

УДК 621.391

К.Д. ГУЛЯЄВ

*Інститут телекомунікацій і глобального інформаційного простору НАНУ, Україна*

## РЕАЛІЗАЦІЯ ПРИКЛАДНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ, ЩО ВИКОРИСТОВУЄ ТЕЛЕКОМУНІКАЦІЙНУ ТЕХНОЛОГІЮ ІЗ ЗМІННИМ РОЗМІРОМ МЕРЕЖНОЇ АДРЕСИ

*Проведено аналіз можливих варіантів реалізації телекомунікаційної технології із змінним розміром мережної адреси (EX) та визначено доцільність її реалізації у вигляді модулю ядра операційної системи сімейства Linux, як протоколу мережного рівня. Представлено архітектурні моделі взаємодії елементів операційної системи із прикладними процесами для стеку протоколів TCP/IP та телекомунікаційної технології із змінним розміром мережної адреси. Розглянуто принцип роботи бібліотеки-завантажувача, що дозволяє використовувати раніше створене програмне забезпечення в новому середовищі.*

**Ключеві слова:** мережна технологія EX, архітектурна модель, бібліотека-завантажувач, програмна реалізація.

Важливим етапом впровадження будь-якої мережної технології є визначення базових принципів реалізації програмного забезпечення, що використовує її переваги. Впровадження запропонованої колективом авторів мережної технології EX, в основу якої покладено принцип використання службових заголовків змінного розміру в яких для адресації вузлів відведено змінну кількість байт [1], також передбачає розробку таких принципів.

На попередніх етапах дослідження було визначено ключові особливості мережної технології EX та розроблено методику оцінки ефективності її застосування для різних випадків [2]. Однак, важливим кроком на шляху подальшого впровадження технології має стати така архітектура програмної реалізації, яка б дозволила використовувати вже існуюче програмне забезпечення, що працює в межах стеку протоколів TCP/IP на новій платформі.

Метою статті є розробка архітектурних моделей взаємодії елементів операційної системи, бібліотеки-завантажувача та прикладних процесів з метою забезпечення функціонування раніше створеного програмного забезпечення в середовищі мережної технології EX.

Серед основних варіантів реалізації будь-якої мережної технології, що заміщує собою протоколи транспортного та мережного рівнів моделі взаємодії відкритих систем (OSI), слід виділити три основних: окремий програмний-шлюз для перетворення класичного TCP/IP навантаження до альтернативного; реалізація протоколу окремим прикладним процесом у просторі користувача; реалізація у модулі ядра, як протоколу третього рівня OSI.

Очевидною перевагою першого з зазначених варіантів є прозорість перетворення для кінцевої операційної системи та повна незалежність від її типу. В свою чергу такий спосіб має два очевидних недоліки. Насамперед, це неможливість створення програмного інтерфейсу для розробки програмного забезпечення, що працює виключно із використанням альтернативної технології. По-друге, це неможливість безпосереднього з'єднання вузлів (для цього необхідно щонайменш два шлюзи).

Реалізація технології окремим прикладним процесом у просторі користувача передбачає створення програмного інтерфейсу для розробки програмного забезпечення, що використовує таку технологію у вигляді окремої бібліотеки. При цьому сумісність із наявним програмним забезпеченням може забезпечуватися, наприклад, локальним проксі-сервером. Незважаючи на очевидну простоту реалізації (що є певною перевагою), такий спосіб значно зменшує швидкодію, не дозволяє створити рішення для декількох операційних систем одночасно та створює додаткові недоліки пов'язані із необхідністю забезпечення сумісності із процесами-посередниками.

Реалізація мережної технології, як протоколу третього рівня OSI, передбачає розробку модуля ядра, що динамічно завантажується та реєструє протокол у мережевому стеку операційної системи. При цьому програмний інтерфейс може забезпечуватись підтримкою абстракції сокетів Берклі [3], а сумісність – окремою бібліотекою, що перехоплює системні виклики сокетів AF\_INET та відображає їх на сокети AF\_EX. Перевагами такого способу є вико-

ристання класичного програмного інтерфейсу (сокетів), повна сумісність з існуючим програмним забезпеченням та швидкодія за рахунок перенесення коду до ядра системи. Очевидним недоліком варіанту є обмеження на використання модулю в інших операційних системах.

