

УДК 338.47:004.057.4 DOI: [https://doi.org/10.31617/tr.knute.2021\(40\)05](https://doi.org/10.31617/tr.knute.2021(40)05)

Yuliia KOSTIUK

Assistant Lecturer at the Department of Software Engineering and Cyber Security, Kyiv National University of Trade and Economics
19, Kyoto str., Kyiv, 02156, Ukraine

E-mail: kostyuk_yu@knute.edu.ua

ORCID ID: 0000-0001-5423-0985

Yaroslav SHESTAK

sen. Lecturer at the Department of Software Engineering and Cyber Security, Kyiv National University of Trade and Economics
19, Kyotostr., Kyiv, 02156, Ukraine

E-mail: shestack@knute.edu.ua

ORCID ID: 0000-0002-5102-9642

ТРАНСПОРТНИЙ РІВЕНЬ МОДЕЛІ ISO/OSI В КОМП'ЮТЕРНИХ МЕРЕЖАХ

Розглянуто призначення та функції транспортного рівня в управлінні передачею даних. З'ясовано, яку роль виконують протоколи надійності на транспортному рівні в комп'ютерних мережах. Охарактеризовано протоколи TCP та UDP, використання номерів портів. Визначено, які операції виконують протоколи транспортного рівня. Пояснено, у який спосіб передаються блоки даних по протоколу TCP і як підтверджено їх гарантовану доставку. Порівняно операції протоколів транспортного рівня при наскрізному каналі зв'язку.

Ключові слова: протоколи, транспортний рівень, дані, управління передачею, сеанс зв'язку, надійна доставка, датаграми.

Постановка проблеми. Мережі створюють світ, у якому кордони країн, відстані між ними втрачають значення і стираються перешкоди. За довгі роки існування комп'ютерних мереж створено велику кількість різних мережевих протоколів, тобто правил, що дають змогу реалізувати з'єднання й обмін даними між двома і більше підключеними до мережі пристроями. Ці протоколи прийнято співвідносити з так званою еталонною моделлю взаємодії відкритих систем (*Open Systems Interconnection Reference Model*), або просто моделлю *OSI*. Її опис опубліковано у 1984 р. Міжнародною організацією зі стандартизації (*International Standards Organization, ISO*), тому для неї часто використовується інша назва – модель *ISO/OSI*. Це набір специфікацій, які описують мережі з пристроями, вимоги до них, а також способи їхньої взаємодії. Модель *OSI* має вертикальну структуру, в якій мережеві функції розподілені між сімома рівнями. Кожному такому рівню відповідають певні операції, пристрої та протоколи. Реальна взаємодія рівнів, тобто передача інформації всередині одного комп'ютера, можлива тільки за вертикаллю та тільки із сусідніми рівнями, які розташовані вище або нижче [1–4]. Кожний більш високий рівень користується послугами більш низького рівня, маючи поняття, у якому вигляді та в який спосіб потрібно передати йому дані. Завдання більш

низького рівня – прийняти дані, додати свою інформацію і передати дані далі. Тільки дійшовши до самого нижнього, фізичного рівня мережевої моделі, інформація потрапляє в середовище передачі та досягає комп'ютера одержувача. Цей рівень пов'язує вищі рівні, які значною мірою залежать від додатків, з нижніми, що більше прив'язані до ліній зв'язку.

На транспортному рівні (TP) відбувається розбиття потоку даних на сегменти під час відправлення даних або збирання вихідного потоку даних із сегментів у процесі отримання. TP призначений для доставки даних без помилок, втрат і дублювання в тій послідовності, у якій вони передані. Він забезпечує передачу даних між двома додатками з необхідним рівнем надійності. Протоколи TP, які гарантують надійну доставку даних, встановлюють перед обміном ними віртуальне з'єднання та у разі втрати або пошкодження сегментів відправляють їх повторно (наприклад *TCP*). Протоколи ненадійної доставки не ретранслюють даних (наприклад *UDP*). Попри це, найпоширеніші протоколи передачі даних транспортного рівня – *Transmission Control Protocol (TCP)* та *User Datagram Protocol (UDP)* – у більшості випадків добре справляються з поставленими перед ними завданнями [5].

