МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
Кафедра електроніки і комп’ютерної техніки

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи магістра
на тему:

«Електронна система моніторингу мережевого трафіку»

Завідувач кафедри ЕКТ А.С. Опанасюк

Керівник роботи О.В. Бережна

Консультант з
техніко-економічної частини О.М. Маценко

Студент групи ЕС-71 В.Р. Васильєв

Суми
2022

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

Факультет електроніки та інформаційних технологій

Кафедра електроніки і комп'ютерної техніки

Напрям підготовки 171 Електроніка

Освітня програма Електронні системи та компоненти

ЗАТВЕРДЖУЮ

Зав. кафедрою Опанасюк А.С.

"___"_____ 202__ р..

З А В Д А Н Н Я

на кваліфікаційну роботу магістра

Васильєва Віталія Романовича

1 Тема роботи «Електронна система моніторингу мережевого трафіку» затверджена наказом по університету "25" жовтня 2022 р. № 0942-VI

2 Термін здачі студентом закінченої роботи 16.12.2022

3 Вихідні дані до роботи Локальна мережа Ethernet. Портативний програмно-апаратний пристрій. Можливість аналізу фізичної та логічної топології мережі.

4 Зміст розрахунково-пояснювальної записки (перелік питань, що належить розробити): Вступ 1. Огляд літератури та постановка завдання. 2. Науково дослідна частина. 3. Розроблення алгоритму функціонування та електрично структурної схеми пристрою. 4. Розробка електрично функціональної схеми пристрою. 5. Розробка та розрахунок принципових електричних схем вузлів та блоків пристрою. 6. Технічно-економічна частина. 7. Розробка програмного забезпечення для пристрою. Висновки.

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) 1. Алгоритм схема; 2. Електрична структурна схема; 3. Електрична функціональна схема; 4. Електрична принципова схема.

6 Консультанти по проекту (роботі), із зазначенням розділів проекту, що стосуються їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Економічна частина	МАЦЕНКО О.М.		

7 Дата видачі завдання 25.10.2022

Керівник _____ Бережна О.В.

Завдання прийняв до виконання _____ Васильєв В.Р.

КАЛЕНДАРНИЙ ПЛАН

№ п/п	1 Назва етапів дипломного проекту (роботи)	2 Термін виконання етапів проекту (роботи)	Примітка
1	Огляд літератури та поставлення задачі проектування	01.11.22-06.11.22	
2	Вибір та обґрунтування алгоритму функціонування та структурної схеми системи	07.11.22-12.11.22	
3	Науково-дослідна частина	13.11.22-23.11.22	
4	Розробка функціональної схеми блоків системи	25.11.22-30.11.22	
5	Вибір елементної бази та розробка принципових електричних схем блоків	01.12.22-11.12.22	
6	Економічна частина	12.12.22-17.12.22	

Студент-дипломник _____ Васильєв В.Р.

Керівник проекту (роботи) _____ Бережна О.В.

"___"_____2022 р.

РЕФЕРАТ

Пояснювальна записка: 94 аркуші, 26 рисунки, 12 таблиць, 23 джерела літератури.

Графічна частина роботи включає в себе: алгоритм роботи пристрою, структурну, функціональну та принципову електричні схеми.

Пояснювальна записка містить 7 розділів: огляд літератури, науково-дослідна частину, розробка алгоритму роботи та структурної схеми пристрою, розробка функціональної схеми пристрою, розроблення принципової електричної схеми пристрою, техніко-економічну частину, розробка програмного забезпечення пристрою електронного моніторингу мережевого трафіку.

Перший розділ містить загальну інформацію про огляд науково-технічної літератури, призначення, аналіз сфери попиту та використання, переваги й недоліки, передумови для прийняття рішення.

Другий розділ присвячений науково-дослідній роботі.

Третій розділ присвячений розробці алгоритму роботи та структурної схеми проектованого пристрою.

Четвертий розділ присвячений розробці функціональної схеми пристрою.

П’ятий розділ присвячений розробленню принципової електричної схеми пристрою.

Шостий розділ присвячений техніко-економічній частині.

Сьомий розділ містить розробку програмного забезпечення прототипу пристрою електронного моніторингу мережевого трафіку.

ЗМІСТ

ВСТУП

Сучасні тенденції розвитку інтелектуальних розподілених мереж вимагають впровадження різних пристроїв контролю, вимірювання та управління, що використовують в мережі для інтеграції послідовні канали передачі інформації. Для реалізації каналів та ефективної підтримки необхідні пристрої для обміну інформацією та моніторингу продуктивності мережі.

При виникненні в мережі будь-яких проблем адміністратору доводиться виступати в ролі лікаря. І без портативного мережевого аналізатора, який легко поміщається в чемоданчик, при цьому не обійтися. Подібний пристрій допоможе виявити неполадки кабельної проводки, виявити мережеві адаптери, що проявляють зайву активність, встановити джерела помилкового функціонування комутаторів або маршрутизаторів. Перераховані події є причиною більшості збоїв, пов'язаних з роботою мережевої інфраструктури, а тому ефективні засоби тестування дозволять уникнути довгих годин, а іноді і днів простою мережі.

Аналіз мережевого трафіку на сьогоднішній день дуже велика тема. Під "аналізом мережевого трафіку" ми будемо розуміти сукупну назву технологій і їх реалізацій, що дозволяють проводити накопичення, обробку, класифікацію, контроль і модифікацію мережевих пакетів в залежності від їх вмісту в реальному часі. Одним з ускладнюючих факторів, при розгляді даного питання, є подвійність розвитку коштів аналізу мережевого трафіку: з одного боку-це розвиток алгоритмів і підходів до аналізу, з іншого-розвиток програмно-апаратних засобів для ефективного вирішення цього завдання. У свою чергу, це призводить як до плутанини в термінології, так і до свідомого маніпулювання фактами і цифрами в маркетингових цілях. У даній роботі зроблена спроба відобразити, як історичний розвиток, так і поточний стан даної області з науковою і прикладної точок зору. Також робиться спроба систематизувати відомості про сукупність технологій, що містяться в публікаціях. [1, 23].

1 ОГЛЯД ЛІТЕРАТУРИ

1.1 Огляд науково-технічної літератури електронних систем моніторингу мережевого трафіку

Моніторинг мережевого трафіку - це процес аналізу, оцінки та управління потоками даних для виявлення всіх невідповідностей або аномалій у роботі мережі. Це процес управління мережею, який досліджує зв'язок, дані, пакетний трафік мережі за допомогою різних ін

Аналізатор трафіку, або сніффер (від англ. to sniff-нюхати)-мережевий аналізатор трафіку, програма або програмно-апаратний пристрій, призначений для перехоплення і подальшого аналізу мережевого трафіку для всіх вузлів мережі. Під час роботи сніффера мережевий інтерфейс перемикається в режим прослуховування, що і дозволяє йому отримувати пакети, адресовані іншим інтерфейсам в мережі.

Перехоплення трафіку може здійснюватися:

- звичайним "прослуховуванням" мережі, що ефективно при використанні в сегменті концентраторів (хабів), але малоефективно при використанні комутаторів (світчей), оскільки на сніффер потрапляють лише окремі фрейми;
- підключенням сніффера в розрив каналу;
- відгалуженням (програмним або апаратним) трафіку і напрямком його копії на сніффер;
- через аналіз побічних електромагнітних випромінювань і відновлення, таким чином, прослуховується трафіку;
- через атаку на канальному (MAC) або мережевому (IP) рівні, що приводить до перенаправлення трафіку на сніффер з подальшим поверненням трафіку в належну адресу.

На початку 1990-х сніффери широко застосовувалися хакерами для захоплення користувальницьких логінів і паролів, які в ряді протоколів передаються в незашифрованому або слабо зашифрованому вигляді, а використання хабів дозволяло захоплювати трафік у великих сегментах мережі практично без ризику бути виявленим.