Проведений аналіз показав, що серед усіх розглянутих варіантів найбільш доцільним буде реалізація модулю ядра, що динамічно завантажується. Це дозволить забезпечити сумісність із існуючим програмним забезпеченням шляхом створення бібліотеки-завантажувача, що буде перехоплювати системні виклики до сокетів.

У якості платформи розробки було обрано дистрибутив Debian [4] операційної системи Linux. Це пояснюється тим, що мережний стек цієї системи добре документовано, у наявності є вихідний код системи та є підтримка динамічного розширення мережевого стеку.

Модуль працюватиме на всіх операційних системах сімейства Linux. Також обраний варіант реалізації дозволяє забезпечити за потреби той самий функціонал також для інших операційних систем у майбутньому.

На рис. 1 зображено схему проходження пакету крізь функції мережевого стеку для класичного стеку TCP/IP (рис. 1, а) та для технології EX (рис. 1, б).

Розглянемо детально шлях проходження пакету від програмного переривання до процесу користувача (рис. 1). Процедура обробки програмного переривання `net_rx_action()` у безперервному циклі вилучає пакети із вхідної черги мережного адаптеру та передає їх до відповідних протоколів верхнього рівня. Спочатку буфер передається до протоколів зареєстрованих у списку `rture_all`, згодом, до усіх протоколів, що зареєстровано у списку `rture_base` із відповідним значенням `eth_ture`. Наприклад: при значенні `eth_ture==0x0800` буфер буде передано до процедури `ip_rcv()`.

У випадку використання стеку TCP/IP (рис. 1, а) пакет надходить до процедури `ip_rcv()`, що реалізує функції екземпляру протоколу IP та передає буфер далі. Далі пакет надходить до процедури `tcp_v4_rcv()`, яка обробляє буфер згідно протоколу TCP, додає корисні дані до вхідної черги відповідного сокету та інформує його про наявність та кількість нових даних у черзі. В свою чергу функція сокету `recvmsg()` протоколу `tcp` здобуває дані з вхідної черги, копіює їх до простору користувача та передає керування процесу користувача.

У випадку використання технології EX (рис. 1, б) пакет надходить до процедури `ex_rcv()`. У залежності від наявності сокету та типу вхідного пакету вона передає його до функцій `ex_rcv_state()`, `ex_defrag()` та `ex_rcv_established()` відповідно.