Аналіз останніх досліджень і публікацій. Протоколи транспортного рівня (*Transport Layer*) забезпечують надійну передачу даних для протоколів більш високих рівнів і прикладних додатків. Транспортний рівень відповідає за наскрізну передачу даних між абонентами мережі. Як правило, всі протоколи, починаючи із транспортного рівня і вище, реалізуються програмними засобами кінцевих вузлів мережі – компонентами їхніх мережних операційних систем. Прикладом транспортних протоколів є протоколи *TCP* і *UDP* стека *TCP/IP* і протокол *SPX* стека *Novell*. Усі роботи в цьому напрямі можна розділити на три види [6–9]. Перші [6; 7] – це використання оптимізованих апаратних рішень у пристроях приймання чи передачі даних, другі [8] – підвищення ефективності протоколів зв'язку, треті [9] – комбіновані.

Аналіз досліджень щодо порушення цілісності протоколів міжмережевої взаємодії [3; 6; 10–12] свідчить, що наразі протоколи *TCP/IP* широко використовуються в безпроводних каналах передачі даних, які характеризуються підвищеним рівнем шуму. Це за певних обставин призводить до виникнення багатьох помилок у прийнятих пакетах даних, а отже, до повторних передач цих пакетів. Крім того, на зниження цілісності протоколів *TCP/IP* впливають віддалені мережеві атаки, наприклад атаки типу "зашумлення". Наявні методи підвищення цілісності передачі даних протоколів *TCP/IP* за певних умов завадової обстановки не забезпечують гарантованої доставки повідомлень.

Основною метою статті є визначення та пояснення призначення транспортного рівня в управлінні передачею даних, встановлення ролі протоколів надійності на транспортному рівні в комп'ютерних мережах.

Передбачено виконання таких завдань: описати роль протоколів надійності на транспортному рівні в комп'ютерних мережах, надати характеристику протоколів *TCP* та *UDP* і використання номерів портів, визначити, які операції виконують протоколи транспортного рівня.

Матеріали та методи. Для досягнення мети у дослідженні використано статистичний аналіз і системний підхід.

Результати дослідження. Стек протоколів *TCP/IP* та модель *OSI* виявили себе як надійна, відносно проста технологія передачі даних у комп'ютерних мережах. Але за наявності надлишкового шуму в каналі передачі даних виникає необхідність у застосуванні додаткових методів підвищення вірогідності передачі інформації.

Протоколи *TCP* і *UDP*, що широко використовуються у комп'ютерних мережах, майже не зазнали змін. У більшості випадків вони добре справляються з поставленими перед ними завданнями, проте виникають ситуації, за яких стандартних рішень недостатньо для створення необхідного рівня надійності передачі даних.

Транспортний рівень семирівневої моделі *OSI* забезпечує передачу даних мережею, підтримку і відстеження кількох сеансів зв'язку. Він відповідає за логічні зв'язки між додатками, що працюють на різних вузлах, а також за поділ програми на блоки відповідного розміру. Отже, залежно від протоколу *TP*, блоки транспортного рівня називаються сегментами або датаграмами.

Протоколи *TP* визначають спосіб передачі повідомлень між вузлами і забезпечують управління надійністю зв'язку, мають важливу інформацію заголовка, що містить двійкові дані, організовані в кілька полів до кожного блока даних. Саме значення в цих полях дають змогу різним протоколам *TP* виконувати завдання з управління передачею даних. Транспортний рівень має кілька функцій: відстеження окремих сеансів зв'язку, сегментація даних і подальша збірка сегментів, додавання інформації до заголовка, визначення додатків, мультиплексування сеансів зв'язку.