У наш час сніффери знаходять застосування в мережевому адмініструванні. Аналіз минулого через сниффер трафіку дозволяє:

- Виявити паразитний, вірусний і закільцьований трафік, наявність якого збільшує завантаження мережевого обладнання і каналів зв'язку. Однак сніффери для цих цілей є не дуже ефективними. Для цих цілей використовують збір різноманітної статистики серверами і активним мережевим обладнанням.
- Виявити в мережі шкідливе і несанкціоноване ПЗ, наприклад, мережеві сканери, флудери, трояни, клієнти пірінгових мереж та інші (для цього використовують спеціалізовані сніфери - Монітори мережевої активності).
- Локалізувати несправність мережі або помилку конфігурації мережевих агентів, що особливо важливо при роботі мережевих адміністраторів.струментів і методів.

Згідно з оглядом джерел, основними областями використання електронних систем моніторингу мережевого трафіку є сервіси провайдерів, корпоративні замовлення, датацентри, державний сектор. Слугують, як засоби контролю трафіку віртуалізованих середовищ, організації мереж SDN і забезпечення інформаційної безпеки мереж на основі систем моніторингу та легального перехоплення трафіку.

Спочатку портативні пристрої моніторингу мережевого трафіку, призначені для тестування роботи мереж, були розраховані виключно на перевірку технічних параметрів кабелю. Однак в останні роки виробники наділили своє обладнання рядом функцій аналізаторів протоколів. Якщо вірити заявам цих компаній, сучасні мережеві аналізатори здатні виявляти найширший спектр можливих неполадок-від фізичного пошкодження кабелю до перевантаження мережевих ресурсів. З метою перевірити істинність подібних тверджень ми запропонували виробникам надати нам продукти для тестування в лабораторних умовах. В першу чергу нас цікавили швидкість і точність визначення збоїв, що виникають в кабелях або інших частинах мережевої інфраструктури, можливість передачі первинних результатів діагностики на ПК для подальшого детального аналізу і генерації звітів, простота роботи і ціна. Особливий акцент був зроблений на здатності кожної

з моделей проводити діагностику і в умовах функціонуючої мережі, а не тільки на етапі прокладки кабелю.

Постійний моніторинг роботи локальної мережі, що лежить в основі корпоративної мережі, необхідний для підтримки її працездатності. Контроль- це перший крок, який ви повинні зробити, керуючи своєю мережею. Через важливість цієї функції її часто відокремлюють від решти системи управління і реалізують особливим чином.

Автономний контроль допомагає адміністраторам мережі виявляти проблемні області та мережеві пристрої. В цьому випадку ви можете в ручну відключити або переналаштувати їх. Поділ функції управління на функцію контролю і власне управління корисно для невеликих мереж, яким установка інтегрованої системи менеджменту економічно недоцільно.

На етапі моніторингу виконується більш проста процедура - процедура збору первинних даних про роботу мережі: статистики про кількість циркулюючих в мережі кадрів і пакетів різних протоколів, стан портів концентраторів, комутаторів і маршрутизаторів і т. п.

Далі виконується етап аналізу, під яким розуміється більш складний і інтелектуальний процес осмислення зібраної на етапі моніторингу інформації, зіставлення її з даними, отриманими раніше, і вироблення припущень про можливі причини повільний або ненадійної роботи мережі.

Завдання моніторингу вирішуються програмними і апаратними вимірниками, тестерами, мережними аналізаторами, вбудованими засобами моніторингу комунікаційних пристроїв, а також агентами систем управління. Завдання аналізу вимагає більш активної участі людини і використання таких складних засобів, як експертні системи, акумулюючі практичний досвід багатьох мережевих фахівців.

Простота, гнучкість та надійність пристроїв доступу до мережевого трафіку надзвичайно важливі, оскільки:

Системи моніторингу дуже складні і не варто ускладнювати їх понад те, що необхідно;

Системи спостереження надзвичайно чутливі до збоїв в роботі своїх компонентів, тому не варто створювати нову потенційну точку відмови для таких систем;

Проектування систем моніторингу - досить складний процес, і навіть при самому ідеальному плані можна втратити якусь важливу деталь, але гнучкість рішення по доступу до мережевого трафіку в багатьох випадках допоможе компенсувати помилки планування;

Система моніторингу коштує недешево, але це з точки зору капітальних і поточних витрат; інтелектуальна система доступу до мережевого трафіку система моніторингу дозволяє істотно знизити загальну вартість володіння (ТСО) системою.

Недоліком систем, що існують є відсутність змоги аналізувати рівень сигналу в інформаційних шинах і шукати причини відмови при передачі даних, аналізу пристроїв систем безпеки (електроніки та інших суміжних областей) таких як:

- сервіси провайдерів;
- датацентри;
- державний сектор;
- системи безпеки;
- обробка інформації, збір і аналіз в системах нагляду.

Проведений аналіз та дослідження технічних даних не виявило великої кількості пропозицій щодо електронних систем моніторингу мережевого трафіку, особливо таких що змогли б , що змогли б аналізувати рівень аналогового сигналу в інформаційних шинах і шукати причини відмови при передачі даних, аналізу пристроїв. Подача сигналу через канал дозволяє прослуховувати канали передачі даних, аналізувати рівні і форму модульованих сигналів, демодулювати їх в терміни символів формату ASCII і аналізувати на логічному рівні протоколи обміну між сервером і кінцевими пристроями. Таким чином, такий прилад має велику перевагу в порівнянні з пристроями і пристроями нашого часу. Тому цей прилад може стати необхідним і незамінним пристроєм для використання як приватним особам, так і в сервісних центрах, системах охорони безпеки.

Робота по створенню цього приладу перспективна, актуальна та конкурентоспроможна.

Завданням та метою розробки «Електронної системи моніторингу мережевого трафіку» розробити пристрій який може підключатися до розривів каналів і мати змогу «прослуховувати» каналів для передачі даних,

аналізувати форму та рівень модульованих сигналів, виконувати демодуляцію, перетворювати в рядки ASCII, аналізувати протоколи обміну, повинно було бути створено забезпечує можливість візуального відображення між серверами і кінцевими пристроями на логічному рівні, мати можливість надати інформацію на маленькому робочому місці, змінювати масштаб одержуваних даних і використовувати прилад в умовах наближених до польових.

В результаті аналізу конструктивних, а особливо економічних показників представлених нижче було визначено, що оптимальну конструкцію пристрою можна буде створити на основі аналогової обробки вхідного сигналу і поєднання цифрового комунікаційного аналізатора і аналогового осцилографа. Такий метод дозволить створити ряд сімейств «Електронних систем моніторингу мережевого трафіку» [1,2].

Завданням магістерської роботи є розробка електронної системи моніторингу мережевого трафіку з малими розмірами, але з використанням ряду функціональних можливостей:

- захист від високої напруги на вході;
- відображення у певних діапазонах функцій сигналу;
- відображення з точністю у встановленому діапазоні діаграм;
- відображення на екрані результатів;
- масштаб відображення на екрані пристрою;
- мобільність
- простота конфігурації
- робота від батареї.

1.2 Аналіз сфери попиту

З аналізу попиту на пристрої даного виду, які здатні проводити моніторинг мережевого трафіку, виявлено, що великим попитом та потребою користуються портативні та особливо мініатюрні пристрої моніторингу, аналізу, контролю, вимірювання, експрес-аналізу та діагностики. Суттєво з’явилася потреба і можливість створення портативного пристрою електронної системи моніторингу мережевого трафіку. Тому пристрої швидко

набирають не аби якої популярності в роботі, де головним чинником є дуже малі розміри пристрою, але з високими вимоги до точності та швидкодії. Цей тип пристроїв надає змогу використовуватися в занадто стислих та некомфортних умовах роботи, а також в польових умовах, саме тоді, коли дуже важлива мобільність приладу.

Дослідження показали, що джерела технічної інформації не виявили широкої наявності приладів із технічними можливостями особливо зазначеними вище. В наш час прилад має великі шанси бути корисним та навіть потрібним.

