, підключений до вашої мобільної мережі), а також програмне забезпечення для доступу до Інтернету, встановлене на цих пристроях (зазвичай веб-браузер, наприклад Firefox або Chrome).

Сервери – це комп'ютери, які зберігають веб-сторінки, сайти або застосунки. Коли клієнтський пристрій хоче отримати доступ до веб-сторінки, копія веб-сторінки завантажується з сервера на клієнтський пристрій, для відображення у веб-браузері користувача.

Клієнт та сервер не єдині елементи, які потрібні для роботи в мережі.

Представимо мережу як вулицю, на одному боці якої знаходиться пристрій клієнта – це ваш будинок. На іншому боці вулиці знаходиться сервер – це магазин, в якому ви хочете щось купити.



Для отримання інформації в мережі (здійснення покупки) крім пристрою клієнта й сервера, потрібні також:

- Інтернет-з'єднання: дозволяє надсилати та отримувати дані в мережі (вулиця між вашим будинком і магазином).

- TCP/IP: Transmission Control Protocol і Internet Protocol — протоколи зв'язку, які визначають, як дані повинні переміщатися через інтернет (транспортний механізми, які дозволяють вам розмістити замовлення, зайти в магазин й придбати товари – машина або велосипед).

- DNS: Domain Name System — це як адресна книга для веб-сайтів. Під час введення веб-адреси в браузері браузер дивиться DNS, щоб знайти IP-адресу веб-сайту, перш ніж він зможе досягти цього веб-сайту. Браузер повинен з'ясувати, на якому сервері знаходиться веб-сайт, для надсилання HTTP-повідомлення в потрібне місце (пошук адреси магазину, щоб ви могли істатися до нього).

– HTTP: Hypertext Transfer Protocol — протокол, що визначає мову для спілкування пристроїв клієнтів та серверів (мова, якою ви користуєтеся для замовлення ваших товарів).

– Файли компонентів (Component files): веб-сайт складається з багатьох різних файлів (різні типи товарів, які ви купуєте в магазині). Ці файли можуть бути двох основних типів:

а) файли кодів (Code Files): веб-сайти побудовані з використанням HTML, CSS і JavaScript та за допомогою інших технологій.

б) активи (Assets): під цією назвою об'єднанні всі інші компоненти, з яких складається веб-сайт, таких як зображення, музика, відео, документи Word, PDFs.

1.2. Процес отримання інформації в мережі

Розглянемо процес отримання інформації в мережі докладніше.

Під час введення веб-адреси в браузері (вихід з дому до магазину):

1 Браузер переходить на DNS-сервер, і знаходить справжню адресу сервера, на якому працює потрібний веб-сайт (ви знаходите адресу магазину).

2 Браузер надсилає серверу повідомлення HTTP-запиту, з проханням надіслати клієнту копію веб-сайту (ви заходите в магазин і замовляєте свій товар). Це повідомлення та всі інші дані, надіслані між клієнтом та сервером, надсилаються через ваше інтернет-з'єднання за допомогою TCP/IP.

3 Якщо сервер схвалює запит клієнта, сервер надсилає клієнту повідомлення "200 ОК", що означає "звичайно, ви можете подивитися на цей веб-сайт! Ось він», а потім починає відправляти файли сайту в браузер як серію невеликих повідомлень – пакетних даних (магазин дає вам ваші товари, а ви відносити їх у свій будинок).

4 Браузер об'єднує окремі повідомлення в повну веб-сторінку та показує її клієнту (товар у дверей вашого будинка).

1.3. Порядок обробки файлів компонентів

Коли браузери надсилають запити на сервери для файлів HTML, ці файли HTML часто містять елементи <link>, що посилаються на зовнішні таблиці стилів CSS і елементи <script>, що посилаються на зовнішні скрипти

JavaScript. Важливо знати порядок, в якому ці файли обробляються браузером під час завантаження сторінки:

1 Браузер спочатку аналізує HTML-файл, і це призводить до того, що браузер розпізнає всі `<link>`-елементні посилання на зовнішні таблиці стилів CSS та всі `<script>`-елементні посилання на скрипти.

