МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

До захисту допускається
Завідувач кафедри
_____ Скарга-Бандурова І.С.
«____»_____ 20__ р.

МАГІСТЕРСЬКА РОБОТА

НА ТЕМУ:

Автоматне моделювання мережевих протоколів

Освітньо-кваліфікаційний рівень "Магістр"
Спеціальність 123 "Комп'ютерна інженерія" (освітня програма - "Комп'ютерні системи і мережі")"

Науковий керівник роботи: _____ (підпис) О.І.Рязанцев (ініціали, прізвище)

Консультант з охорони праці: _____ (підпис) Я.О.Критська (ініціали, прізвище)

Студент: _____ (підпис) І.С. Зінченко (ініціали, прізвище)

Група: КСМ-16дм

Сєвєродонецьк 2018

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки
Кафедра Комп'ютерних наук та інженерії
Освітньо-кваліфікаційний рівень магістр
Напрям підготовки
(шифр і назва)
Спеціальність 123 "Комп'ютерна інженерія" (освітня програма - "Комп'ютерні системи і
(шифр і назва)
мережі")

ЗАТВЕРДЖУЮ:

Завідувач кафедри
І.С. Скарга-Бандурова
«_____» _____20____р.

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

Зінченку Ігорю Сергійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Автоматне моделювання мережевих протоколів

керівник проекту (роботи) Рязанцев Олександр Іванович, д.т.н., проф.
(прізвище, м.'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом вищого навчального закладу від «18» 10 2018 р. № 208/48
2. Строк подання студентом роботи 21.01.2018
3. Вихідні дані до роботи Матеріали науково-дослідної практики, математичний апарат – булева алгебра, об'єкт тестування - мережеві протоколи, тип автоматної моделі – Милі, тестування - конформність

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Аналіз методів моделювання та тестування мережевих протоколів, розробка автоматної моделі мережевих протоколів, автоматне моделювання мережевих протоколів, аналітична форма продання протоколу TCP, охорона праці та безпека в надзвичайних ситуаціях, висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Електронні плакати

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці та безпека в надзвичайних ситуаціях	Критська Я.О. ст. викл. кафедри КНІ		

7. Дата видачі завдання 18.10.2017

Керівник _____
(підпис)

Завдання прийняв до виконання _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Аналіз методів моделювання і тестування мережевих протоколів	10.09.2017-15.09.2017	
2	Розробка технічного завдання	16.09.2017-22.09.2017	
3	Розробка автоматної моделі мережевих протоколів	23.09.2017-25.09.2017	
4	Автоматне моделювання мережевих протоколів	26.09.2017-06.10.2017	
5	Аналітична форма подання протоколу TCP	07.10.2017-25.10.2017	
6	Розробка частини проекту "Охорона праці та безпеки в надзвичайних ситуаціях"	26.10.2017-13.11.2007	
7	Оформлення пояснювальної записки та презентації	14.11.2017-30.11.2017	
8	Оформлення автореферату	01.12.2017-31.12.2017	

Студент _____ Зінченко І.С.
(підпис) (прізвище та ініціали)

Науковий керівник _____ Рязанцев О.І.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Зінченко І. С. Автоматне моделювання мережевих протоколів.

Метою дипломної роботи є розробка методу тестування мережевих протоколів на основі теорії булевих похідних. Розробка і аналіз методів моделювання мережевих протоколів для забезпечення перевірки конформності розроблюваних протоколів заданої специфікації на етапі створення протоколу з метою підвищення якості обслуговування та достовірності передачі інформації в комп'ютерних мережах, ефективності проектування, створення, і функціонування мережевих протоколів.

Результатом роботи став алгоритм тестування мережевих протоколів, в даному випадку TCP протоколу.

Ключові слова: тестування, діагностика, протокол, алгоритм, булеві похідни.

АННОТАЦИЯ

Зинченко И.С. Автматное моделирование сетевых протоколов.

Целью дипломной работы является разработка метода тестирования сетевых протоколов на основе теории булевых производных. Разработка и анализ методов моделирования сетевых протоколов для обеспечения проверки конформности разрабатываемых протоколов заданной спецификации на этапе создания протокола с целью повышения качества обслуживания и достоверности передачи информации в компьютерных сетях, эффективности проектирования, создания, и функционирования сетевых протоколов.

Результатом работы стал алгоритм тестирования сетевых протоколов, в данном случае TCP протокола.

Ключевые слова: тестирование, диагностика, протокол, алгоритм, булеви производные

THE ABSTRACT

Zinchenko I.S. Automated simulation of network protocols.

The aim of the thesis is to develop a method for testing network protocols using Boolean derivative. Development and analysis of methods for modeling network protocols to ensure the verification of conformity given specification developed protocols on the stage of the protocol in order to increase quality of service and reliability of information transmission in computer networks, the effectiveness of the design, creation, and operation of the network protocols.

The work will be ready network protocol testing algorithm, in this case, the TCP protocol.

Key words: testing, diagnostic, protokols, algorithm, boolean derivative.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

TCP - Transmission Control Protocol - протокол управління передачею

OSI - Open Systems Interconnection - взаємодія відкритих систем

ISO - International Organization for Standardization - міжнародна організація по стандартизації

ГПС - графік послідовності повідомлень

ГГПС - ГіперГПС

FSM - Finite-State Machine - кінцевий автомат

МП - мережу Петрі

ЖЦ ПО - життєвий цикл програмного забезпечення

TTCN - Testing and Test Control Notation - нотація тестування і управління тестами

ЛТК - лабораторія тестування конформности

HDL - Hardware Description Language - мова опису апаратури

ВСТУП

Сьогодні гостро стоїть завдання створення мережевих протоколів нового покоління. Помилки проектування, що призводять до порушення конформности- невідповідності протоколу його задекларованої специфікації -подовжують і здорожують процес створення протоколів. Особливо гостро ця проблема стоїть в зв'язку з великим обсягом тиражування мережевих протоколів. Крім того, часто виникає необхідність в діагностуванні вже створених і використовуються протоколів, помилки у функціонуванні яких знижують якість обслуговування і достовірність передачі інформації [1].

Одним з широко використовуваних способів, що дозволяють діагностувати стан протоколів є використання експертних оцінок, проте він має суттєвий недолік - вимагає наявності великої кількості висококваліфікованих фахівців в області мережевих технологій, що далеко не завжди можливо на робочих майданчиках користувачів Інтернету [2].

Також в діагностуванні протоколів використовується підхід з використанням апарата мереж Петрі. Недоліки застосування мереж Петрі в якості мови програмування укладені в процесі їх виконання в обчислювальній системі. У мережах Петрі немає строго поняття процесу, який можна було б виконувати на зазначеному процесорі. Немає також однозначної послідовності виконання мережі Петрі, так як вихідна теорія представляє нам мову для опису паралельних процесів. Також недоліком методу є відсутність використання принципів об'єктно-орієнтованого підходу.

Найбільш перспективним є автоматний підхід до вирішення задачі моделювання мережевих протоколів. Однак зараз цей підхід використовується в основному на етапі експлуатації протоколів, а не при їх створенні та реалізації. До того ж автори обмежують можливості автоматного моделювання зводячи його лише до подання протоколу у вигляді класичного автомата Мілі, не використовуючи в повній мірі теорію синтезу мікропрограмних автоматів [4, 5].

Вищесказане наочно свідчить про необхідність подальшого наукового дослідження в області моделювання і діагностування мережевих протоколів [6].

В умовах стрімкого розвитку web-простору і збільшення кількості вузлів користувачів в тому числі і мобільних, необхідності оптимальної агрегації мережевих адрес і захисту інформації, що передається з'являються нові проблеми, пов'язані з необхідністю переходу на стек протоколів нового покоління -TCP / IPv6. Зокрема, це- проблема тестування і верифікації новостворюваних мережевих протоколів, а разом з цим і необхідність

моделювання протоколів з метою перевірки їх конформності задекларованої специфікації і формування відповідних тестових послідовностей в системі діагностування протоколів [1].

При необхідності формування тестових послідовностей для діагностування протоколів можуть використовуватися методи експертних оцінок. Однак даний підхід може використовуватися лише на кінцевому етапі життєвого циклу протоколу, тобто На етапі його експлуатації, що у багато обмежує сферу застосування даного методу. До того ж складність підготовки експертної інформації унеможливлює її широке застосування на робочих майданчиках користувачів Інтернету [2, 7].

Моделювання мережевих протоколів дозволяє уявити неформализованное опис функціонування протоколів (специфікацій) у вигляді формальної моделі, придатної для комп'ютерної обробки [39-42]. У ряді робіт аналізується використання апарату моделей Петрі для вирішення цього завдання [3, 8-10].

Є ряд робіт, присвячених вирішенню проблем моделювання протоколів [4, 5, 12-14]. Роботи присвячені в основному використання отриманих моделей в системах діагностування для побудови тестів перевірки протоколів, які вже перебувають в експлуатації. Таке застосування дозволяє вирішувати досить обмежене коло завдань, що не включають рішення проблем поліпшення якості та надійності створюваних протоколів на початкових етапах проектування протоколів.

Незважаючи на істотні досягнення в області моделювання і діагностування мережевих протоколів, залишається ряд завдань, які ще далекі від свого остаточного вирішення. До таких завдань належить задача моделювання мережевих протоколів на всьому протязі їх життєвого циклу з метою використання в системах діагностування мережевих протоколів [15, 16].

У даній роботі введено поняття автоматної моделі мережевих протоколів; запропонований метод побудови автоматних моделей на основі класичної теорії синтезу мікропрограмних автоматів; проаналізована і модифікована структура життєвого циклу типового мережевого протоколу і показана доцільність застосування запропонованого методу моделювання в якості етапу життєвого циклу; показана можливість застосування розроблених моделей для перевірки конформності протоколів на прикладі транспортного TCP-протоколу; надано рекомендації щодо використання запропонованого підходу в системах діагностування мережевих протоколів на початкових етапах життєвого циклу протоколу.

Метою роботи є підвищення якості обслуговування і достовірності передачі інформації в комп'ютерних мережах, ефективності проектування, створення, і

функціонування мережевих протоколів шляхом використання автоматного методу моделювання протоколів на ранніх етапах їх життєвого циклу.

Досягнення поставленої мети здійснюється **вирішенням** наступних основних завдань:

– аналіз методів моделювання протоколів і розробка автоматного підходу до моделювання мережевих протоколів з використанням класичної теорії синтезу мікропрограмних автоматів;

– аналіз структури життєвого циклу протоколу і його доопрацювання для використання еталонної моделі мережевого протоколу на ранніх етапах циклу з метою підвищення ефективності проектування протоколів;

– аналіз і розробка принципів перевірки конформності створюваних мережевих протоколів задекларованої специфікації з використанням для цієї мети розробленої автоматної моделі для верифікації протоколів;

– аналіз і розробка принципів побудови системи діагностування мережевих протоколів з використанням еталонної моделі, отриманої на ранніх етапах проектування протоколів;

– дослідження можливості застосування розробленого підходу до моделювання протоколів для вирішення прикладних завдань діагностування протоколів на прикладі мережевого протоколу транспортного рівня TCPv4;

– аналіз методів побудови тестів для перевірки правильності функціонування мережевих протоколів в сучасних комп'ютерних мережах;

– розробка структури та функцій окремих модулів системи генерації тестів мережевих протоколів для вирішення прикладних завдань тестування протоколів з метою підвищення ефективності та надійності функціонування сучасних розподілених обчислювальних мереж;

– практична реалізація розроблених методів автоматного моделювання мережевих протоколів.

Об'єктом дослідження є процес проектування, створення, і функціонування мережевих протоколів з урахуванням особливостей окремих етапів життєвого циклу протоколу. Процес створення протоколу досить тривалий. Його результати кардинально впливають на якість функціонування мережевих систем, оскільки раз створений протокол широко тиражується і використовується у великій кількості мережевих вузлів і пристроїв.

Предметом дослідження є методи моделювання мережевих протоколів з урахуванням адекватності моделі та ефективності її подальшого використання в системах діагностування протоколів.

Основними методами дослідження є методи комп'ютерного моделювання, методи синтезу мікропрограмних кінцевих автоматів за допомогою класичної моделі автомата Мілі, булева алгебра і булеві похідні, натурні експерименти на реальних мережевих протоколах.

Метод моделювання мережевих протоколів з використанням автоматної моделі Мілі на базі класичної теорії кінцевих автоматів дає можливість виявляти і часто ідентифікувати помилки в програмній реалізації як нових протоколів, так і вже існуючих.

Побудова аналітичного виразу для опису перехідних процесів в автоматної моделі протоколу дає можливість тестової послідовності для перевірки працездатності мережевих протоколів.

Публікації. Основні результати магістерської роботи доповідались на на Всеукраїнській науково-практичній конференції «Електронні апарати та системи. Проблеми створення. Перспективи розвитку».

Структура та обсяг роботи. Магістерська робота складається зі вступу, 5 розділів, висновків, переліку джерел посилань, додатку. Загальний обсяг становить 100 сторінок, 7 таблиць, 34 рисунки.

1 АНАЛІЗ МЕТОДІВ МОДЕЛЮВАННЯ ТА ТЕСТУВАННЯ МЕРЕЖЕВИХ ПРОТОКОЛІВ

1.1 Сучасний стан та тенденції розвитку систем тестування мережевих протоколів

Інтенсивна розробка мережевих протоколів нового покоління останнім часом призводить до гострої необхідності тестування нових (що розробляються) протоколів як на відповідність технічним специфікаціям, так і на можливість їх коректної взаємодії між собою і з раніше розробленими і використовуються в даний час протоколами. Вирішення цих питань набуває особливої актуальності у зв'язку з переходом на нове покоління мережних протоколів стека версії TCP / IP6 [1, 17, 18].

В літературі [2-5, 7, 8, 10-14] ми бачимо, що існують різні шляхи вирішення цього завдання:

– підготовка тестів експертами - фахівцями в області мережевих технологій;

– розробка ручного методу побудови тестів;

– створення спеціалізованої автоматизованої системи генерації тестових послідовностей.

В [19-21] розглядаються принципи подання та обробки знань в сучасних системах штучного інтелекту та підтримка прийняття рішень. Більш детально питання розробки експертних систем і програмування показані в [22].

Слід зазначити, що в літературі експертні системи розглядаються спільно з базами знань як моделі поведінки експертів в певній галузі знань з використанням процедур логічного висновку і прийняття рішень, А бази знань - як сукупність фактів і правил логічного висновку в обраній предметній області діяльності [2, 22].

В [22] також наголошується, що для того, щоб забезпечити стислі терміни виведення програмних рішень на ринок, необхідно проводити тестування на якомога більш ранній стадії життєвого циклу розробки програмного забезпечення. Це ж цілком справедливо і для розробки нових мережевих протоколів. Тобто, завдання тестування мережевих протоколів на стадії їх розробки є актуальною. Її сенс полягає в необхідності автоматизації процесу перевірки практичної реалізації протоколу його задекларованої специфікації конформності.

Така можливість з'являється при проведенні експертного тестування. Засноване на ряді дій, які націлені на управління якістю інформаційної системи (в нашому випадку, інформаційної розподіленої мережі), а також її функціональними можливостями і ресурсами,

воно може проводитися в тому числі на самих ранніх стадіях розробки програмного продукту.

Проведення експертного тестування дозволяє виявляти дефекти, гарантуючи при цьому відповідність функціональних можливостей системи заявленим вимогам. У той же час така процедура дозволяє скорочувати витрати на вдосконалення інформаційних систем, а також проводити аналіз потенційних загроз, пов'язаних з виробництвом продукту недостатню якість.

Відзначимо, що підхід розглянутий у літературі, наприклад, в [2-5] був розроблений для тестування програмних продуктів, що певною мірою є досить вузьким завданням з точки зору його застосування для тестування мережевих протоколів оскільки останні являють собою складну систему взаємодії фізичних вузлів в розподіленої інформаційної системи та спеціального програмного забезпечення, зокрема, операційних систем. У зв'язку з цим потрібно розробити підхід, що враховує всі особливості функціонування таких систем.

Тестування програмного забезпечення враховує особливості технології і передбачає проведення наступних заходів [7]:

– перевірка працездатності SOAсистем. SOA (Service-oriented Architecture) Це - підхід до розробки програмного забезпечення на основі слабосвязанних компонентів, що взаємодіють за допомогою стандартизованих інтерфейсів;

– тестування клієнт-серверних додатків;

– проведення тестування продуктів без графічного інтерфейсу користувача;

– перевірка систем реального часу;

– перевірка текстових і графічних редакторів, експертних систем;

– тестування веб-додатків.

Проведення експертного тестування дозволяє здійснити оптимізацію витрат на досягнення необхідного рівня якості продукту, що розробляється, а також усунення виявлених дефектів інформаційної системи вже на ранніх етапах її розробки. Саме ця особливість експертного тестування становить інтерес для проектувальників і розробників мережевих протоколів.

Інша можливість полягає в підготовці тестів експертами - професіоналами в області мережевих технологій. Цю можливість ми не будемо розглядати оскільки дане рішення є приватним і не дозволяє широко його використовувати в повсюдної практиці проектування мережевих протоколів так як, перш за все, воно вимагає наявності висококваліфікованих і досвідчених експертів, які повинні тісно співпрацювати з розробниками протоколів, що накладає досить серйозні обмеження на застосування даного методу на практиці [2,7, 23].

Ще одне рішення лежить, так би мовити, на поверхні - розробка ручного методу побудови тестів [24]. Аналіз цього підходу не представляє інтересу, оскільки цей підхід так само, як і в попередньому випадку, є приватним рішенням і повністю залежить від кваліфікації, знань і досвіду проектувальників.

Таким чином, створення спеціалізованої автоматизованої системи генерації тестових послідовностей для тестування мережевих протоколів є найбільш прийнятним і доцільним. Підтвердження цьому можна знайти в роботах [6, 25-31].

Розглянемо систему тестування мережевих протоколів, запропоновану автором в [29], і наведену на рис. 1.1. До складу даної системи входять наступні блоки:

– TG (Test Generator) - генератор тестових наборів;

– TC (Test Converter) - перетворювач тестових наборів, що приводить їх до єдиного стандартизованого виду;

– CORE (Main functional block) - основний функціональний блок, ядро системи;

– RA (Results Analyzer) - аналізатор результатів тестування;

– RVR (Results View Rendering) - блок візуалізації результатів.

В якості вхідних інтерфейсів можна назвати:

– tech-docs, dev-notes - технічна документація, замітки розробників;

– protocol's model - логічна модель протоколу;

– protocol's specification - специфікація протоколу.

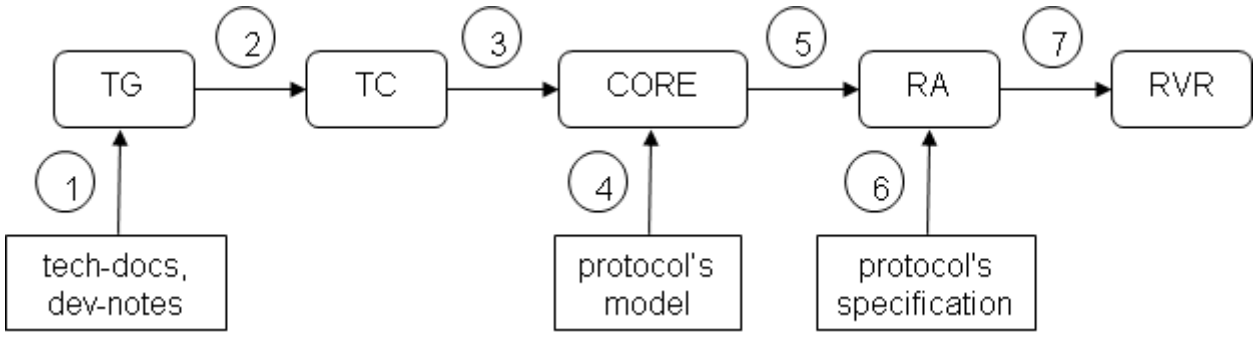


Рисунок 1.1 - Загальна структура системи тестування

Крім перерахованих блоків система містить наступні типи потоків даних:

– оброблювач чорнових матеріалів на базі UniTESK;

– таблиця тестових вхідних впливів;

– нормалізувати набори тестів;

– графічна модель протоколу;

– результати тестування у вигляді таблиці значень;

– заявлені навантаження і характеристики у вигляді таблиці значень;

– єдина таблиця порівнянь результатів з очікуваними значеннями.

Відзначимо, що дана структура представляє універсальний механізм, що дозволяє провести перевірку правильності функціонування розроблюваного тесту на конформність задекларованої специфікації як на етапі розробки, так і після її закінчення грунтуючись на затвердженої специфікації.

Першою ланкою цього механізму є блок генерації тестів (TG), вхідні значення якого отримані грунтуючись на нотатках розробників на етапі виробництва і на затвердженої специфікації після випуску останньої стабільної версії протоколу. Вихідні дані цього блоку необхідно представити у вигляді структурованої таблиці критичних значень з параметром навантаження.

Другим блоком моделі є аналізатор вихідних значень генератора (TC), який повинен інтерпретувати їх в єдиний формат для покриття всіх логічних елементів протоколу.

Виконавчою ланкою системи тестування (ядром системи) є третій блок (CORE). Це блок з двома потоками вхідних даних, наведених до єдиного формату, тобто тестові набори і модель протоколу. Дані, які отримані після аналізу специфікації покривають модель протоколу і результаті ми отримуємо набір вихідних значень тестування класифікованих за заявленим навантаженням.

Аналіз отриманих результатів тестування здійснює кінцевий блок (RA). Результати, отримані при випробуванні, порівнюються з заявленими технічними характеристиками і визначаються можливі дефекти, які можуть виникнути при реальних навантаженнях після введення протоколу в масову експлуатацію.

Також планується використовувати додатковий блок для відображення результатів тестування у вигляді графіків і діаграм (RVR), які необхідно накласти на заявлені в специфікації характеристики.

На етапі концептуального проектування системи тестування мережевих протоколів необхідно вирішити ряд проблемних питань, першим з яких є - в якому вигляді представляти вхідні дані першого блоку (замітки розробників і чорнові варіанти документацій)? Як ми знаємо, стандартним форматом нормативної документації протоколів Інтернет є документи RFC (Request for Comment) як це показано, наприклад, в [16]. Вимоги в цих документах викладені англійською мовою і є неформальний текст, що описує бажану поведінку системи. В рамках технології UniTESK [32, 33] для формального запису вимог використовуються специфікаційні розширення мов програмування - Java або C.

Технологія UniTESK пронизує весь життєвий цикл розробки програмного забезпечення від збору і аналізу вимог до його супроводу. Заснована на досвіді реальних

промислових проектів в компаніях зі сформованою культурою розробки, технологія UniTESK не вимагає для впровадження докорінної перебудови процесів. Вона легко поєднується з іншими підходами до тестування і забезпечення якості, збагачуючи їх можливості і збагачуючись при цьому сама. Всі елементи технології UniTESK підпорядковуються вимозі простежуваності вимог від етапу аналізу потреб користувачів до випуску кінцевого продукту, в нашому випадку мережевого протоколу. Технологія являє собою поєднання добре зарекомендували себе технік, які можуть застосовуватися в різних комбінаціях, взаємно поєднуючись, і підсилюючи один одного.життєвого циклу розробки протоколу від збору і аналізу вимог до супроводу.

Запис неформальних вимог нормативної документації на формальній мові являє собою модель протоколу. У підході UniTESK формальна модель протоколу будується в термінах кінцевих автоматів [18]. Переходи між станами можуть задаватися як в явному вигляді, так і в неявному. У разі явного завдання переходу модель містить алгоритм обчислення наступного стану та реакції протоколу; неявне завдання переходу є предикат, який накладає обмеження на допустимі кінцеві стану і реакції протоколу.

Другим питанням, що вимагає рішення на етапі концептуального проектування системи тестування мережевих протоколів, є представлення моделі тестованого протоколу, грунтуючись на одній з його найбільш стабільних версій [16].

В даний час на практиці часто використовується автоматний підхід до моделювання мережевих структур, а саме, використовується теорія кінцевих автоматів і теорія графів [34-38]. Такий підхід є виправданим, оскільки будь-яка специфікація протоколу може бути досить легко представлена у вигляді змістовного графа автомата, який описується наступними множинами: безліччю станів автомата, безліччю вхідних подій, безліччю вихідних подій, а також безліччю вихідних функцій і безліччю переходів автомата. Потім отримана модель представляється у вигляді графа, вузли якого є станами системи, дуги позначаються вхідними подіями, які переводять автомат з одного стану в інший, а також вихідними подіями.

Зазначений підхід дає можливість досить адекватно описувати поведінку системи під впливом мережевого протоколу. Разом з тим, на цьому автори зупиняються не використовуючи в повній мірі всіх можливостей теорії кінцевих автоматів, яка широко застосовується при проектуванні і синтезі цифрових автоматів [18, 19].

Очевидно, що даний підхід на етапі проектування і створення протоколів дозволяє значно підвищити ефективність проектування і уникнути цілого ряду помилок на етапі програмної реалізації мережевих протоколів в тому числі і стека протоколів нового покоління TCP / IP 6, про що йде мова в [16].

1.2 Класифікація тестів перевірки мережевих протоколів

У загальному вигляді процес аналізу конформности мережевих протоколів може бути представлений як сукупність двох основних складових: моделювання протоколів, і їх тестування з використанням отриманої на першому етапі моделі [15, 41, 43].

На рис. 1.2 наведено класифікацію тестів перевірки мережевих протоколів.

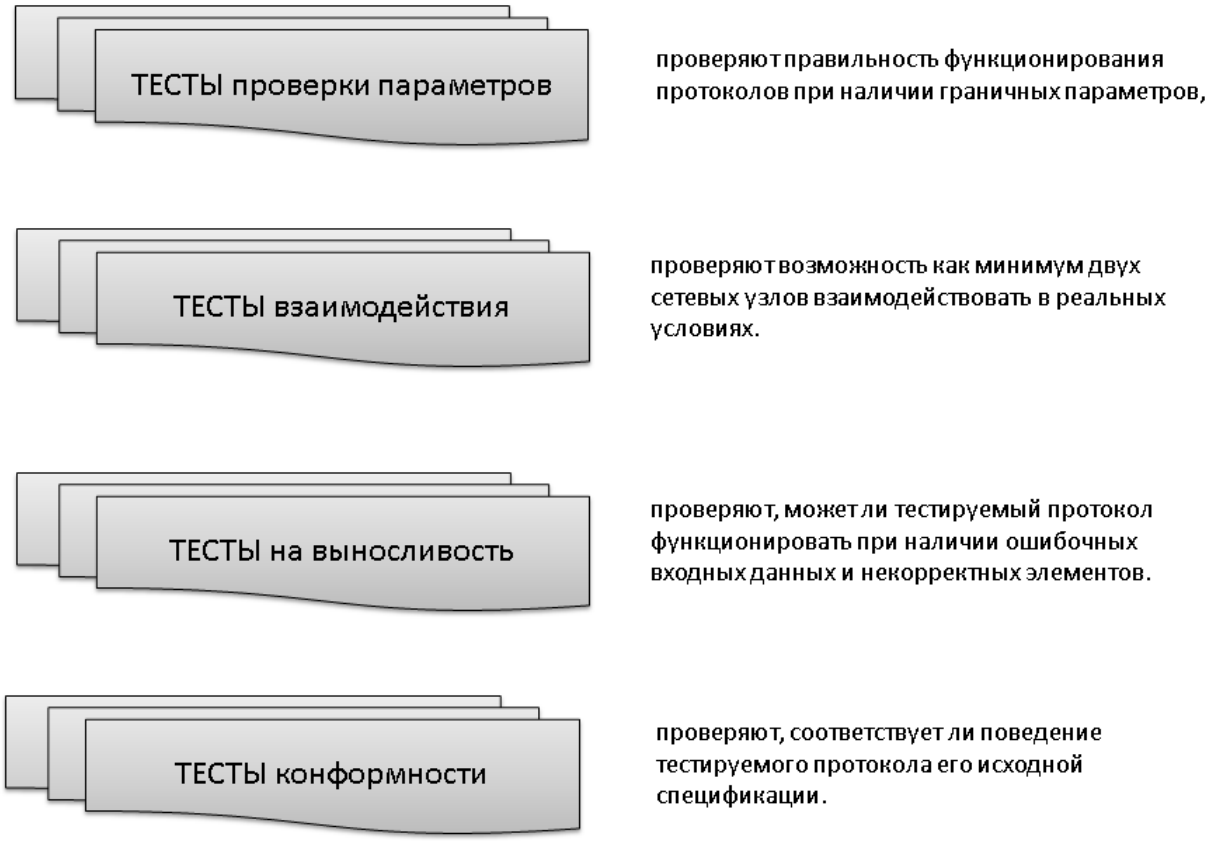


Рисунок 1.2 - Класи тестів

Тести перевірки параметрів - перевіряють правильність функціонування протоколів при наявності граничних параметрів, тобто максимальну кількість паралельних підключень, максимальні значення характеристик (в залежності від величини кешу, від значення таймерів і т.п.).