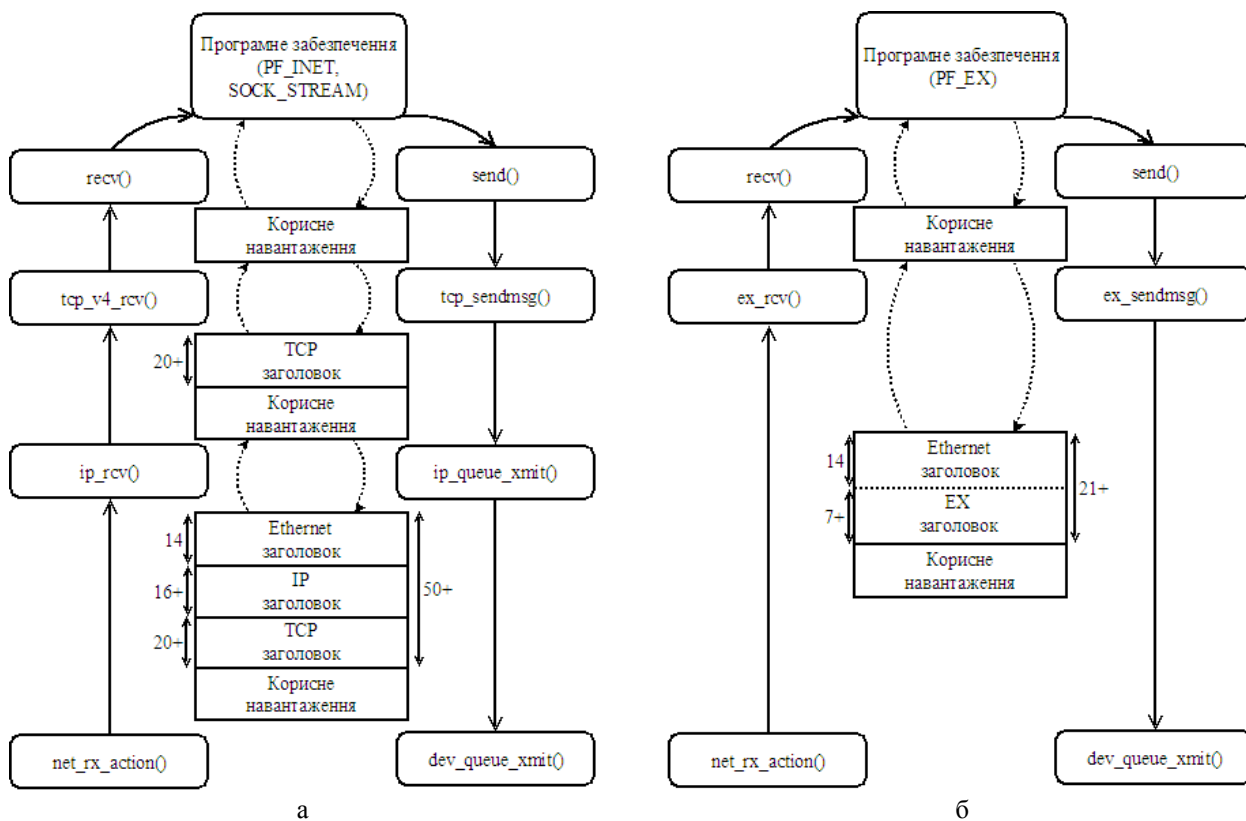


Рис. 1. Схема проходження пакету крізь функції мережевого стеку:  
а – для стеку TCP/IP; б – для технології EX

В свою чергу функція `ex_rcv_established()` додає пакет до вхідної черги відповідного сокету та інформує систему про наявність нових даних у черзі. Функція сокету `recvmsg()` технології EX здобуває дані з вхідної черги, копіює їх до простору користувача та передає керування процесу користувача.

В свою чергу шлях від процесу користувача до мережного адаптера виглядає таким чином. Процес користувача викликає бібліотечні функції `send()`, `sendto()`, `sendfile()` тощо, що в свою чергу викликає функцію `sendmsg()` відповідну до відкритого типу сокету. У разі використання стеку TCP/IP (рис. 1, а) дані надходять до процедури `tcp_sendmsg()`, яка формує відповідний tcp-заголовок, та передає буфер сокету до протоколу рівнем нижче. В свою чергу буфер сокету надходить до процедури `ip_queue_xmit()`, що формує одну або кілька відповідних IP-дейтаграм, додає заголовок мережного рівня та передає до мережного адаптера за допомогою функції `dev_queue_xmit()`.

У випадку використання технології EX (рис.1, б) дані надходять до процедури `ex_sendmsg()`. Вона формує відповідні EX-дейтаграми, додає до них заголовки мережного рівня та передає до мережного адаптера за допомогою функції `dev_queue_xmit()`.

На рис. 2 зображено принцип роботи бібліотеки завантажувача. Зазначена бібліотека підміняє бібліотечні виклики `libc` або `libconnect` обраних процесів своїми, що є цілком прозорим для процесу та не створює додаткового навантаження на операційну систему.

У разі відкриття TCP або UDP сокету процесом, бібліотека заміщує тип сокету (на EX) та викликає оригінальну бібліотечну функцію.

У подальшому, якщо викликається функція, вхідними або вихідним параметрами якої є IP-адреси, бібліотека реалізує трансформацію адреси у потрібний бік, та передає її до сокету EX або до процесу користувача.