2 Коли браузер обробляє HTML, він посилає запити назад на сервер для CSS-файлів, які він знайшов у елементах `<link>`, і будь-яких JavaScript-файлів, які він знайшов у елементах `<script>`, потім аналізує CSS та JavaScript.

3 Браузер генерує в пам'яті DOM дерево з аналізу HTML, генерує в пам'яті CSSOM структуру з розпакованої CSS, компілює та виконує розпакований JavaScript.

4 Під час того, як браузер будує дерево DOM, застосовує стилі з дерева CSSOM та виконує JavaScript, візуальне представлення сторінки з'являється на екрані пристрою клієнта, користувач бачить контент веб-сторінки і може почати взаємодіяти з нею.

Об'єктна модель документа (Document Object Model, DOM) з'єднує веб-сторінки зі скриптами або мовами програмування, представляючи структуру документа, наприклад HTML, яка представляє веб-сторінку в пам'яті. Зазвичай це стосується JavaScript, хоча моделювання HTML, SVG або XML документів як об'єктів не є основними задачами програмування JavaScript.

DOM є документом з логічним деревом. Кожна гілка дерева закінчується у вузлі, який містить певний об'єкт. Методи DOM дозволяють програмувати доступ до дерева. За допомогою них можна змінити структуру документа, стиль або зміст.

Логічне дерево DOM

Основними елементами HTML-документа є теги. Згідно з Document Object Model (DOM), кожен HTML-тег є об'єктом. Вкладені теги – це 'діти' батьківського елемента. Текст всередині тегу також є об'єктом. Всі ці об'єкти доступні за допомогою JavaScript та використовуються для зміни сторінки. Наприклад, `document.body` – це об'єкт, що представляє тег `<body>`.

Запуск наведеного нижче коду зробить `<body>` червоним протягом 3 секунд:

```
1 document.body.style.background = 'red'; // make the background red
2
3 setTimeout(() => document.body.style.background = '', 3000); // return back
```

Тут використано `style.background` для зміни кольору фону документа `body`, але код може містити й багато інших властивостей, таких як: `innerHTML` – HTML вміст, `offsetWidth` – ширина тощо.

Перш ніж розглядати способи маніпулювання DOM слід з'ясувати його структуру.

Приклад логічного дерева ДОМ

Наведемо приклад простого документа:

```
1 <!DOCTYPE HTML>
2 <html>
3 <head>
4   <title>About elk</title>
5 </head>
6 <body>
7   The truth about elk.
8 </body>
9 </html>
```

DOM представляє HTML як структуру дерева тегів. Для наведеного у прикладі документа таке дерево матиме виглядає:



Об'єктна модель CSS (CSS Object Model, CSSOM) — набір API для читання і модифікації інформації, пов'язаної зі стилем документа (CSS). Іншими словами, подібно до того, як DOM дозволяє читати та змінювати структуру та вміст документа з JavaScript, CSSOM дозволяє корегувати стиль документа зчитуючи його з JavaScript.

CSS Object Model — набір API, що дозволяє маніпулювати CSS за використання JavaScript. Він багато в чому схожий на DOM, але для CSS не використовують HTML. Він дозволяє користувачам динамічно читати і змінювати стиль CSS.

1.3. Призначення DNS

Справжні веб-адреси мають вигляд не прийнятний для запам'ятовування та введення до рядка адреси браузера, для пошуку веб-сайтів. Вони є спеціальними наборами чисел, які мають наступний вигляд: 63.245.215.20.

Це називається IP-адресою, і вона є унікальним розташуванням в мережі. Проте, такий набір чисел не є читабельним, тому було винайдено сервери доменних імен. Це спеціальні сервери, які відтворюють відповідність

між веб-адресою, яку ви вводите до вашого браузера (наприклад, "mozilla.org"), та справжньою IP-адресою веб-сайту.