Як відомо, протокол *IP* відповідає тільки за структуру, адресацію і маршрутизацію пакетів, але він не визначає способу доставки або передачі пакетів. А протоколи транспортного рівня визначають спосіб передачі повідомлень між вузлами і відповідають за управління надійністю передачі. На *TP* працюють два протоколи – *TCP* і *UDP*. Різні програми висувають різні вимоги до надійності передачі даних. Отже, *TCP/IP* забезпечує два протоколи транспортного рівня, як показано на *рис. 1*.

Протокол *TCP* вважається надійним і повнофункціональним протоколом *TP*, який забезпечує передачу всіх даних на вузол призначення та надійність доставки, підтримує управління потоком даних. Крім того, *TCP* гарантує надійність управління потоком, використовуючи такі основні операції: відстеження кількості сегментів, відправлених на той чи інший хост тим чи іншим додатком; підтвердження отриманих даних; повторна передача даних після закінчення певного часу очікування; забезпечення послідовності даних, які можуть надійти в неправильному порядку; передача даних зі швидкістю, прийнятною для одержувача.

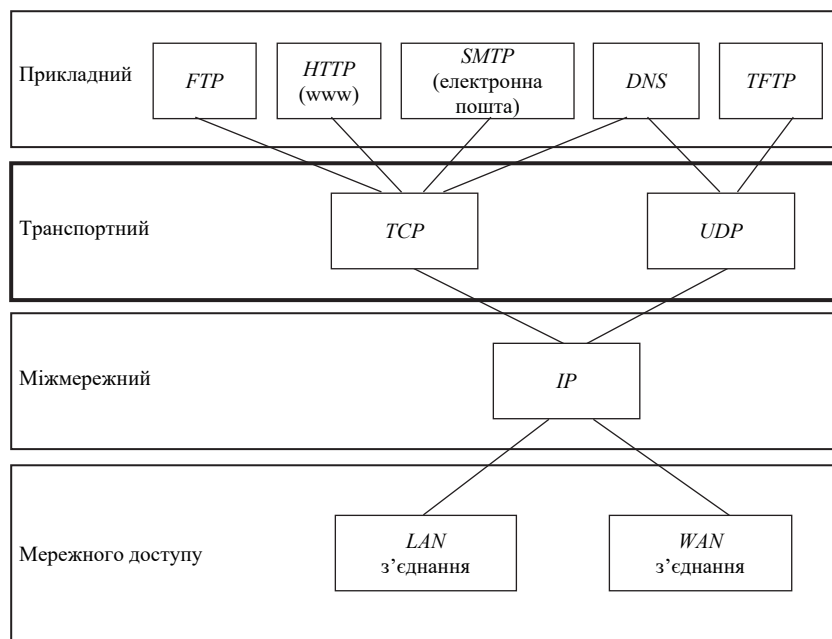


Рис. 1. Протоколи транспортного рівня [13]

Щоб підтримувати стан зв'язку і відстежувати інформацію, *TCP* має спочатку встановити з'єднання між відправником і отримувачем, тому *TCP* є протоколом зі встановленням з'єднання. Для встановлення стану сеансу зв'язку протокол *TCP* фіксує, яку інформацію він відправив і яку інформацію підтверджено. Сеанс зв'язку з контролем стану починається із сеансу обміну даними і припиняється з його завершенням. Фактично *TCP* сам виконує всі завдання, пов'язані з розбивкою потоку даних на сегменти, з надійністю їх передачі й управління потоком. Сегмент *TCP* становить 20 байт (тобто 160 біт). Поле заголовка *TCP* містить порти джерела і призначення, порядковий номер, номер підтвердження, довжину заголовка, керуючі біти і розмір вікна.

Як правило, використовують *TCP* для електронної пошти й Інтернету [3; 4; 10]. Можна виділити такі приклади відомих застосувань, що використовують протокол *TCP*: *HTTP* (протокол передачі гіпертексту), *FTP* (протокол передачі файлів), *SMTP* (протокол простої передачі електронної пошти), *Telnet*.

На рис. 2 наведено поля заголовка *TCP*.