Підбиваючи підсумок, можна сказати, що головні сфери використання електронної системи моніторингу мережевого трафіку - це автоматизовані виробничі лінії. Мережа підприємства телекомунікацій, спеціалізована мультисервісна інформаційна мережа зв'язку, система безпеки, передача, збір та аналіз даних у системах спостереження, ремонт і налагодження електронної апаратури в перерахованих вище областях, але в стислих умовах роботи, і аналіз рівня і форми модульованих сигналів [2,3].

1.3 Недоліки та переваги існуючих технологій

Будь-яка складна обчислювальна мережа вимагає додаткових спеціальних засобів управління крім тих, які є в стандартних мережевих операційних системах. Це обумовлено великою кількістю різноманітного комунікаційного обладнання, від надійності роботи якого залежить робота всієї мережі.

Аналіз показав, що системи моніторингу, пропоновані на світовому ринку, схожі по виконуваних функцій. Всі вони надають майже однаковий мінімальний набір можливостей, проте кожна з них характеризується певними недоліками: в більшості систем взагалі не реалізовані можливості прогнозування трендів, а в системах, де вони реалізовані, побудова відбувається на основі застарілої статистичної інформації. Подібне прогнозування не враховує фрактальність трафіку, нелінійність характеристик і нестаціонарність процесів. Узагальнивши запропоновані вище рішення, можна синтезувати загальну архітектуру системи моніторингу та управління. Всі розглянуті системи моніторингу засновані на використанні агентного

підходу. Агенти збирають статистичну інформацію про роботу елементів мережі і передають її в центральну базу даних, потім зібрана інформація обробляється керуючими модулями. До складу системи моніторингу повинні введіть наступні компоненти: формування звітів, Модуль управління SNMP, архів та консоль управління. Модуль формування звітів дозволяє формувати з наявних даних інформацію для прийняття управлінських рішень. Модуль управління SNMP відповідає за збір інформації з агентів моніторингу та взаємодія з системами управління. Архів дозволяє упорядкувати зберігання статистичної інформації і організувати подальшу роботу з нею. Консоль управління реалізує функції конфігурації і управління системою. Недоліком є те, що зменшення розмірів й здешевлення конструкції впливає на послаблення технічних вимог [1-4].

1.4 Передумови для прийняття рішення з реалізації пристрою

Аналізуючи пропозиції мінімізувати електронні системи моніторингу мережевого трафіку визначили, що портативні пристрої, особливо невеликі пристрої, користуються великим попитом і потребою. Потреба і потенціал для створення портативних пристроїв для електронного моніторингу мережевого трафіку значно зросли. Тому основним фактором є дуже малий розмір приладу, але високі вимоги до точності і швидкодії роблять прилад швидко затребуваним в роботі. Пристрої цього типу можуть використовуватися в дуже обмежених і незручних умовах роботи і полях, коли велике значення має мобільність пристрою. Завдяки досягненням у використанні цифрових АЦП і мікроконтролерів для мінімізації кількості деталей світу, необхідних для створення, ми знайшли можливість створювати портативні міні-пристрої для електронних систем моніторингу мережевого трафіку і згодом розширювати їх технічні характеристики.

Згідно з дослідженнями, навіть джерела технічної інформації не виявили, що пристрої з наведеними вище технічними характеристиками широко доступні. У наш час дуже ймовірно, що цей пристрій буде корисним і навіть потрібно буде використовувати.

1.5 Постановка завдання

Метою даної роботи є розробка пристроїв для електронних систем моніторингу мережевого трафіку. Цей пристрій повинен з достатньою точністю аналізувати рівень сигналу передачі даних на інформаційній шині, шукати причину збоїв передачі даних і виконувати аналіз пристрою. Пристрій повинен містити швидкодіючі АЦП для вимірювання тимчасових діаграм в точках підключення кабельної системи дротової локальної мережі. Вимірювання повинні дозволити оцінити рівні напруги і якість фронтів імпульсів при модуляції сигналів за допомогою коду Манчестер-II. На базі цієї інформації пристрій повинен сформувати двійкову послідовність зареєстрованої інформації з подальшою розшифровкою всіх полів IP-пакетів. Обладнання повинно бути компактним і придатним для використання в обмежених і некомфортних умовах роботи.

Для досягнення цієї мети ви повинні:

1. Визначити завдання та основні функції які буде виконувати система електронного контролю мережевого трафіку.
2. Розробити алгоритми роботи пристрою.
3. Створити електричну схему пристрою.
4. Розробити електричну схему функціональності пристрою.
5. Створити електричну схему на основі функціональності пристрою системи моніторингу мережевого трафіку.

2. НАУКОВО-ДОСЛІДНА ЧАСТИНА

Сьогодні, в той час як політичні IT-кордони стають все більш явними (держави створюють цілі екосистеми, що включають свій незалежний інтернет, свої сервіси, своє ПО), в корпоративному середовищі процес прямо протилежний: периметри організацій все сильніше розчиняються в інфосфері, забезпечуючи неабиякий головний біль безпечникам. Недовірені середовища та ентропія тіньової інфраструктури (Shadow IT), а на іншій стороні барикад — все більш витончені методи з ланцюжка загроз (kill chain) і ретельне приховування своєї присутності. Стандартні засоби ІБ-моніторингу не можуть дати повну картину того, що відбувається, що спонукає шукати додаткові джерела інформації — і в першу чергу аналізувати мережевий трафік.

2.1 Глибина аналізу мережевих пакетів

По цій «осі» технології аналізу трафіку розвивалися послідовно, кожна наступна успадковувала частину попередніх механізмів і додавала свої. Можна виділити три рівня розвитку технології, які наведені на рис. 1. Розглянемо ці рівні більш детально.

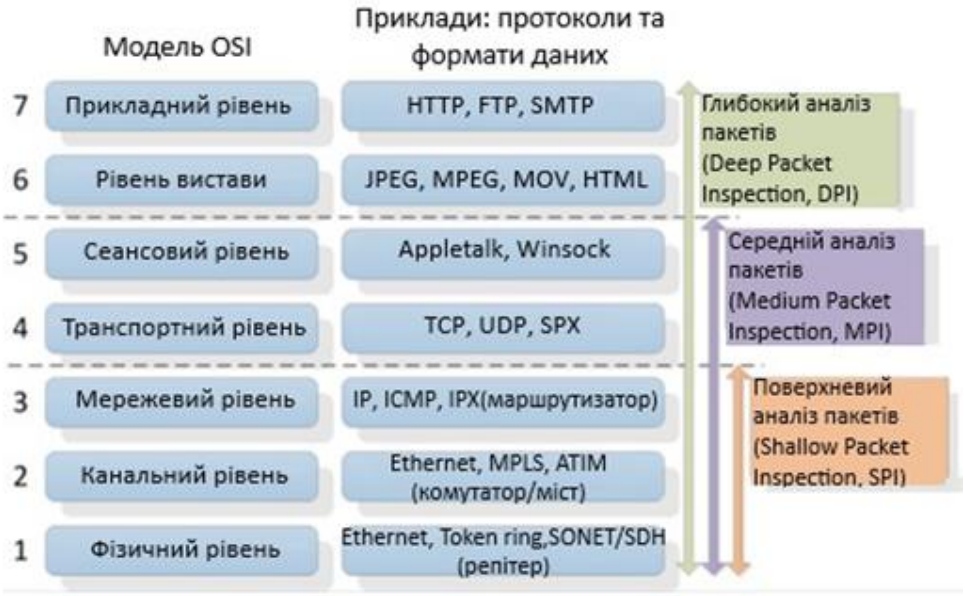


Рис. 1 - Рівні розвитку технології аналізу мережевого трафіку по «глибині».

2.1.1 Поверхневий аналіз пакетів (SPI)

Технологія аналізу трафіку, що грунтується виключно на заголовках пакету рівнів L1-L3 по моделі OSI. Висуває низькі вимоги до обчислювальних ресурсів, що дозволяє аналізувати великі обсяги трафіку. Технологія широко поширена, на її основі працює більшість міжмережевих екранів операційних систем (зокрема в ОС Windows XP/Vista і OS X), маршрутизаторів та інших мережевих пристроїв. На її основі реалізовані мережеві списки контролю доступу на рівні IP адрес і портів (Access Control List, ACL). Таким чином, дана технологія добре підходить для розмежування доступу ззовні до окремих комп'ютерів (IP) і сервісів (порти) внутрішньої мережі.