Domain Name System, DNS – це комп'ютерна розподілена система (БД, частини якої розташовані на різних комп'ютерах мережі) для отримання інформації про домени.

Веб-сайти можуть бути досягнуті безпосередньо через їх IP-адреси. Для пошуку IP-адреси веб-сайту можна скористатися інструментом пошуку DNS (<https://www.nslookup.io/website-to-ip-lookup/>).

1.4. Призначення пакетів

В основному, коли дані надсилаються по мережі, вони надсилаються тисячами невеличких фрагментів. Для цього є кілька причин, наприклад, якщо дані були втрачені або пошкоджені, їх легше замінити невеликими порціями. Крім того, пакети можуть маршрутизуватися по різних шляхах, що робить обмін швидшим і дозволяє багатьом різним користувачам завантажувати один і той же веб-сайт одночасно. Якщо кожен веб-сайт був відправлений як один великий шматок, лише один користувач може завантажити його в один й той самий момент часу, що зробить веб дуже неефективним та зручним у використанні.

1.5. HTTP – протокол прикладного рівня

У Бутані, коли люди зустрічаються, вони зазвичай вітають один одного з «чи добре ваше тіло?» У Японії вони можуть вклонятися, залежно від обставин. В Омані чоловіки часто цілують один одного в ніс після потискання рук. У Камбоджі та Таїланді вони складають руки, ніби моляться. Все це комунікаційні протоколи, проста послідовність кодів, які мають сенс і готують дві сторони для змістовного обміну.

В Інтернеті протокол, який готує комп'ютери для обміну даними – протокол гіпертекстової передачі або HTTP.

Протокол передачі гіпертексту HTTP (HyperText Transfer Protocol) – протокол прикладного рівня передачі даних. Раніше використовувався для

передачі даних у вигляді гіпертекстових документів у форматі HTML, сьогодні використовується для передачі довільних даних.

Протокол прикладного рівня – протокол, який забезпечує взаємодію між мережею та користувачем. На цьому рівні застосунки користувача мають доступ до мережеслужб: обробнику запитів до БД, доступу до файлів, пересиланню електронної пошти. Протокол цього рівня відповідає за надання службової інформації: передає у застосунки інформацію про помилки, формує запити до рівня представлення. Приклади таких протоколів: HTTP, POP3, SMTP.

1.6. Модель OSI

Модель OSI (Open Systems Interconnection) або базова еталонна модель взаємодії відкритих систем (EMBBS) Open Systems Interconnection Basic Reference Model (1978 р.) – абстрактна мережева модель для комунікацій та розроблення мережеслужб. Представляє рівневий підхід до мережі. Кожен рівень обслуговує свою частину процесу взаємодії. Завдяки такій структурі спільна робота мережевого обладнання й програмного забезпечення стає набагато простішою, прозорішою й зрозумілішою.

На сьогодні основним використовуваним стеком протоколів є TCP/IP, розроблення якого не було пов'язане з моделлю OSI і до того ж було здійснено до її прийняття. За увесь час існування моделі OSI вона не була реалізована, і, очевидно, не буде реалізована ніколи. Сьогодні використовується тільки деяка підмножина моделі OSI. Вважається, що модель занадто складна, а її реалізація займе забагато часу.