Рис. 2. Поля заголовка *TCP* [13]

Протокол *UDP* є більш простим за своїми функціями протоколом транспортного рівня, ніж *TCP*. Але датаграми *UDP* можуть оброблятися набагато швидше, ніж сегменти *TCP*. Фактично протокол *UDP* забезпечує тільки основні функції для обміну даними між додатками, водночас практично відсутня перевірка даних. *UDP* ділить дані на датаграми (сегменти). Заголовок *UDP* має поля: порт призначення, довжина і контрольна сума. На *рис. 3* показано поля заголовка *UDP*.



Рис. 3. Поля заголовка *UDP* [13]

Програми, що використовують *UDP*, включають *DHCP*, *DNS*, *SNMP*, *TFTP*, *VoIP* і відеоконференції. Вважають, що *UDP* – протокол транспортного рівня без встановлення з’єднання, який не забезпечує надійності управління потоком даних, не відстежує інформації щодо її отримання або надсилання. Роботу протоколу *UDP* можна порівняти з розсилкою рекламних повідомлень. Ніякого встановлення зв’язку та підтвердження отримання повідомлення немає, оскільки листи з рекламною інформацією просто кидають до нашої поштової скриньки. Водночас ні відправника, ні одержувача надійність доставки інформації чи її цілісність особливо не турбують. На *рис. 4* представлено стислий опис відмінностей між протоколами *TCP* і *UDP*.





<i>UDP</i>		<i>TCP</i>	
 <i>VoIP</i> (<i>IP</i> -телефонія)	 <i>DNS</i> (Перетворення доменних імен)	 <i>SMTP / IMAP</i> (Електронна пошта)	 <i>HTTP / HTTPS</i> (<i>www</i>)
Необхідні властивості протоколу: <ul style="list-style-type: none"> • швидкість; • низькі накладні витрати; • не вимагає підтвердження про доставку; • повторно не надсилає втрачених даних; • доправляє дані в порядку надходження 		Необхідні властивості протоколу: <ul style="list-style-type: none"> • надійність; • підтвердження доставки даних; • повторне надсилання втрачених даних; • впорядкована доставка даних 	

Рис. 4. Відмінності між протоколами *TCP* і *UDP* [13]

Порт у *TCP* або *UDP* – це логічний канал з певним номером (від 0 до 65 536), який забезпечує поточну взаємодію між відправником та одержувачем. Порти уможливають обмін даними комп’ютеру з однією *IP*-адресою паралельно з великою кількістю інших комп’ютерів. Деякі номери портів (так звані добре відомі, або *well-known*, порти з номерами від 0 до 1024) прив’язані до певних служб і додатків, ця можливість дає змогу клієнтам без проблем звертатися до потрібних їм мережевих сервісів.

Крім підтримки базових функцій, протокол *TCP* також містить такі можливості:

- встановлює і налаштовує постійне з'єднання (або сеанс) між джерелом і призначенням, при цьому сеансі узгоджується обсяг трафіку, який можна ретельно контролювати;
- якщо з різних причин під час передачі мережею один із сегментів буде пошкоджений або повністю втрачений, *TCP* забезпечить гарантовану доставку на вузол призначення всіх без винятку сегментів даних, відправлених одержувачем;
- управління потоком передачі даних.

Існують три типи додатків, які є оптимальними для роботи з протоколом *UDP*: *по-перше*, мультимедійні програми і передача відео в реальному часі (прикладом можуть слугувати *VoIP* і потокове відео), *по-друге*, прості додатки запитів і відповідей (*DNS* і *DHCP*), *по-третє*, додатки, що забезпечують надійність передачі даних, – ненаправлений обмін даними, за якого управління потоком, виявлення помилок, відправка підтверджень і відновлення після збоїв не потрібні або виконуються самим додатком (наприклад, *SNMP* і *TFTP* включають *DHCP*, *DNS*). Отже, можна виділити основні приклади відомих застосувань, що використовують протокол *UDP*: *DHCP*, *DNS*, *SNMP*, *TFTP*. Оскільки деякі додатки забезпечують надійність доставки самостійно, в такому разі сервіси *TCP* не потрібні, а використання *UDP* як протоколу транспортного рівня буде правильним рішенням. Щодо протоколу *TFTP*, то він має власні способи управління потоком даних, відправкою підтверджень, сам виявляє помилки і виправляє їх, тому не потребує використання *TCP* [13; 14]. На *рис. 5* продемонстровано додатки, для роботи яких потрібні протоколи *TCP* і *UDP*.