2.1.2 Середній аналіз пакетів (MPI)

Технологія аналізу трафіку, що грунтується на інспектуванні сесій і сеансів зв'язку, ініційованих додатком, але встановлюються шлюзом - посередником (див. 2). Також застосовується термін "проксі додатків" (application proxy). В рамках даної технології вміст пакетів аналізується частково і за зумовленими правилами. Не використовуються складні методи аналізу типу сигнатурного. Пристрої, що реалізують даний функціонал розміщуються між провайдером інтернету і кінцевим користувачем. Дані пристрої розбирають заголовки аж до транспортного рівня і невелику частину даних пакета для зіставлення розібраної частини з деяким списком розбору (parse list), з подальшою реакцією в разі їх виявлення. Дані списки зазвичай коротші за списки ACL і надають ширший діапазон дій на відміну від «дозволити/заборонити» у випадку ACL. Ці списки також більш виразні, так як дозволяють прив'язуватися не до IP-адресами, а до формату даних пакетів і даними деяких протоколів рівня додатки, наприклад, URL-адресами в разі протоколу HTTP. За допомогою MPI можна, наприклад, заблокувати можливість отримання flash-файлів або картинок з певних Інтернет сервісів (на рівні представлення OSI) або заблокувати частину команд (на рівні програми OSI) в окремих протоколах. Набір протоколів, як правило, дуже обмежений. Наприклад, в перших версіях CheckPoint FireWall-1 (CheckPoint FW-1) підтримувалися протоколи Telnet, FTP, HTTP, а в Cisco Private Internet

Exchange (Cisco PIX) - FTP, HTTP, H. 323, RSH, SMTP і sqlnet. Згодом дані набори незначно розширювалися. Також відомо, що дана технологія використовується в продуктах компаній McAfee і Symantec. Міжмережеві екрани, що використовують дану технологію, відносяться до другого покоління [1, 4].

Дана технологія більш гнучка в порівнянні з SPI і, крім розмежування доступу, підходить для більшого числа завдань — кешування вмісту, аналіз стисненого/шифрованого трафіку, обмеження функціоналу окремих протоколів шляхом заборони окремих команд. Завдяки підключенню в режимі проксі, може служити в якості Wan Optimizer'a (див.вище). Основний недолік MPI-погана масштабованість: кожна команда і протокол вимагають окремого «шлюзу» (вхідний-вихідний порти). Крім того, робота в режимі проксі сильно знижує швидкість обробки. Для зниження навантаження на проксі-сервер був розроблений протокол ICAP [4], що дозволяє проксі-серверам відправляти проходять через них дані для проведення аналізу стороннім серверам на предмет безпеки або аналізу вмісту. Ця схема реалізована в антивірусному продукті ClamAV, який може підключатися до згаданих вище проксі-серверів Squid та NetCache.

Ці фактори сильно обмежують застосування даної технології на рівні провайдерів інтернету внаслідок необхідності аналізу великого числа протоколів і команд на широких каналах зв'язку.

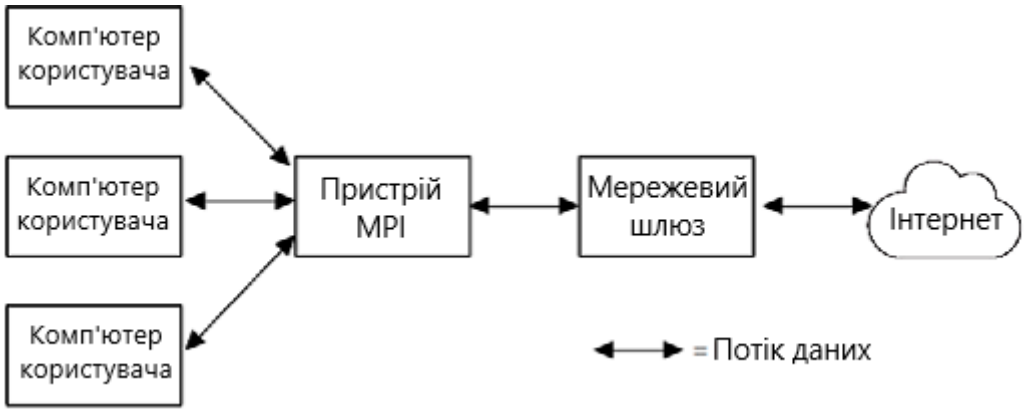


Рис. 2 - Схема застосування пристроїв аналізу на основі технології MPI.

2.1.3 Глибокий аналіз пакетів (DPI)

Іноді вживають більш вузький термін-DPP (Deep Packet Processing), який має на увазі такі дії над пакетами, як модифікація, фільтрація або

перенаправлення. Сьогодні обидва терміни часто використовуються як взаємозамінні [2]. Дана технологія є логічним розвитком MPI. В рамках даного підходу аналізатор переглядає вміст кожного пакета повністю. Одним з важливих відмінностей від попередніх технологій є те, що системи на базі DPI можуть приймати рішення не тільки по вмісту пакетів, але і за непрямими ознаками, властивим якимось певним мережевим програмам і протоколам. Для цього може використовуватися Статистичний аналіз. Наприклад, аналіз частоти зустрічі певних символів, довжин пакетів, відстань між мітками часу послідовних пакетів і т. д. також, в порівнянні з попередніми підходами, значно розширено список застосувань технології: Класифікація, обмеження смуги, пріоритезація, маркування, кешування і т. Д. технологія dpi отримала розвиток, перш за все, через стрімке зростання обчислювальних здібностей процесорів, їх швидкодії і, відповідно, можливостей для більш повного і точного аналізу мережевих даних.

На відміну від MPI, дана технологія спочатку розроблялася для високошвидкісної обробки та ідентифікації великого числа додатків в реальному часі. Таким чином, рішення на основі DPI добре масштабуються як по ширині мережевого каналу (відомі рішення, що працюють на каналах порядку 100 Гбіт/сек), так і по числу ідентифікованих додатків (в існуючих рішеннях — порядку декількох тисяч). З точки зору реалізації, основний компонент будь - якого рішення DPI-модуль класифікації, що відповідає за класифікацію мережевих потоків. При цьому в залежності від цілей застосування DPI, класифікація може виконуватися з різною точністю:

- тип протоколу або програми (наприклад, Web, p2p, VoIP)
- конкретний протокол рівня програми (HTTP, BitTorrent, SIP)
- додаток, що використовує протокол (Google Chrome, µTorrent, Skype)

Важливо відзначити, що відповідність між класами різних рівнів точності не однозначно, що показано на рис. 3.

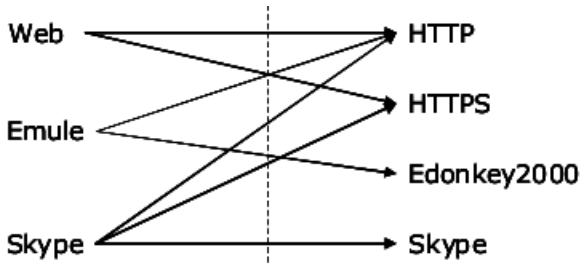


Рис. 3 - Різниця між ідентифікацією додатків (ліворуч) і протоколів (праворуч).

Технологія dpi на даний момент є поточним стандартом де факто для засобів аналізу мережевого трафіку і відноситься до області критично важливих технологій необхідних для забезпечення, як мережевої безпеки, так і вимог законодавства. Внаслідок цього останнім часом на міжнародному рівні було прийнято ряд стандартів, вимог та рекомендацій щодо особливостей реалізації, внутрішнього устрою та набору функцій відповідних засобів [2]. Ця технологія рідко застосовується в міжмережевих екранах-це скоріше область IDS/IPS систем, як винятки можна вказати екрани Hogwash і Shield. Однак брандмауери, що належать до четвертого покоління [1] можуть враховувати дані IDS/IPS систем у процесі аналізу.