Модель OSI	
Дані	Рівень
Дані	<p>7. Прикладний [сховати] <i>доступ до мережевих служб</i></p> <p>NNTP · SIP · SSI · DNS · FTP · Gopher · HTTP · NFS · NTP · SMPP · SMTP · SNMP · Telnet · DHCP · NETCONF · <i>більше...</i></p>
Дані	<p>6. Представлення [сховати] <i>представлення і кодування даних</i></p> <p>MIME · XDR</p>
Дані	<p>5. Сеансовий [сховати] <i>керування сеансом зв'язку</i></p> <p>Named pipe · NetBIOS · SAP · PPTP · RTP · SOCKS · SPDY</p>
Блоки	<p>4. Транспортний [сховати] <i>безпечне та надійне з'єднання «точка - точка»</i></p> <p>TCP · UDP · SCTP · DCCP · SPX</p>
Пакети	<p>3. Мережевий [сховати] <i>визначення маршруту та логічних адрес</i></p> <p>IP (IPv4 · IPv6) · ICMP · IGMP · IPX · AppleTalk · X.25</p>
Кадри	<p>2. Канальний [сховати] <i>MAC та LLC (фізична адресація)</i></p> <p>ATM · ARP · IS-IS · SDLC · HDLC · CSLIP · SLIP · GFP · PLIP · IEEE 802.2 · LLC · MAC^[en] · L2TP · IEEE 802.3 · Frame relay · ITU-T G.hn DLL^[en] · PPP · X.25 LAPB</p>
Біти	<p>1. Фізичний [сховати] <i>кабель, сигнали, бінарна передача</i></p> <p>RS-232 · RS-449 · List of ITU-T V-series recommendations^[en] · PDH · SONET/SDH^[en] · PON · OTN · DSL · IEEE 802.3 · IEEE 802.11 · IEEE 802.15^[en] · IEEE 802.16 · IEEE 1394 · USB · Bluetooth</p>

Рисунок 1.2 – Рівні моделі OSI

Модель					
Уровень (layer)		Тип данных (PDU ^[15])	Функции	Примеры	Оборудование
Host layers	7. Прикладной (application)	Данные	Доступ к сетевым службам	HTTP, FTP, POP3, SMTP, WebSocket	Хосты (клиенты сети), Межсетевой экран
	6. Представления (presentation)		Представление и шифрование данных	ASCII, EBCDIC, JPEG, MIDI	
	5. Сеансовый (session)		Управление сеансом связи	RPC, PAP, L2TP, gRPC	
	4. Транспортный (transport)	Сегменты (segment) / Датаграммы (datagram)	Прямая связь между конечными пунктами и надёжность	TCP, UDP, SCTP, Порты	
Media ^[16] layers	3. Сетевой (network)	Пакеты (packet)	Определение маршрута и логическая адресация	IPv4, IPv6, IPsec, AppleTalk, ICMP	Маршрутизатор, Сетевой шлюз, Межсетевой экран
	2. Канальный (data link)	Биты (bit)/Кадры (frame)	Физическая адресация	PPP, IEEE 802.22, Ethernet, DSL, ARP, сетевая карта.	Сетевой мост, Коммутатор, точка доступа
	1. Физический (physical)	Биты (bit)	Работа со средой передачи, сигналами и двоичными данными	USB, RJ («витая пара», коаксиальный, оптоволоконный), радиоканал	Концентратор, Повторитель (сетевое оборудование)

7. Прикладний рівень (Application layer)

Верхній (7-й) рівень моделі, забезпечує взаємодію мережі й користувача. Рівень дозволяє прикладним програмам користувача доступ до мережних служб, таких як обробник запитів до баз даних, доступ до файлів, пересилання електронної пошти. Також відповідає за передачу службової інформації, надає програмам інформацію про помилки й формує запити до рівня представлення.

6. Рівень представлення (Presentation layer)

Цей рівень відповідає за перетворення протоколів і кодування/декодування даних. Запити програм, отримані з прикладного рівня, він перетворює у формат для передачі по мережі, а отримані з мережі дані перетворює у формат, зрозумілий для застосунків. На цьому рівні може здійснюватися стиснення/розпакування або кодування/декодування даних, а також перенаправлення запитів іншому мережевому ресурсу, якщо вони не можуть бути оброблені локально.