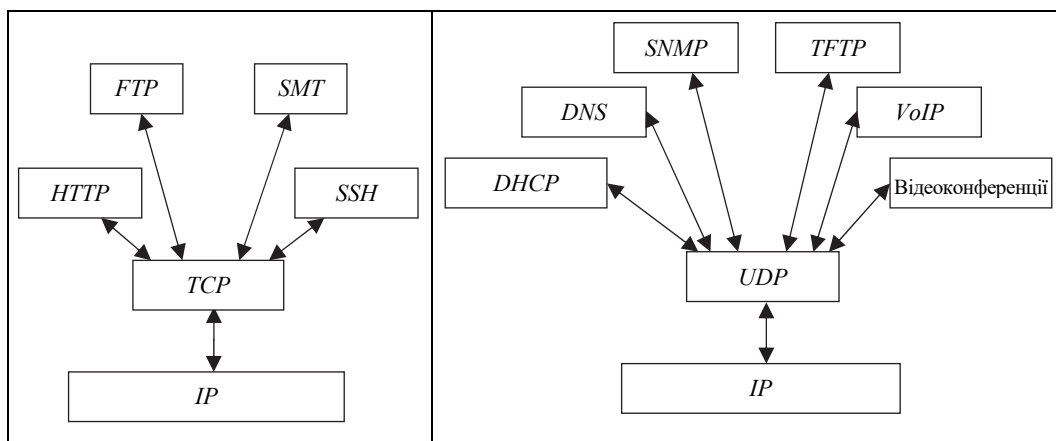


Рис. 5. Додатки, що використовують протоколи *TCP* і *UDP* [13]

TCP є найкращим протоколом для деяких додатків завдяки тому, що, на відміну від *UDP*, він повторно відправляє відкинуті пакети і нумерує пакети, щоб вказати їхній правильний порядок перед доставкою. Аби одержувач зміг розшифрувати вихідне повідомлення, дані в цих сегментах повторно збираються у вихідному порядку.

Для цього у заголовку кожного пакета вказуються порядкові номери. Порядковий номер відповідає порядковому номеру першого байта даних сегмента *TCP*. Під час налаштування сеансу зв'язку задається початковий порядковий номер сеансу (*ISN*), який є стартовим значенням лічильника. У міру передачі даних під час сеансу порядковий номер збільшується на число переданих байтів. Таке відстеження байтів дає змогу однозначно визначати і підтверджувати кожен сегмент. У такий спосіб є можливість з'ясувати, які ж сегменти відсутні. Номер *ISN* не обов'язково має починатися з "1", фактично це будь-яке випадкове число. Це уможливило запобігання шкідливим атакам.

Іноді відбувається втрата даних, у такому разі протокол *TCP* забезпечує можливість для управління втраченими сегментами. Серед них – механізм повторної передачі сегментів з даними, отримання яких не підтверджено. Тобто *TCP* може підтвердити тільки очікуваний наступний байт.

Протокол *UDP* не встановлює з'єднання, але відповідає за передачу даних з меншими накладними витратами, оскільки має невеликий заголовок датаграми і не обмінюється керуючим трафіком. Цей протокол не відстежує порядкових номерів, як це робить протокол *TCP*. Отже, він просто повторно збирає дані в тому порядку, в якому вони прийняті, й пересилає їх з додатком. Якщо послідовність даних важлива для роботи програми, то програма має сама визначити правильну послідовність і вибрати спосіб обробки даних [7].