2.2 Облік стану потоку при аналізі мережевого трафіку

Другим напрямком розвитку технології аналізу можна назвати облік стану протоколу (потоку) в процесі аналізу — Т.зв. stateless/statefull види аналізу. Даний напрямок актуально тільки для протоколів, що використовують транспортний протокол з встановленням з'єднання (connection-oriented). Це означає, що перед будь-яким обміном командами і даними відбувається процес «з'єднання», в ході якого сторони обмінюються фіксованою послідовністю пакетів, яка часто називається «рукостисканням» (handshake), а після завершення обміну відбувається аналогічний процес «закриття з'єднання». До connection-oriented протоколам, зокрема, відноситься протокол TCP, але не UDP. Однак слід врахувати, що поверх UDP може бути реалізований інший транспортний протокол, з встановленням з'єднання. Як приклад можна привести протокол Quick UDP Internet Connections (QUIC) — протокол транспортного рівня з встановленням з'єднання, що використовує

UDP. З цього випливає, що, в загальному випадку, не можна повністю виключити statefull аналіз для UDP пакетів.

Для опису відмінностей описаних підходів потрібно дати визначення поняттю»потік пакетів". Відомі різні визначення даного поняття. Частина з найбільш широко використовуваних наведена на сайті Center for Applied Internet Data Analysis (CAIDA). У даній роботі ми будемо використовувати "односторонній потік транспортного рівня" -послідовність пакетів передаються з заданого IP-адреси і TCP/UDP порту на даний IP-адреса і TCP/UDP порт, із зазначенням протоколу транспортного рівня (TCP/UDP). Таким чином, потік задається п'ятіркою <srcip, srcport, dstip, dstport, protocol>. З урахуванням даного визначення, можна сформулювати відмінність statefull від stateless підходу. Воно полягає в тому, що в разі statefull підходу враховується той факт, до якого саме потоку відноситься аналізований пакет, і результат (стан) аналізу попередніх пакетів цього ж потоку, якщо даний пакет не перший. У разі якщо пакет перший — перевіряється, що він є коректним пакетом встановлення з'єднання. Слід також зазначити, що поняття «statefull» не цілком чітке і може мати різні градації з різним «станом», що призводить до різного балансу точність аналізу/ресурсоємність/швидкість роботи [2]. Один з варіантів градації можна бачити на рис. 4. Список рівнів обліку стану потоку, який там відображений-наступний:

– Аналіз окремих пакетів без урахування потоків і станів (Packet Based No State, PBNS).

– Аналіз пакетів в рамках потоків (Packet Based Per Flow State, PBFS).

– Аналіз повідомлень в рамках потоку (Message Based per Flow State, MBFS), тобто проведена збірка IP-фрагментів в IP-пакети (IP нормалізація) і збірка TCP-сегментів в TCP-сеанси (TCP - нормалізація).

– Аналіз повідомлень в рамках протоколу (Message Based per Protocol State, MBPS), тобто враховується стан автомата протоколу (можливість приймати той чи інший тип повідомлень). Приклад автомата станів протоколу HTTP наведено на рис. 5. Вершини відповідають станам, ребра-умовам переходу, до яких можуть відноситися прийом/відправка повідомлення, результати обробки повідомлень, закінчення таймауту.

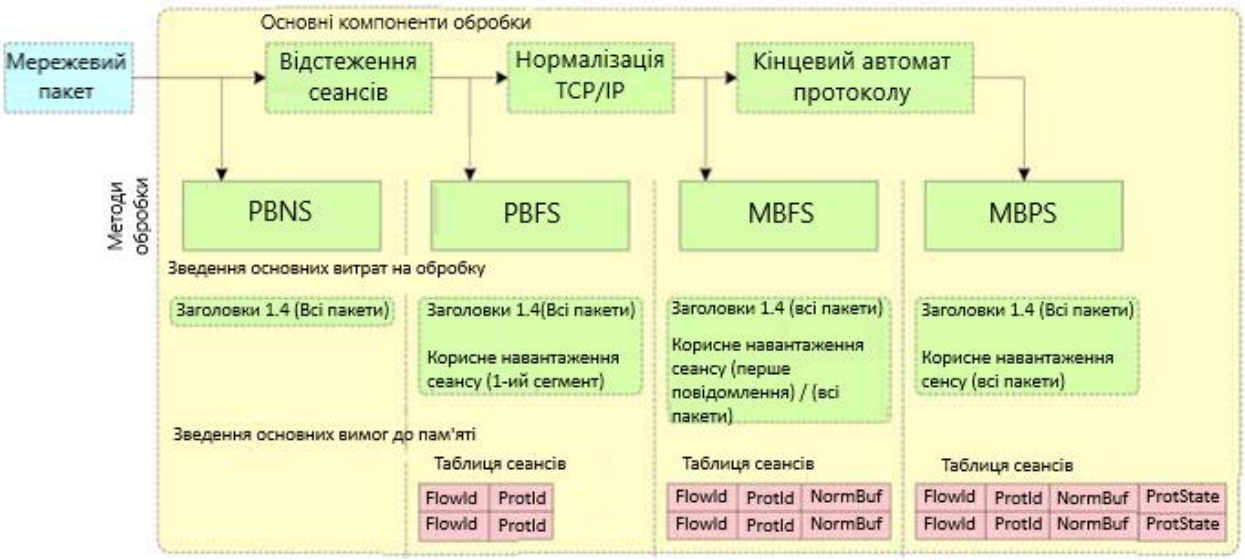


Рис. 4 - Градації повноти обліку стану потоку.

Базові реалізації технології DPI часто відносяться до stateless-аналізу, тобто аналіз виконується на рівні окремих пакетів, стан між аналізом декількох пакетів одного мережевого потоку не зберігається. Цього рівня точності вистачає для багатьох практичних додатків і дозволяє значно економити ресурси (див. 4). У той же час, існують завдання, для яких такого рівня точності не достатньо. Як приклади можна навести дві технології, що використовують statefull підхід-інспекція пакетів зі зберіганням стану (statefull packet inspection, SPI) і глибокий аналіз вмісту (deep content inspection, DCI).

2.2.1 Аналіз мережевих пакетів з урахуванням стану потоків

В рамках SPI підходу, програма або пристрій, який його реалізує, в момент відкриття нового з'єднання перевіряє його на відповідність заданій політиці безпеки і до закриття зберігає параметри цього з'єднання в пам'яті. За допомогою таких рішень, зокрема, здійснюється перевірка коректності з'єднання, наприклад відсутність пакетів на відкритому мережевому порту після завершення з'єднання. Реалізації SPI містяться в більшості сучасних маршрутизаторів у вигляді SPI-брандмауерів. Також ця технологія використовується в програмних міжмережевих екранах, що враховують стан (stateful firewalls), компанії CheckPoint і ряді IDS/IPS систем. Міжмережеві екрани, що використовують цю технологію, відносять до третього покоління.

При даному підході відслідковуються не тільки вхідні та вихідні пакети,але і стан окремих з'єднань, яке зберігається в динамічних таблицях. Завдяки цьому при аналізі чергового пакета можуть враховуватися не тільки задані правила і політики по відношенню до адрес і вмісту пакетів, а й стан з'єднання, до якого відноситься пакет і попередніх пакетів, які до нього відносяться, а також і інших, пов'язаних з даними, з'єднань. Класичний приклад переваги міжмережевого екрану підтримує стан потоку в порівнянні з міжмережевими екранами без такої підтримки — обробка FTP протоколу. Даний протокол відкриває новий потік передачі даних на кожну відповідну команду, причому потік відкривається на випадковому порту, більшому 1024. Так як міжмережевий екран не має можливості дізнатися, що новий потік відноситься до допустимого FTP протоколу — цей потік буде заблокований. У разі наявності підтримки станів потоків — адресна інформація нового потоку буде додана в таблицю легітимних потоків і сесія буде пропущена в мережу.