5. Сеансовий рівень (Session layer)

Відповідає за підтримку сеансу зв'язку, дозволяючи програмам взаємодіяти між собою тривалий час. Рівень керує створенням/завершенням сеансу, обміном інформацією, синхронізацією завдань, визначенням права на передавання даних і підтримкою сеансу в періоди неактивності програм. Синхронізація передавання забезпечується розміщенням у потік даних контрольних точок, починаючи з яких відновлюється процес при порушенні взаємодії.

4. Транспортний рівень (Transport layer)

Транспортний рівень (Transport layer) — 4-й рівень моделі OSI, призначений для доставлення даних без помилок, втрат і дублювання в тій послідовності, у якій вони були передані. При цьому немає значення, які дані передаються, звідки й куди, тобто він визначає сам механізм передачі. Блоки даних він розділяє на фрагменти, розмір яких залежить від протоколу, короткі об'єднує в один, довгі розбиває. Протоколи цього рівня призначені для взаємодії типу точка-точка.

3. Мережевий рівень (Network layer)

3-й рівень мережної моделі OSI, призначений для визначення шляху передавання даних. Відповідає за трансляцію логічних адрес й імен у фізичні, визначення найкоротших маршрутів, комутацію й маршрутизацію пакетів, відстеження неладів і заторів у мережі. На цьому рівні працює такий мережний пристрій, як маршрутизатор.

2. Канальний рівень (Data Link layer)

Цей рівень призначений для забезпечення взаємодії мереж на фізичному рівні й контролю за помилками, які можуть виникнути. Отримані з фізичного рівня дані він упаковує в кадри даних[джерело?], перевіряє на цілісність, якщо потрібно — виправляє помилки й відправляє на мережний рівень. Канальний рівень може взаємодіяти з одним або декількома фізичними рівнями, контролюючи цю взаємодію й керуючи нею. Специфікація IEEE 802 поділяє цей рівень на 2 підрівня — MAC (Media Access Control) регулює доступ до поділюваного фізичного середовища, LLC (Logical Link Control) забезпечує

обслуговування мережного рівня. На цьому рівні працюють комутатори, мости й мережеві адаптери.

MAC-підрівень забезпечує коректне спільне використання загального середовища, надаючи його в розпорядження тієї або іншої станції мережі. Також додає адресну інформацію до фрейму, позначає початок і кінець фрейму.

Рівень LLC відповідає за достовірне передавання кадрів даних між вузлами, а також реалізує функції інтерфейсу з мережевим рівнем за допомогою фреймування кадрів. Також здійснює ідентифікування протоколу мережевого рівня.

У програмуванні цей рівень представляє драйвер мережної карти, в операційних системах є програмний інтерфейс взаємодії канального й мережного рівня між собою, це не новий рівень, а просто реалізація моделі для конкретної ОС. Приклади таких інтерфейсів: NDIS, ODI.

1. Фізичний рівень (Physical layer)

Найнижчий рівень моделі, призначений безпосередньо для передавання потоку даних. Здійснює передавання електричних або оптичних сигналів у кабель і відповідно їхнє приймання і перетворення на біти даних відповідно до методів кодування цифрових сигналів. Інакше кажучи, здійснює інтерфейс між мережним носієм і мережним пристроєм. На цьому рівні працюють концентратори й повторювачі (ретранслятори) сигналу. Фізичний рівень визначає електричні, процедурні і функціональні специфікації для середовища передавання даних, в тому числі роз'єми, розпаювання і призначення контактів, рівні напруги, синхронізацію зміни напруги, кодування сигналу.

Цей рівень приймає кадр даних від канального рівня, кодує його в послідовність сигналів, які потім передаються у лінію зв'язку. Передавання кадру даних через лінію зв'язку вимагає від фізичного рівня визначення таких елементів: тип середовища передавання (дротовий або бездротовий, мідний кабель або оптичне волокно) і відповідних конекторів; як мають бути представлені біти даних у середовищі передавання; як кодувати дані; якими мають бути схеми приймача і передавача.