Щоб забезпечити надійність, протокол *TCP* використовує процес рукостискання, таймери, підтвердження повідомлень і динамічну зміну вікна. Зворотна сторона забезпечення надійності пов'язана з додатковими навантаженнями на мережу, оскільки процес потребує використання великих заголовків сегментів і пересилки великої кількості трафіку між вузлами джерела і призначення. У разі, якщо потрібна швидка доставка даних або якщо пропускна здатність мережі не витримує додаткового навантаження, що пов'язане з надсиланням повідомлень, кращий варіант – застосування протоколу *UDP*, оскільки він не відстежує і не підтверджує отримання датаграм на вузлу призначення, тобто просто передає одержані датаграми на рівень застосувань у міру їх постування [14; 15].

Висновки. Забезпеченням надійності інформаційного зв'язку між двома кінцевими вузлами займається транспортний рівень, а протоколи *TCP/IP* надають транспортні послуги прикладним процесам.

Транспортні послуги цих протоколів мають відмінності. Протокол *UDP* доставляє датаграми без встановлення з'єднання і не гарантує доставки. Протокол *TCP*, навпаки, забезпечує надійну доставку байтових потоків (сегментів).

Для розв'язання транспортних питань протоколи *TCP* та *UDP* під час передачі даних формують і додають до них свої заголовки обсягом 20 та 8 байт відповідно. Кожний прикладний процес взаємодіє з модулем транспортного рівня *TCP* або *UDP* через окремий порт, що

дає змогу однозначно їх ідентифікувати. Фактично функції *UDP* зводяться до операцій мультиплексування і демультиплексування, а також нескладної перевірки наявності помилок у даних. Відтак, при використанні *UDP* додаток майже безпосередньо взаємодіє з протоколом мережевого рівня *IP*.

Протокол *TCP* забезпечує надійну передачу повідомлень внаслідок утворення логічних з'єднань, даючи змогу рівноранговим об'єктам на комп'ютері-відправнику і на комп'ютері-одержувачі підтримувати обмін даними в дуплексному режимі.

Протокол *UDP* забезпечує передачу прикладних пакетів за допомогою датаграм і виконує тільки роль мультиплексора між мережевим протоколом і системами прикладного рівня.

Різниця між протоколами *TCP* і *UDP* – у так званій гарантії доставки. Протокол *TCP* вважається надійним, на відміну від протоколу *UDP*, який отримав назву "протокол ненадійних датаграм", він виключає втрати даних, затримки, дублювання і перемішування пакетів. Своєю чергою, *UDP* все це допускає, водночас з'єднання для роботи йому абсолютно не потрібне. *TCP* контролює також завантаженість з'єднання, *UDP* ж не контролює нічого, крім цілісності отриманих датаграм.

Завдяки своїм функціям протокол надійності *TCP* успішно застосовується в поштових програмах, контролюючи не тільки трафік, але й швидкість обміну трафіком і довжину самого повідомлення.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Нові комп'ютерні засоби, обчислювальні машини та мережі: зб. наук. пр. Київ: НАНУ, Ін-т кіберн. ім. В. М. Глушкова, Наук. рада НАН з проблеми "Кібернетика", 2001. Т. 2. 176 с.
2. Кучернюк П. В. Методи і технології захисту комп'ютерних мереж (мережний, транспортний та прикладний рівні). *Microsystems, Electron & Acoust.* Київ: Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського", 2018. № 23. С. 52-58.
3. Струбицький Р. П. Порівняльний аналіз швидкості роботи протоколів гарантованої передачі даних, базованих на UDP у міжконтинентальній мережі. *Вісн. Хмельн. нац. ун-ту.* 2015. № 2. С. 173-177.
4. Kumar S., Rai S. Survey on Transport Layer Protocols: TCP & UDP. *International Journal of Computer Applications.* 2012. N 46 (7). P. 20-25.
5. Тимченко О. В., Верхола Б. М., Кирик М. І., Самі Аскар. Співвідношення між якістю послуг мереж передачі даних і рівнями OSI: зб. наук. пр. ІПМЕ НАН України. Вип. 49. Київ, 2008. С. 150-157.
6. Thanh V. T., Urano Y. Mobile TCP socket for secure applications. *Advanced Communication Technology (ICACT).* The 12th International Conference. Nangang, China, 2010. P. 971-974.
7. Hong R. L. Research and application of TCP/IP protocol in embedded system. 2011 IEEE 3rd International Conference on Communication Software and Networks. 2011. P. 584-587.