Тести взаємодії - перевіряють можливість як мінімум двох мережевих вузлів взаємодіяти в реальних умовах.

Тести на витривалість - перевіряють, чи може тестований протокол функціонувати у «ворожих» умовах, тобто при наявності помилкових вхідних даних і некоректних елементів.

Природно, що специфікація тесту не може передбачити заздалегідь всі подібні випадки. В цьому і є основна відмінність тестів відповідності і тестів на витривалість.

Тести конформності - дозволяють перевірити, чи відповідає поведінка тестованого протоколу його вихідної специфікації.

Відповідно до стандарту ISO 9646 тести конформності, в свою чергу, поділяються на такі типи [44-46]:

– тести базового взаємодії - перевіряють протокол на виконання елементарних операцій;

– тести перевірки здібностей - перевіряють протокол на дієздатність, тобто перевіряється здатність протоколу на правильне виконання вимог.

– тести перевірки робочих властивостей - перевіряють поведінку протоколу в динаміці, особливо помилкові або несвоєчасні елементи, значення параметрів, базові механізми, такі як управління потоками;

– тести відповідності - призначені для перевірки спеціальних властивостей протоколу найчастіше на основі спеціальних вимог програми. Зауважимо, що визначення цих тестів залишається сьогодні досить туманним;

– тести конформности дозволяють вирішувати поставлену задачу перевірки створюваних протоколів на їх відповідність задекларованим специфікаціям, тобто перевірка на конформність.

Надалі саме тестам конформности ми приділимо основну увагу.

Разом з тим можна виділити кілька типів тестів конформности мережевих протоколів. На рис. 1.3 наведені різні типи тестів конформності в залежності від того, на що спрямований процес перевірки конформності.

Відзначимо, що багато фірм - фахівці в своїй області пропонують свої тести, розроблені їхніми експертами, хоча в більшості випадків дані тести не стандартизовані і становлять інтерес лише при вирішенні вузького класу задач. Тому, як зазначалося вище, дані тести (точніше, методи їх побудови) ми не будемо брати до уваги, оскільки поставлена задача розробки підходу до тестування довільних мережевих протоколів [47-51].

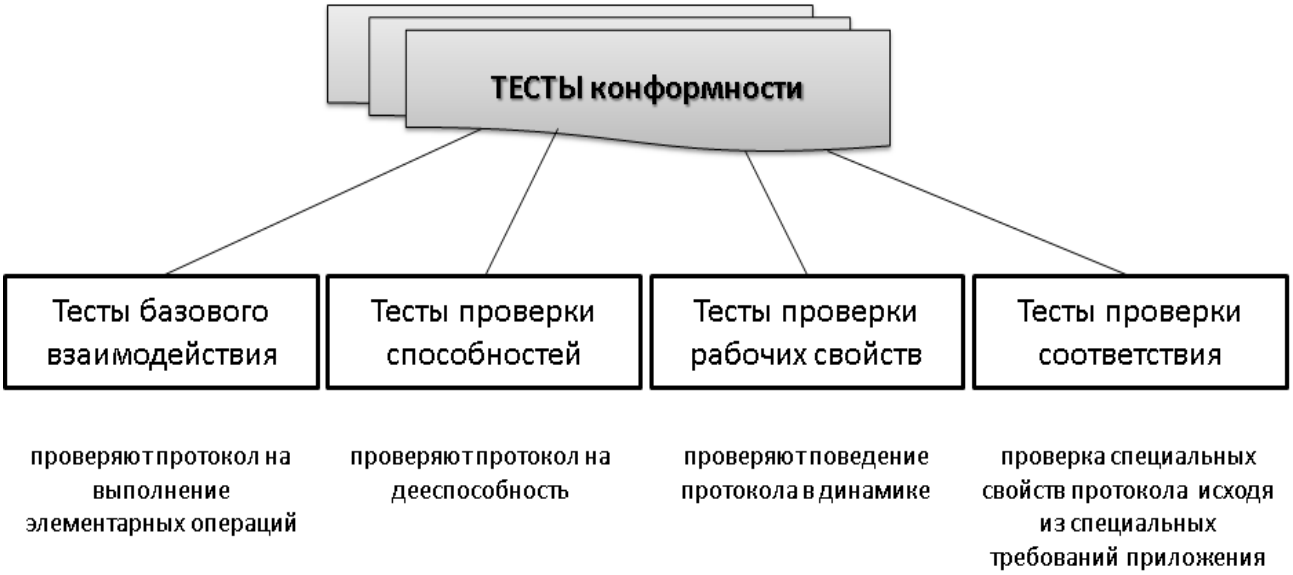


Рисунок 1.3 - Типи тестів конформности

1.3 Структура системи тестування конформности

В області тестування реалізацій Інформаційних Технологій (ІТ) одним з основних понять є поняття Лабораторія Тестування Конформності (ЛТК) або атестаційного тестування. ЛТК призначена для надання гармонізованих послуг в області атестаційного тестування і сертифікації реалізацій (продуктів) ІТ, з метою встановлення їх ступеня відповідності стандартам і профілів. В рамках нашої роботи під «реалізацією» ми розуміємо створення мережевих протоколів.

Основною вимогою при створенні ЛТК є вимога можливості акредитації ЛТК органами стандартизації, що відображає факт відповідності ЛТК вимогам міжнародних стандартів і наділення її відповідними повноваженнями, що підвищує ступінь довіри до діяльності ЛТК і дозволяє розглядати її як елемент міжнародної системи центрів тестування. Концептуальна структуру ЛТК приведена на рис. 1.4.

Розробники (specifiers) тестових специфікацій, використовуючи стандартні специфікації, розробляють структуру комплектів тестів і затвердження для цілей тестових ситуацій, а потім на цій основі створюють стандартизовані абстрактні комплекти тестів.

Творці (реалізатори) тестів здійснюють розробку для заданих абстрактних комплектів тестів відповідні їм засоби тестування (Means of Testing). Засоби тестування можуть містити одну або більше систем тестування (тестових оточень) і здійсненний (на цих системах тестування) комплект тестів.

Рисунок 1.4 - Концептуальна структура ЛТК

Структура ЛТК представляє для нас великий інтерес і подальші дослідження спрямовані на наповнення її конкретним змістом. Серед найважливіших завдань можна виділити наступні:

– аналіз існуючих та розробка модифікованих методів моделювання мережевих протоколів на базі задекларованих специфікацій протоколів;

– розробка загальної концепції і структури системи тестування мережевих протоколів з використанням розроблених методів моделювання;

– проведення експериментального дослідження розробленого підходу до вирішення проблеми тестування мережевих протоколів.

Тестові лабораторії надають сервіс в області тестування конформности реалізацій IT.

Процес встановлення конформності, що реалізовується ЛТК, може бути представлений наступними трьома основними фазами:

– підготовка до процесу тестування;

– тестові операції;

– складання звіту про тестування.

1.4 Моделі опису мережевих протоколів

Одним з базових і найбільш опрацьованих мережевих протоколів сьогодні є протокол транспортного рівня TCP, розроблений ще у вісімдесятих роках минулого століття [1, 52-56]. В даний час можна, мабуть, говорити про «сімействі протоколів TCP» оскільки за минулі роки було розроблено і реалізовано на практиці багато варіантів цього протоколу. При цьому створення нових модифікацій як цього протоколу, так і ряду інших ставить перед

розробниками завдання перевірки відповідності створюваних протоколів їх задекларованої специфікації.

У світлі вищесказаного, мабуть, найбільший інтерес при розробці методів моделювання і тестування мережевих протоколів представляє аналіз базового протоколу транспортного рівня TCP. Природно, результати, отримані при цьому, можуть бути поширені на інші протоколи різних рівнів семиуровневой моделі (OSI) [45, 57].

Дослідження властивостей транспортних протоколів, зокрема різних версій протоколу TCP, є важливим і актуальним завданням, яка розглядалася в цілому ряді робіт [6]. Основним об'єктом досліджень були алгоритми управління потоком транспортних протоколів, а основним методом досліджень, який використовувався в цих роботах - імітаційне моделювання [39]. Наприклад, розглядаються різні версії алгоритму управління потоком протоколу TCP, де видається новий протокол TCP Westwood, і на ряді прикладів обгрунтовується його ефективність також, як і новий протокол ARTCP (Adaptive Rate TCP) і на ряді модельних експериментів обґрунтовується його перевага перед стандартним TCP.

Іншим популярним методом дослідження є побудова аналітичних моделей. Наприклад, розглядається модель алгоритму управління потоком і питання породження протоколом TCP самоподібного трафіку. Методи імітаційного моделювання можуть бути вельми економічними для виявлення багатьох помилок.

Одним з підходів до вирішення завдання коректності транспортних протоколів може бути побудова формальних моделей і їх подальший аналіз за допомогою формальних методів (наприклад, методів model checking). Значний інтерес в цьому напрямку представляють роботи В. В. Кузьмука, присвячені формалізму мереж Петрі і методам аналізу їх властивостей. Основним підходом в роботах [1, 4] є побудова моделей протоколів за допомогою автоматів і їх подальша верифікація, а в багатьох роботах наводиться метод побудови моделей мережевих протоколів за допомогою мереж Петрі (а також і розфарбованих мереж Петрі) [3]. Застосування цих підходів ускладнене тим, що стандартні документи, які визначають специфікацію сімейства транспортних протоколів TCP, викладені на неформальному мовою. У роботах [36] представляються результати моделювання процесу обміну даними ряду версій протоколу TCP, зокрема аналізується модель процесів установки і завершення з'єднань. Як видно, ці роботи розглядають деякі фрагменти оригінального протоколу в той час як на практиці стоїть завдання промоделювати стандарт протоколу цілком.

Іншою особливістю цих робіт є те, що основний акцент вони роблять саме на побудові моделей, а не на розробці методів їх дослідження з метою перевірки конформності протоколів. У нашій роботі крім завдання побудови моделей, розглядаються питання аналізу

можливостей їх використання на етапі проектування і створення протоколів, а також при вирішенні задач тестування (діагностування) мережевих протоколів.

У своїй дипломній роботі на здобуття наукового ступеня кандидата наук «Моделювання та валідація комунікаційних протоколів, представлених на мовах Estelle і SDL, за допомогою мереж Петрі високого рівня» автор, Чурина Т. Г. [47], досить повно проаналізувала можливості використання мереж Петрі для вирішення задачі моделювання мережевих протоколів. Разом з тим, дана робота також не позбавлена недоліків, зазначених вище.

Чуріна Т. Г. зазначає, що «... розвиток методів трансляції SDL-специфікацій здійснювалося за двома напрямками. При першому використовуються мережі Петрі високого рівня, такі як PrT (predicate-transition) - TH і M-мережі. Запропоноване в цих роботах моделювання з використанням PrT-мереж не є повним моделюванням всієї специфікації, а здійснюється лише для верхніх рівнів специфікації, в яких відображаються потік управління і зв'язку між об'єктами специфікації. У роботах SDL-специфікації з динамічними конструкціями транслюються в мережеві моделі, в яких екземпляри процесів відображаються фішками. У роботі описана трансляція з діалекту мови SDL88 - TNSDL. При цьому для верифікації використовується аналізатор графа досяжності. В даному випадку використовуються нові класи мереж Петрі високого рівня - SDL-мережі, орієнтовані на мову. Однак їх застосування вимагає розробки спеціальних методів аналізу, неминуче трудомістких в силу складності мереж. Для застосування цих методів потрібні подальші дослідження, оскільки графи досяжності дуже громіздкі і звичайні способи їх обробки неефективні ».

Мережі Петрі знаходять досить широке застосування при моделюванні мережевих протоколів [8, 9]. Однак, проведений вище аналіз стану справ в цій галузі дозволяє зробити висновок, що такий підхід (використання мереж Петрі) має швидше за все великий теоретичний інтерес. Практичне ж його застосування в системах тестування мережевих протоколів досить спірно, особливо на етапі проектування і створення протоколів.

1.5 Можливості використання мов програмування в системі тестування мережевих протоколів

1.5.1 Мова SDL

Особливості побудови і функціонування систем генерації тестових послідовностей свідчить про те, що в даному випадку немає суворої кордону між поняттям моделі і поняттям

мови моделювання. Обидва можуть бути трактовані і як моделі і як мови в той же самий час [50]. Це зауваження стосується тільки термінологічної боку і нічого суттєво не міняє.

Серед наявних сьогодні мов моделювання певний інтерес представляє спеціалізований мова SDL (Specification and Description Language) - мова специфікації і опису, розроблений колишнім Консультативною Міжнародним Комітетом по телеграфування і телефонії (CCITT). Він використовується сьогодні як формальна мова для опису поведінки мережевих протоколів [50].

Спочатку він був призначений для опису структури і функціонування систем реального масштабу часу особливо мереж зв'язку. Відзначимо, що SDL побудований на базі моделі кінцевих автоматів по об'єктно-орієнтованої схемою. При цьому вершини графа моделі представляють процеси, а дуги між ними представлені сигналами, якими, або процеси обмінюються між собою, або процеси обмінюються із зовнішнім по відношенню до моделі системи середовищем. Ця мова непогано пристосований для опису поведінки мережевих протоколів, так як останні також часто використовують модель кінцевих автоматів для їх опису.

Разом з тим, зазначимо, що зазначена вище орієнтація SDL на графічну форму представлення специфікації мережевих протоколів вносить суттєві обмеження на його використання. Такі моделі як, наприклад, мережі Петрі погано адаптовані до використання цієї мови. Те ж можна сказати і про деяких інших способах представлення моделей мережевих протоколів.

1.5.2 Мова TTCN

Ще однією можливістю для подання тестових послідовностей мережевих протоколів є TTCN (Tree and Tabular Combined Notation) [стандарт ISO9646, частина 3]. Це уявлення дає опис тестових послідовностей незалежне від тестової архітектури. Можна сказати, що форма TTCN є мова TTCN, який визначає всю тестову послідовність [5, 46].

Розрізняють такі частини опису TTCN:

– test suite overview є список тестової послідовності;

– declarations part описує тип повідомлення і даних;

– constraints part представляє умови параметрів запиту;

– dynamic part описує послідовність запитів обміну для кожної тестової послідовності.

Існують дві різні синтаксичні форми подання TTCN: TTCN / MP (TTCN Machine Processible form) - текстова форма і TTCN / GR (TTCN Graphical form) - графічна форма. Обидві форми еквівалентні і вони можуть взаємно трансформуватися.

Структура TTCN описана і детально розглянута в літературі. Тут, ми даємо тільки кілька головних мовних характеристик TTCN. Зазвичай, TTCN передбачає дві частини.

– декларує і обмежує частину;

– динамічна частина.

Перша частина описує умови, які треба виконувати, щоб реалізувати цей тест. Друга - представляє послідовність подій, які повинні мати місце на якому тестують обладнанні.

Теоретично, можна очікувати три результату тестування:

– passed - позитивний результат тестування (має місце відповідність специфікації);

– inconclusive - безрезультатний тест;

– fail - негативний результат тестування (порушено відповідність специфікації).

– На практиці результат Inconclusive може бути інтерпретований як Fail. Тоді, в цьому випадку буде тільки два результати тесту: Passed і Fail.

Зазвичай базовий модуль складається з двох модулів: module definition part і module control part.

Module definition part задає визначення високого рівня модуля. Ці визначення можуть бути використані або в самому модулі, або в module control part так само як вони можуть бути імпортовані і з іншого модуля.

Module control part описує порядок виконання тестової послідовності.

Є кілька стали вже класичними нотацій формальних специфікацій: VDM, Z, B, CCS, LOTOS і ін. [46]. Деякі з них, наприклад, VDM, використовуються переважно для швидкого прототипування. Мова B зручний для аналізу, зокрема для аналітичної верифікації моделей. Всі ці мови активно використовуються в рамках університетських програм. У реальній практиці для опису архітектурних моделей використовується UML, а для побудови поведінкових моделей - мови SDL / MSC, виконані діаграми UML і близькі до них нотації. Поєднання TTCN-3 і MBT призвело до створення комбінації цих методологій тестування.

Використання мови TTCN передбачає різні способи. Так, наприклад, можливо ручне програмування - введення моделі мережевого протоколу в середу TTCN. Природно, ручне тестування є не раціональним для моделей з безліччю станів і переходів, тому основний упор слід зробити на метод автоматизованого тестування. Для цього необхідно згенерувати набір тестів, використовуючи ще один вбудований в платформу інструмент - TestCast Generator.

Розвиток мови TTCN спричинило за собою також і поява його модифікацій, наприклад, TTCN-3 [5].

Ще одна можливість полягає в розробці такого підходу як візуальне моделювання. Загалом итеративном процесі розробки протоколів інформаційного обміну, візуальне моделювання знаходиться на самому початку життєвого циклу програмного забезпечення [1].

У роботах [95] розглядається розробка і застосування ще однієї мови візуального моделювання протоколів інформаційного обміну, що володіє ключовими можливостями по опису таких базових конструкцій протоколів як:

- початкові умови, що передують початку роботи протоколу;
- дані учасників (в тому числі секретні і загальні дані);
- обмін повідомленнями між учасниками протоколу;
- відносини довіри між учасниками протоколу;
- циклічні процеси і умовні переходи.

Опис протоколу на даному мовою можна привести до моделі кінцевого автомата за рахунок чого можливий автоматизований аналіз описаних цією мовою протоколів. Можливість автоматизованого аналізу є надзвичайно важливою, дозволяючи виключити ряд неспроможність протоколів ще на фазі проектування.

Однак використання нових мов візуального моделювання поки що знаходиться на початковій стадії свого розвитку і не знайшла широкого застосування в області моделювання і тестування мережевих протоколів на стадії їх проектування та розробки.

1.6 Життєвий цикл мережевого протоколу

Актуальним завданням на етапі створення програмного продукту, в тому числі і мережевих протоколів, є завдання тестування протоколів перед впровадженням їх у «виробництво» оскільки тестування на етапі виробництва і супроводу ПЗ вимагає значно більших витрат у порівнянні з витратами на тестування на етапі проектування і розробки протоколів .

Як відомо, «життєвий цикл» мережевого протоколу містить наступні етапи^

– етап ескізного проектування, на якому пропонується структура майбутнього протоколу (або його модифікації) і який є одним з початкових етапів створення протоколу;

– специфікація протоколу, Яка є по суті його стандартом. В даний час в якості специфікацій використовується інформація, представлена в документах RFC (англ. - Request For Comments);

– програмна реалізація протоколу;

– етап тестової експлуатації протоколу дозволяє користувачам ознайомитися з новим протоколом на практиці і зробити висновок про його працездатності і взаємодії з іншими протоколами стека протоколів;

– етап впровадження в експлуатацію розробленого протоколу передбачає початок його широкого використання на практиці шляхом включення як сотавного компонента в поширені мережеві операційні системи.

На рис. 1.5 приведена спрощена структура життєвого циклу протоколу. У ній для стислості опущені такі етапи, як пропозиція чорнових варіантів протоколу і вибір кращого з них, а також етап старіння протоколу і виведення його з експлуатації.

Рисунок 1.5 - Спрощена структура життєвого циклу мережевого протоколу

З метою підвищення ефективності проектування мережевих протоколів, а також надійності і безперебійності їх роботи пропонується доповнити рас-смотренії структуру життєвого циклу протоколу додатковими етапами, які дозволяють значно спростити процес створення мережевих протоколів і підвищити їх ефективність і надійність функціонування шляхом аналізу конформности мережевих протоколів задекларованої специфікації. Для цього зробимо деякі зміни в структурі розглянутого життєвого циклу. Модифікована структура життєвого циклу мережевого протоколу показана на рис. 1.6.

Як бачимо, перші етапи циклу («Ескізний проект» і «Технічна специфікація протоколу») залишаються без зміни. Наступним етапом після створення специфікації протоколу пропонується здійснювати моделювання протоколу на підставі його специфікації. У роботі були проаналізовані різні можливості моделювання. У тому числі метод мереж Петрі, а також метод автоматного моделювання на основі теорії кінцевого автомата Милі. Останній метод був узятий за основу подальшого дослідження як найбільш адекватний і перспективний. В результаті була отримана на практиці автоматна модель одного з найбільш використовуваних мережевих протоколів транспортного рівня - TCP (англ. Transmission

Control Protocol). Як апробації отриманої моделі вона була реалізована на мові VHDL, який є базовим мовою при розробці апаратури сучасних обчислювальних систем.

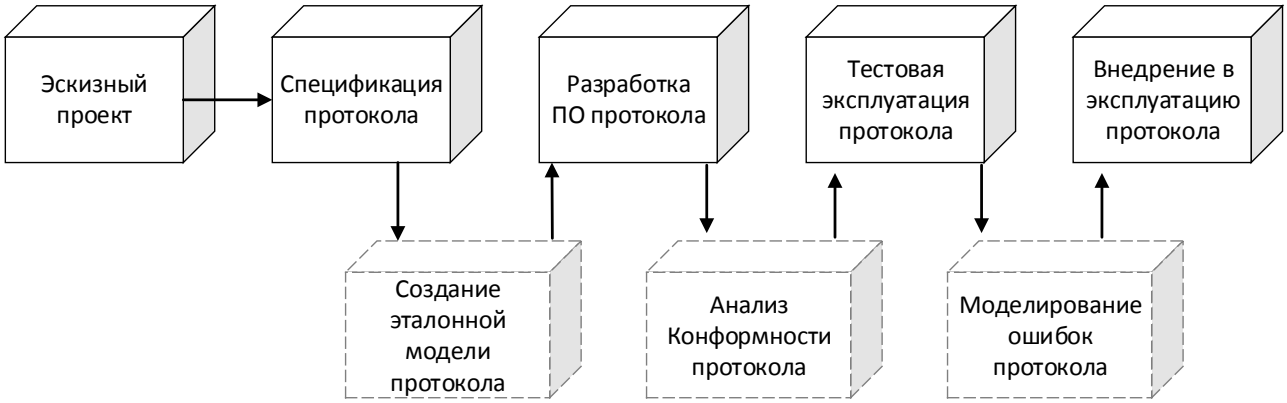


Рисунок 1.6 - Модифікована структура життєвого циклу мережевого протоколу

Як бачимо, перші етапи циклу («Ескізний проект» і «Технічна специфікація протоколу») залишаються без зміни. Наступним етапом після створення специфікації протоколу пропонується здійснювати моделювання протоколу на підставі його специфікації. У роботі були проаналізовані різні можливості моделювання. У тому числі метод мереж Петрі, а також метод автоматного моделювання на основі теорії кінцевого автомата Милі. Останній метод був узятий за основу подальшого дослідження як найбільш адекватний і перспективний. В результаті була отримана на практиці автоматна модель одного з найбільш використовуваних мережевих протоколів транспортного рівня - TCP (англ. Transmission Control Protocol). Як апробації отриманої моделі вона була реалізована на мові VHDL, який є базовим мовою при розробці апаратури сучасних обчислювальних систем.

Етап «Аналіз конформности проколу» дозволяє зробити висновок про відповідність деякій імплементації протоколу його задекларованої специфікації. Висновки про конформності протоколу можна зробити порівнюючи реакції заданої реалізації протоколу з реакцією еталонної моделі на однакові вхідні впливу (команди).

Відомо наступне визначення конформності: «Conformance (конформність) - це властивість документа, що означає повну відповідність його розмітки як синтаксичним, так і семантичним вимогам заявленого стандарту». В контексті діагностування мережевих протоколів потрібно уточнити, що конформність протоколу визначає додаток, яке оголошується конформних одному або більшій кількості профілів документа OSI, має використовувати тільки засоби, описані в даних профілях, а також в базових стандартах, на які є посилання в цих профілях. Таким чином, для перевірки конформності розробляється

протоколу задекларованої специфікації досить порівняти вихідні реакції протоколу у відповідь на вхідні дії з еталонні ми реакціями моделі специфікації на ті ж вхідні впливу.

Одним з базових понять, яким користуються проектувальники і розробники програмного забезпечення є поняття життєвий цикл програмного забезпечення (ЖЦ ПЗ), яке має на увазі включення в себе всі етапи розвитку ПО: від виникнення потреби в ньому до повного припинення його використання внаслідок морального старіння або втрати необхідності вирішення відповідних задач. оскільки мережеві протоколи створюються і використовуються у вигляді програмного забезпечення, що включається до складу як широко поширених операційних систем (таких як Windows, наприклад) в локальних мережах різного масштабу, так і спеціального серверного мережевого програмного забезпечення, то цілком очевидно, що на них поширюються всі основні положення , пов'язані з ЖЦ ПО.

Як зазначається в [11] «... по тривалості життєвого циклу програмні вироби можна розділити на два класи: з малим і великим часом життя. Цим класам програм відповідають гнучкий (м'який) підхід до їх створення і використання і жорсткий промисловий підхід регламентованого проектування і експлуатації програмних виробів. У наукових організаціях і вузах, наприклад, переважають розробки програм першого класу, а в проектних і промислових організаціях - другого ».

У цьому ж джерелі говориться, що програмні вироби з малою тривалістю експлуатації створюються в основному для вирішення наукових і інженерних задач, для отримання конкретних результатів обчислень. При цьому їх життєвий цикл характеризується великим часом, який відводився на системний аналіз і проектування. У той же час етап експлуатації і отримання результатів має порівняно малу тривалість як це показано на рис. 1.7.

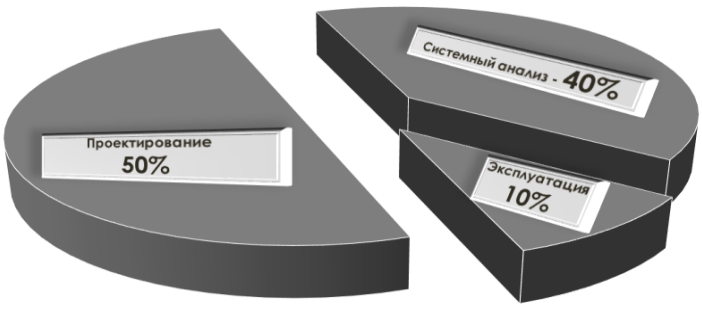


Рисунок 1.7 - ПЗ з малою тривалістю експлуатації

Характерною особливістю ПЗ з малою тривалістю експлуатації є те, що супровід та модифікація таких програм не обов'язкові.

Для регулярної обробки інформації та управління створюється ПЗ з великою тривалістю експлуатації. Програмні вироби цього класу допускають тиражування, вони представляють собою відчужувані від розробника програмні продукти. Звідси можна зробити висновок, що мережеві протоколи являють собою ПЗ з великою тривалістю експлуатації, для якого характерно розподіл етапів ЖЦ, представлене на рис. 1.8.