Фізичним рівнем в лінію зв'язку кадр даних (фрейм) не передається як єдине ціле. Кадр представляється як послідовність сигналів, що передаються один за одним. Сигнали, в свою чергу, представляють біти даних кадру.

В сучасних мережах використовуються 3 основних типа середовища передавання: мідний кабель (copper), оптичне волокно (fiber) та бездротове середовище передавання (wireless). Тип сигналу, за допомогою якого здійснюється передавання даних, залежить від типу середовища передавання. Для мідного кабелю сигнали, що представляють біти даних є електричними імпульсами, для оптичного волокна — імпульсами світла. У випадку використання бездротових з'єднань сигнали є радіохвилями (електромагнітними хвилями).

Коли пристрій, що працює на фізичному рівні кодує біти кадру в сигнали для конкретного середовища передавання, він має розрізняти кадри. Тобто позначати, де закінчується один кадр і починається інший. Інакше мережеві пристрої, що здійснюють приймання сигналів, не зможуть визначити, коли кадр буде отриманий повністю. Відомо, що початок і кінець кадру позначається на каналному рівні, але в багатьох технологіях фізичний рівень також може додати спеціальні сигнали, що використовуються тільки для позначення початку і кінця кадру даних.

Технології фізичного рівня визначаються стандартами, що розробляються такими організаціями: The International Organization for Standardization (ISO), The Institute of Electrical and Electronics Engineers (IEEE), The American National Standards Institute (ANSI), The International Telecommunication Union (ITU), The Electronics Industry Alliance/Telecommunications Industry Association (EIA/TIA) тощо. Дані стандарти охоплюють 4 області, що належать фізичному рівню: фізичні та електричні властивості середовища передавання, механічні властивості (матеріали, розміри, розпаювання контактів конекторів), кодування (представлення бітів сигналами), визначення сигналів для управління інформацією. Всі компоненти апаратного забезпечення такі, як мережеві карти (Network interface card, NIC), інтерфейси і конектори, матеріали кабелів та їхня конструкція визначаються стандартами фізичного рівня. Можна зазначити, що функції фізичного рівня вбудовані у мережеве обладнання (hardware).

Основними функціями фізичного рівня є: фізичні компоненти, кодування даних, передавання даних. Фізичні компоненти — електронне обладнання, середовище передавання і конектори, через які передаються сигнали, що представляють біти даних.