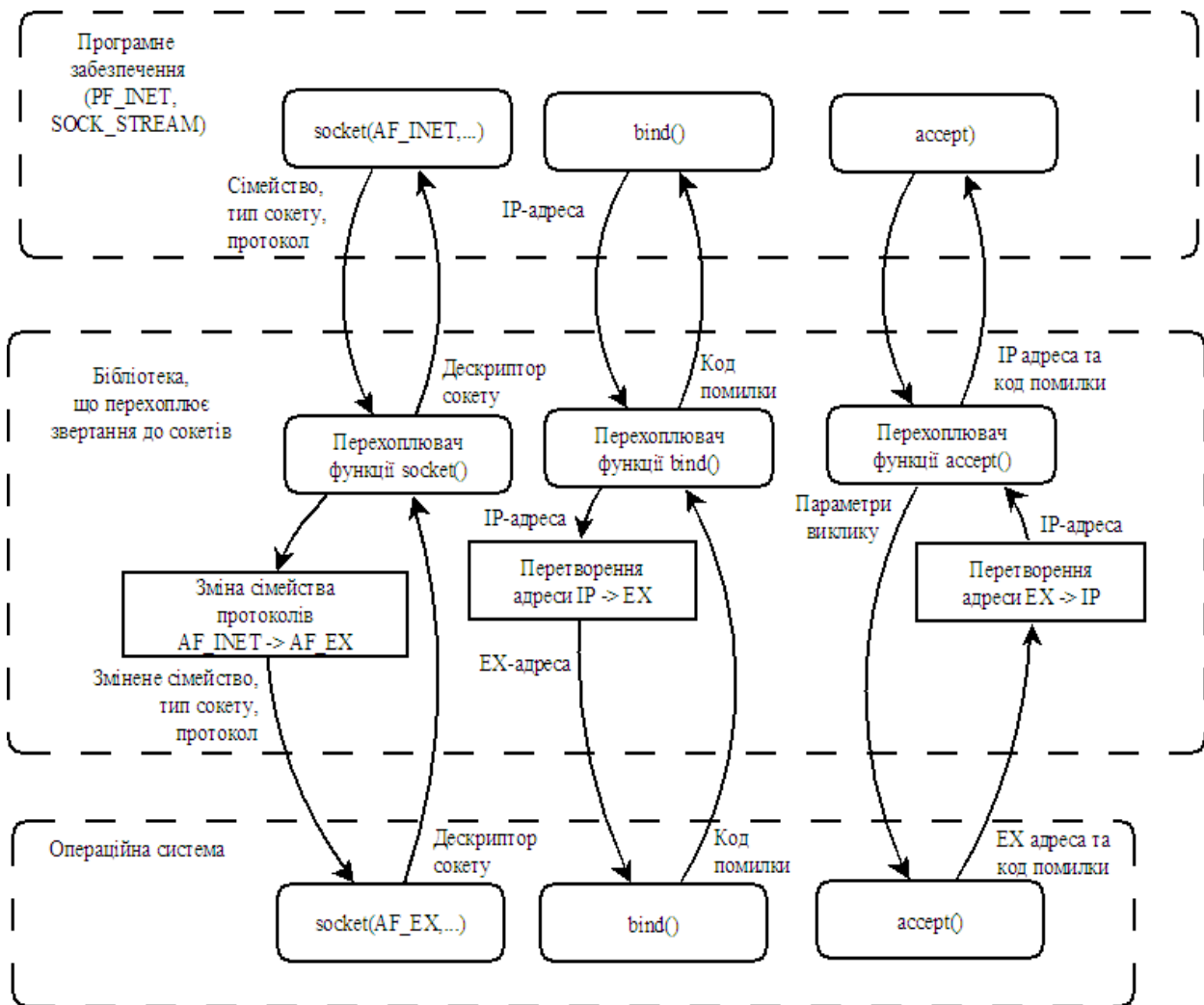


Рис. 2. Принцип роботи бібліотеки-завантажувача

## Висновки та результати

1. Важливим кроком на шляху подальшого впровадження нової телекомунікаційної технології EX має стати її програмна реалізація, яка б дозволила використовувати вже існуюче програмне забезпечення, що працює в межах стеку протоколів TCP/IP.