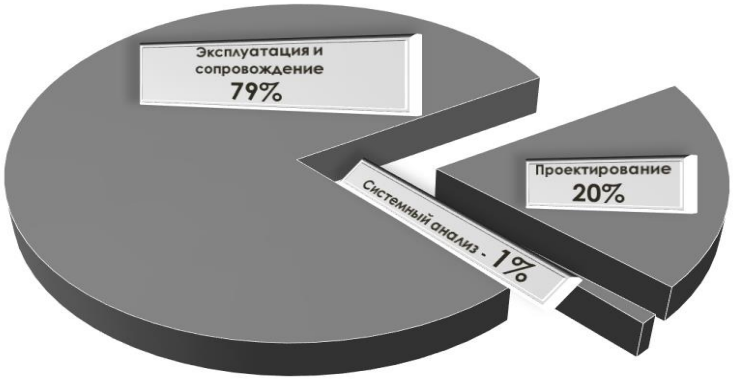


Рисунок 1.8 - ПЗ з великою тривалістю експлуатації

Внаслідок масового тиражування і тривалої експлуатації мережевих протоколів, сукупні витрати в процесі їх експлуатації та супроводу значно перевищують витрати на системний аналіз і проектування. З іншого боку, в літературі [15] відзначається, що ПЗ вважається придатним до випуску, якщо в ньому усунені всі критичні помилки та усунуто 85% некритичних помилок. Вважається, що в загальному випадку подальше тестування економічно недоцільно.

Беручи до уваги все вищесказане зазначимо, що в нашому випадку справедливим буде твердження, що однією з актуальних завдань на етапі створення програмного продукту, в нашому випадку - мережевих протоколів, є завдання тестування протоколів перед впровадженням їх у «виробництво» оскільки тестування на етапі виробництва і супроводу пО вимагає значно більших витрат у порівнянні з витратами на тестування на етапі проектування і розробки протоколів.

1.7 Структура життєвого циклу мережевих протоколів

Як зазначається в літературі [17], «ПЗ вважається придатним до випуску, якщо в ньому усунені всі критичні помилки та усунуто 85% некритичних помилок». При цьому аналітики відзначають, що розподіл витрат по стадіях життєвого циклу ПЗ приблизно таке:

на аналіз вимог, створення специфікації, проектування та кодування припадає близько 18% витрат. Решта 82% витрат припадають на тестування, промислове виробництво і супровід.

Таким чином, ми бачимо, що одним із актуальних завдань на етапі створення програмного продукту, в тому числі і мережевих протоколів, є завдання тестування протоколів перед впровадженням їх у «виробництво» оскільки тестування на етапі виробництва і супроводу ПЗ вимагає значно більших витрат у порівнянні з витратами на тестування на етапі проектування і розробки протоколів.

Як відомо, «життєвий цикл» програмного забезпечення, в тому числі і мережевого протоколу містить наступні етапи:

– етап ескізного проектування, на якому пропонується структура майбутнього протоколу (або його модифікації) і який є одним з початкових етапів створення протоколу;

– специфікація протоколу, яка є по суті його стандартом. В даний час в якості специфікацій використовується інформація, представлена в документах RFC (англ. Request For Comments). Наприклад, на сторінціhttp://www.protocols.ru/files/RFC/rfc793.pdf ми знаходимо специфікацію протоколу TCP - Transmission Control Protocol;

– програмна реалізація протоколу здійснюється на мовах, що використовуються в поширених мережевих операційних системах;

– етап тестової експлуатації протоколу дозволяє користувачам ознайомитися з новим протоколом на практиці і зробити висновок про його працездатності і взаємодії з іншими протоколами стека протоколів;

– етап впровадження в експлуатацію розробленого протоколу передбачає початок його широкого використання на практиці шляхом включення як складового компонента в поширені мережеві операційні системи.

На рис. 1.9 приведена спрощена структура життєвого циклу протоколу, що використовується в даний час на практиці. У ній для стислості опущені такі етапи, як пропозиція чорнових варіантів протоколу і вибір кращого з них, а також етап старіння протоколу і виведення його з експлуатації.

Створений на початковому етапі проектування ескізний проект покладено в основу специфікації протоколу, яка стає стандартом для заданого протоколу. Наступний потім етап розробки програмного забезпечення протоколу передбачає, що отриманий в результаті розробки протокол повністю відповідає його специфікації. Підкреслимо, що це пред'являє серйозні вимоги до забезпечення конформности програмного продукту.

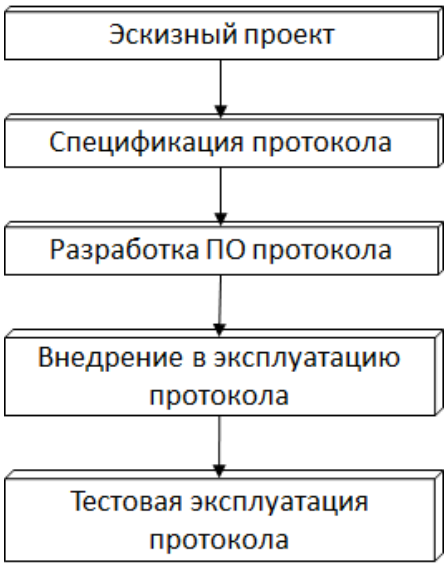


Рисунок 1.9 - Спрощена структура життєвого циклу протоколу

1.7.1 Модифікація структури життєвого циклу протоколу

З метою підвищення ефективності проектування мережевих протоколів, а також надійності і безперебійності їх роботи пропонується доповнити розглянуту структуру життєвого циклу протоколу додатковими етапами, які дозволяють значно підвищити ефективність і надійність функціонування мережевих протоколів шляхом аналізу конформности протоколів задекларованої специфікації. Для цього зробимо деякі зміни в структурі розглянутого життєвого циклу. Пропонується наступна модифікована структура життєвого циклу мережевого протоколу (рисунок 1.10).

Як бачимо, перші етапи циклу («ескізний проект» і «специфікація протоколу») залишаються без зміни. Наступним етапом після створення специфікації протоколу пропонується здійснювати «створення еталонної моделі протоколу» на основі його специфікації. Процес побудови еталонної моделі протоколу базується на використанні автоматного принципу, запропонованого в розділі 2 цієї роботи.

Відзначимо два очевидних застосування еталонної моделі мережевого протоколу для вирішення завдання підвищення якості функціонування розподілених інформаційних мереж, що використовують розробляються мережеві протоколи.

По-перше, порівняння реакції розробленого ПЗ протоколу з реакцією еталонної моделі на однакові вхідні впливу дозволяє зробити висновок про відповідність протоколу його специфікації, тобто про його конформності.

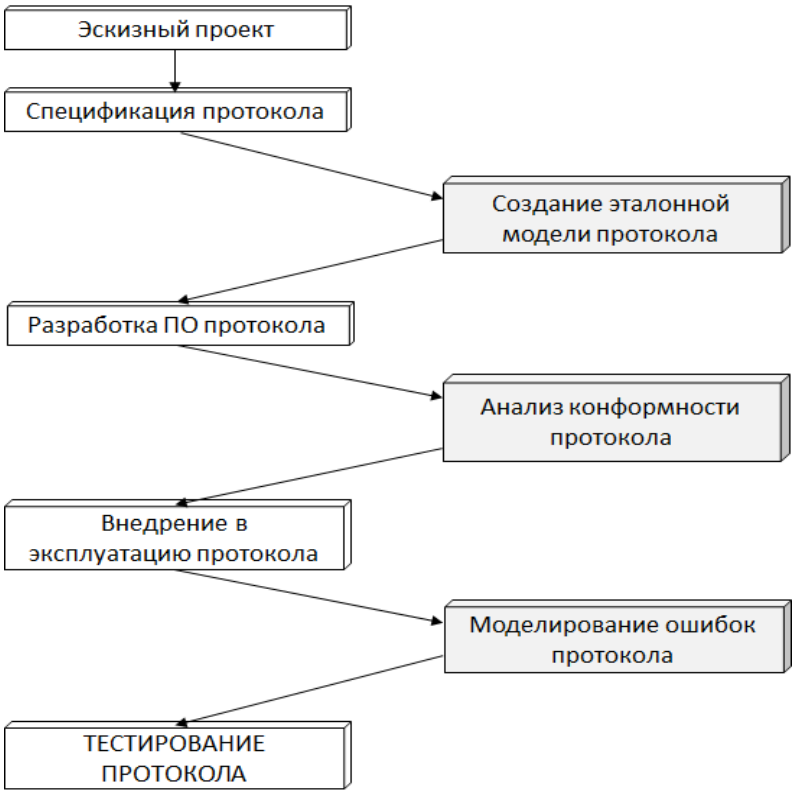


Рисунок 1.10 - Модіфіцірованная структура життєвого циклу протоколу

По-друге, використання еталонної моделі дозволяє заздалегідь скласти таблицю реакцій протоколу на наперед заданий або сгенерированное деяким чином (можливо випадковим) безліч помилок з тим, щоб перевірити реалізацію протоколу на наявність або відсутність цих помилок. Промоделювати на еталонної моделі заданий безліч помилок протоколу ми отримуємо можливість їх виявляти і класифікувати на етапі проектування і створення протоколу.

Таким чином, створення еталонної моделі протоколу дає можливість реалізувати два нових етапу життєвого циклу: етап аналізу конформности проколу і етап моделювання помилок протокол »з метою його тестування.

Етап аналізу конформности проколу дозволяє зробити висновок про відповідність деякій імплементації протоколу його задекларованої специфікації. Як зазначалося, висновки про конформності протоколу можна зробити порівнюючи реакції заданої реалізації протоколу з реакцією еталонної моделі на однакові вхідні впливу (команди).

Етап «моделювання помилок протоколу» передує завершального етапу життєвого циклу, пов'язаного з впровадженням в експлуатацію розробленого протоколу. Відзначимо, що протокол може бути представлений або у вигляді графа потоку керуючих команд контролера або у вигляді графа потоку даних між вхідними параметрами і вихідними змінними контексту.

1.7.2 Використання етапів модифікованого ЖЦ в системі перевірки конформності протоколу

Розроблена вище модифікація життєвого циклу протоколу є універсальною і дозволяє використовувати її етапи як для перевірки конформності протоколу, так і для його тестування. Як ми бачимо з рис. 1.5, перше завдання, перевірка конформності, повинна здійснюватися на перших етапах створення протоколу в той час як друга задача, тестування, буде вирішуватися при необхідності перевірки працездатності вже створеного протоколу в разі наявності в ньому деяких помилок, що з'явилися в його програмній реалізації в процесі експлуатації протоколу.

Пропонується наступна структура системи перевірки конформності протоколу побудована з використанням запропонованих етапів модифікованого ЖЦ (рис. 1.11).

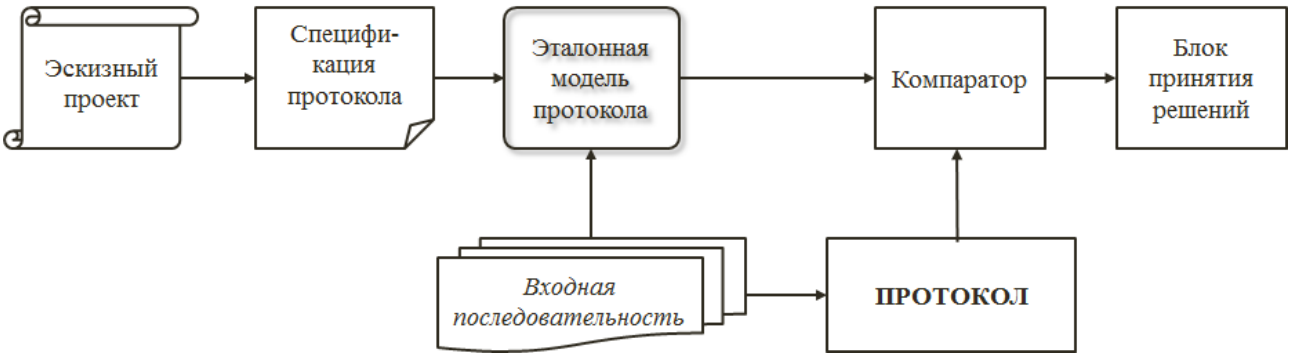


Рисунок 1.11 - Система перевірки конформності протоколу

Спочатку виходячи з ескізного проекту протоколу створюється його специфікація, яка є детальним описом структури та закону функціонування створюваного протоколу і грає роль офіційного стандарту для даного протоколу. Потім, використовуючи метод автоматного моделювання, ми одержуємо еталонну модель протоколу. Як зазначалося там же, еталонна модель може бути задана декількома способами: у вигляді програмної моделі, в аналітичному вигляді, у вигляді апаратної моделі і т.п.

Принцип перевірки програмної реалізації протоколу на його конформність задекларованої специфікації полягає в наступному: необхідно порівняти реакцію еталонної моделі протоколу і його конкретної реалізації на певну послідовність вхідних впливів і на підставі отриманих результатів порівняння зробити висновок про конформності.

Для вирішення цього завдання на входи отриманої еталонної моделі і побудованого протоколу подається деяка вхідна послідовність. Характерно, що дана послідовність в

принципі може бути випадковою або псевдослучайной. Вихідні реакції з еталонною моделі і з перевіряється протоколу надходять на блок порівняння - компаратор, який спільно з блоком прийняття рішень дозволяє зробити висновки про конформності розробленого протоколу.

1.7.3 Використання модифікованого ЖЦ в системі діагностування протоколу

Нові етапи, введені в ЖЦ протоколу, дозволяють крім розглянутої вище перевірки конформності протоколу його задекларованої специфікації дозволяють також створити систему тестування протоколу, що дозволяє виробляти тестування протоколу на наявність в його функціонуванні деякого безлічі помилок і їх ідентифікацію. Такий підхід до перевірки працездатності протоколу дає можливість в разі виявлення помилки зробити її діагностику, тобто визначити тип помилки, що дозволить значно спростити подальший процес усунення помилки в розробці протоколу.

Як відомо, завдання постановки діагнозу може бути вирішена якщо ми маємо так званий «словник помилок», який представляє собою таблицю, де вказана взаємозв'язок кожної конкретної помилки з деяким вхідним словом або вхідним безліччю, сформованим для виявлення заданої помилки, і відповідної вихідний реакцією об'єкта діагностування . Такий підхід до діагностування цифрових пристроїв є класичним. Пропонується застосувати такий же підхід і до діагностування мережевих протоколів з урахуванням модифікації ЖЦ, запропонованої вище.

Систему перевірки конформності протоколу, наведену на рісунке 3.4, ми кілька перетворимо - усунемо несуттєві для даного випадку блоки і додамо нові. Структура системи діагностування протоколів з використанням модифікованого життєвого циклу протоколів представлена на рис. 1.12.

Для складання словника помилок протоколу ми визначаємо то безліч помилок, на яке ми будемо орієнтуватися в подальшому при діагностуванні протоколу. Як правило, вирішення питання складання переліку типових помилок протоколу можна віддати на відкуп досвідченому системному адміністратору чи іншому експерту, добре розбирається в особливостях функціонування об'єкта, що діагностується.

Подаючи цю послідовність на входи еталонної моделі ми формуємо безліч реакцій протоколу на вхідні дії. Ставлячи у відповідність ці значення вхідних сигналів і вихідних реакцій система формує так званий «словник помилок протоколу», який потім перетворюється в тест.

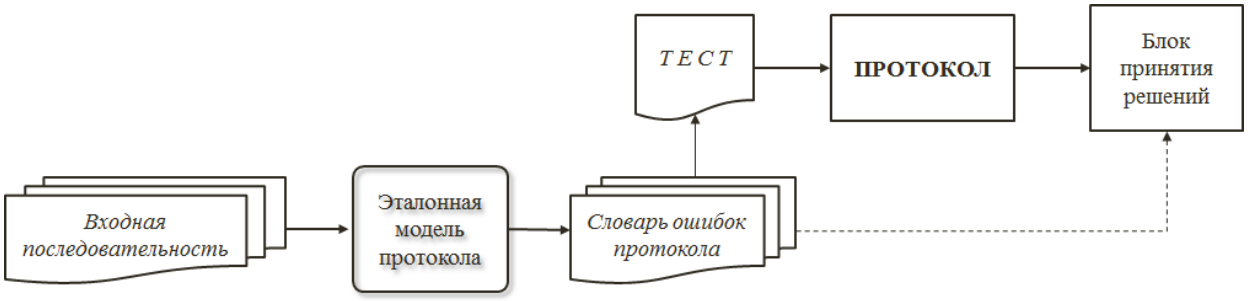


Рисунок 1.12 - Система діагностування протоколу

Особливістю тестової послідовності є те, що вона повинна бути неізбиточной і по можливості повно покривати граф, однозначно описує специфікацію протоколу. Тестова послідовність, що подається на об'єкт діагностування, викликає його відповідні вихідні реакції, які надходять на блок прийняття рішень. Аналізуючи поведінку протоколу, і вибираючи зі словника помилок потрібні записи, блок прийняття рішень визначає клас і тип помилок, що мають місце в роботі протоколу.

Відзначимо, що блоки, наведені в схемах на рис. 1.10 і 1.11, в даній роботі слід інтерпретувати швидше, як блоки алгоритму функціонування системи, ніж як фізичні блоки електронної схеми хоча останнім в принципі не виключається.

Структура узагальненої системи тестування мережевих протоколів показана на рис. 1.13.

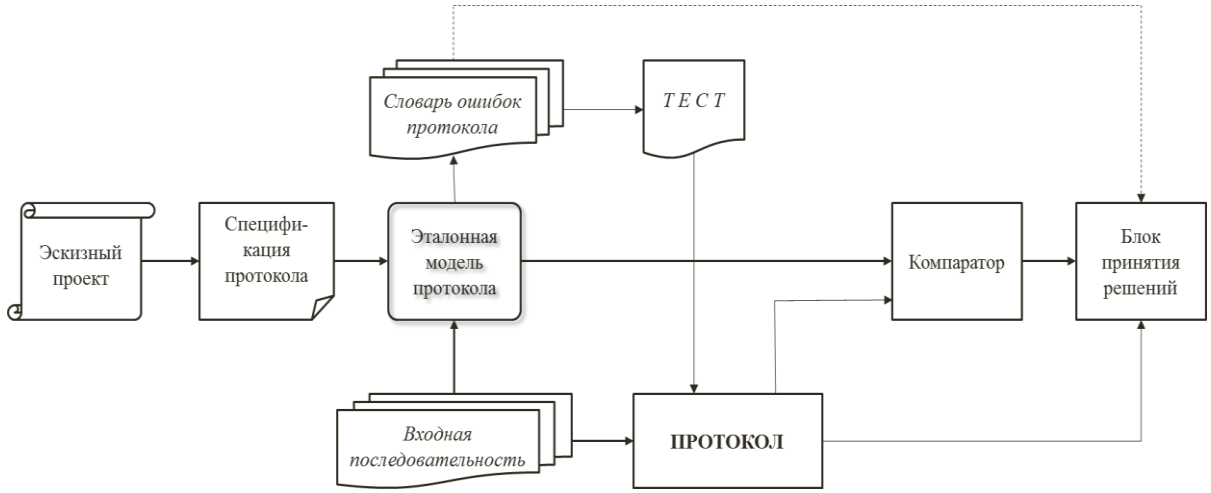


Рисунок 1.13 - Узагальнена система тестування мережевих протоколів

Очевидно, що дана система є комбінацією двох систем, розглянутих вище: зистема перевірки конформності і системи діагностування протоколу (рис. 1.11, 1.12). Таке розбиття на дві підсистеми є чисто умовним і служить лише цілям спрощення розуміння суті розробки.

1.8 Постановка задач досліджень дипломної роботи

В умовах стрімкого розвитку web-простору і збільшення кількості вузлів користувачів, в тому числі і мобільних, в необхідності оптимальної агрегації мережевих адрес і захисту інформації, що передається, з'являються нові проблеми, пов'язані з необхідністю переходу на новий стек протоколів шостої версії (далі TCP / IP 6). Зокрема, це проблема тестування і верифікації нових мережевих протоколів, а, як наслідок, необхідність моделювання протоколів з метою формування відповідних тестових послідовностей для їх перевірки на відповідність задекларованої специфікації.

Рішення таких завдань можливе за допомогою методів моделювання, що дозволяють уявити неформализованное опис функціонування мережевих протоколів (специфікацій) у вигляді формальної моделі, придатної для подальшої комп'ютерної обробки з метою отримання тестової послідовності для перевірки реалізації протоколу з точки зору її відповідності вихідної специфікації.

Аналіз життєвого циклу мережевих протоколів показує, що в «класичному», існуючому сьогодні вигляді процес розробки і створення мережевих протоколів є досить трудомістким і не гарантує створення протоколів адекватно відображають вимоги задекларованої специфікації. Це, в свою чергу, часто призводить до необхідності циклічного повторення або всього циклу, або його значної частини, починаючи з етапу розробки ПЗ, що є неприпустимим особливо при переході на новий стек протоколів TCP / IP версії 6. При цьому ми неминуче приходимо до необхідності включення в життєвий цикл таких етапів, як створення еталонної моделі, аналіз конформности протоколу, а також етапу моделювання можливих помилок створюваного протоколу, що дозволяє значно спростити процес створення протоколу,

Аналіз можливих видів і типів тестування мережевих протоколів показує, що підготовка тестів експертами - фахівцями в області мережевих технологій є вельми трудомісткою і дорогою операцією, що вимагає роботи висококваліфікованих фахівців у своїй галузі. Даний підхід є неприйнятним у разі, коли мова йде про потреби порівняно невеликих робочих майданчиках - користувачів мережевими ресурсами. У цьому випадку доцільним є використання на практиці заздалегідь створених еталонних моделей протоколу.

Аналогічні зауваження можна зробити і по відношенню до ручного методу побудови тестів з тією лише різницею, що такий підхід може бути застосований швидше в разі малих робочих площадок.

Аналіз показує, що застосування спеціалізованих автоматизованих систем генерації тестових послідовностей для тестування мережевих протоколів може бути виправдано лише на завершальних етапах життєвого циклу протоколу, що не дозволяє впливати на розробку і створення мережевих протоколів в його початку.

Метою дипломної роботи є розробка і аналіз методів моделювання мережевих протоколів для забезпечення перевірки конформності розроблюваних протоколів заданої специфікації на етапі створення протоколу з метою підвищенняякості обслуговування і достовірності передачі інформації в комп'ютерних мережах, ефективності проектування, створення, і функціонування мережевих протоколів.

Досягнення поставленої мети здійснюється вирішенням наступних основних завдань:

– аналіз основних методів моделювання протоколів і розробка автоматного підходу до моделювання мережевих протоколів з використанням класичної теорії синтезу мікропрограмних автоматів.

– аналіз структури життєвого циклу протоколу і його модифікація для використання еталонної моделі мережевого протоколу на ранніх етапах циклу з метою підвищення ефективності проектування протоколів.

– аналіз і розробка принципів перевірки конформності мережевих протоколів і можливості використання автоматної моделі для верифікації конформности задекларованої специфікації створюваних протоколів.

– аналіз і розробка принципів побудови системи діагностування мережевих протоколів з використанням еталонної моделі, отриманої на ранніх етапах проектування протоколів.

– дослідження можливості застосування розробленого підходу до моделювання протоколів для вирішення прикладних завдань діагностування протоколів на прикладі мережевого протоколу транспортного рівня TCP / IPv.4.

– аналіз методів побудови тестів для перевірки правильності функціонування мережевих протоколів в сучасних комп'ютерних мережах.

– розробка структури та функцій окремих модулів системи генерації тестів мережевих протоколів для вирішення прикладних завдань тестування протоколів з метою підвищення ефективності та надійності функціонування сучасних розподілених обчислювальних мереж.

2 РОЗРОБКА АВТОМАТНОЇ МОДЕЛІ МЕРЕЖЕВИХ ПРОТОКОЛІВ

Перш ніж приступити питань моделювання мережевих протоколів проведемо аналіз стану справ в питаннях формалізації завдання протоколів, відзначаючи як перспективні напрямки в дослідженнях, так і невирішені в даний час питання. Такий аналіз ми почнемо з огляду способів обміну даними в розподілених системах, яскравим представником яких зараз є Інтернет. Розділ присвячений розробці автоматної моделі мережевих протоколів на базі класичної теорії синтезу кінцевих автоматів. Наведено приклад використання автоматних моделей. Показана можливість використання аналітичної форми подання протоколу з метою його моделювання, а також проведено порівняння з методом мереж Петрі для моделювання мережевих протоколів.

2.1. Способи обміну даними, типи і призначення блоків даних

Прийнята в 1984 році еталонна модель OSI / RM (англ. Open System Interconnections / Reference Model, master model) являє собою метод опису мережевих середовищ і відкритих архітектур (рисунок 2.1).

Рисунок 2.1 - Архітектура еталонної моделі OSI RM

Основною метою даної моделі є стандартизація і простота написання драйверів певного рівня, можливість організації стеків протоколів. Дана архітектура еталонної моделі є найбільш поширеною і широко застосовується.

Характеристики кожного з рівнів еталонної моделі OSI RM наведені в таблиці 2.1.

Таблиця 2.1 - Рівні моделі OSI RM

рівень моделі	опис
прикладний	Загальний доступ до мережі, потік даних, наприклад, telnet.
подання даних	Визначає формат для обміну даними (перекладач), переклад даних понад в загальноприйнятий стандарт, шифрування, зміна кодової таблиці, стиснення даних.
сеансовий	Встановлення, використання і завершення сеансу зв'язку, розпізнавання імен та захист, розстановка checkpoints, щоб в разі невдалої передачі починати з поганого місця, некоректне завершення сеансу.
транспортний	Гарантує доставку пакетів без помилок, в тій же послідовності, без втрат і дублювання.
Мережевий	Адресація і маршрутизація в глобальних мережах. Комутація пакетів, маршрутизація, перевантаження.
канальний	Передача кадрів з мережевого рівня в середу передачі (паралельної в послідовну і навпаки), іноді спеціальне кодування.
фізичний	Потік бітів. Електричний, оптичний, механічний і функціональний інтерфейси мережевої плати з кабелем.

У даній роботі нас більше цікавлять процеси, пов'язані з передачею повідомлень по каналах зв'язку. Виходячи зі структури мережевої моделі OSI RM, відзначимо, що формування пакета відбувається послідовно на всіх рівнях, тобто на кожному рівні здійснюється додавання відповідного префікса (заголовка) до переданому пакету. Потім, при отриманні пакету відбувається аналіз заголовків і їх відсікання. Цей процес детально описаний в [2] і добре ілюструється рис. 2.2.

Як видно з рис.2.2, повідомлення, сформоване на прикладному рівні, передається на рівень представлення, який часто називають «представницьким». При цьому до вихідного

пакету приєднується префікс - заголовок, властивий даному рівню. Далі пакет послідовно передається на нижче лежачі рівні, кожен раз з приєднанням відповідного префікса. По суті справи, процес додавання додаткового заголовка можна назвати процесом «інкапсуляції», який широко використовується при тунелюванні пакетів.