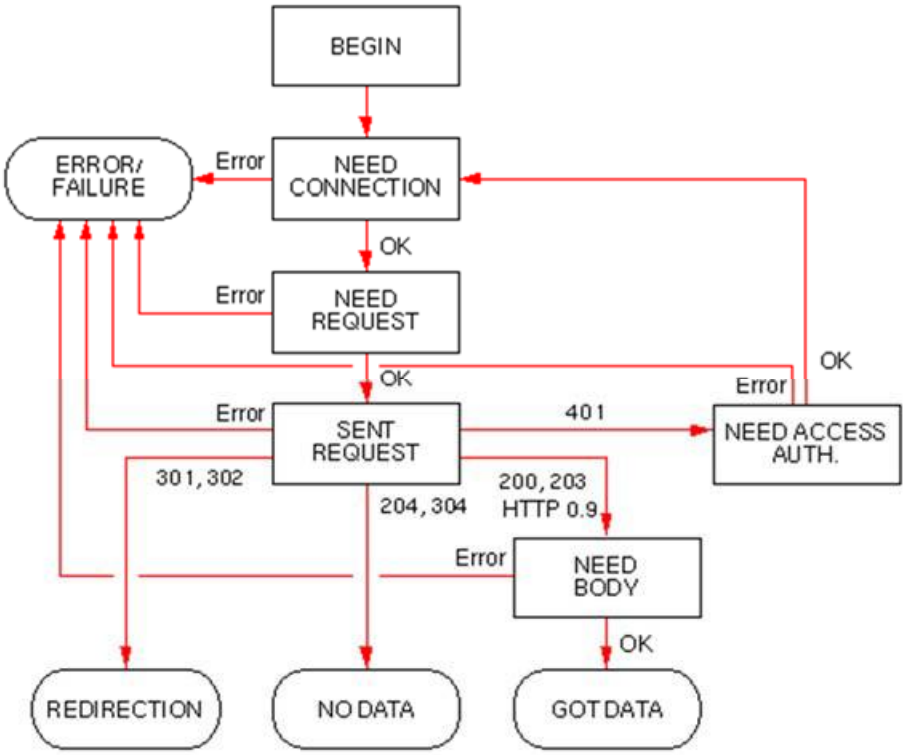


Рис. 5 - Приклад автомата станів протоколу HTTP.

2.2.2 Аналіз вмісту мережевих протоколів прикладного рівня

В рамках технології DCI виконується не тільки ідентифікація протоколу конкретного мережевого потоку, але і угруповання потоків в групи, що відповідають за надання деякого сервісу, наприклад сигнального протоколу, наприклад, SIP і протоколу передачі даних, наприклад, RTP, в разі VoIP. Також в процесі застосування DCI, аналіз не зупиняється на ідентифікації протоколу, наприклад, HTTP, але також робиться спроба визначити додаток, яке його використовує (наприклад, Gmail) і зібрати контент цього додатка в тому вигляді, в якому воно було передано додатком для відправки по мережі (електронний лист). Прикладом використання даної технології може служити функція прослуховування VoIP дзвінків по перехопленому трафіку в аналізаторі WireShark .

З точки зору функціоналу, основний внесок DCI на додаток до модуля класифікації (основний функціонал DPI) — набір модулів розбору для різних протоколів прикладного рівня і різних видів даних в різних кодуваннях (наприклад, MIME), які вони містять. Функції модулів розбору, зводяться до двох основних:

1. Розбір буфера даних (мережевого пакету або зібраної сесії), відповідно до формату повідомлень протоколу, описаним, як правило, на одному зі спеціальних мов типу ASN.1 і P4.

2. Збірка сесій для протоколів з встановленням з'єднання і їх подальший розбір (пункт 1).

Однією з тенденцій останнього часу в розвитку засобів DPI / dci є універсалізація і централізація аналізу. Дана концепція може бути позначена як «DPI як сервіс» - під цією назвою вона була наведена в роботі. Суть концепції полягає в тому, що якщо в мережі використовується велика кількість різних засобів, що реалізують той чи інший аналіз трафіку (міжмережеві екрани, системи IDS, оптимізатори трафіку і ін.), то має сенс винести весь аналіз в окремий пристрій. Це пристрій буде виконувати повний розбір мережевих даних і розсилати результати аналізу всім пристроям в залежності від їх потреб, а ті, в свою чергу, реалізовувати тільки реакцію на дані, що надходять. Перехід до цієї концепції в чомусь аналогічний переходу до програмно-конфігурованих мереж (Software Defined Networks, SDN) в питаннях управління трафіком, при якому всі рішення по використовуваних

алгоритмах маршрутизації і рівні її виконання переходять від конкретних маршрутизаторів до виділених пристроїв - SDN-контролерів. Такі підходи спрощують масштабування систем і дозволяють ефективно розширювати функціонал без додаткових робіт з інтеграції та перенастроювання обладнання. Концепція "DPI як сервіс" може бути ефективно реалізована в рамках систем уніфікованого управління загрозами (Unified threat management, UTM) і уніфікованого управління безпекою (Unified security management, USM). Ці системи також є відображенням тенденції централізації у вигляді об'єднання функціоналу міжмережевих екранів, мережевих систем IDS / IPS, антивірусів, VPN-серверів, фільтрів вмісту, балансування навантаження і запобігання витоків даних в рамках єдиної системи.

Демонстрацією цих тенденцій є виділення функціоналу розпізнавання протоколів і вилучення метаданих у вигляді окремих модулів. Причому ці модулі можуть бути, як чисто програмними, так і прив'язуватися до деякої апаратури. Прикладами програмних реалізацій є Qosmos Intelligence Engine, ipoque PACE, Windriver Content Inspection Engine, Procera PacketLogic Content Intelligence. Серед прив'язаних до апаратури модулів можна вказати Cisco Network Based Application Recognition (NBAR) і Junos OS Next-Generation Application Identification. Використання цих модулів у вигляді складової частини систем контролю і управління трафіком дозволяє формулювати політики безпеки та інші види політик в набагато більш високорівневих термінах, наприклад, у термінах URL, імен додатків, окремих функцій у рамках цих програм (наприклад, блокування передачі голосу в рамках Skype, при збереженні можливості обміну текстовими повідомленнями). По суті, набір функцій даних модулів аналогічний розширенню функціоналу технології MPI на довільну безліч протоколів, їх команд і даних, яке підтримуються конкретним модулем розпізнавання протоколів. Типова схема використання такого рішення наведена на рис. 6, де» зовнішній інтерфейс « — рішення типу» DPI як сервіс«, PCRF - Policy and Charging rules Function — пристрій, що зберігає політики і правила, що застосовуються до трафіку,» внутрішній інтерфейс " — пристрій зберігає статистику, журнали, результати застосування правил до трафіку, і т. д.