2. Проведений в роботі аналіз показав, що серед усіх розглянутих варіантів реалізації найбільш доцільним є розробка модулю ядра операційної системи та створення бібліотеки-завантажувача, що буде перехоплювати системні виклики до сокетів.

3. Отриманий в результаті механізм є надбудовою над стандартним програмним інтерфейсом та забезпечує прозоре перетворення між протоколами для процесів користувача. Зокрема внутрішня архітектура мережної технології EX розроблена таким чином, щоб модулі реалізації не були пов'язані із реалізацією стеку TCP/IP та IP-адресацією. Трансформація зазначених протоколів для сумісності реалізується цілковито бібліотекою.

## Література

1. Каптур, В.А. Базові принципи практичної реалізації систем адресації із змінним розміром мережної адреси в Ethernet мережах [Текст] / В.А. Каптур, К.Д. Гуляєв, П.С. Кравченко // *Радіоелектронні і комп'ютерні системи*. – 2012. – №1. – С. 51 – 54.

2. Оцінювання ефективності впровадження телекомунікаційних технологій зменшення протокольної надлишковості [Текст] / В.А. Каптур, К.Д. Гуляєв, П.С. Кравченко, О.О. Яніна // *Открытые информационные и компьютерные интегрированные технологии: сб. науч. тр. Нац. аэрокосм. ун-та им. Н.Е. Жуковского «ХАИ»*. – Вып. 52. – Х., 2011. – С. 77 – 89.

3. *Linux: сетевая архитектура. Структура и реализация сетевых протоколов в ядре* [Текст] / К. Вейрле, Ф. Пэльке, Х. Риттер, Д. Мюллер и др. – КУДИЦ-Образ, 2006. – 656 с.

4. *Debian – Универсальная операционная система* [Электронный ресурс]. – Режим доступа: <http://www.debian.org>. – 7.11.2012 г.

Поступила в редакцію 7.11.2012

**Рецензент:** д-р техн. наук, професор, ректор П.П. Воробієнко, Одеська національна академія зв'язку ім. О.С. Попова, Одеса.

### РЕАЛИЗАЦИЯ ПРИКЛАДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИСПОЛЬЗУЮЩЕГО СЕТЕВУЮ ТЕЛЕКОММУНИКАЦИОННУЮ ТЕХНОЛОГИЮ ПЕРЕМЕННЫМ РАЗМЕРОМ СЕТЕВОГО АДРЕСА

*К.Д. Гуляєв*

Проведен анализ возможных вариантов реализации телекоммуникационной технологии с переменным размером сетевого адреса (EX) и определена целесообразность реализации в виде модуля ядра операционной системы семейства Linux, как протокола сетевого уровня. Представлены архитектурные модели взаимодействия элементов операционной системы с прикладными процессами для стека протоколов TCP/IP и телекоммуникационной технологии с переменным размером сетевого адреса. Рассмотрен принцип работы библиотеки-загрузчика, позволяющий использовать ранее созданное программное обеспечение в новой среде.

**Ключевые слова:** сетевая технология EX, архитектурная модель, библиотека-загрузчик, программная реализация.

### SOFTWARE IMPLEMENTATION OF TELECOMMUNICATION TECHNOLOGY WITH VARIABLE SIZE OF NETWORK ADDRESS

*K.D. Gulyayev*

The analysis of the possible implementation of telecommunication technology with variable size of network address (EX) and determine the feasibility of implementation as a kernel module of Linux operating system (as protocol of network layer). Presented the architectural model for the interaction of the operating system elements with the application process for the TCP/IP and telecommunication technology with variable size of network address. The principle of operation of the library-loader allows use of previously developed software in the new environment.

**Key words:** EX network technology, architectural model, library-loader, software implementation.

**Гуляєв Кирило Дмитрович** – канд. техн. наук, в.о. завідуючого відділу інформаційних та інноваційних технологій в освіті, Інститут телекомунікацій і глобального інформаційного простору НАН України, e-mail: kirill@gulyayev.com.ua.