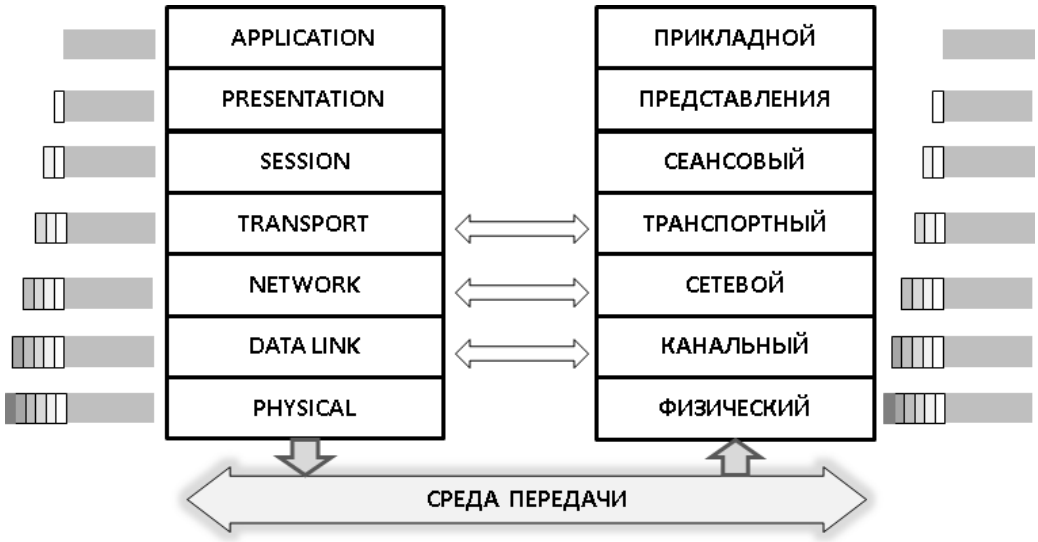


Рисунок 2.2 - Передача даних по мережі

Передача даних по мережі здійснюється за допомогою мережевих протоколів. Сьогодні в глобальній мережі Інтернет дуже широко використовується стек протоколів TCP / IP (Transmission Control Protocol / internet Protocol). Це стандарт для гетерогенних мереж. Для цього стека спеціально розроблені такі протоколи як SMTP, FTP, SNMP. Одним з найбільш використовуваних протоколів цього стека є транспортний протокол TCP. Надалі мова буде йти в основному про протоколах цього стека, а саме про протокол TCP. Природно, розроблені принципи моделювання можна досить просто адаптувати до їх використання для інших стеків протоколів.

2.2. Угода про специфікації протокольних сервісів

2.2.1 Поняття методу і нотації специфікації протокольних сервісів

Стандартизація взаємозв'язку відкритих систем включає три рівні опису засобів інформаційного обміну, а саме: концептуальний рівень, зміст якого визначається моделлю OSI RM, рівень специфікацій функціональних можливостей (або сервісів) елементів архітектури OSI RM і рівень специфікацій протоколів інформаційного обміну між функціональними елементами еталонної моделі.

Розглянемо модель і угоду про специфікації сервісів:

– (N)-сервіс ((N) -service): Функціональні можливості, які можуть бути надані на кордоні між (N + 1) - і (N) - рівнями.

– Користувач (N) -Сервис ((N) -service-user): Абстрактне уявлення всієї множини тих логічних об'єктів в деякій (N + 1) -подсістеме, які використовують деякий (N)-сервіс через деяку (N) -точку доступу.

– Постачальник (N) -Сервис ((N) -service-provider): Абстрактний автомат, який моделює поведінку сукупності логічних об'єктів, що забезпечують (N)-сервіс, з точки зору користувача.

– (N) - сервісний примітив або примітив ((N) -service-primitive; primitive): Абстрактне, що не залежне від реалізації неподільне взаємодія між користувачем (N) -Сервис і постачальником (N) -Сервис, що відбувається на кордоні між ними.

У разі, коли зміст поняття поширюється на всі сім рівнів еталонної моделі, воно префексіруется словом OSI, наприклад, ((OSI) -service, (OSI) -service-user, (OSI) -service-provider, (OSI) -service- primitive).

В теорії протоколів і в документації по мережевим протоколам сервісні примітиви, за допомогою яких описуються міжрівневого взаємодії в моделі OSI RM, називаються також абстрактними сервісними примітивами або ASPs (Abstract Service Primitives).

Далі вводяться визначення двох загальних примітивів: запросити (submit) і доставити (deliver), а також двох загальних типів користувачів сервісу:

– ініціатор запиту (requestor) і одержувач (acceptor);

– запросити (submit): Сервісний примітив, що ініціюється користувачем сервісу (направляється від користувача до постачальника сервісу);

– доставити (deliver): Сервісний примітив, що ініціюється постачальником сервісу (направляється від постачальника сервісу до користувача);

– ініціатор запиту (requestor): Користувач, який видає (постачальнику сервісу нижчого рівня) примітив запросити, і в результаті чого може отримати від постачальника один або кілька примітивів доставити;

– одержувач (acceptor): Користувач, який примітив доставити, в результаті чого може видати постачальнику як відповідь один або кілька примітивів запросити.

Введені вище класифікація примітивів не надто зручна для використання через свою спільності. Тому при визначенні стандартів сервісів мережних протоколів застосовується більш практична класифікація примітивів, що виводиться з визначень загальних примітивів і класів користувачів.

2.2.2 Модель сервісу рівнів

Під сервісом рівня ми розуміємо функціональні можливості відповідного постачальника сервісу, які він може запропонувати користувачам на своєму кордоні (в точках доступу до сервісу) для реалізації взаємозв'язку між користувачами. Сервіс рівня визначається в термінах абстрактної моделі, що містить наступні елементи: (N)-користувачів сервісу, (N) -постачальники сервісу, кордон взаємодії ((N) -SAP) і спостерігаються на кордоні елементарні акти взаємодії (події, пов'язані з передачею через кордон сервісних примітивів). Загальна структура моделі сервісу рівнів показана на рис. 2.3.

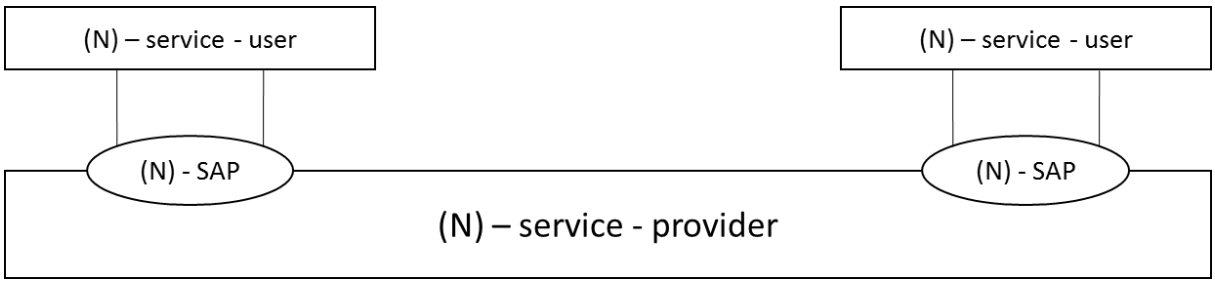


Рисунок 2.3 - Модель сервісу рівнів

Це модель надання (N) -Сервис з боку постачальника (N) -Сервис двом рівнозначним або однорангові (N) -Користувач (peer (N) -users), тобто (N + 1) -сущностям-кореспондентам, які взаємодіють один з одним за допомогою деякого (N + 1) протоколом. Дана модель легко розширюється.

2.2.3 Склад і основні властивості сервісних примітивів

Визначимо чотири типи сервісних примітивів:

– примітив запит (request);

– примітив індикація (indication);

– примітив відповідь (response) - може бути позитивним або негативним;

– примітив підтвердження (confirm) - може бути позитивним або негативним.

Основними властивостями примітивів є:

– атомарность - кожен сервісний примітив являє собою логічно самостійне неподільне взаємодія, яке не може бути перервано іншим взаємодією;

– спрямованість - сервісний примітив має напрямок або від користувача сервісу до постачальника сервісу, або навпаки;

– перенесення семантики- з сервісним примітивом можуть бути пов'язані один або кілька параметрів, кожен з яких має певний діапазон значень. Для опису сервісних примітивів використовується функціональна форма запису.

Таким чином, процес реалізації (N) -Сервис може розглядатися у вигляді упорядкованого в часі набору атомарних подій, що відбуваються на кордоні (N) -рівня (в точках (N) -SAP), і пов'язаних з передачею і прийомом примітивів запит, індикація, відповідь, підтвердження.

2.2.4 Угоди про часові діаграмах

Для опису логічних і часових зв'язків між примітивами сервісу будемо використовувати метод діаграм, званих TS-діаграмами (time-sequence diagrams). За допомогою часових діаграм можна відобразити:

– послідовність подій на кордоні (N + 1) -Користувач / (N)-постачальник в деякій відкритій системі;

– послідовність подій між (N + 1) -Користувач.

Кожна діаграма розділена двома вертикальними лініями на три області. Центральна область представляє постачальника сервісу, а дві області, розташовані зліва і праворуч від центральної, відповідають користувачам сервісу. Вертикальні лінії представляють точки доступу до сервісу. Вони ж є і часовими осями.

Послідовності подій, що відбуваються в кожній точці доступу до сервісу, розміщуються на вертикальних лініях. Причому події, які розташовується на лінії нижче іншого, відбувається в більш пізній момент часу. Стрілки в призначеній для користувача області діаграми сервісу, вказують напрямок передачі примітиву.

Перебіг часу на діаграмі відображається за допомогою нахилу вниз ліній зв'язку в області, що представляє постачальника сервісу. На рис. 2.4 показані способи побудови таких діаграм.

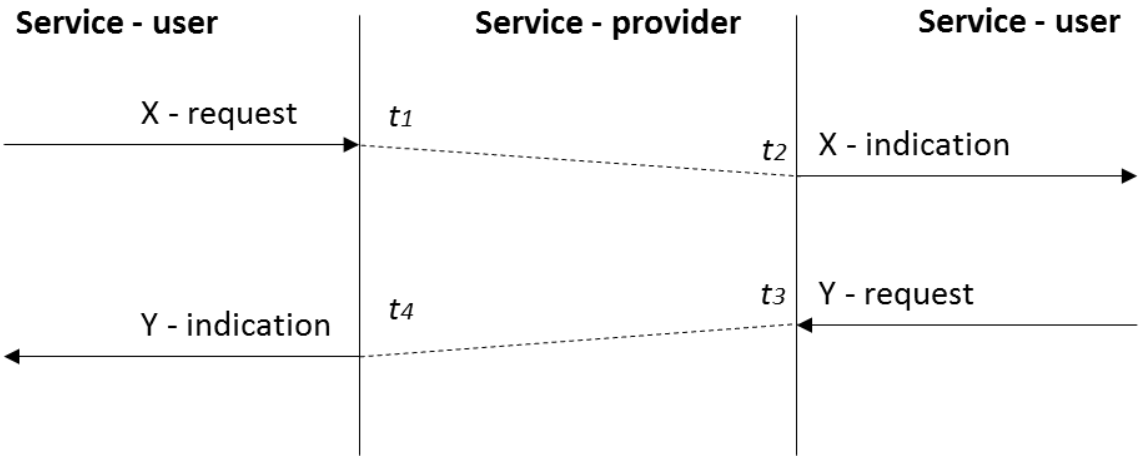


Рисунок 2.4 - Часові діаграми

2.2.5 Графічна модель ГПС

У даній роботі пропонується використовувати деяку модифікацію часових діаграм, розглянутих вище. Метою даної розробки є формалізація описового способу завдання мережевих протоколів - використання графіків послідовності повідомлень (ГПС). Даний підхід є виправданим для опису функціонування мережевих протоколів, оскільки головною особливістю останніх є те, що вони досить легко можуть бути представлені у вигляді деякої послідовності повідомлень: запитів (request) і відповідей (response).

Як приклад використання моделі ГПС розглянемо її використання для опису одного з фрагментів протоколу транспортного рівня TCP. Як відомо, першою фазою даного протоколу є встановлення логічного з'єднання між двома вузлами. Ця фаза носить назву Handshake (рукостискання) і полягає в аутентифікації вузлів, що вступають в зв'язок [98].

Нехай вузол А є ініціатором (Initiator User A) зв'язку з вузлом В, а вузол В в нашому випадку є відповідачем (Responder User B). Блоки Initator і Responder це TCP протоколи вузлів А і В відповідно, а в якості Medium виступає провідна (або бездротова) понеділок мережі (Рисунок 2.5).

На наведеному вище рис. використані наступні скорочення: CON - connection; MDAT - Medium Data; req - Request; ind - Indication; conf - Confirmation; resp - Respons.

У загальному випадку ми маємо два напрямки проходження запитів (Request) і (Respons). Це так звана Служба примітивів - Service Primitives (SPs). При цьому одиниця інформації, що передається називається Protocol Data Units (PDU). Тобто ми маємо наступну послідовність запитів:

«Initiator User A ↔ Initiator ↔ Medium ↔ Responder ↔ Responder User B».

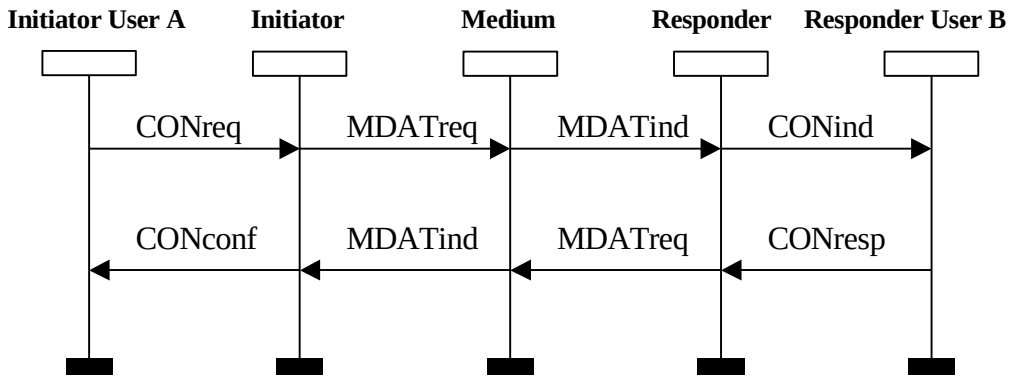


Рисунок 2.5 - Модель ДПС встановлення логічного зв'язку

Скориставшись принципами опису процесу, наведеними вище (Рисунок 2.6), для побудови ДПС протоколу TCP, який встановлює логічне з'єднання (віртуальний канал) з подальшою передачею повідомлень. Очевидно, що навколишнє середовище Medium повинна розміщуватися між Initiator і Responder. Але в нашому прикладі ми її проігноруймо для спрощення загальної структури ДПС. В результаті отримуємо спрощений ДПС протоколу TCP (рисунок 2.6). Зауважимо, що фаза Handshake представлена в повному обсязі, а решта фази дані лише схематично.

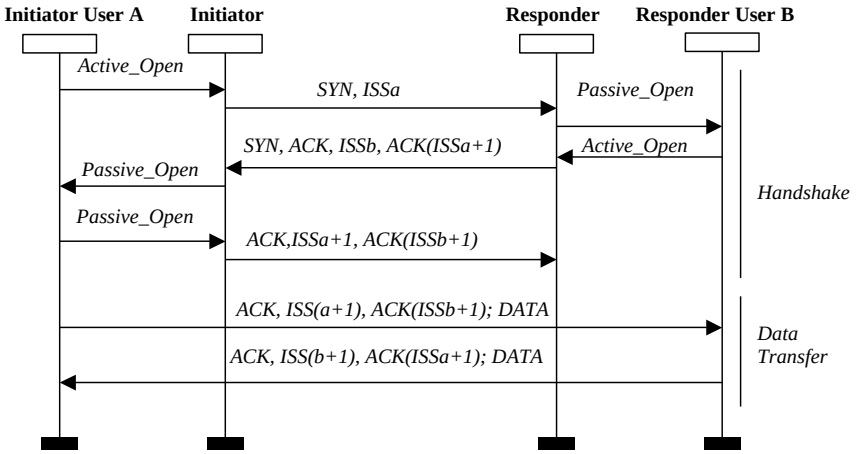


Рисунок 2.6 - ГПС встановлення віртуального каналу протоколом TCP

Встановлення віртуального каналу складається в реалізації трьох послідовних кроків процедури, званої "Handshake". Припустимо, вузол А инициализирует підключення до вузла В. Розглянемо послідовність кроків, що реалізують таке підключення.

Крок 1: Вузол А посилає свого протоколу TCP запит на активне відкриття порту - Active_Open. У відповідь протокол TCP посилає вузлу В запит SYN (Synchronize Sequence Number) і поміщає в поле Sequence Number початкове значення ISNa (Initial Sequence

Number). Надалі дане число буде служити вирішення завдання аутентифікації вузла B. Протокол TCP вузла B посилає вузлу B підтвердження про пасивному відкритті порту Passive_Open.

Крок 2: Вузол B в свою чергу посилає свого протоколу TCP також запит на активне відкриття порту Active_Open. Протокол TCP вузла B посилає протоколу TCP вузла A сигнали SYN, ACK (Acknowledgment) і ISNb (свій власний Initial Sequence Number), а також підтвердження про отримання ISNa тобто ACK (ISNa + 1).

Крок 3: На завершення "рукостискання" вузол A посилає свого протоколу TCP підтвердження про пасивному відкритті порту Passive_Open, а також за аналогією з вищесказаним посилає вузлу B сигнали ACK, ISNa + 1 та ACK (ISNb + 1).

На цьому віртуальний канал між вузлами A і B встановлено, і вони можуть обмінюватися інформацією (Data Transfer). При цьому підкреслимо, що насправді встановлено два канали зв'язку для передачі інформації від A до B і назустріч - від B до A.

Використання моделі ГПС для опису складних систем призводить до необхідності її спрощення, тому що ця модель стає наочною через її розвиненою і надто деталізованої структури. Це спрощення може бути здійснено завдяки використанню моделі ГіперГПС (ГГПС).

Головна особливість ГГПС складається в так званій гіпертекстової формі подання інформації. У цьому випадку деякі частини ГПС представлені в текстовій формі, пов'язаної з відповідними посиланнями. Таким чином, одні частини ГПС деталізуються, а інші представлені в скороченій формі.

Наприклад, опис протоколу TCP або, точніше, його частини, присвяченій створенню віртуального каналу протоколом TCP, може бути здійснено у вигляді ГГПС, як це представлено на рис. 2.7. Як бачимо, ГГПС складається з двох основних блоків R1: Handshake і R2: Data Transfer.

Для нашого прикладу можна навести ще три можливості подання ГГПС (рисунок 2.8, a, b, c). Природно, кожен скорочений блок повинен бути супроводжений коментарями, що описують його структуру. Вибір однієї з структур, наведених на рис.2.7, повністю залежить від складності модельованої структури.

Таким чином, при моделюванні мережевого протоколу ми послідовно проходимо наступні етапи:

«Граф кінцевих станів => ГПС => ГГПС».

Наступним етапом буде представлення моделі на формальній мові.

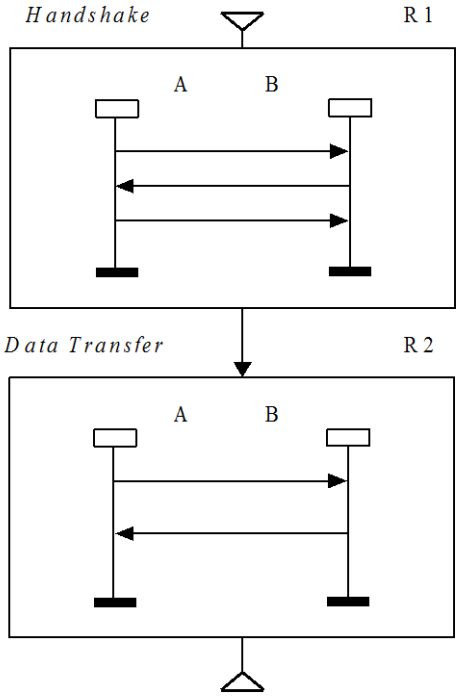


Рисунок 2.7 - ГіперГПС створення віртуального каналу протокол TCP

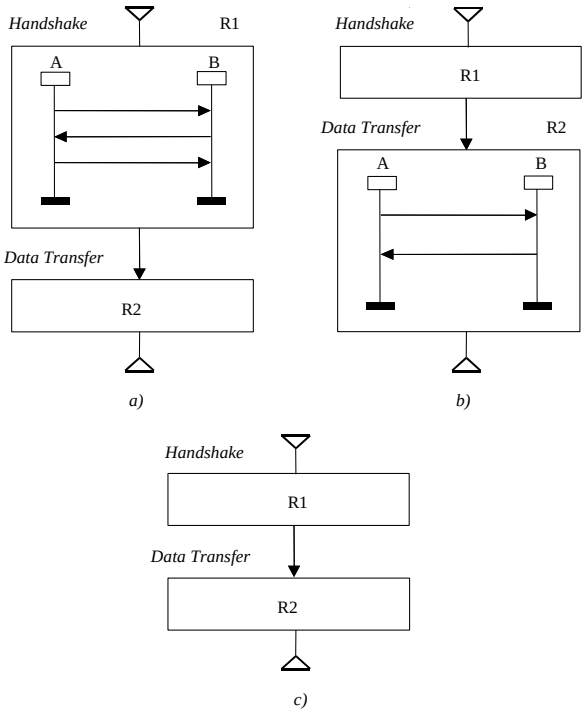


Рисунок 2.8 - Форми подання ГіперГПС

3 АВТОМАТНЕ МОДЕЛЮВАННЯ МЕРЕЖЕВИХ ПРОТОКОЛІВ

Описані вище моделі забезпечують методичну основу для розробки теорії мережевих протоколів. У ній передбачається наявність двох рівнів специфікації мережевих протоколів, а саме:

– специфікації сервісу протоколу або його абстрактного інтерфейсу,

– опису процедур, що виконуються постачальником сервісу для реалізації сервісу або його частин, тобто опис власне протоколів взаємодії сутностей, що реалізують функції постачальника.

З огляду на те, що процес взаємодії постачальника сервісу і його користувача носить дискретно-подієвий характер, моделювання поведінки постачальника може здійснюватися за допомогою використання понять абстрактних автоматів. В цьому випадку згадана вище номенклатура подій стає основою для формування вхідних і вихідних алфавітів абстрактних пристроїв, що моделюють роботу постачальника сервісу.

Іншим важливим достоїнством моделі сервісу є те, що в ній закладено підхід до визначення семантики переносяться сервісними примітивами даних через опис типів і значень параметрів. Зокрема, саме, через механізм параметрів здійснюється реалізація відображення (N + 1) -протокольних блоків даних в (N) - сервісні блоки даних.

До основних завдань теорії мережевих протоколів відносяться наступні:

– розробка засобів і методів формальної специфікації мережевих протоколів і сервісів;

– розробка метод верифікації та аналізу функціонування реалізацій мережевих протоколів;

– розробка засобів і методів автоматизації тестування реалізацій мережевих протоколів;

– автоматизація програмування реалізацій мережевих протоколів та ін.

Таким чином, зі сказаного вище ми бачимо, що тема розробки засобів і методів верифікації реалізацій мережевих протоколів автоматизації тестування реалізацій мережевих протоколів є досить актуальною.

3.1 Модель кінцевого автомата

Проаналізувавши принципи завдання специфікацій для мережевих протоколів [12], ми можемо констатувати, що вони задаються безліччю стійких станів розподіленої інформаційної системи, а також безліччю вхідних команд і вихідних реакцій системи. З іншого боку, з теорії кінцевих автоматів ми знаємо, що будь-який кінцевий автомат задається безліччю вхідних і вихідних слів, безліччю стійких станів, а також функціями переходів і виходів. Зі сказаного випливає наступне твердження: «Всякий мережевий протокол, заданий безліччю стійких станів інформаційної системи, а також безліччю вхідних команд і вихідних реакцій системи, може бути описаний автоматної моделлю Милі.»

Справедливість цього очевидна і не потребує доказів, оскільки вона випливає з збігу базових визначень як способу завдання специфікації мережевих протоколів, так і з теорії кінцевих автоматів.

Це дозволяє визначити клас мережевих протоколів, які можна задавати за допомогою автоматної моделі.

Відомо визначення моделі кінцевих автоматів, що застосовується при моделюванні поведінки протокольних сутностей. Модель кінцевого автомата (Finite-State Machine - FSM) в загальному випадку описує поведінку протокольної суті, і є безліч наступних параметрів:

$$S = \{A, X, Y, a0, f, \psi\}, \tag{3.1}$$

де A - безліч станів автомата, X - вхідний алфавіт автомата, Y - вихідний алфавіт автомата, a0 - початковий стан автомата, f - функції переходів станів автомата, ψ - функції виходів автомата.

Для представлення функцій переходів і виходів FSM використовуються, як правило, таблиці станів або діаграми станів автомата. З теорії кінцевих автоматів відомо, що серед безлічі різних типів особливо виділяються моделі автоматів типу Мілі і Мура.

Відрізняються вони за законом формування вихідних сигналів: у автомата Мілі вихідні сигнали є функція від безлічі вхідних сигналів і поточного стану автомата; у автомата Мура - вихідні сигнали залежать тільки від попереднього стану автомата. При цьому значення кожного наступного стану в обох автоматів є функція від значення попереднього стану автомата і від безлічі вхідних сигналів.

Модель Милі:
$$\begin{cases} a(t+1) = f\{a(t), x(t)\}, \\ y(t) = \psi\{a(t), x(t)\}. \end{cases} \quad (3.2)$$

Модель Мура:
$$\begin{cases} a(t+1) = f\{a(t), x(t)\}, \\ y(t) = \psi\{a(t)\}. \end{cases} \quad (3.3)$$

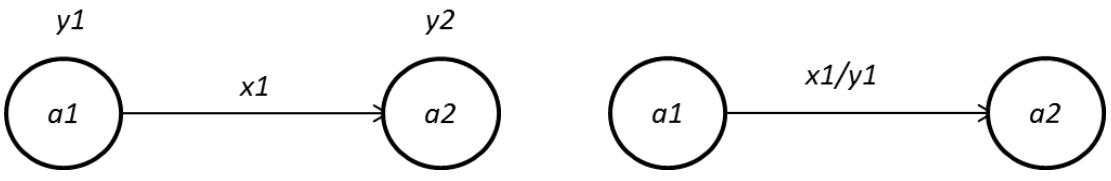
Принципово можливе подання даних автоматів або у вигляді графа, або у вигляді таблиці переходів - виходів, або в аналітичній формі. Для початку розглянемо перший спосіб представлення - графічний.

3.2 Графічна модель FSM - автомата

Опис наведених вище моделей може здійснюватися у вигляді графа, чиї вершини відповідають стійким станам автомата, а переходи з одного стійкого стану в інше здійснюються під впливом вхідних сигналів. У нашому випадку вхідні слова будемо позначати символами xi, а вихідні - символами yj (див. Систему рівнянь 3.2 і 3.3).

В автоматі Мура вхідні слова відповідають переходам автомата, а вихідні слова приписуються стійким станам, тобто вузлів графа. Відзначимо, що область використання даної моделі з метою побудови тестів мережевих протоколів є вельми обмеженою і представляє для нас чисто теоретичний інтерес. Приклад графа автомата Мура наведено на рис. 3.1, а).

З причини вищесказаного, перейдемо до аналізу моделі Мілі з точки зору її використання в цікавій для нас області діагностики. Характерною особливістю моделі Мілі є те, що вихідні слова відповідають переходам автомата з одного стійкого стану в інше. Цим же дуг приписуються відповідні вхідні слова (рисунок 3.1, б).



а) б)

Рисунок 3.1- Загальний вигляд графічної моделі: а) Мура і б) Мілі

Підкреслимо ще раз, що структура FSM-моделей автомата, розглянутих вище, говорить про те, що поведінка мережевих протоколів найбільш повно і адекватно описує модель автомата Мілі. Надалі ми будемо постійно звертатися до цього типу моделей, за відсутності іншої домовленості.