8. Assar A., Hofmann K. A hardware implementation of the TCP protocol applying TCP-BIC and TCP-CUBIC standards. *Microelectronics (ICM). 28th International Conference. Giza, Egypt, 2016. P. 37-40.*
9. Liu Q., Zhiqiang X., Zhengying L. Implementation of hardware TCP/IP stack for DAQ systems with flexible data channel. *Electronics Letters. 2017. Vol. 53. Issue 8. P. 530-532.*
10. Eddy W. M., Ishac J. Location Management in a Transport Layer, *Mobility Architecture. NASA/TM-2005-213844, 2005.*
11. Павликевич М. Телекомунікаційні мережі. Ч. 2. Мережі IP. Львів: Вид-во нац. ун-ту "Львівська політехніка", 2009. 216 с.
12. Якименко І. З. Критерії оцінки рівня захисту комп'ютерних мереж з врахуванням їх архітектури. *Інформатика та математичні методи в моделюванні. 2013. Т. 3. № 1. С. 82-90.*
13. CCNAv7: Introduction to Network. URL: <https://www.netacad.com/ru/courses/networking/ccna-introduction-networks>.
14. Тимченко О. В., Верхола Б. М., Самі Аскар. Критерії ефективності функціонування комп'ютерних мереж. Моделювання та інформаційні технології: зб. наук. пр. ПІМЕ НАН України. Київ, 2007. Вип. 43. С. 184-190.
15. Зайченко О. Ю., Зайченко Ю. П. Комп'ютерні мережі. Київ: Видавничий Дім "Слово", 2010. 520 с.

Стаття надійшла до редакції 12.05.2021.

Kostiuk Yu., Shestack Ya. The transport layer of the ISO/OSI model in computer networks.

Background. The transport layer is designed to deliver data without errors, losses and duplication in the order in which they were transmitted. It provides data transfer between two applications with the required level of reliability. Transport layer protocols, which guarantee reliable data delivery, establish a virtual connection before data exchange and resend segments in case of loss or damage.

The aim of the study was to determine the role of transport security protocols in computer networks.

Materials and methods. To achieve the goal, the study used statistical analysis and a systematic approach.

Results. TCP provides reliable message transmission through the formation of logical connections, while allowing peers on the sending computer and the receiving computer to support data exchange in duplex mode. It also has the ability to seamlessly send a byte stream generated on one of the computers to any other computer connected to the network. In addition, TCP controls the connection load, UDP does not control anything but the integrity of the received datagrams.

Conclusion. The difference between TCP and UDP is the so-called "delivery guarantee". TCP requires a response from the client to whom the data packet is delivered, confirmation of delivery, and for this he needs a pre-established connection. TCP is also considered reliable, unlike UDP, which is called "unreliable datagram protocol". TCP eliminates data loss, duplication and shuffling of packets, delays, UDP allows all this, and it does not need a connection to work., as a result of which the data is transferred on UDP, should manage received, even with losses.

Keywords: protocols, transport layer, data, transmission control, communication session, reliable delivery, datagrams.