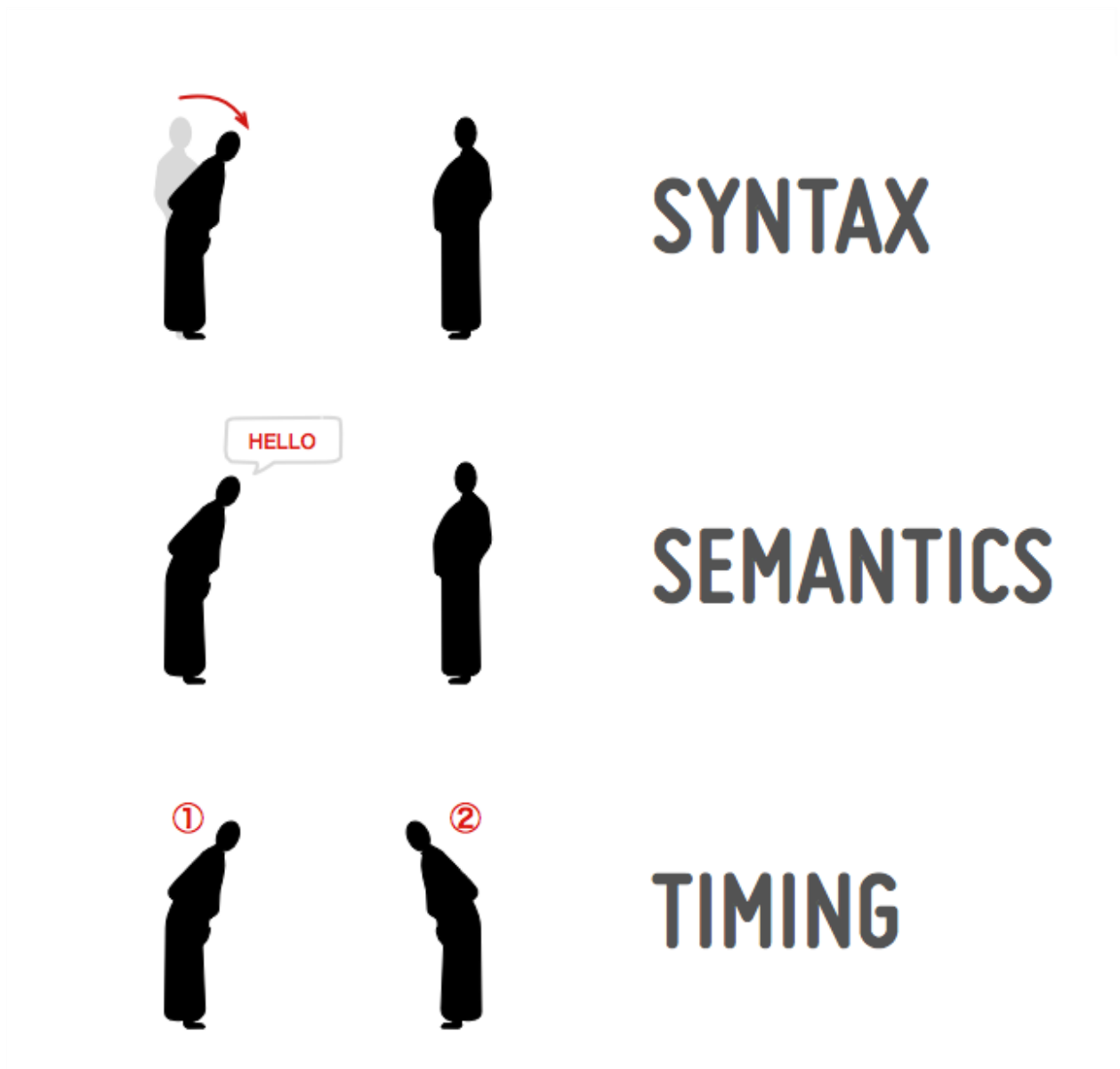
HTTP – протокол прикладного рівня для протоколу TCP/IP, комунікаційний протокол. Розуміння HTTP протоколу допомагає визначити кращі шляхи взаємодії між користувачем та сайтом, досягти кращої продуктивності сайту і створити ефективні інструменти для управління інформацією в Інтернеті.

1.7. Призначення комунікаційного протоколу

Розглянемо базовий сценарій комунікації. Щоб мати можливість спілкуватися, має бути дві сторони (будь це програмне забезпечення, пристрої, люди тощо) для порозуміння між якими потрібні:

- Синтаксис (формат даних і кодування)
- Семантика (контрольна інформація та обробка помилок)
- Таймінг (відповідна швидкість та визначена послідовність)

Коли двоє людей зустрічаються, вони починають використовувати комунікаційний протокол: наприклад, в Японії людина робить специфічний жест з тілом. Одним з таких жестів є уклін, який є синтаксисом, що використовується для взаємодії. Таким чином, коли одна людина вклониться іншій людині, між ними встановлюється послідовність подій в конкретний момент часу.



1.8. Протокол зв'язку

Протокол онлайн-зв'язку містить ті ж елементи. Синтаксис буде послідовністю символів, таких як ключові слова, які використовуються для написання протоколів. Семантика є значенням, пов'язаним з кожним з цих ключових слів і, нарешті, час є порядком, в якому два або більше сутностей обмінюються цими ключовими словами.