3.3 Табличне представлення FSM - автомата

Поряд з графічним методом уявлення FSM - автомата на практиці найчастіше застосовується таблична форма, оскільки вона ближча до машинного виду. Іншими словами, оскільки граф в його початковому вигляді не може бути заданий для введення в ЕОМ, на практиці він повинен бути представлений в «легкому для читання» для машини вигляді: або у вигляді матриці, або у вигляді таблиці переходів. Друга форма подання є краще, оскільки матриця має більш громіздкий вигляд і в кінцевому підсумку буде представлена у вигляді таблиці для введення в ЕОМ.

У загальному випадку таблиця переходів для автомата Мілі має такий вигляд (таблиця 3.1).

Таблиця 3.1 - Таблиця переходів автомата Мілі

a (t)	x (t)	a (t + 1)	y (t)
ai	xk	aj	yz
...	...	...	...

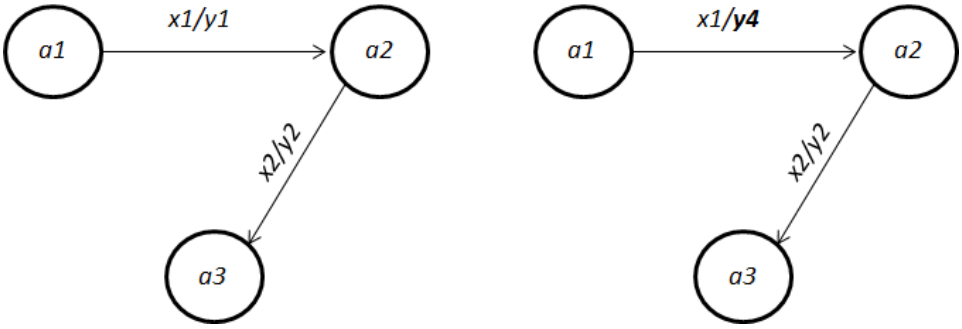
Стовпець a (t) являє поточний попередній стан автомата, a (t + 1) - поточний подальше. У стовпці x (t) записуються умови переходу зі стану a (t) в стан a (t + 1). При цьому на переході виробляється вихідний слово (вплив) y (t). Іншими словами, в даному випадку мова йде про функції переходів з одного стану в інший (f) і про функції виходів автомата (ψ) (дивись вище систему рівнянь 3.2).

Слід додати, що відсутність умови переходу в таблиці позначається символом «1», а при відсутності вихідного сигналу ставиться прочерк. Крім того, зауважимо, що в загальному випадку одночасно може існувати кілька переходів зі стану ai в стан aj під впливом різних вхідних умов. На підставі таблиці переходів автомата будується система булевих рівнянь в диз'юнктивній нормальній формі (ДНФ). Подібним способом може бути побудована таблиця переходів автомата Мура. Однак в цьому випадку система булевих рівнянь буде представлена в кон'юнктивній нормальній формі (КНФ).

3.4. Моделі помилок мережевих протоколів на базі FSM - автомата

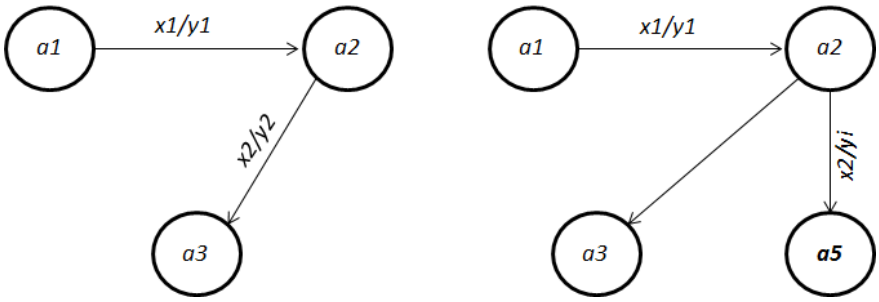
Слід зазначити, що протокол може бути представлений або у вигляді графа потоку керуючих команд контролера або у вигляді графа потоку даних між вхідними параметрами і вихідними змінними контексту. Як і зазначалося вище, ми будемо розглядати модель Милі в якості робочої моделі мережевих протоколів. На основі цієї моделі проведемо класифікацію помилок в кінцевому автоматі. Будемо виділяти такі класи помилок.

– помилка виходу автомата (рисунок 3.2, а).
– помилка переходу (рисунок 3.2, б).
– помилка стану (рисунок 3.2, в).



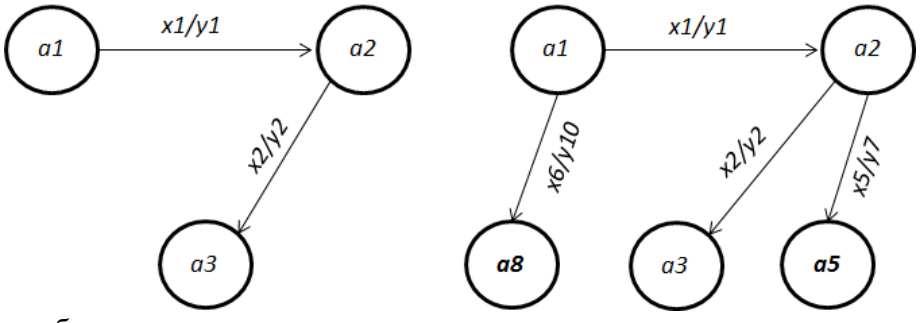
без помилки з помилкою виходу

а)



без помилки з помилкою переходу

б)



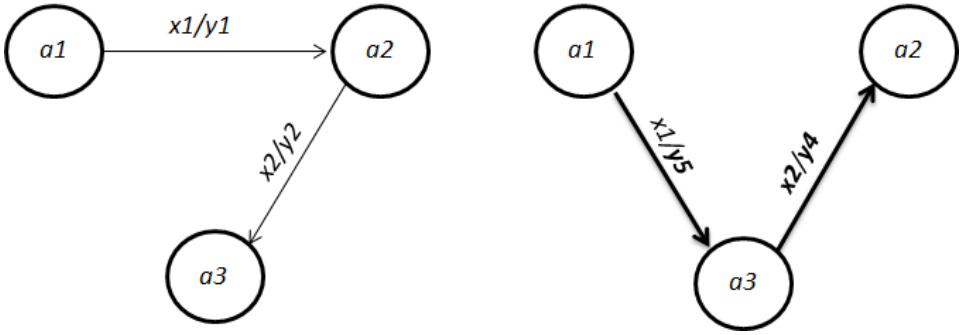
без помилки з помилкою стану

в)

Рисунок 3.2 - Класи помилок

Помилка на виході автомата має місце в разі, коли отриманий вихідний сигнал не збігається з очікуваним. При цьому перехід автомата з одного стану в інший є правильним (рисунок 3.2, а). Помилка переходу має місце в разі, коли автомат під впливом правильного (заданого) вхідного слова переходить в деякий непередбачене алгоритмом стан (рисунок 3.2, б). Нарешті, під помилкою стану ми будемо розуміти випадок, коли безліч станів автомата A = {a0, a1, ..., an} не збігається з безліччю, заданим специфікацією (рисунок 3.2, в).

Відзначимо, що на практиці в більшості випадків в автоматах можуть мати місце різні поєднання помилок, як це показано на рис. 3.3. В даному прикладі в результаті непередбачених збоїв у функціонуванні автомата сталося наступне: замість запланованих переходів з'явилися два нових. Зі стану a1 під впливом вхідного слова x1 автомат перейшов в стан a3 замість стану A2. При цьому було вироблено вихідне слово У5, а потім зі стану a3 під впливом вхідного слова x2 автомат перейшов в стан a2 і виробив при цьому незаплановане вихідне слово y4. Оскільки стан A2 є одним зі стійких станів автомата, то подальше функціонування автомата може піти хибним шляхом.



без помилки з комбінованої помилкою

Рисунок 3.3 - Приклад комбінованої помилки

3.5 Використання FSM-моделей

В даний час в теорії розподілених інформаційних мереж відбувається перехід до використання стека протоколів нового покоління TCP / IP v6.

Практичні питання, які виникли після розробки нового сімейства протоколів IPv6 це - генерація тестів відповідності специфікації протоколів, а також генерація тестів взаємодії новостворюваного програмного забезпечення (тестів) з уже існуючим. Дана проблема в загальному випадку може вирішуватися шляхом: використання загальних вхідних параметрів; використання загальних баз даних; вироблення вихідних значень в заданому

форматі; реалізації «єдиної політики», прийнятої в стеці IPv6 для всіх протоколів. При цьому слід враховувати такі особливості нового покоління протоколів як мобільність, безпека і multicast. У загальному випадку процедура побудови тестів мережевого протоколу може бути представлена, як це показано на рис. 3.4. Охарактеризуємо основні етапи цієї процедури.

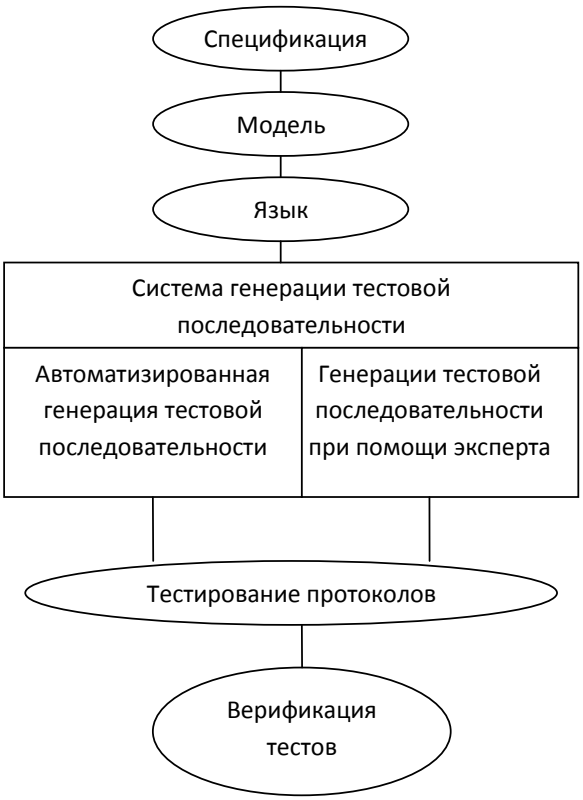


Рисунок 3.4 - Узагальнена процедура тестування мережевих протоколів

Вихідною інформацією при побудові тестів є специфікація, яка називається RFC (Request for Comments). У зв'язку з цим цілком логічним є рішення прийняти RFC в якості первісної інформації для функціонування системи побудови тестів. Зауважимо, що при цьому специфікація задається в описовому не формалізована вигляді. Тому потрібен перехід на етап подання RFC у вигляді деякої формальної моделі.

Наступним етапом є опис моделі на деякій формальній мові. Принципи створення тестових послідовностей для перевірки протоколів будуть розглянуті докладніше нижче. І, нарешті, після цього йде етап тестування протоколів з подальшою перевіркою тестів на їх повноту і адекватність.

Покажемо на конкретних прикладах реалізацію перших двох етапів процедури тестування мережевих протоколів.

3.6 Модель протоколу BGP

Протокол граничного шлюзу Border Gateway Protocol (BGP) зазнав кілька змін з моменту виходу його першої версії BGP-1 в 1989 році. Повсюдне впровадження BGP-4 почалося в 1993 році. Це перша з версій BGP, в якій з'явилися можливості агрегації (об'єднання), що дозволило реалізувати безкласову міждоменну маршрутизацію (classless interdomain routing - CIDR), і забезпечити підтримку Суперсети.

Протокол BGP не пред'являє ніяких вимог до топології мережі. Принцип його дії передбачає, що маршрутизація всередині автономної системи виконується за допомогою внутрішніх протоколів маршрутизації, або, як їх ще називають, інтра- протоколів (наприклад, Interior Gateway Protocol - IGP). Протокол BGP активно використовує інформацію про маршрутах до певного пункту призначення, що дозволяє уникнути утворення петель маршрутизації між доменами.

Уявімо даний протокол спочатку в спрощеній описовій формі, опускаючи детальний опис RFC, з яким можна ознайомитися на відповідних сторінках в Інтернеті.

Виходячи з опису протоколу BGP, ми можемо уявити модель кінцевих станів при переговорах по протоколу BOP між сусідніми вузлами в наступному вигляді (рисунок 3.5).

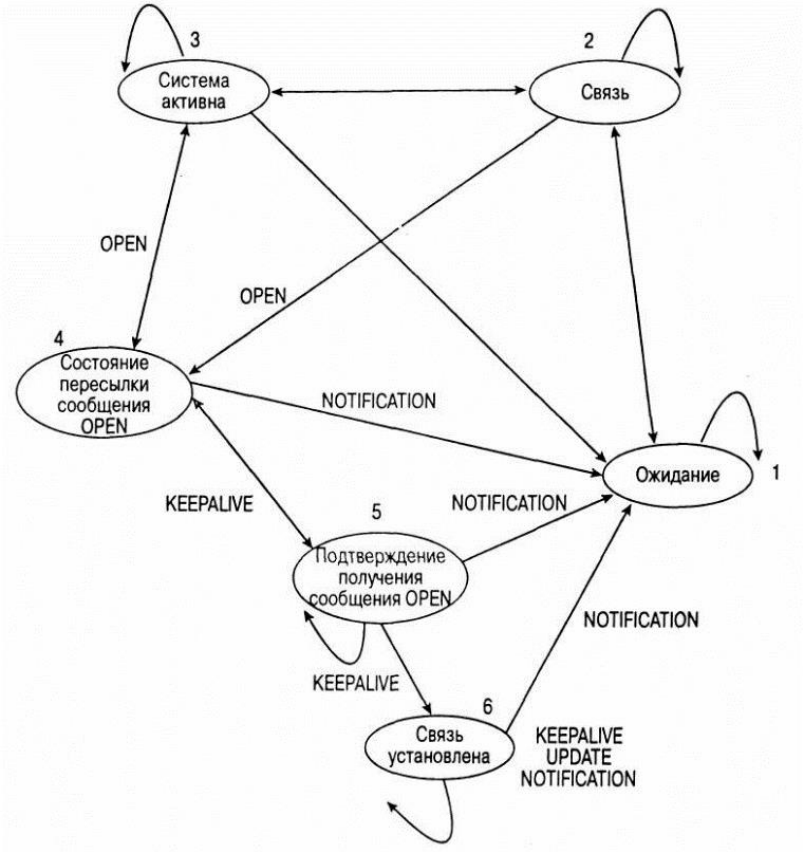


Рисунок 3.5- Приклад FSM-моделі фрагмента BGP-протоколу

Перейшовши від змістовного графа, представленого на рис. 3.5 до більш простому і зручному для обробки увазі, ми отримуємо граф, наведений на рис. 3.6.

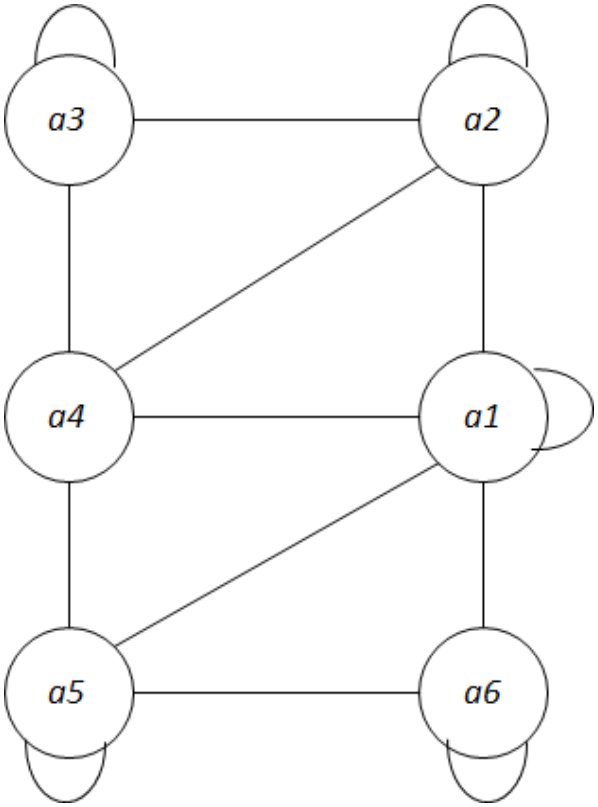


Рисунок 3.6 - Спрощений граф фрагмента BGP-протоколу

Як ми бачимо, вершини a1, a2, a3, a5, a6 є «чекають» тобто такими, один з виходів яких зводиться на вершину його породжує.

Далі розглянемо побудову та використання графічної моделі на прикладі одного з базових протоколів - протоколу TCP.

3.7 Графічна модель протоколу TCP

Перш за все, відзначимо, що протокол TCP (Transmission Control Protocol) - протокол транспортного рівня моделі OSI є одним з найстаріших мережевих протоколів (70-ті роки XX століття). Він забезпечує надійну доставку повідомлень по мережі завдяки використанню механізму квітірованія, тобто на кожну посилку повинна прийти квитанція, яка свідчить про те, що повідомлення доставлено. Відсутність такої квитанції говорить про необхідність повторної посилки повідомлення.

Як відомо, в даний час відбувається поступовий перехід від використання стека протоколів TCP / IPv4 (нині діючої четвертої версії) до стека TCP / IPv6. Надалі ми будемо вести мову в основному про версії TCP / IP v4, за відсутності іншої домовленості.

Аналізуючи протокол TCPv4, ми бачимо, що дії TCP можна розглядати як відгуки на події. Події, що відбуваються можна розбити на три категорії - призначені для користувача виклики, доставка сегментів і тайм-аути. У багатьох випадках необхідна у відповідь на подію обробка залежить від стану соединенія. Наведемо перелік подій:

– призначені для користувача виклики: OPEN, SEND, RECEIVE, CLOSE, ABORT, STATUS;

– доставка сегментів: SEGMENT ARRIVES;

– тайм-аути: TIMEOUT, RETRANSMISSION TIMEOUT, TIME-WAIT TIMEOUT.

Модель призначеного для користувача інтерфейсу TCP базується на негайному поверненні з користувальницьких викликів і можливо затриманих відгуках на виклик за допомогою події або псевдо переривання. Повідомлення про помилки наводяться в формі символьних рядків. Наприклад, при виклику команди, яка звертається до неіснуючого з'єднанню, буде повертатися повідомлення "error: connection not open" (помилка: з'єднання не відкрито).

Природним варіантом процесу обробки вхідних сегментів є спочатку перевірка коректності порядкового номера (тобто його "попадання" у вікно прийому), розміщення в черзі і подальша обробка в порядку зростання номерів. Коли сегмент перекривається з отриманим раніше сегментом, він реконструюється таким чином, щоб в сегменті містилися тільки нові дані (поля заголовків змінюються відповідно до нового вмістом). Відзначимо, що якщо зміна стану TCP не вказано, це говорить про збереження колишнього стану.

На рис. 3.7 ми наводимо формально описане в RFC графічне представлення структури протоколу TCP v4. Очевидно, що процес моделювання деякого мережевого протоколу буде адекватним, якщо ми врахуємо всі тонкощі процесів їм викликаються. У цьому плані не зайвим буде уточнити особливості виконання основних подій провокованих протоколом.

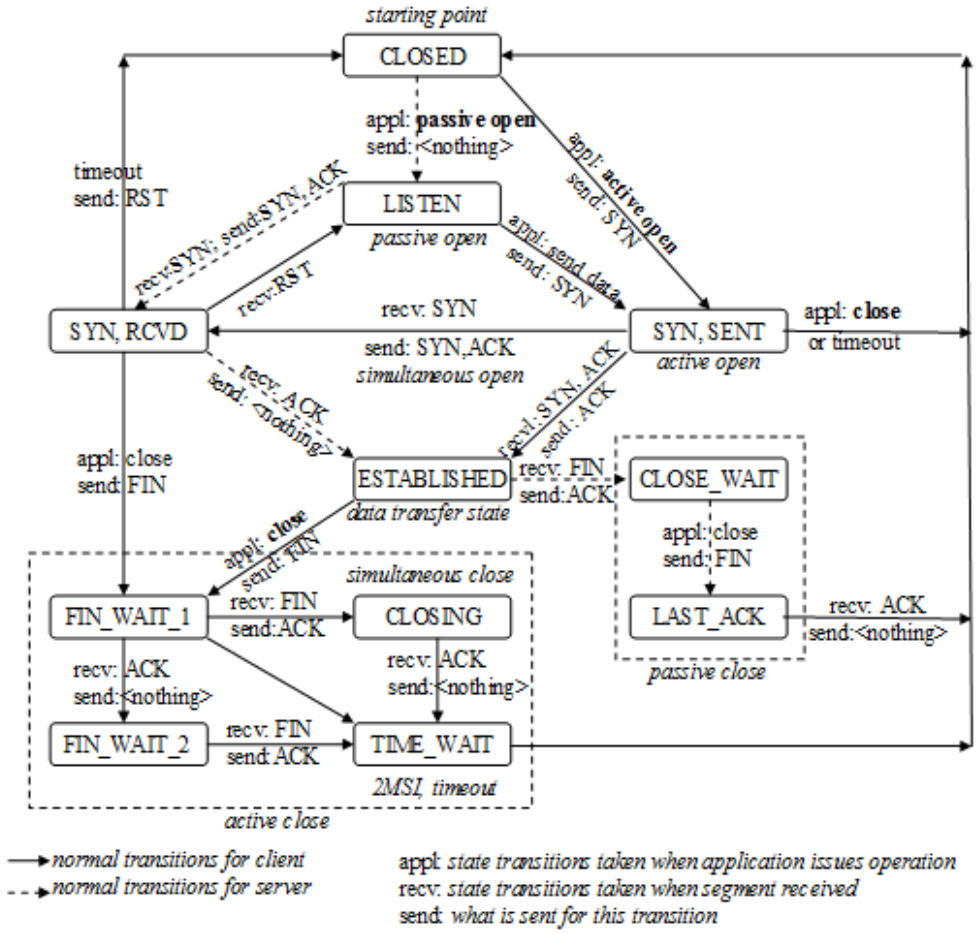


Рисунок 3.7- Графічна форма завдання TCP-протоколу

3.8 Табличне представлення протоколу TCP

Покажемо на прикладі протоколу TCP побудова таблиці переходів для автомата заданого на рис. 3.7 в графічній формі. Перш за все, уявімо протокол TCP у вигляді зазначеного графа кінцевого автомата. Тут під «зазначеним» ми розуміємо граф, у якого відзначені всі стійкі стану автомата. При цьому початковий стан згідно з формулою 1) ми відзначаємо як a0 (рисунок 3.8).

Зауважимо, що вершина a0 має двоякий сенс: вона є одночасно і початковою, і кінцевою вершиною. Однак, щоб не ускладнювати наочність прикладу ми будемо використовувати одну вершину, маючи на увазі, що наш кінцевий автомат починає своє функціонування з цієї вершини і в ній же і закінчується його робота. Таким чином, після завершення кожного робочого циклу кінцевий автомат, а значить і протокол, описаний даною моделлю, приходить в початковий стан і знову готовий до роботи з початкового стану.

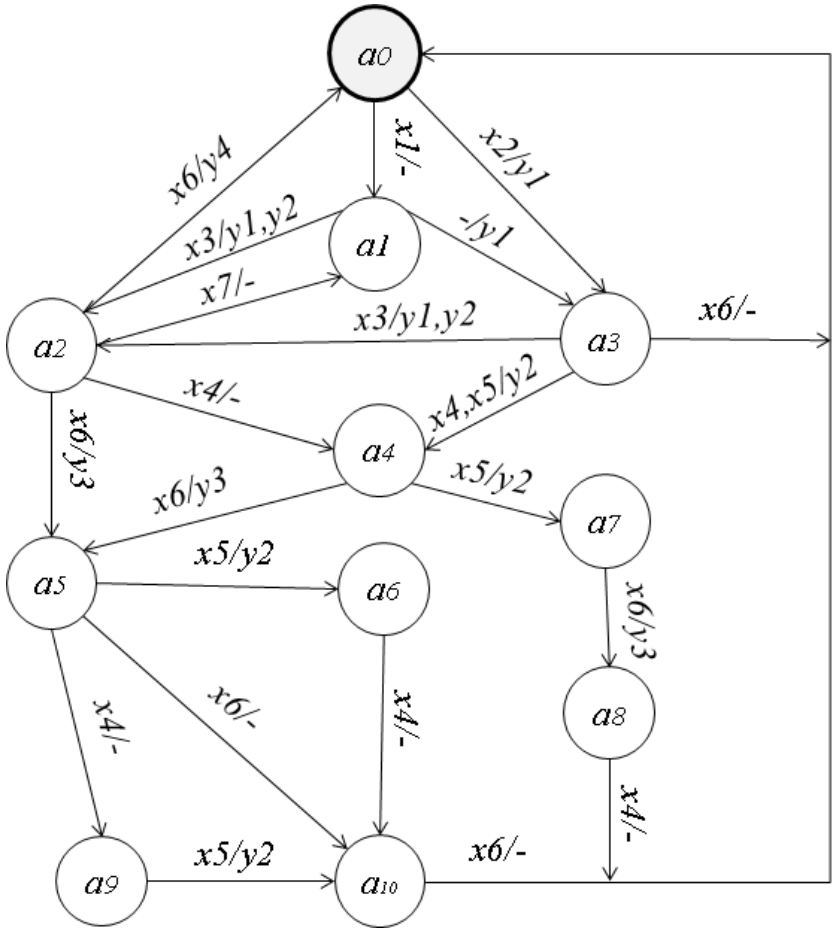


Рисунок 3.8- Зазначений граф кінцевого автомата для протоколу TCP

На рис. 3.8 ми використовували позначення, зміст яких розкрито в таблиці 3.2.

Таблиця 3.2 - Значення символів на зазначеному графі TCP - протоколу

Безліч станів {A}		Безліч вхідних команд {X}		Безліч вихідних реакцій {Y}	
a0	CLOSED	x1	Passive Open	y1	SYN
a1	LISTEN	x2	Active Open	y2	ACK
a2	SYN_RCVD	x3	SYN	y3	FIN
a3	SYN_SENT	x4	ACK	y4	RST
a4	ESTABLISHED	x5	FIN		
a5	FIN_WAIT_1	x6	close or timeout		
a6	CLOSING	x7	RST		
a7	CLOSE_WAIT				
a8	LAST_ACK				
a9	FIN_WAIT_1				
a10	TIME_WAIT				

4 АНАЛІТИЧНА ФОРМА ПОДАННЯ ПРОТОКОЛУ TCP

Певний інтерес представляє можливість використання апаратної форми реалізації протоколу. Для такої реалізації будемо використовувати аналітичну форму подання протоколу, яку розглянемо на прикладі моделювання протоколу TCP.

Надалі для однаковості термінології замість термінів безліч «вхідних команд» і «вихідних реакцій» ми будемо використовувати терміни «безліч вхідних (і вихідних) слів» відповідно.

На основі зазначеного графа протоколу TCP (рисунок 3.8) ми будуємо зворотну таблицю переходів (таблиця 4.1).

Таблиця 4.1 - Зворотній таблиця переходів автомата

№	a (t + 1)	x (t)	a (t)	y (t)
1.	a10	x6	a5	-
2.	a10	x4	a6	-
3.	a10	x5	a9	y2
4.	a9	x4	a5	-
5.	a8	x6	a7	y3
6.	a7	x5	a4	y2
7.	a6	x5	a5	y2
8.	a5	x6	a2	y3
9.	a5	x6	a4	y3
10.	a4	x4	a2	-
11.	a4	x4, x5	a3	y2
12.	a3	x2	a0	y1
13.	a3	1	a1	y1
14.	a2	x2	a1	y1, y2
15.	a2	x3	a3	y1, y2
16.	a1	x7	a2	-
17.	a1	x1	a0	-
18.	a0	x6	a2	y4
19.	a0	x6	a3	-
20.	a0	x4	a8	-
21.	a0	x6	a10	-

Нагадаємо, що стан a (t) є попереднім по відношенню до стану a (t + 1). При цьому зміст кожної окремої рядки таблиці показує умови переходу в автоматі під впливом вхідного слова і яке вихідне слово виробляється на цьому переході.