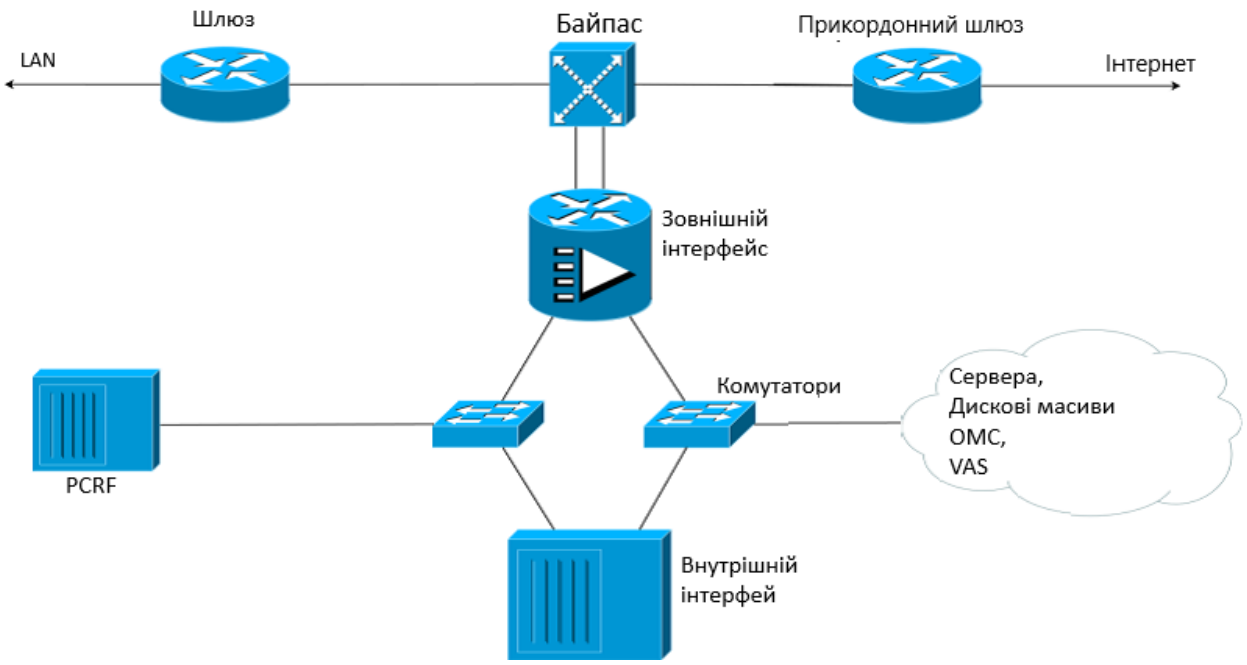


Рис. 6 - Схема використання системи DPI для застосування політик до мережевого трафіку.

Концепція "DPI як сервіс" може також розглядатися як відділення інфраструктурної частини аналізу мережевого трафіку від бізнес-логіки в рамках окремих прикладних завдань (збір статистики, міжмережевий екран, IDS/IPS системи та ін.). У наступному розділі буде розглянута схема роботи саме інфраструктурної частини аналізу, так як вона є, з невеликими варіаціями, ідентичною в різних рішеннях для аналізу мережевого трафіку. Зокрема, будуть виділені окремі етапи аналізу з коротким описом їх особливостей, а в наступних розділах кожен етап буде розглянуто більш детально.

2.3. Загальна схема інфраструктурних алгоритмів аналізу мережевого трафіку

Загальна схема аналізу мережевого трафіку складається з наступної послідовності кроків, кожен з яких призводить до підвищення рівня представлення об'єкта аналізу.

1. Захоплення пакетів, що проходять через контрольоване мережеве з'єднання. Результатом даного кроку є отримання об'єкта аналізу у вигляді

мережевих пакетів. Залежно від необхідної точності і швидкості подальшого аналізу, а також доступних обчислювальних потужностей можуть використовуватися різні підходи.

Слайсинг (slicing), при якому аналізу піддаються не весь вміст пакетів, а тільки деякий префікс (n перших байт). Ряд досліджень (наприклад,) показує, що цей підхід добре працює для подальшої класифікації трафіку за протоколами. У окремому випадку, якщо перехоплюваний розмір дорівнює сумарному розміру мережевих заголовків (L1-L3) є реалізацією технології SPI.

Самплінг (sampling), при якому перехоплюються не всі пакети, а тільки їх частина, яка може вибиратися за різними умовами, в залежності від потреб. У процесі розвитку технології було запропоновано велику кількість стратегій відбору. Наприклад, для завдань моніторингу типів трафіку підходить варіант з вибором кожного n-го пакету (uniform sampling), де n може вибиратися в залежності від співвідношення ширини каналу і пропускної здатності системи аналізу. Завдання отримання інформації про повний стан мережі за результатами самплінгу відома як inversion problem, зокрема, при застосуванні uniform sampling відбувається недооцінка середнього розміру пакетів, так як частіше будуть відбиратися пакети меншого розмір. Для передачі перехоплених даних використовується протокол PSAMP [2].

Нарешті, для завдань, в яких потрібно максимально точний аналіз трафіку, наприклад для систем забезпечення мережевої безпеки, потрібно перехоплювати всі дані всього надходить трафіку без втрат — для позначення цього підходу використовується термін lossless capture або deep packet capture (DPC).

2. Агрегування пакетів в потоки за деякими адресними ознаками (flow generaion), отримання нового об'єкта для аналізу — мережевого потоку. Якщо при цьому дані пакетів в подальшому аналізі не враховуються, то такий вид аналізу називається «аналіз потоків» - flow. based analysis (на відміну від packet-based аналізу, при якому аналізуються дані пакетів). На рис. 7 Показані відмінності типових схем packet і flow-based аналізу. Flow-based аналіз широко використовується в силу значно менших вимог до потужності обчислювача і пропускної здатності, за рахунок значного зниження обсягу даних для обробки. Такий вид аналізу може виконуватися як локально [43], так і віддалено від точки збору даних. Для передачі зібраних даних від точки збору

до точки аналізу використовується велика кількість протоколів, частина з яких стандартизована у вигляді IPFIX, а частина розроблена окремими виробниками — Cisco NetFlow, Juniper Jflow. В рамках підходу записи, що описують потік можуть містити різний набір даних.

Найбільш загальним набором таких даних є наступний: IP адреси джерела і адресата, протокол транспортного рівня, в разі протоколів TCP/UDP — номери портів джерела/адресата, набір лічильників: кількість переданих пакетів і байт, час створення і завершення потоку.

Слід зазначити, що хоча даний метод дійсно значно знижує вимоги до аналізатора, проте, він не є досить гнучким, так як на відміну від слайсинга і самплинга не дозволяє варіювати кількість даних, що надходять (воно залежить від вхідних даних). Більш того в більшості реальних завдань кількість потоків незначно менше кількості пакетів (приблизно на порядок) через велику кількість дуже коротких потоків, що складаються з декількох пакетів-flash flows. Для вирішення цієї проблеми було запропоновано використовувати самплінг для потоків .

Іншою особливістю даного методу є те, що, внаслідок обмеженості пам'яті, пристрій, що здійснює агрегацію пакетів, не може відстежувати один потік протягом довільного проміжку часу. Для вирішення цієї проблеми в конкретному рішенні зазвичай присутня настройка, що обмежує максимальну тривалість потоку (5 хвилин, у випадку Cisco NetFlow). Після закінчення цього часу вважається, що потік закінчився, і інформація про наступні пакети агрегується в рамках «нового» потоку. Дослідження точності flowbased підходу і впливу цього ефекту на точність аналізу міститься в роботі. Також в цій публікації описаний інструмент FLOW-REDUCE, що здійснює "збірку" повної інформації про потік з фрагментів, на які вона була розбита через обмеження за часом.

3. Виконання класифікації по протоколу прикладного рівня або конкретному мережевому додатку. Результатом даної операції є отримання нового об'єкта для аналізу — мережевого потоку конкретного протоколу або додатки (в цьому випадку пов'язаних потоків може бути кілька, наприклад, в разі VoIP Додатки це потоки SIP і RTP). Після виконання даної операції можлива наступна додаткова обробка отриманого об'єкта, конкретний вид якої залежить від розв'язуваної прикладної задачі:

– розбір полів протоколу (protocol parsing),
– збірка сесії протоколу для протоколів з встановленням з'єднання,
– витяг даних Додатки (content extraction) — сторінок сайтів (HTML), файлів різних типів (виконувані, зображення, текстові документи, і т. д.), електронних листів, аудіо-відео потоків і т. д.,
– розбір даних програми (application content parsing).