REFERENCES

1. *Novi komp'juterni zasoby, obchysljuval'ni mashyny ta merezhi [New computer means, computers and networks]*. (2001). Zbirnyk naukovykh prac'. Vol. 2. Kyi'v: NANU, Instytut kibernetiky imeni V. M. Glushkova, Naukova rada NAN z problemy "Kibernetyka" [in Ukrainian].
2. Kuchernjuk, P. V. (2018). Metody i tehnologii' zahystu komp'juternykh merezh (mereshnyj, transportnyj ta prykladnyj rivni) [Methods and technologies of computer network protection (network, transport and application levels)]. *Microsystems, Electron & Acoust*, 23, 52-58. Kyi'v: Nacional'nyj tehnicnyj universytet Ukrai'ny "Kyi'vs'kyj politehnicnyj instytut imeni Igorja Sikors'kogo" [in Ukrainian].
3. Strubyc'kyj, R. P. (2015). Porivnjal'nyj analiz shvydkosti roboty protokoliv garantovanoi' peredachi danyh, bazovanyh na UDP u mizhkontynental'nij merezhi [Comparative analysis of the speed of UDP-based guaranteed data transmission protocols in the intercontinental network]. *Visnyk Hmel'nyc'kogo nacional'nogo universytetu – Bulletin of Khmelnytsky National University*, 2, 173-177 [in Ukrainian].
4. Kumar, S., & Rai, S. (2012). Survey on Transport Layer Protocols: TCP & UDP. *International Journal of Computer Applications*, 46 (7), 20-25 [in English].
5. Tymchenko, O. V., Verhola, B. M., Kyryk, M. I., & Sami, Askar. (2008). *Spivvidnoshennja mizh yakistju poslug merezh peredachi danyh i rivnjamy OSI [The correlation between the quality of data network services and OSI levels]*. Zbirnyk naukovykh prac' IPME NAN Ukrai'ny. Issue 49, 150-157. Kyi'v [in Ukrainian].
6. Thanh, V. T., & Urano, Y. (2010). *Mobile TCP socket for secure applications. Advanced Communication Technology (ICACT)*. The 12th International Conference. (pp. 971-974). Nangang, China [in English].
7. Hong, R. L. (2011). *Research and application of TCP/IP protocol in embedded system*. 2011 IEEE 3rd International Conference on Communication Software and Networks. (pp. 584-587) [in English].
8. Assar, A., & Hofmann, K. (2016). *A hardware implementation of the TCP protocol applying TCP-BIC and TCP-CUBIC standards. Microelectronics (ICM)*. 28th International Conference. (pp. 37-40). Giza, Egypt [in English].
9. Liu, Q., Zhiqiang, X., & Zhengying, L. (2017). Implementation of hardware TCP/IP stack for DAQ systems with flexible data channel. *Electronics Letters*. Vol. 53, Issue 8, 530-532 [in English].
10. Eddy, W. M., & Ishac, J. (2005). *Location Management in a Transport Layer, Mobility Architecture*. NASA/TM-2005-213844 [in English].
11. Pavlykevych, M. (2009). *Telekomunikacijni merezhi. Ch. 2. Merezhi IR [Telecommunication networks. Part 2. IP networks]*. L'viv: Vydavnyctvo nacional'nogo universytetu "L'vivs'ka politehnika" [in Ukrainian].
12. Jakyenko, I. Z. (2013). Kryterii' ocinky rivnja zahystu komp'juternykh merezh z vrakhuvannjam i'h arhitektury [Criteria for assessing the level of protection of computer networks, taking into account their architecture]. *Informatyka ta matematychni metody v modeljuvanni – Informatics and mathematical methods in modeling*. Vol. 3, 1, 82-90 [in Ukrainian].
13. CCNAv7: Introduction to Network. *Networking Academy*. Retrieved from <https://www.netacad.com/ru/courses/networking/ccna-introduction-networks> [in English].
14. Tymchenko, O. V., Verhola, B. M., & Sami, Askar. (2007). *Kryterii' efektyvnosti funkcionuvannja komp'juternykh merezh. Modeljuvannja ta informacijni tehnologii' [Criteria for the effectiveness of computer networks. Modeling and information technologies]*. Zbirnyk naukovykh prac' IPME NAN Ukrai'ny. Issue 43, 184-190. Kyi'v [in Ukrainian].
15. Zajchenko, O. Ju., & Zajchenko, Ju. P. (2010). *Komp'juterni merezhi [Computer networks]*. Kyi'v: Vydavnychyj dim "Slovo" [in Ukrainian].