Повертаючись протоколу TCP, відзначимо, що в кожен момент часу він знаходиться в деякому стійкому стані, з якого він може перейти в інший стійкий стан під впливом певного

вхідного слова, виробляючи на переході вихідне слово (керуючу команду). Ця послідовність переходів однозначно задається специфікацією протоколу. Також з таблиці видно, що кожна попередня рядок «зачіпається» з попередньої з точки зору нерозривності послідовності станів автомата. Тобто, наведена автоматна модель є адекватною заданої специфікації.

Перейдемо до написання аналітичної форми подання автомата, що описує функціонування протоколу TCP. Для цього скористаємося класичною теорією синтезу кінцевих автоматів [34].

Використовуючи інформацію з таблиці 4.1, запишемо систему булевих рівнянь, заданих в загальному вигляді формули (3.2) (автомат Мілі).

Зауважимо, що наш автомат має одинадцять стійких станів (a0, ... A10) і десять переходів між ними, система рівнянь для завдання функції переходів автомата приймає наступний вигляд:

$$\begin{cases} a0 = a10x6 \vee A8x4 \vee a3x6 \vee a2x6 = x6\,(a2 \vee a3 \vee A10) \vee A8, \\ a1 = a2x7 \vee a0x1, \\ a2 = a1x2 \vee A3x3, \\ a3 = a0x2 \vee A1, \\ a4 = a2x4 \vee A3x4x5 = x4\,(a2 \vee A3x5), \\ a5 = a2x6 \vee A4x6 = x6\,(a2 \vee A4), \\ a6 = a5x5, \\ a7 = a4x5, \\ a8 = a7x6, \\ a9 = a5x4, \\ a10 = a5x6 \vee A6x4 \vee a9x5. \end{cases} \quad (4.1)$$

Подібним же чином запишемо систему функцій виходів автомата:

$$\begin{cases} y1 = a0x2 \vee a1 \vee a1x2 \vee a3x3, \\ y2 = a9x5 \vee a4x5 \vee a5x5 \vee A3x4x5 \vee a1x2 \vee A3x3, \\ y3 = a2x6 \vee a4x6 = x6\,(a2 \vee A4), \\ y4 = a2x6. \end{cases} \quad (4.2)$$

Зауважимо, що отримані системи булевих функцій 4.1 і 4.2 в подальшому при необхідності можуть бути мінімізовані шляхом суперпозиції повторюваних фрагментів рівнянь.

Подальша апаратна реалізація протоколу шляхом синтезу пристрою на основі отриманих булевих рівнянь не є складним і відома з класичної теорії синтезу кінцевих автоматів.

4.1 Формування таблиці несправностей FSM протоколу TCP

Раніше ми розглядали типи тестів і відмов мережевих протоколів. Далі для простоти викладу і приведення теорії тестування протоколів ми будемо використовувати поняття «несправність» мережевого протоколу. При цьому під «несправністю» мережевого протоколу ми будемо розуміти порушення правильного перебігу його виконання, тобто його відмова.

Одним з базових понять теорії тестування комп'ютерних пристроїв є поняття «таблиці несправностей» - ТН. Поширимо це поняття на теорію тестування мережевих протоколів мережевих протоколів.

В теорії булевої алгебри існує поняття булевої похідної як по одній, так і по декількох змінним. Це поняття широко використовується в теорії тестування цифрових схем оскільки булева похідна від функції f (x1, x2, ..., xi, ..., xn) за змінної xi визначає умови активізації шляху в схемі від входу xi до виходу f (x) . Булевой похідною від функції f (x) = f (x1, x2, ..., xn) по xi називається функція:

$$df(x) / dxi = f(x1, x2, ..., xi, ..., xn) \oplus f(x1, x2, ..., \bar{xi}, ..., xn), \qquad (4.3)$$

де $\oplus$ - сума по модулю 2.

Перейшовши до двійкового значення змінної xi ми можемо обчислити булеву похідну за такою формулою:

$$df(x) / dxi = f(x1, x2, ..., 0, ..., xn) \oplus f(x1, x2, ..., 1, ..., xn). \qquad (4.4)$$

Іншими словами, булева похідна визначає значення логічних змінних x1, ..., xn (крім xi), при яких зміна стану xi призводить до зміни значення функції f (x).

Зі сказаного вище випливає принцип побудови тесту для виявлення несправності по змінної xi. Ставлячи значення булевої похідної рівним одиниці (df (x) / dxi = 1), ми тим самим визначаємо умови при яких зміна даної змінної буде змінювати значення функції на протилежне заданому. Як ми бачимо, тут використовується той же принцип активізації

шляхів у перевіреній схемі, який лежить в основі класичного D-алгоритму з тією лише різницею, що в нашому випадку активізація шляхів проводиться за допомогою трансформації булевої функції заданої схеми, а не шляхом обробки кубічних покриттів схеми як у випадку D-алгоритму [119]. Тобто тестові набори для несправності xi ≡ 0 (xi ≡ 1) визначають значення логічних змінних x1, ..., xn (крім xi), при яких задана несправність буде транспортуватися до виходу.

Сказане можна поширити і на внутрішні змінні. Тест для несправностей z ≡ 0 (z ≡ 1) внутрішньої лінії схеми визначають значення логічних змінних, при яких: $z \times df(x) / dz = 1$; $(\bar{Z} \times df(x) / dz = 1)$.

Нижче ми розглянемо додаток викладеної теорії на практиці для тестування автоматної моделі мережевого протоколу.

4.2 Використання методу булевих похідних при тестуванні автоматної моделі протоколу TCP

Використання принципу активізації шляхів у перевіреній схемі передбачає побудову тесту шляхом послідовного визначення перевіряючих вхідних наборів для окремих несправностей. Тест утворюється шляхом об'єднання обраних наборів. Алгоритм вибору перевіряючих наборів можна представити таким чином:

– записуємо булеву функцію f (x), в якій присутня змінна i, яку треба перевірити на наявність несправностейxi ≡ 1 і xi ≡ 0;

– обчислюємо булеву похідну df (x) / dxi і наводимо отриманий вираз до диз'юнктивній формі (ДФ);

– вибираємо один з умов (наприклад, T), отриманої ДФ;

– несправність w ≡ 0 перевіряється на впливі, при якому значення змінних x1, ..., xn забезпечують умова wT = 1;

– несправність w ≡ 1 перевіряється на впливі, при якому значення змінних x1, ..., xn забезпечують умова $\bar{w}T = 1$.

Таким чином, активізуючи шлях від входу автомата до його виходу ми перевіряємо наявність несправності типу w ≡ 0 або w ≡ 1 на керуючі шляху.

Розглянемо метод активізації шляхів на прикладі автомата, що реалізує протокол TCP. Із системи рівнянь (2.5) ми бачимо, що значення на виходах в1, в2, в3, у4 є функція від

вхідних змінних x і станів автомата a.Розглянемо процес отримання тесту для функції виходу y1. Провівши мінімізацію функції отримуємо такий вираз:

$$y1 = a0x2 \setminus /a1 \setminus /a1x2 \setminus / a3x3 = a0x2 \setminus /a1 \setminus / A3x3.$$

Як бачимо, функція y1 залежить від вхідних змінниххx2 і x3, а також від станів автомата a0, a1, a3.

Знайдемо булеву похідну df (x) / dx2 дляy1:

$$dy1 / Dx2 = (a0x2 \setminus /a1 \setminus / A3x3) \oplus (a0\ \overline{x}2 \setminus /a1 \setminus / A3x3) = (a01 \setminus /a1 \setminus / A3x3) \oplus (a0\ 0 \setminus /a1 \setminus / A3x3) =$$
$$=(a0 \setminus /a1 \setminus / A3x3) \oplus (a1 \setminus / A3x3) =. . . = a0\overline{a}1\overline{a}3 \setminus / \overline{a}1\overline{x}3.$$

Для того, щоб визначити умови активізації шляху від входу x2 до виходу y1 (x2 → в1) прирівняємо отриману похідну до «1» і визначимо значення всіх змінних, що входять в дану похідну.

$$dy1 / Dx2 = a0\overline{a}1\overline{a}3 \setminus / \overline{a}1\overline{x}3 = 1.$$

Множимо обидва терма рівняння спочатку на $\overline{x}2$ (щоб перевірити на несправність типу x2≡ 1), а потім - на x2 (для перевірки на несправність x2≡0):

$$\overline{x}2a0\overline{a}1\overline{a}3 \setminus / \overline{x}2\overline{a}1\overline{x}3;\ x2a0\overline{a}1\overline{a}3 \setminus / x2\overline{a}1\overline{x}3.$$

Результат представимо в кубічної форми (табл.. 4.2):

Таблиця 4.2 - Вектори тесту для перевірки входуx2

несправність	x2≡ 1	x2≡ 0
Перевірка по входу x2	$\overline{x}2a0\overline{a}1\overline{a}3 \setminus / \overline{x}2\overline{a}1\overline{x}3$	$x2a0\overline{a}1\overline{a}3 \setminus / x2\overline{a}1\overline{x}3$
вектори тесту	x2 x3 a0 a1 a3 в1 1 U 1 0 0 1 0 U 1 0 0 0 DU 1 0 0 D	

Аналіз отриманих векторів тесту говорить про те, що нами знайдено умову, за якої зміна значення змінної x2 призводить до зміни значення функціїy1, причому напрямок зміни однаково. Надалі для простоти ми будемо позначати буквою D зміна значення з «1» на «0», а зміна з «0» на «1» - буквою N. Тим самим ми підкреслюємо ще раз схожість даного підходу до теорії активізації шляхів за допомогою D -алгоритми Рота (Roth).

Для наочності покажемо цифрову схему, що реалізує дану функцію за умови активізації шляху від входу x2 (рисунок 4.1). При цьому змінних3 приймає значення U з безлічі {0, 1} (тобто x3 не впливає на умови активізації заданого шляху).

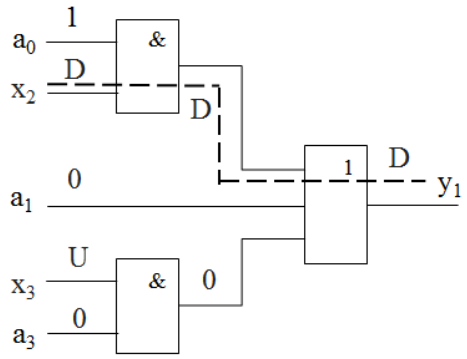


Рисунок 4.1 - Схема, що реалізує функцію y1, з активним шляхом x2 → в1

Для активізації виходу автомата в1по входу x3 пріравняем булеву похідну dy1 / dx3 до 1 і визначимо значення інших змінних (крім x3) створюють умови для активізації шляху x3 → в1. Опускаючи проміжні обчислення отримуємо:

dy1 / Dx3 = (a0x2 \ /a1\ / A31)⊕(a0 $\overline{x2}$\ /a1\ / A30)= ... = a0$\overline{a}$1$\overline{x}$2 \ / $\overline{a0}\overline{a}$1a3x2.

Виходячи з умови, що обидва терма рівняння повинні бути рівні 1, отримуємо тест, в якому символ D вказує на активізацію (змінних і функції), а під U, як це було зазначено вище, розуміється довільне значення змінної з множини {0, 1} :

x2 x3a0 a1a3 y1
0 1 U 0 1 1
0 0 U 0 1 0
0 DU 0 1 D

Подібним чиномвиявляємо умови активізації шляхів від кожного входу xi до кожного виходу yj. В результаті отримуємо безліч тестових наборів ti, складових результуючий тест T (табл. 4.3) на підставі яких будується таблиця несправностей.

Таблиця 4.3 - Структура тесту T

T	x1	x2	x3	x4	x5	x6	a0	a1	a2	a3	a4	a9	в1	y2	y3	y4
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
t1	U	D	U	U	U	U	1	0	U	0	U	U	D	-	-	-
t2	U	0	D	U	U	U	U	0	U	1	U	U	D	-	-	-
t3	U	D	0	0	0	U	U	1	U	U	U	U	-	D	-	-
t4	U	0	D	0	0	U	U	U	U	1	U	U	-	D	-	-
t5	U	0	0	D	1	U	U	U	U	1	0	0	-	D	-	-
t6	U	0	0	1	D	U	U	U	U	1	0	0	-	D	-	-

Продовження табл. 4.3

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
t7	U	0	0	1	D	U	U	U	U	0	1	0	-	D	-	-
t8	U	0	0	0	D	U	U	U	U	0	0	1	-	D	-	-
t9	U	U	U	U	U	D	U	U	1	U	1	U	-	-	D	-
t10	U	U	U	U	U	D	U	U	1	U	1	U	-	-	-	D

Дамо інтерпретацію отриманих результатів в додатку до протоколу TCP. Покажемо структурну схему автомата протоколу (рис. 4.2).

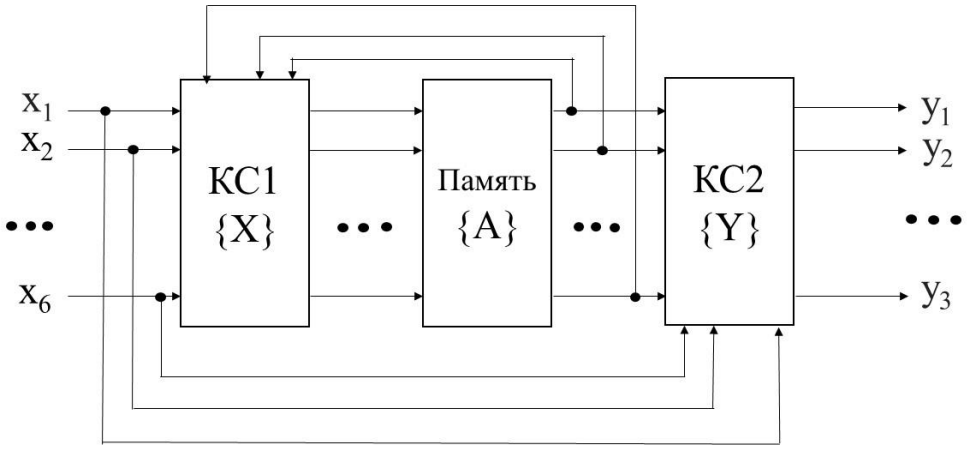


Рисунок 4.2 - Структурна схема автомата протоколу TCP

Уявімо структуру тесту T, наведену в таблиці 4.3, з урахуванням інформації з таблиці 4.1. Отримана таблиця 4.4 містить тестову послідовність для перевірки автомата протоколу TCP.

Таблиця 4.4 - Тестова послідовність автомата протоколу TCP

Вектори тесту T	{X}						{A}						{Y}			
	Passive Open	Active Open	SYN	ACK	FIN	close or timeout	CLOSED	LISTEN	SYN_RCVD	SYN_SENT	ESTABLISHED	FIN_WAIT_1	SYN	ACK	FIN	RST
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
t1	-	D	-	-	-	-	1	0	-	0	-	-	D	-	-	-
t2	-	0	D	-	-	-	-	0	-	1	-	-	D	-	-	-
t3	-	D	0	0	0	-	-	1	-	-	-	-	-	D	-	-
t4	-	0	D	0	0	-	-	-	-	1	-	-	-	D	-	-
t5	-	0	0	D	1	-	-	-	-	1	0	0	-	D	-	-
t6	-	0	0	1	D	-	-	-	-	1	0	0	-	D	-	-
t7	-	0	0	1	D	-	-	-	-	0	1	0	-	D	-	-

Продовження табл. 4.4

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
t8	-	0	0	0	D	-	-	-	-	0	0	1	-	D	-	-
t9	-	-	-	-	-	D	-	-	1	-	1	-	-	-	D	-
t10	-	-	-	-	-	D	-	-	1	-	1	-	-	-	-	D

На практиці в двійковому вигляді наведена в таблиці 4.4 тестова послідовність містить двадцять векторів оскільки, як зазначалося вище, символD введений для спрощення і наочності. Він задає в кожному векторі ti двійкові значення 1 і 0 на двох часових тактах. Символ «-» свідчить про те, що відповідна складова у формуванні даного вектора не бере.

4.3 Моделювання протоколів методом мереж Петрі

4.3.1 Базові поняття теорії мереж Петрі

Ще один з підходів до моделювання мережевих протоколів будується на основі використання теорії мереж Петрі (МП). Розглянемо деякі вихідні поняття, властиві МП. Мережі Петрі використовуються для моделювання асинхронних систем, що функціонують як сукупність паралельних взаємодіючих процесів. Аналіз мереж Петрі дозволяє отримати інформацію про структуру та динамічному поведінці модельованої системи [3, 8, 9]. Причинно-наслідковий зв'язок подій в асинхронних системах задається безліччю відносин виду "умови-події".

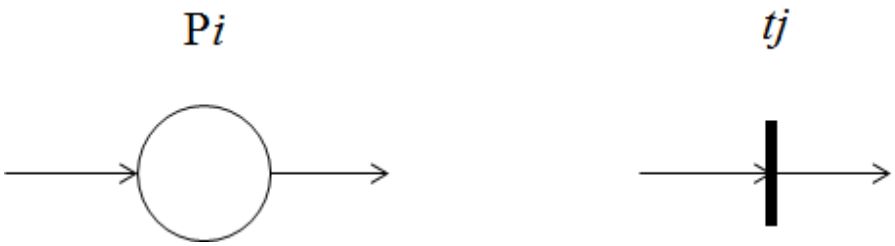
Побудова моделей систем у вигляді мереж Петрі полягає в наступному:

а) моделюються процеси описуються безліччю подій (дій) і умов визначають можливість настання цих подій, а також причинно-наслідкових відносин, що встановлюються на безлічі пар "події-умови";

б) визначаються події - дії, послідовність виконання яких управляється станами системи.

Стани системи задаються безліччю умов, які формуються у вигляді предикатів. Кількісно умови характеризуються величиною, яка виражається числами натурального ряду. Умови, в залежності від значень їх кількісних характеристик, можуть виконуватися чи ні. Виконання умов забезпечує можливість реалізації подій. Умови, з фактом виконання яких пов'язується можливість реалізації подій, називаються передумови. Реалізація події забезпечує можливість виконання інших умов, які перебувають з передумовою в причинно-наслідкового зв'язку.

У мережах Петрі умови - це позиції, а події - переходи. Відповідно до цього граф мережі Петрі є дводольним орієнтованим мультиграфом. Зображення позиції і переходу на графі показано на рис. 4.3.



а) б)

Рисунок 4.3 - Зображення позиції -а) і переходу б)

Орієнтовані дуги можуть з'єднувати тільки позиції і переходи в прямому і зворотному напрямку (властивість дводольних). Мережа Петрі є мультиграфом, так як допускається кратність дуг між позиціями і переходами (вершинами графа). В графах мережі Петрі кількісні характеристики умов (числа натурального ряду) прийнято зображати числом міток у відповідних позиціях.

4.3.2 Приклад моделі Петрі мережевого протоколу TCP

У нашому випадку при моделюванні мережевих протоколів мережу Петрі повинна бути живою, тобто вона не повинна породжувати такі маркування, для яких інші маркування недосяжні. Крім того, для опису мережевих процесів слід застосовувати тільки безпечні мережі Петрі, тобто Такі мережі, в яких при будь-маркування в кожній позиції не може бути більше однієї позначки. В якості моделі, яка описує процеси, що протікають в мережевих структурах, в даній роботі пропонується використовувати правильні МП (безпечні і живі).

Для ілюстрації можливості застосування методу мереж Петрі для моделювання мережевих протоколів нами був узятий приклад розглянутого вище протоколу TCP. При цьому використовувалася програмне середовище Visual Petri. Її перевагами є простота і доступність графічного інтерфейсу, широкий інструментарій для графічного представлення мережі, можливість промоделювати її стану.

Використовуючи програмне середовище Visual Petri 1.2 ми отримуємо модель Петрі [104] для мережевого протоколу TCP, специфікація якого була приведена вище. Модель Петрі протоколу TCP показана на рис. 4.4. Наступним кроком є аналіз МП-моделі, який

виконується за допомогою Analysis Module Manager, State Spaces Analysis Module - по трьом параметрам Bounded (перевірка обмеженості), Safe (перевірка на безпеку), Deadlock (відсутність безвихідного становища або глухого кута).

Повний аналіз мережі Петрі можна провести за допомогою вивчення і аналізу її поведінкових властивостей: досяжність, обмеженість, активність, оборотність і досяжність тупикової розмітки.

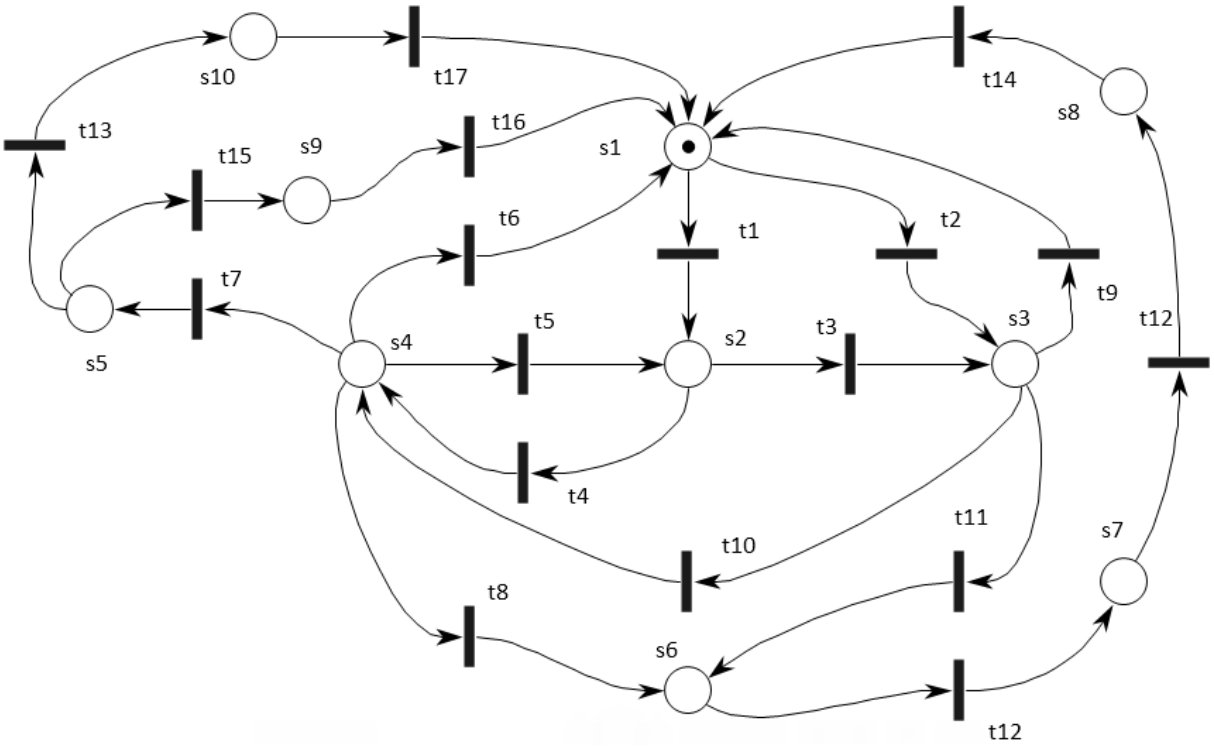


Рисунок 4.4 - Модель Петрі протоколу TCP

Отримана модель може бути використана в подальшому для аналізу протоколу, тестування його конформності, а також для програмної реалізації протоколу з метою його подальшого дослідження.

Разом з тим проведені дослідження говорять про те, що на практиці використання методу МП для цілей моделювання протоколів наштовхується на ряд труднощів, головна з яких полягає в нетривіальністю отримання тестів з МП-моделі.

Крім розглянутого вище методу моделювання та аналіз мережевих транспортних протоколів також може проводитися за допомогою розфарбованих МП [3]. Розфарбована мережу Петрі (РМП) - це графоорієнтірованний мову для проектування, опису, імітації та контролю, розподілених і паралельних систем, до яких відносяться також і мережі передачі даних. Графічними примітивами показується протягом процесу, а конструкціями спеціальної мови імітується необхідна обробка даних.

Аналіз апарату розфарбованих мереж Петрі показує недоцільність використання даного апарату для моделювання мережевих протоколів з метою подальшої побудови системи генерації тестових послідовностей. Базуючись на засадах угоди про специфікації протокольних сервісів розглянуті різні можливості моделювання мережевих протоколів. В результаті приходимо до висновку, що в основі моделювання повинна лежати специфікація протоколу, яка легко бути подана в графічній формі.

В якості методів моделювання протоколів можна використовувати як теорію кінцевих автоматів, так і метод мереж Петрі. Аналіз пропонованих моделей показав їх адекватність і можливість використання в системі генерації тестових послідовностей для перевірки мережевих протоколів на їх відповідність специфікації. Обидва підходи можуть бути використані для вирішення даної задачі.

Разом з тим, проведені дослідження однозначно свідчать на користь використання методу автоматних моделей для побудови адекватної моделі мережевого протоколу, яка в подальшому може бути використана в автоматизованій системі тестування мережевих протоколів. На користь такого висновку також говорить те факт, що на базі теорії FSM-автоматів ми отримуємо адекватні моделі помилок мережевих протоколів.

При цьому особливої уваги заслуговує той факт, що використання базової теорії синтезу цифрових автоматів дає можливість застосувати на практиці такий математичний апарат з алгебри логіки як приватні булеві похідні для моделювання конкретних несправностей автомата і побудови тестових послідовностей для їх виявлення і діагностування за допомогою сформованих таблиць несправностей мережевих протоколів.

Також показано, що використання теорії автоматного моделювання дає можливість використання розроблених моделей для перевірки конформності програмної реалізації мережевих протоколів їх задекларованої специфікації, що є перспективним для модифікації життєвого циклу протоколу.

Серед основних переваг використання автоматних моделей відзначимо:

– наявність глибокого опрацювання класичної теорії синтезу цифрових автоматів, що дозволяє використовувати багатий апарат даної теорії в процесі перевірки працездатності і діагностування несправностей мережевих протоколів;

– простотуа переходу від заданої специфікації протоколу до його автоматною моделі, оскільки завдання специфікації, і автоматною моделі Мілі мають схожі принципи подання;

Сильною стороною автоматного моделювання є використання теорії алгебри логіки (булевої алгебри), а саме приватних булевих похідних для формування таблиць несправностей протоколів, що дозволяють виявляти і діагностувати можливі несправності (помилки) протоколів.

Великий інтерес представляє також можливість апаратної реалізації моделі розробляються мережевих протоколів поряд з використанням їх програмної моделі.

Пропонований автоматний підхід до моделювання протоколів дозволяє легко перейти від аналітичної форми завдання моделі до апаратної реалізації. Як відомо, апаратна реалізація автоматів має ряд переваг, таких як висока швидкодія і надійність функціонування.

5 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

5.1 Аналіз потенційних небезпечних і шкідливих виробничих чинників проектованого об’єкту, що мають вплив на персонал

У даному дипломному проекті розробляється математична модель в апараті мереж Петрі за допомогою програмного забезпечення. Розроблене програмне забезпечення орієнтоване на роботу з персональним комп'ютером. Експлуатовані для вирішення внутрішньовиробничих завдань ПЕОМ типу IBM PC мають наступні характеристики:

споживана потужність	220 Вт;
робоча напруга	220 В;
напруга джерел живлення	+12 В; - 12 В; +5 В;
робоча частота	50 Гц.