Роль HTTP в цьому процесі в тому, що він автономно працює поверх інших протоколів. При підключенні до веб-сайту, наприклад на www.example.org, агент користувача користується пакетом протоколів TCP/IP. Модель TCP/IP була розроблена в 1970 році з 4 різними шарами:

1. Посилання описує доступ до фізичних носіїв (наприклад, за допомогою мережевої карти).
2. Інтернет описує конверт і маршрутизацію даних - к він упакований (IP).
3. Транспортний шар описує спосіб доставки даних з початкової точки до кінцевого пункту призначення (TCP, UDP).
4. Прикладний шар описує значення або формат переданих повідомлень (HTTP).

HTTP – прикладна програма, що знаходиться над протоколом зв'язку. Розділення моделі на незалежні шари допомагає еволюціонувати частинам платформи без необхідності переписувати все. Наприклад, TCP, транспортний протокол, міг еволюціонувати, не змінивши HTTP, прикладного протоколу.

1.9. Призначення сервера

Порт 80 є типовим портом для з'єднання з веб-сервером. Можна переглянути його роботу використовуючи оболонку. Для цього слід відкрити командний рядок та відкрити з'єднання з www.opera.com за портом 80 командою:

```
telnet www.opera.com 80
```

Маємо отримати наступний результат:

```
Trying 195.189.143.147...
Connected to front.opera.com.
Escape character is '^]'.
Connection closed by foreign host.
```

Спроба 195.189.143.147...

Підключений до front.opera.com.

Символ виходу «^】».

З'єднання закрито закордонним вузлом.

Тобто термінал намагається зв'язатися з сервером, розташованим за IP-адресою 195.189.143.147. Якщо ми нічого не робимо, сервер закриє з'єднання

автоматично. Цілком можливо використовувати інший порт і інший комунікаційний протокол, але це найбільш поширені.

Спробуймо ще раз зв'язатися з сервером з використанням HTTP. Для цього вводимо у командний рядок таке повідомлення:

```
telnet www.opera.com 80
```

Після того, як це буде зроблено, а зв'язок буде встановлено, швидко введіть наступне повідомлення HTTP (перед тим, як з'єднання автоматично закриється), після чого двічі натисніть клавішу ENTER/Return:

```
GET / HTTP/1.1
```

```
Host: www.opera.com
```

Це повідомлення вказує:

- GET: що ми хотіли отримати представлення інформації.
- /: що інформація, яку ми хочемо отримати, зберігається в корені сайту.
- HTTP/1.1: ми «спілкуємося» за допомогою HTTP версії 1.1.
- Host::намагаємося досягти конкретного сайту.
- www.opera.com: назва сайту — www.opera.com.

Тепер настав час дати відповідь серверу. Вікно термінала заповнюється вмістом наступних рядків:

```
HTTP/1.1 200 OK
```

```
Date: Wed, 23 Nov 2011 19:41:37 GMT
```

```
Server: Apache
```

```
Content-Type: text/html; charset=utf-8
```

Set-Cookie: language=none; path=/; domain=www.opera.com;
expires=Thu, 25-Aug-2011 19:41:38 GMT

Set-Cookie: language=en; path=/; domain=.opera.com; expires=Sat,
20-Nov-2021 19:41:38 GMT

Vary: Accept-Encoding

Transfer-Encoding: chunked

<!DOCTYPE html>

<html lang="en"> ...

Тут сервер говорить: «Я «спілкуюся» за допомогою HTTP версії 1.1. Ваше прохання було успішно виконано, тому я відповів кодом 200». Рядок OK є необов'язковим і призначений для пояснення того, що цей код означає для людини — у цьому випадку все гаразд і наше повідомлення було прийнято. Потім надсилається серія заголовків HTTP, щоб описати, що таке повідомлення, і як його слід розуміти. Нарешті, включається вміст сторінки, розташованої у корені сайту, починаючи з **<!DOCTYPE html>**.