Виходячи з приведених характеристик, вочевидь, що для людини існує небезпека поразки електричним струмом, унаслідок недбалого поводження з комп'ютером і порушення правил експлуатації, залишення частин ПЕОМ, що знаходяться під напругою, відкритими або знятих для ремонту вузлів.

Відповідно до [56] до легкої фізичної роботи відносяться всі види діяльності, виконувані сидячи і ті, що не потребують фізичної напруги. Робота користувача ПК відноситься до категорії 1а.

При роботі на ПЕОМ користувач піддається ряду потенційних небезпек. Унаслідок недотримання правил техніки безпеки при роботі з машиною(невиконання огляду відкритих частин ПЕОМ, що знаходяться під напругою або знятих для ремонту вузлів) для користувача існує небезпека поразки електричним струмом.

Джерелами підвищеної небезпеки можуть служити наступні елементи:

- розподільний щит;
- джерела живлення;
- блоки ПЕОМ і друку, що знаходяться в ремонті.

Ще одна проблема полягає у тому, що спектр випромінювання комп'ютерного монітора включає рентгенівську, ультрафіолетову і інфрачервону області, а також широкий діапазон хвиль інших частот. Небезпека рентгенівського проміння мала, оскільки цей вид випромінювання поглинається речовиною екрану. Проте велику увагу слід приділяти біологічним ефектам низькочастотних електромагнітних полів(аж до порушення ДНК).

Відповідно до [57], при обслуговуванні ПЕОМ мають місце фізичні і психофізичні небезпечні, а також шкідливі виробничі чинники:

– підвищене значення напруги в електричному ланцюзі, замикання якої може відбутися через тіло людини;

– підвищений рівень статичної електрики;

– підвищений рівень електромагнітних випромінювань;

– підвищена або знижена температура повітря робочої зони;

– підвищений або знижений рух повітря;

– підвищена або знижена вологість повітря;

– відсутність або недостатність природного світла;

– підвищена пульсація світлового потоку;

– недостатня освітленість робочого місця;

– підвищений рівень шуму на робочому місці;

– розумове перенапруження;

– емоційні навантаження;

– монотонність праці.

5.2 Заходи щодо техніки безпеки

Основним небезпечним чинником при роботі з ЕОМ є небезпека поразки людини електричним струмом, яка посилюється тим, що органи чуття людини не можуть на відстані знайти наявності електричної напруги на устаткуванні.

Проходячи через тіло людини, електричний струм чинить на нього складну дію, що є сукупністю термічної(нагрів тканин і біологічних середовищ), електролітичної(розкладання крові і плазми) і біологічної(роздратування і збудження нервових волокон і інших органів тканин організму) дій.

Тяжкість поразки людини електричним струмом залежить від цілого ряду чинників:

– значення сили струму;

– електричного опору тіла людини і тривалості протікання через нього струму;

– роду і частоти струму;

– індивідуальних властивостей людини і навколишнього середовища.

Розроблений дипломний проект передбачає наступні технічні способи і засоби, що застерігають людину від ураження електричним струмом [8]:

– заземлення електроустановок;

– занулення;

– захисне відключення;

– електричне розділення ятерів;

– використовування малої напруги;

– ізоляція частин, що проводять струм;

– огорожа електроустановок.

Занулення зменшує напругу дотику і обмежує година, протягом якого людина, ткнувшись до корпусу, може потрапити під дію напруги.

Струм однофазного короткого замикання визначається по наближеній формулі:

$$I_K = \frac{\text{Uф}}{\text{Zп} + \frac{\text{Zт}}{3}} , \qquad (5.1)$$

де Uф - номінальна фазна напруга мережі, В;

Zп - повний опір петлі, створене фазними і нульовими дротами, Ом;

Zт - повний опір струму короткого замикання на корпус, Ом.

Згідно таблиці 4 [59]: Zт /3 = 0,1 Ом.

Для провідників і жил кабелю для розрахунку повного опору петлі використовуємо формулу (5.2.) :

$$\text{Zп} = \sqrt{\text{Rп}^2 + \text{Xп}^2} , \qquad (5.2)$$

де Rп = Rф + Ro - сумарний активний опір фазного Rф і нульового Ro дротів, Ом;

Xп - індуктивний опір паяння дротів, Ом.

Перетин 1 км мідного дроту S = 2.5 мм, тоді згідно таблицям 5 і 6 [59], має такий опір:

Xп = 0,11 Ом;

Rф = 7,55 Ом;

Ro = 7,55 Ом.

Отже, Rп = 7,55 + 7,55 = 15,1 Ом.

Тоді по формулі (5.2) знаходимо повний опір петлі :

$$Zп = \sqrt{15,1^2 + 0,11^2} \approx 15,1 \text{ (Ом)}.$$

Струм однофазного короткого замикання рівний:

$$Iк = \frac{220}{15,1 + 0,1} = 14,47 \text{ (А)}.$$

Дія плавкої вставки на ПЕОМ забезпечується, якщо виконується співвідношення:

$$Iк \geq k * Iн, \quad (5.3)$$

де Iн - номінальний струм спрацьовування плавкої вставки, А;

k - коефіцієнт кратності нелінійного струму Iн, А.

Коефіцієнт кратності нелінійного струму Iн розраховується по формулі (5.4.) :

$$Iн = P / U, \quad (5.4)$$

де P = 220 Вт - споживана потужність;

U = 220 В - робоча напруга;

k = 3 А - для плавких вставок.

Отже, Iн = 220 / 220 = 1 А.

Підставивши значення у вираз (5.3), одержимо:

$$14,47 > 3*1.$$

Таким чином, доведено, що апарат забезпечить спрацьовування(і захист) при підвищенні номінального струму.

4.3 Заходи, що забезпечують виробничу санітарію і гігієну праці

Вимоги до виробничих приміщень встановлюються [60], СНіП, відповідними ГОСТами і ОСТамі з урахуванням небезпечних і шкідливих чинників, що утворюються в процесі експлуатації електроустаткування.

Підвищення працездатності людини і збереження її здоров'я забезпечується стабільними метеорологічними умовами. Мікроклімат виробничих приміщень [61] визначається діючими на організм людини поєднаннями температури, вологості і швидкості руху повітря, а також температури навколишніх поверхонь. Значне коливання параметрів мікроклімату приводити до порушення систем кровообігу, нервової і потовидільної, що може викликати підвищення або пониження температури тіла, слабкість, запаморочення і навіть непритомність.

Відповідно до [62] встановлюють оптимальну і допустиму температуру, відносну вологість і швидкість руху повітря в робочій зоні . За відсутності надмірного тепла, вологи, шкідливих речовин в приміщенні досить природної вентиляції.

У приміщенні для виконання робіт операторського типу(категорія 1а), пов'язаних з нервово-емоційною напругою, проектом передбачається дотримання наступних нормованих величин параметрів мікроклімату (табл. 5.1).

Таблиця 5.1 - Санітарні норми мікроклімату робочої зони приміщень для робіт категорії 1а.

Пора року	Температура, С	Відносна вологість, %	Швидкість руху повітря, м/с
Холодна	22…24	40…60	0,1
Тепло	23…25	40…60	0,1

У приміщенні, де знаходиться ПЕОМ, повітрообмін реалізується за допомогою природної організованої вентиляції(з пристроєм вентиляційних каналів в перекриттях будівлі і вертикальних шахт) й установленого промислового кондиціонера фірми Mitsubishi, який дозволяє вирішити переважну більшість завдань по створінню та підтримці необхідних параметрів повітряного середовища. Цей метод забезпечує приток потрібної кількості свіжого повітря, визначеного в СНіП (30 $м^3$ в годину на одного працівника).

Шум на виробництві має шкідливу дію на організм людини. Стомлення операторів через шум збільшує число помилок при роботі, призводить до виникнення травм. Для оператора ПЕОМ джерелом шуму є робота принтера. Щоб усунути це джерело шуму, використовують наступні методи. При покупці принтера слід вибирати найбільш шумозахисні матричні принтери або з великою швидкістю роботи(струменеві, лазерні). Рекомендується принтер поміщати в найбільш віддалене місце від персоналу, або застосувати звукоізоляцію та звукопоглинання(під принтер підкладають демпфуючі підкладки з пористих звукопоглинальних матеріалів з листів тонкої повсті, поролону, пеноплену).

При роботі на ПЕОМ, проектом передбачені наступні методи захисту від електромагнітного випромінювання : обмеження часом, відстанню, властивостями екрану.

Обмеження годині роботи на ПЕОМ складає 3,5-4,5 години. Захист відстанню передбачає розміщення монітора на відстані 0,4-0,5 м від оператора. Передбачений монітор 20" TFT, Samsung 2043BW відповідає вимогам стандарту [63].

Стандарт [63] пред’являє жорсткі вимоги в таких областях: ергономіка(фізична, візуальна і зручність користування), енергія, випромінювання(електричних і магнітних полів), навколишнє середовище і екологія, а також пожежна та електрична безпека, які відповідають всім вимогам [64].

Для зниження стомлюваності та підвищення продуктивності праці обслуговуючого персоналу в колірній композиції інтер'єру приміщень для ПЕОМ дипломним проектом пропонується використовувати спокійні колірні поєднання і покриття, що не дають відблисків.

У проекті передбачається використання сумісного освітлення. У світлий час доби приміщення освітлюватиметься через віконні отвори, в решту часу використовуватиметься штучне освітлення.

Як штучне освітлення необхідно використовувати штучне робоче загальне освітлення. Для загального освітлення необхідно використовувати люмінесцентні лампи. Вони володіють наступними перевагами: високою світловою віддачею, тривалим терміном служби, хоча мають і недоліки: високу пульсацію світлового потоку.

При експлуатації ПЕОМ виробляється зорова робота. Відповідно до [65] ця робота відноситься до розряду 5а. При цьому нормоване освітлення на робочому місці(Ен) при загальному освітленні рівна 200 лк.

Приміщення завдовжки 12 м, шириною 10 м, заввишки 4 м обладнується світильниками типу ЛП02П, оснащеними лампами типу ЛБ зі світловим потоком 3120 лм кожна.

Виконаємо розрахунок кількості світильників в робочому приміщенні завдовжки a=12 м, шириною b=10 м, заввишки з=4 м, використовуючи формулу (5.5) розрахунку штучного освітлення при горизонтальній робочій поверхні методом світлового потоку:

$$n = (E \cdot S \cdot Z \cdot k)/(F \cdot U \cdot M), \qquad (5.5)$$

де F - світловий потік = 3120 лм;

E - максимально допустима освітленість робочих поверхонь = 200 лк;

S - площа підлоги = 120 м2;

Z - поправочний коефіцієнт світильника = 1,2;

k - коефіцієнт запасу, що враховує зниження освітленості в процесі експлуатації світильників = 1,5;

n - кількість світильників;

U - коефіцієнт використовування освітлювальної установки = 0,6;

M - кількість ламп у світильнику =2.

З формули (5.5) виразимо n (5.6) і визначимо кількість світильників для даного приміщення:

$$n = (E \cdot S \cdot Z \cdot k)/(F \cdot U \cdot M), \quad (5.6)$$

Отже, n = (200·120·1,2·1,5)/(3120·0,6·2) = 12 .

Виходячи з цього, рекомендується використовувати 12 світильників. Світильники слід розміщувати рядами, бажано паралельно стіні з вікнами. Схема розташування світильників зображена на рис. 5.1.

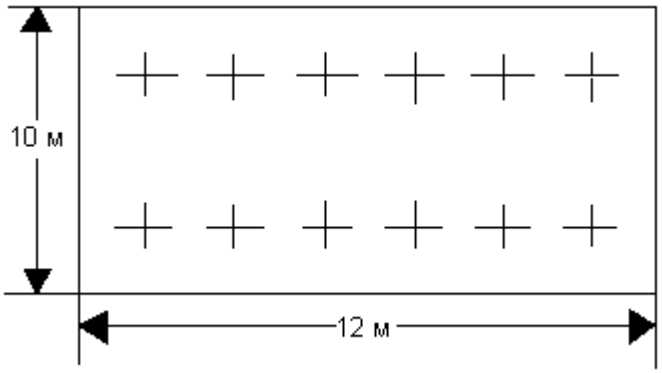


Рисунок 5.1 - Схема розташування світильників

4.4 Рекомендації по пожежній безпеці

Пожежі в приміщеннях, де встановлена обчислювальна техніка, представляють небезпеку для життя людини. Пожежі також пов’язані як з матеріальними втратами, так і з відмовою засобів обчислювальної техніки, що у свою чергу спричиняє за собою порушення ходу технологічного процесу.

Пожежа може виникнути при наявності горючої речовини та внесення джерела запалювання в горюче середовище. Пальними матеріалами в приміщеннях, де розташовані ПЕОМ, є:

– поліамід - матеріал корпусу мікросхеми, горюча речовина, температура самозаймання аерогелю 420 З ;

– полівінілхлорид - ізоляційний матеріал, горюча речовина, температура запалювання 335 З, температура самозаймання 530 З, кількість енергії, що виділяється при згоранні - 18000 - 20700 кДж/кг;

– стеклотекстоліт ДЦ - матеріал друкарських плат, важкозаймистий матеріал, показник горючості 1.74, не схильний до температурного самозаймання;

– пластика кабельний №489 - матеріал ізоляції кабелю, горючий матеріал, показник горючості більш 2.1;

– деревина - будівельний і обробний матеріал, матеріал з якого виготовлені меблі, горючий матеріал, показник горючості більше 2.1, теплота згорання 18731 - 20853 кДж/кг, температура запалювання 399 З, схильна до самозаймання [66].

Згідно [67] приміщення відносяться до категорії В(пожежовибухонебезпечним) і згідно правилам побудови електроустановок простір усередині приміщення відноситься до вогненебезпечної зони класу П - IIа (зони, розташовані в приміщеннях, в яких зберігаються тверді горючі речовини).

Потенційними джерелами запалення при роботі ПЕОМ є:

– іскри при замиканні і розмиканні ланцюгів;

– іскри і дуги коротких замикань;

– перегріви від тривалого перевантаження і наявності перехідного опору.

Продуктами згорання, що виділяються при пожежі, є : оксид вуглецю, сірчистий газ, оксид азоту, синильна кислота, акропеін, фосген, хлор та ін. При горінні пластмас, окрім звичайних продуктів згорання, виділяються різні продукти термічного розкладання: хлорангідрідні кислоти, формальдегіди, хлористий водень, фосген, синильна кислота, аміак, фенол, ацетон, стирол та ін., що шкідливо впливають на організм людини.

Для захисту персоналу від дії небезпечних і шкідливих чинників пожежі проектом передбачається застосування промислового протигаза з коробкою марки В(жовта).

Пожежна безпека об’єктів народного господарства регламентується [68] і забезпечується системами запобігання пожежам і протипожежному захисту. Для успішного гасіння пожеж вирішальне значення має швидке виявлення пожежі і своєчасний виклик пожежних підрозділів до місця пожежі.

Зменшити горюче навантаження не представляється можливим, тому проектом передбачається застосувати наступні способи і їх комбінації для запобігання утворенню(внесення) джерел запалення :

– застосування устаткування, що задовольняє вимогам електростатичної безпеки;

– застосування в конструкції швидкодіючих засобів захисного відключення можливих джерел запалення;

– виключення можливості появи іскрового заряду статичної електрики в горючому середовищі з енергією, рівної і вище мінімальної енергії запалення;

– підтримка температури нагріву поверхні машин, механізмів, устаткування, пристроїв, речовин і матеріалів, які можуть увійти до контакту з пальним середовищем, нижче гранично допустимої, становить 80% якнайменшої температури самозаймання пального.

– заміна небезпечних технологічних операцій більш безпечними;

– ізольоване розташування небезпечних технологічних установок і устаткування;

– зменшення кількості пальних і вибухонебезпечних речовин, що знаходяться у виробничих приміщеннях;

– запобігання можливості утворення пальних сумішей на лінії, вентиляційних системах і ін.;

– механізація, автоматизація та справність(потокова) виробництва;

– суворе дотримання стандартів і точне виконання встановленого технологічного режиму;

– запобігання можливості появи в небезпечних місцях джерел запалення;

– запобігання розповсюдженню пожеж і вибухів;

– використовування устаткування і пристроїв, при роботі яких не виникає джерел запалення;

– виконання вимог сумісного зберігання речовин і матеріалів;

– наявність громовідводу;

– організація автоматичного контролю параметрів, що визначають джерела запалення;

– ліквідація можливості самозаймання речовин і матеріалів .

– Для запобігання пожежі в обчислювальних центрах проектом пропонується виконання наступних вимог :

– електроживлення ЕОМ повинно мати автоматичне блокування відключення електроенергії на випадок зупинки системи охолоджування і кондиціонування;

– система вентиляції обчислювальних центрів повинна бути обладнана блокуючими пристроями, що забезпечують її відключення на випадок пожежі;

– робочі місця повинні бути оснащені пожежними щитами, сигналізацією, засобами для сповіщення про пожежну небезпеку (телефонами), медичними аптечками для надання першої медичної допомоги, розробленим планом евакуації.

Для зниження пожежної небезпеки в приміщеннях використовуються первинні засоби гасіння пожеж, а також система автоматичної пожежної сигналізації, яка дозволяє знайти початкову стадію загоряння, швидко і точно оповістити службу пожежної охорони про час і місце виникнення пожежі.

Відповідно до [69] приміщення категорії В підлягають устаткуванню системами автоматичної пожежної сигналізації. Проектом передбачається застосування датчика типу ІДФ - 1(димовий фотоелектричний датчик), оскільки специфікою пожеж обчислювальної техніки і радіоапаратури є, в першу чергу, виділення диму, а потім - підвищення температури.

При виникненні пожежі в робочому приміщенні обслуговуючий персонал зобов'язаний негайно вжити заходи по ліквідації пожежі. Для ліквідації пожежі використовують вогнегасники (хімічно-пінні, пінні для повітря ОП-5, ОП-6, ОП-9, вуглекислотні ОУ-5), пісок, пожежний інвентар(сокири, ломи, багри, шерстяну або азбестову ковдри) [70]. Як засіб індивідуального захисту проектом передбачається використовування промислового протигаза з маскою, фільтруючої коробки В.

В якості організаційно-технічних заходів рекомендується проводити навчання робочого персоналу правилам пожежної безпеки.

5.5 Охорона навколишнього природного середовища

5.5.1 Загальні дані з охорони навколишнього природного середовища

Діяльність за темою магістерської роботи, а саме розробці автоматизованої системи моделювання рівноважного складу впливає на навколишнє природнє середовище і регламентується нормами діючого законодавства [71 - 26].

Основним екологічним аспектом в процесі діяльності за даними спеціальностями є процеси впливу на атмосферне повітря та процеси поводження з відходами, які утворюються, збираються, розміщуються, передаються на віддалення (знешкодження), утилізацію, тощо в ІТ галузі.

В процесі створення/розробки програми на робочому місці виникають процеси поводження з відходами ІТ галузі. Нижче надано перелік відходів, що утворюються в процесі роботи:

– Відпрацьовані люмінесцентні лампи - I клас небезпеки
– Змінні носії інформації - IV клас небезпеки
– Відпрацьовані вогнегасники - IV клас небезпеки
– Макулатура - IV клас небезпеки
– Відпрацьовані фільтрувальні засоби індивід. захисту (респіратори, протигази) - IV клас небезпеки
– Побутові відходи - IV клас небезпеки

5.5.2 Вимоги до збору, пакування та розміщення відходів ІТ галузі

Наводяться вимоги зберігання виявлених за своєю роботою відходів відповідно до вимог Державних санітарних правил і норм [77].

Відходи в міру їх накопичення збирають у тару, відповідну класу небезпеки, з дотриманням правил безпеки, після чого доставляють до місця часового зберігання відходів відповідно до затвердженої схеми їх розміщення. Зазначені для зберігання відходів місця чи об'єкти повинні використовуватися лише для заявлених відходів.

Способи часового зберігання відходів визначаються видом, агрегатним станом і класом небезпеки відходів:

- відходи I класу небезпеки зберігаються в герметичній тарі (сталеві бочки, контейнери). У міру наповнення тару з відходами закривають герметично сталевий кришкою;

- відходи IV класу небезпеки можуть зберігатися відкрито на промисловому майданчику у вигляді конусоподібної купи, звідки їх автонавантажувачем перевантажують у самоскид і доставляють на місце утилізації або захоронення;

Особливий контроль наділяється збору і зберіганню відпрацьованих ртутьвмісних ламп (енергоощадних) як відходам I класу небезпеки, що збираються і обов’язково передаються на утилізацію підприємствам, що мають ліцензію на поводження з такими небезпечними відходами.

Всі відходи, що утворюються в процесі діяльності/роботи, підлягають обліку.

Побутові та будівельні відходи вивозяться на полігон твердих побутових відходів міста, також відповідно до договору з комунальним дорожньо-експлуатаційним управлінням.

Особи, винні в порушенні встановленого порядку поводження з відходами (порушення правил обліку відходів, самовільне складування і видалення відходів, передача відходів в інші підприємства/організації з порушенням встановлених правил), згідно законодавства несуть дисциплінарну, адміністративну або кримінальну відповідальність.

З метою визначення та прогнозування впливу відходів на навколишнє середовище, своєчасного виявлення негативних наслідків, їх запобігання відповідно до Закону України «Про відходи» повинен здійснюватися моніторинг місць утворення, зберігання, і видалення відходів.

У розділі «Охорона праці та безпека в надзвичайних ситуаціях» виконано аналіз потенційних небезпек при роботі із засобами обчислювальної техніки і механізмами, розроблені заходи щодо техніки безпеки, заходи, які забезпечують виробничу санітарію і гігієну праці, розраховане штучне освітлення, виконані рекомендації по пожежній безпеці.

ВИСНОВКИ

У магістерськіій роботі вирішена актуальна науково-практична задача тестування і верифікації створюваних мережевих протоколів шляхом їх моделювання з метою перевірки конформності задекларованої специфікації, а також завдання формування тестових послідовностей в системі тестування мережевих протоколів з використанням апарату булевих похідних.

Аналіз життєвого циклу мережевих протоколів показав, що існуючі сьогодні методи розробки і створення мережевих протоколів є досить трудомістким процесом, що не гарантує створення протоколів адекватно відображають вимоги задекларованої специфікації. Це завдання особливо актуальне зараз при переході на використання стека протоколів нового покоління TCP / IP версії 6.

Необхідність включення в життєвий протоколу таких етапів, як створення еталонної моделі, аналіз конформности протоколу, а також етапу моделювання можливих помилок створюваного протоколу привела до необхідності модифікації життєвого циклу. Розроблена модифікація значно спрощує процес створення протоколу, а також підвищує надійність і безвідмовність функціонування всієї системи телекомунікацій в цілому.

Використання апарату булевих похідних для побудови тесту, який перевіряє правильність функціонування протоколу дозволило формалізувати і спростити процес генерації тестових послідовностей.

У процесі вирішення поставлених завдань отримано такі наукові результати:

Проаналізовано та доопрацьована типова структура життєвого циклу мережевого протоколу. Введено нові етапи циклу, які передбачають створення еталонної моделі протоколу з метою перевірки конформності протоколу задекларованої специфікації ще на початкових стадіях створення протоколу.

Обґрунтовано використання класичної теорії кінцевих автоматів для моделювання мережевих протоколів. В уваги береться те, що кожне наступне стан протоколу є функція від попереднього стану і вхідного слова. Зроблено обґрунтований висновок про доцільність використання моделі Мілі для моделювання мережевих протоколів.

Опрацьовані аналітичні вирази для опису перехідних процесів в автоматної моделі мережі під впливом функціонування мережевого протоколу в залежності від стану мережі і вхідних впливів, що стало можливим завдяки використанню теорії кінцевих автоматів при моделюванні.

Розроблено метод побудови автоматизованої системи генерації тестових послідовностей для тестування мережевих протоколів в тому числі і протоколів нового покоління стека TCP/ IPv6. При цьому акцент зроблений на верифікації відповідності створюваних протоколів їх специфікації. Також крім перевірки правильності реалізації протоколів проаналізована можливість побудови так званих «словників несправностей« помилок »протоколу для подальшої його діагностики та ідентифікації помилок.

Отримав подальший розвиток метод використання булевих похідних однієї змінної для генерації тестових послідовностей спрямованих на діагностування несправностей (помилок) в роботі мережевих протоколів. За основу взято принцип активізації шляхів від кожного входу протоколу до виходів в об'єкті, що перевіряється.

Розроблено модель об'єкта діагностування і методика проведення діагностичного експерименту, які використовуються на підприємстві для перевірки мережевих протоколів на їх конформність задекларованої специфікації, а також в навчальному процесі університету.

Розглянуті методи можуть бути застосовані для підвищення якості обслуговування і достовірності передачі інформації в комп'ютерних мережах.

Модифікована структура життєвого циклу мережевих протоколів з нововведеними етапами циклу дозволяє підвищити ефективність проектування нових протоколів в тому числі і протоколів нового покоління TCP / IPv.6.

Метод моделювання мережевих протоколів з використанням автоматної моделі Мілі на базі класичної теорії кінцевих автоматів дає можливість виявляти і ідентифікувати помилки в програмній реалізації як нових протоколів, так і вже існуючих.

На основі автоматного методу моделювання розроблено принципи побудови автоматизованої системи тестування мережевих протоколів.

У розділі «Охорона праці та безпека в надзвичайних ситуаціях» виконано аналіз потенційних небезпек при роботі із засобами обчислювальної техніки і механізмами, розроблені заходи щодо техніки безпеки, заходи, які забезпечують виробничу санітарію і гігієну праці, розраховане штучне освітлення, виконані рекомендації по пожежній безпеці.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Schaff A. Test of the new generation internet protocols IPv6 [Текст] / A. Schaff. V. Nemchenko // Журн. Радіоелектроніка та інформатика. - 2001. - №12. - С. 87-89.

2. Джарратано Д. Експертні системи: принципи розробки та програмування [Текст] / Д. Джарратано, Г. Райлі // пров. з англ. - М.: Видавничий дім «Вільямс», 2006., - 1152 с.

3. Chaly D.Ju. An extensible coloured petri net model of a tansport protocol for packet switched networks [Текст] / D. Ju. Chaly, VA Sokolov, Ed. V. Malyshkin // Proc. of PaCT'2003. Lecture Notes in Computer Science. - Springer-Verlag.- 2003.

4. Petrenko A. On fault coverage of tests for finite state specifications [Текст] / A. Petrenko, G. Bochmann, M. Yao // Computer Networks and ISDN Systems.- # 29 (1) .- 1996. - P. 81 106.

5. Grabowski J. On the Design of the newTesting LanguageTTCN-3 [Текст] / J. Grabowski, H. Ural, RL Probert, G. von Bochmann // Testing of Communicating Systems, Kluwer, 2000. - P. 161-176.

6. Немченко В. П. Автоматне моделювання в системі діагностування мережевих протоколів [Текст] / В. П. Немченко, А. С. Ізотов // Журн. Інформаційно-керуючі системи на залізничному транспорті.- 2012 .- №4 (додаток). - С. 51-52.

7. Частиков А. П. Розробка експертних систем. Середа CLIPS. Розробка [Текст] / А. П. Частиков, Т. А. Гаврилова, Д. Л. Белов // Спб .: БХВ-Петербург, 2003. - 608 с.

8. Лескін А.А. Мережі Петрі в моделюванні та управлінні [Текст] / А. А. Лескін, П. А. Мальцев, А. М. Спиридонов. -Л .: Наука, 1989. - 133 с.

9. Дворянський Л. В. Імітаційне моделювання та верифікація вкладених мереж Петрі з використанням CPNTools [Текст] / Дворянський Л. В., Ломазова І. А. // Модел. і аналіз інформ. систем. - 2012. - Т. 19.- № 5. - С. 115-130

10. Utting, M. Practical Model-Based Testing: A Tools Approach [Текст] / M. Utting B. Legeard. - Morgan Kaufmann: San Francisco, 2007. - 112-143.

11. Пітерсон Дж. Теорія мереж Петрі і моделювання систем [Текст] / Пітерсон Дж. - М.: світ.- 1984.- 264 с.

12. Основи діагностики мережі [Електронний ресурс] / "LAN / ЖУРНАЛ МЕРЕЖЕВИХ РІШЕНЬ" Грудень 1998. - Режим доступу: http://www.prolan.ru/company/article/magazine/lan121998.html. - тисяча дев'ятсот дев'яносто вісім.

13. Dumas Joseph S., Janice CA Practical Guide to Usability Testing [Текст] / UK: Intellect, 1999. - 256 p.

14. Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests [Текст]. Jeffrey Rubin. - Wiley, 1994 - 386 p.

15. Ізотов А. С. Аналіз конформности мережевих протоколів / А. С. Ізотов, В. П. Немченко [Текст] // Журн. Інформаційно-керуючі системи на залізнічному транспорті, 2013, №4, (додаток). - С. 55-57.

16. Ізотов А. С. Аналіз конформности на етапі проектування мережевих протоколів [Текст] / А. С. Ізотов, В. П. Немченко // Журн. Проблеми інформаційних технологій, 2014 року, №1 (015). - С. 206-210.

17. Uri Shani, Ariel Landau. Tools Interoperability Platform for Model-Based Systems-Engineering [Текст]: MBSDPTI workshop SECOOP'13, July 2013.

18. Проектування і діагностика комп'ютерних систем і мереж [Текст]: навч. посібник / М. Ф. Бондаренко, Г. Ф. Кривуля, В. І. Хаханов В.І. та ін.: К.: НМЦ ВО, 2000. - 306 с.

19. Гаврилова Т. А. Основи знань інтелектуальних систем [Текст]: навч. / Т. А. Гаврилова, В. Ф. Хорошевський - МПб .: Питер, 2000. - 215 с.

20. Milner R. Lectures on a calculus for communicating systems [Текст] // Seminar on Concurrency, LNCS 197Springer-Verlag, 1999. P. 197-220.

21. Гольдштейн Б.С. Інтелектуальні мережі [Текст]: Б. С. Гольдштейн, И. М. Ехріель, Р. Д. Рерля. - М.: Радио и связь, 2005. - 502 с.

22. Субботін С. О. Подання й обробка знань у системах штучного інтелекту та ПІДТРИМКИ Прийняття РІШЕНЬ [Текст]: навч. посібник / С. О. Субботін - Запоріжжя: ЗНТУ, 2008. - 341 с.

23. Clarke E. / E. Clarke, O. Grumberg, D. Long, Model Checking, // In Springer-Verlag, Nato ASI Series F. - 1996 року, Volume 152. - P. 287-299.

24. Model-based functional and load testing of the web server of a street lighting system [Електронний ресурс] / K. Raiend (Elvior), A. Kull (Elvior) URL: http: // http://www.elvior.com/ model-based-testing-uc-2012 - ETSI Model Based Testing User Conference. - September 2012.

25. Bourdonov IB UniTesK Test Suite Architecture [Текст] / IB Bourdonov, AS Kossatchev, VV Kuliamin, AK Petrenko // Proceedings of the International Symposium of Formal Methods Europe, July 22-24, 2002. - Copenhagen, LNCS 2391, 2002 - P. 121-152.

26. Bourdonov IB Java Specification Extension for Automated Test Development [Текст] / IB Bourdonov, AV Demakov, AA Jarov, AS Kossatchev, VV Kuliamin, AK Petrenko, SV Zelenov. // Proceedings of the 4-nd International Andrei Ershov Memorial Conference Perspectives of System Informatics, July 2-6, 2001. - Novosibirsk, LNCS 2244, 2001., - P. 301-307.

27. Кулямин В. В. Формалізація вимог на практиці [Текст] / В. В. Кулямин, Н. В. Пакуліна, О. Л. Петренко, А. А. Сортів, А. В. Хорошилов // Препринт 13 ІМП РАН, 2006 .

28. Аналіз мережевих протоколів як метод оптимізації мережі [Електронний ресурс] / "LAN / ЖУРНАЛ МЕРЕЖЕВИХ РІШЕНЬ" Январь 1999. - Режим доступу: URL: http://www.prolan.ru/article/magazine/lan011999.html. - 1 999.

29. Немченко В. П. Побудова системи генерації тестових послідовностей для мережевих протоколів [Текст] / В. П. Немченко, А. С. Ізотов // Журн. Інформаційно-керуючі системи на залізнічному транспорті, 2011, №4. - С. 73-80.

30. Іванніков В. П. Використання контрактних специфікацій для подання вимог і функціонального тестування моделей апаратури [Текст] / Іванніков В. П., Камкін А. С., Косачев А. С., Кулямин В. В., Петренко А. К .. // Програмування, тому 33, №5. М.: Маїк «Наука / Интерпериодика», 2007. - С. 47-61.

31. Брауер В. Введення в теорію кінцевих автоматів [Текст] / В. Брауер. - М.: Радио и связь, 1987 - 384 с.

32. Wagner F. Modeling Software with Finite State Machines: A Practical Approach [Текст] / F.Wagner - Auerbach Publications, 2006 - ISBN 0-8493-8086-3.

33. Palnitkar S. Finite state machine trace analysis program [Текст] / Palnitkar S. // Proc. Int. Verilog HDL Conf. - 1994.

34. Kuliamin VV Practical Approach to Specification and Conformance Testing of Distributed Network Applications [Текст] / VV Kuliamin, AK Petrenko, NV Pakoulin // Proceedings of the 2-nd International Service Availability Symposium, April 25-26, 2005: Додати Berlin, 2005. - P. 68-83.

35. Kuliamin VV Extended Design-by-Contract Approach to Specification and Conformance Testing of Distributed Software [Текст] / VV Kuliamin, AK Petrenko, NV Pakoulin // Proceedings of the 9-th World Multiconference on Systemics, Cybernetics, and Informatics, Model Based Development and Testing Workshop, July 10-13, 2005: Додати Orlando, Florida, 2005. - P. 65-70.

36. Бурдонов І. Б. Формалізація тестового експерименту [Текст] / І. Б. Бурдонов, А. С. Косачев, В. В. Кулямин // Журн. Програмування, 2007. No. 5. - С. 121-134.

37. Бурдонов І. Б. Теорія відповідності для систем з блокуваннями і руйнуванням [Текст] / І. Б. Бурдонов, А. С. Косачев, В. В. Кулямин - М .: Наука, 2008. - 251 с.

38. Бурдонов І. Б. Теорія конформності для функціонального тестування програмних систем на основі формальних моделей [Текст]: Дисертація на здобуття наукового ступеня д.ф.-м.н. / І.Б. Бурдонов - М., 2008. - 596 с.

39. Бурдонов І. Б. Повне тестування з відкритим станом обмежено недетермінірованних систем [Текст] // І. Б. Бурдонов, А. С. Косачев. // Журн. Програмування, 2009. No. 5. - С. 98-109.

40. ITU-T Recommendation Q.3903. Formalized presentation of testing results [Текст] - 2008.

41. ISO / IEC 9646. Information technology - Open Systems Interconnection - Conformance testing methodology and framework - Part 1: General concepts [Текст] / Geneva: ISO, 1994. - 178 с.

42. ISO / IEC 10168-1. Information technology Open Systems Interconnection - Conformance Test Suite for the session protocol [Текст]: - Part 1: Test suite structure and test purposes / Структура тестового комплекту і тестові мети ISO / IEC 19501: 2005 (E) Unified Modeling Language Specification Version 1.4. 2 formal / 05-04-01.

43. ETSI Methods for Testing and Specifcation (MTS); The Testing and Test Control Notation [Текст] / ES 201 873-1 V3.1.1. version 3; Part 1: TTCN-3 Core Language. - Sophia-Antipolis: ETSI, 2009. - 34 p.

44. Гордієнко А. В. Про один підхід до процесу розробки протоколів інформаційного обміну [Текст] / Гордієнко А. В., Михайлов А. С., Пашков А. Ю., Чернишов М. Е .: сб. науч. тр. в 16 т. М .: МІФІ, 2006. - т. 15. с. 21.

45. Dalci E. Requirements for GSC-IS Reference Implementations / E. Dalci, E. Fong, A. Goldfine // National Institute of Standards and Technology, Information Technology Laboratory, 2003. - 56 p.

46. Гольдштейн Б.С. Протоколи GPRS і їх тестування [Текст] - М: "Мобільні системи", №10, 2002. - 10 с.

47. Гойхман В.Ю. Дослідження SDL-специфікацій протоколів мереж TDM і NGN [Текст] / В. Ю. Гойхман // Матеріали 61 науково-технічної конференції професорсько - викладацького складу, наукових співробітників та аспірантів МПбГУТ ім. проф. М.А. Бонч-Бруєвича: МПб, 2009. - С. 28-35.

48. Lehtmets A. Automated Model-based WEB testing FiSTB Testing Assembly [Текст] / A. Lehtmets, D. Felmly-Leesment, August 2013.

49. Вейрле К. Linux: мережева архітектура. Структура і реалізація мережевих протоколів в ядрі [Текст] / К. Вейрле, Ф. Пельке, Х. Ріттер, Д. Мюллер, М. Бехлер: КУДИЦ - Образ, 2006. - 656 с.

50. Стівенс У. Р. UNIX. Розробка мережевих протоколів [Текст] / У. Р. Стівенс, Б. Феннер, Е. М. Рудофф: пров. з англ. - МПб. : Пітер 2007 р 1040 стор.

51. Goldstein BS Evolution of Telecommunication Protocols [Текст]: BHV-Piter, 2002. - 360 p ..

52. Гольдштейн Б.С. Тестування телекомунікаційних протоколів: проблеми і підходи [Текст] / Б. С. Гольдштейн, И. М. Ехріель, Р. Д. Рерля. - М: "Мережі і системи зв'язку", №12, 2002. - 10 с.

53. G.D. Abowd, J.P.G. Sterbenz, Final report on the interagency workshop on research issues for smart environments // . IEEE Personal Communications (October 2000) 36–40.

54. J. Agre, L. Clare, An integrated architecture for cooperative sensing networks // IEEE Computer Magazine (May 2000) 106–108.

55. I.F. Akyildiz, W. Su, A power aware enhanced routing (PAER) protocol for sensor networks. // Georgia Tech Technical Report, January 2002

56. ГОСТ 12.1.005-88. Общие санитарно-гигиенические требования к воздуху рабочей зоны.

57. ГОСТ 12.0.003-74 Опасные и вредные производственные факторы. Классификация.

58. НПАОП 40.1-1.21-98. Правила безпечної експлуатації електроустановок споживачів

59. ГОСТ 12.1.009-76. ССБТ. Электробезопасность. Термины и определения.

60. *ДМП 173-96. Державні санітарні правила планування та забудови населених пунктів.*

61. ДСН 3.3.6.042-99 Санітарні норми мікроклімату виробничих приміщень.

62. ГОСТ 12.1.005-88. Система стандартов безопасности труда. Общие санитарно-гигиенические требования к воздуху рабочей зоны.

63. TCO’ 07 Certified Displays. © 2007 Copyright TCO Development AB

64. ДСанПіН 3.3.2.007-98, Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин.

65. ДБН В.2.5-28-2006. Природне і штучне освітлення

66. ГОСТ 12.1.044-89 Система стандартов безопасности труда. Пожаровзрывоопасность веществ и материалов. Номенклатура показателей и методы их определения.

67. НАПБ Б.03.002-2007. Норми визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою.

68. ГОСТ 12.1.004-91. "Система стандартов безопасности труда. Пожарная безопасность. Общие требования".

69. НАПБ А.01.001-2014 “Правила пожежної безпеки в Україні”

70. НАПБ Б.03.001-2004. Про затвердження Типових норм належності вогнегасників.

71. Закон України «Про охорону навколишнього природного середовища»

72. Закон України «Про забезпечення санітарного та епідемічного благополуччя населення»

73. Закон України «Про відходи»

74. Закон України «Про охорону атмосферного повітря»

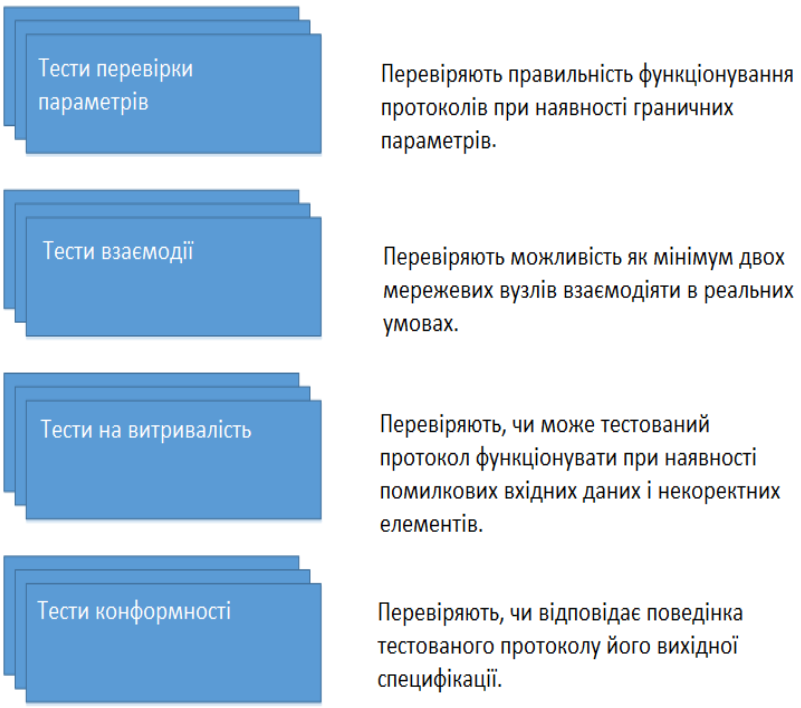
75. Закон України «Про захист населення і територій від надзвичайних ситуацій техногенного та природного характеру»

76. Водний кодекс України

77. ДСанПіН 2.2.7.029-99. Гігієнічні вимоги щодо поводження з промисловими відходами та визначення їх класу небезпеки для здоров'я населення.

ДОДАТОК А.

Комп’ютерна презентація

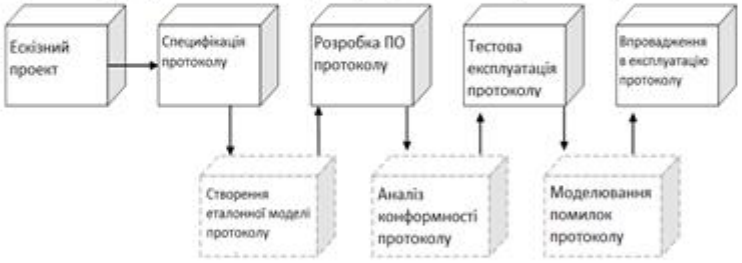


Класифікація тестів перевірки мережевих протоколів

3

Життєвий цикл мережевого протоколу

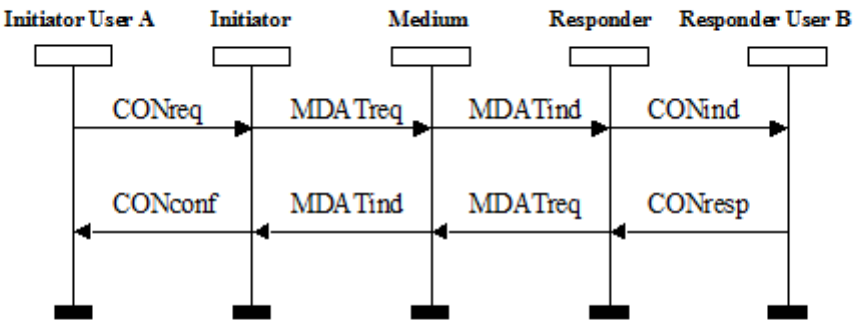
Спрощена структура життєвого циклу мережевого протоколу



Модифікована структура життєвого циклу мережевого протоколу

4

Графічна модель графіка послідовності повідомлень

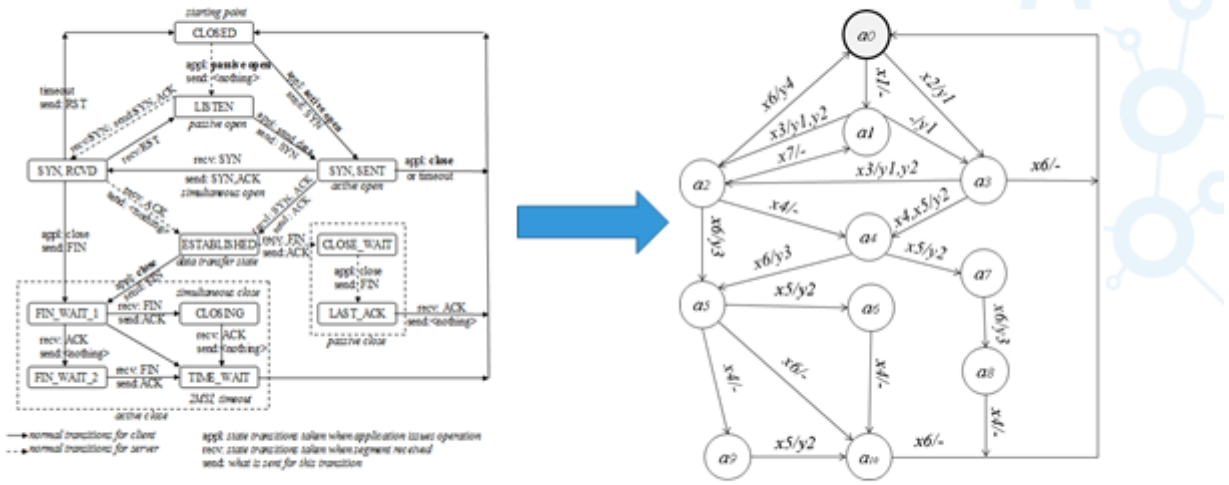


5

Узагальнена процедура тестування мережевих протоколів

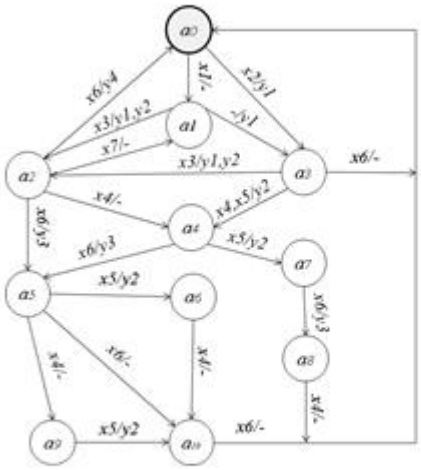
6

Графічна модель протоколу TCP



7

Табличне представлення протоколу TCP



Безліч станів {A}		Безліч вхідних команд {X}		Безліч вихідних реакцій {Y}	
a_0	CLOSED	x_1	Passive Open	y_1	SYN
a_1	LISTEN	x_2	Active Open	y_2	ACK
a_2	SYN_RCVD	x_3	SYN	y_3	FIN
a_3	SYN_SENT	x_4	ACK	y_4	RST
a_4	ESTABLISHED	x_5	FIN		
a_5	FIN_WAIT_1	x_6	close or timeout		
a_6	CLOSING	x_7	RST		
a_7	CLOSE_WAIT				
a_8	LAST_ACK				
a_9	FIN_WAIT_1				
a_{10}	TIME_WAIT				

8

Запишемо систему булевих рівнянь з зворотної таблиці переходів автомата

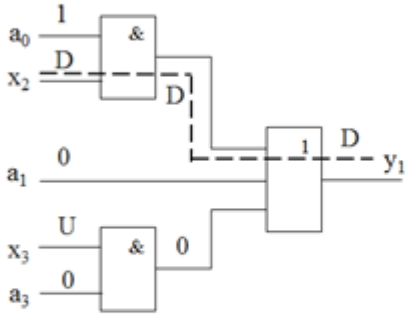
№	a(t+1)	x(t)	a(t)	y(t)
1.	a_{10}	x_6	a_5	-
2.	a_{10}	x_4	a_6	-
3.	a_{10}	x_5	a_9	y_2
4.	a_9	x_4	a_5	-
5.	a_8	x_6	a_7	y_2
6.	a_7	x_5	a_4	y_2
7.	a_6	x_5	a_5	y_2
8.	a_5	x_6	a_2	y_2
9.	a_5	x_6	a_4	y_2
10.	a_4	x_4	a_2	-
11.	a_4	x_4, x_5	a_3	y_2
12.	a_3	x_2	a_0	y_1
13.	a_2	1	a_1	y_1
14.	a_2	x_2	a_1	y_1, y_2
15.	a_2	x_2	a_3	y_1, y_2
16.	a_1	x_7	a_2	-
17.	a_1	x_1	a_0	-
18.	a_0	x_6	a_2	y_4
19.	a_0	x_6	a_3	-
20.	a_0	x_4	a_8	-
21.	a_0	x_6	a_{10}	-

$$
\begin{cases}
a_0 = a_{10}x_6 \vee a_8x_4 \vee a_5x_6 \vee a_2x_6 = x_6(a_2 \vee a_5 \vee a_{10}) \vee a_8, \\
a_1 = a_2x_7 \vee a_0x_1, \\
a_2 = a_1x_2 \vee a_3x_3, \\
a_3 = a_0x_2 \vee a_1, \\
a_4 = a_2x_4 \vee a_3x_4x_5 = x_4(a_2 \vee a_3x_5), \\
a_5 = a_2x_6 \vee a_4x_6 = x_6(a_2 \vee a_4), \\
a_6 = a_5x_5, \\
a_7 = a_4x_5, \\
a_8 = a_7x_6, \\
a_9 = a_5x_4, \\
a_{10} = a_5x_6 \vee a_6x_4 \vee a_9x_5.
\end{cases}
$$

$$
\begin{cases}
y_1 = a_0x_2 \vee a_1 \vee a_1x_2 \vee a_3x_3, \\
y_2 = a_9x_5 \vee a_4x_5 \vee a_5x_5 \vee a_3x_4x_5 \vee a_1x_2 \vee a_3x_3, \\
y_3 = a_2x_6 \vee a_4x_6 = x_6(a_2 \vee a_4), \\
y_4 = a_2x_6.
\end{cases}
$$

9

Узагальнена процедура тестування мережевих протоколів



Несправність	$x_2 \equiv 1$	$x_2 \equiv 0$
Перевірка по входу x_2	$\bar{x}_2a_0\bar{a}_1\bar{a}_3 \vee \bar{x}_2\bar{a}_1\bar{x}_3$	$x_2a_0\bar{a}_1\bar{a}_3 \vee x_2\bar{a}_1\bar{x}_3$
Вектори тесту	$x_2\ x_3\ a_0\ a_1\ a_3\ y_1$ 1 U 1 0 0 1 0 U 1 0 0 0 D U 1 0 0 D	

10

Структура тесту T

T	x_1	x_2	x_3	x_4	x_5	x_6	a_0	a_1	a_2	a_3	a_4	a_5	y_1	y_2	y_3	y_4
t_1	U	D	U	U	U	U	1	0	U	0	U	U	D	-	-	-
t_2	U	0	D	U	U	U	U	0	U	1	U	U	D	-	-	-
t_3	U	D	0	0	0	U	U	1	U	U	U	U	-	D	-	-
t_4	U	0	D	0	0	U	U	U	U	1	U	U	-	D	-	-
t_5	U	0	0	D	1	U	U	U	U	1	0	0	-	D	-	-
t_6	U	0	0	1	D	U	U	U	U	1	0	0	-	D	-	-
t_7	U	0	0	1	D	U	U	U	U	0	1	0	-	D	-	-
t_8	U	0	0	0	D	U	U	U	U	0	0	1	-	D	-	-
t_9	U	U	U	U	U	D	U	U	1	U	1	U	-	-	D	-
t_{10}	U	U	U	U	U	D	U	U	1	U	1	U	-	-	-	D

11

Кінцева тестова послідовність автомата протоколу TCP

Вектори тесту T	{X}						{A}						{Y}			
	Passive Open	Active Open	SYN	ACK	FIN	close or timeout	CLOSED	LISTEN	SYN_RCVD	SYN_SENT	ESTABLISHED	FIN_WAIT_1	SYN	ACK	FIN	RST
t_1	-	D	-	-	-	-	1	0	-	0	-	-	D	-	-	-
t_2	-	0	D	-	-	-	-	0	-	1	-	-	D	-	-	-
t_3	-	D	0	0	0	-	-	1	-	-	-	-	-	D	-	-
t_4	-	0	D	0	0	-	-	-	-	1	-	-	-	D	-	-
t_5	-	0	0	D	1	-	-	-	-	1	0	0	-	D	-	-
t_6	-	0	0	1	D	-	-	-	-	1	0	0	-	D	-	-
t_7	-	0	0	1	D	-	-	-	-	0	1	0	-	D	-	-
t_8	-	0	0	0	D	-	-	-	-	0	0	1	-	D	-	-
t_9	-	-	-	-	-	D	-	-	1	-	1	-	-	-	D	-
t_{10}	-	-	-	-	-	D	-	-	1	-	1	-	-	-	-	D

12

Наукові результати

Проаналізовано та доопрацьована типова структура життєвого циклу мережевого протоколу. Введено нові етапи циклу, які передбачають створення еталонної моделі протоколу з метою перевірки конформності протоколу задекларованої специфікації ще на початкових стадіях створення протоколу.

Обґрунтовано використання класичної теорії кінцевих автоматів для моделювання мережевих протоколів. В уваги береться те, що кожне наступне стан протоколу є функція від попереднього стану і вхідного слова. Зроблено обґрунтований висновок про доцільність використання моделі Мілі для моделювання мережевих протоколів.

Опрацьовані аналітичні вирази для опису перехідних процесів в автоматної моделі мережі під впливом функціонування мережевого протоколу в залежності від стану мережі і вхідних впливів, що стало можливим завдяки використанню теорії кінцевих автоматів при моделюванні.

14

Наукові результати

Розроблено метод побудови автоматизованої системи генерації тестових послідовностей для тестування мережевих протоколів в тому числі і протоколів нового покоління стека TCP/ IPv6. При цьому акцент зроблений на верифікації відповідності створюваних протоколів їх специфікації. Також крім перевірки правильності реалізації протоколів проаналізована можливість побудови так званих «словників несправностей« помилок »протоколу для подальшої його діагностики та ідентифікації помилок.

Отримав подальший розвиток метод використання булевих похідних однієї змінної для генерації тестових послідовностей спрямованих на діагностування несправностей (помилок) в роботі мережевих протоколів. За основу взято принцип активізації шляхів від кожного входу протоколу до виходів в об'єкті, що перевіряється.

Розроблено модель об'єкта діагностування і методика проведення діагностичного експерименту, які використовуються на підприємстві для перевірки мережевих протоколів на їх конформність задекларованої специфікації, а також в навчальному процесі університету.

15

Наукові результати

Розглянуті методи можуть бути застосовані для підвищення якості обслуговування і достовірності передачі інформації в комп'ютерних мережах.

Модифікована структура життєвого циклу мережевих протоколів з нововведеними етапами циклу дозволяє підвищити ефективність проектування нових протоколів в тому числі і протоколів нового покоління TCP / IPv.6.

Метод моделювання мережевих протоколів з використанням автоматної моделі Мілі на базі класичної теорії кінцевих автоматів дає можливість виявляти і ідентифікувати помилки в програмній реалізації як нових протоколів, так і вже існуючих.

На основі автоматного методу моделювання розроблено принципи побудови автоматизованої системи тестування мережевих протоколів.

16