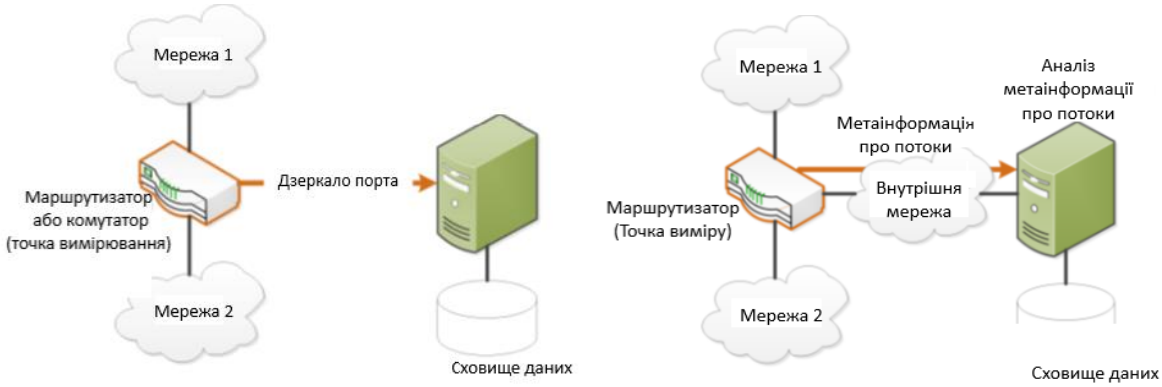


Рис. 7 - Відмінності типових схем packet (зліва) і flow-based (праворуч) аналізу.

Для повноти картини, слід сказати, що крім зазначених вище packetbased і flow-based підходів існує ще одне джерело даних про мережевий трафік — т.зв. база керуючої інформації (Manage Information Base, MIB) – віртуальна база даних, яка використовується для управління об'єктами в мережі зв'язку.

Модулі для накопичення, зберігання та обміну даними в форматі MIB реалізовані в більшості пристроїв. Передача даних здійснюється за протоколом SNMP. Дані одержувані таким шляхом мають низький обсяг і неспецифічні для протоколів. Наприклад, в рамках даного підходу, можна отримати відомості про загальну кількість пакетів і байт пройшли через конкретний мережевий інтерфейс конкретного мережевого пристрою. Слід сказати, що однією з причин розвитку MIB і flow-based підходів, незважаючи на їх порівняно низьку точність, послужила досі йде Глобальна дискусія про законність і допустимість глибокого аналізатрафіку з точки зору порушення безпеки, прав на приватне життя і т. д.

На даний момент одним із наслідків даної дискусії є, зокрема, те, що в наукових роботах, трафік, який піддається глибокому аналізу попередньо проходить процедуру «анонімізації» за допомогою спеціальних засобів [1]. Далі будуть більш детально розглянуті окремі кроки з наведеної загальної

схеми аналізу мережевого трафіку, методи, алгоритми і підходи, а також їх особливості та обмеження застосовності.

2.3.1 Захоплення мережевих пакетів

Програмні та апаратні засоби, що здійснюють захоплення трафіку відносяться до класу сніферів (sniffers). Для вирішення завдання захоплення трафіку можуть використовуватися як стандартні серверні мережеві карти, так і спеціалізовані мережеві карти, призначені для перехоплення трафіку на граничних швидкостях без втрат. Спеціалізовані карти, як правило, реалізовані на базі FPGA або ASIC і мають вбудовані засоби для проставлення тимчасових міток, апаратної фільтрації, зняття деяких заголовків низькорівневих протоколів, балансування навантаження між процесорами на багатопроцесорних комп'ютерах з урахуванням IP-потоків, виявлення помилкових і дублюються пакетів. При цьому вся обробка (в тому числі і копіювання даних в пам'ять комп'ютера з пам'яті мережевої карти) здійснюється без залучення ресурсів ЦПУ. У міру розвитку технологій багато з описаних властивостей реалізуються і на базі стандартних мережевих карт. Технологія реалізації таких додаткових функцій носить назву TCP Offload Engine (toe). Вона включає в себе наступні різні технології, базовими з яких є наступні:

- Large Segment Offload (LSO) або Giant send offload (GSO) - сегментація великих TCP-пакетів при відправці
- Large Receive Offload (LRO) - збірка приходять окремих мережевих пакетів у великі сегменти
- Checksum Offload-перевірка контрольних сум в заголовках IPv4, IPv6, TCP і UDP
- IP Security (IPSec) Offload-шифрування / дешифрування трафіку протоколу IPSec

Основною проблемою для стандартних мережевих адаптерів є не швидкість передачі даних, як така, а кількість пакетів в одиницю часу. Це обумовлено особливостями внутрішньої реалізації обробників пакетів на мережевих картах, драйверів мережевих карт і програмних мережевих стеків ОС. Внаслідок цього, стандартні мережеві карти без спеціалізованих драйверів

і мережевих стеків не забезпечують перехоплення трафіку без істотних втрат на швидкостях більше 3 Mpps (мільйонів пакетів в секунду). Причини такого обмеження будуть розглянуті нижче. Ще однією проблемою є точне проставлення тимчасових міток.

Проблеми, що виникають при переході до мережевих з'єднань, що підтримують більш високі швидкості передачі даних, пов'язані в основному з декількома факторами:

- Обмеженої пропускною здатністю апаратури.
- З архітектурними обмеженнями при взаємодії апаратури з ОС і ОС з призначеними для користувача додатками.
- Об'ємом пам'яті, необхідним для зберігання одержуваних даних.

Більшість поширених систем аналізу трафіку працюють, використовуючи бібліотеки Libpcap (ОС Linux) і WinPcap (ОС Windows). Дані бібліотеки працюють в режимі користувача. Для забезпечення своєї роботи з боку ОС вони використовують драйвери рівня ядра Berkeley Packet Filter (BPF) і Netgroup Packet Filter (NPF) відповідно. Основна різниця між цими драйверами полягає в схемі їх роботи з буферами пам'яті, що використовуються для тимчасового зберігання пакетів, одержуваних від мережевої карти. Драйвер BPF використовує схему подвійного буферизації, тоді як драйвер NPF використовує кільцевий буфер.

Серед проблем цих рішень, що призводять до зниження продуктивності можна виділити:

- Подвійне копіювання даних пакета (з карти в пам'ять ядра, з пам'яті ядра в пам'ять користувацького процесу).
- Велике число переривань від мережевої карти (на кожен пакет, щоб він був скопійований в буфер ядра).
- Велике число перемикань між режимами ядра і Користувача (на кожен пакет при його копіюванні в пам'ять користувальницького процесу).
- Недостатнє використання паралелізму на рівні окремих ядер і процесорів (за замовчуванням всі переривання обробляються одним ядром).

– Проблеми з синхронізацією при доступі до даних з декількох потоків виконання. У разі, якщо отримані дані повинні оброблятися в кілька потоків між цими потоками виникає ситуація змагання за ресурси.
– Залежно від кількості копіювань даних пакетів, які виконуються в процесі перехоплення, рішення поділяються наступним чином.

0-copy (zero-copy). Для реалізації підходу з нульовим копіюванням потрібна апаратна підтримка з боку мережевої карти-вона повинна містити власний DMA контролер, копіює дані з карти в пам'ять програми Користувача, без додаткового копіювання через пам'ять ядра. Прикладом може служити бібліотека PF_RING ZC в зв'язці з мережевими картами Intel або Napatech

1-copy. Для реалізації цього підходу можливі кілька варіантів-розробка аналізатора на рівні ядра, що є досить складним завданням або пряме відображення пам'яті ядра в пам'ять користувальницького процесу.

2-copy. Стандартне рішення на базі LibPcap або WinPcap.

Для вирішення перерахованих проблем було реалізовано деяку кількість спеціалізованих драйверів і мережевих стеків, до яких відносяться, наприклад, комерційне рішення Sniffer10G від Emulex і Myricom, а також відкрита розробка Pf_ring компанії Ntop. Ці рішення використовують схему з кільцевим буфером, як більш ефективну, а також оптимізовані для багатопроцесорних і багатоядерних комп'ютерів. Зокрема вони реалізують наступний функціонал:

– Обробка перехоплення пакетів з використанням великого числа ниток виконання (одна нитка на вхідну чергу).
– Балансування навантаження між ядрами (одне ядро – одна вхідна черга).
– Пакетна фільтрація всередині мережевої карти.

Для реалізації цих функцій використовується як апаратна підтримка з боку архітектури, так і підтримка з боку ОС (спеціалізоване API). Серед використовуваних технологій можна виділити наступні:

– Набір близьких технологій Interrupt Moderation, Adaptive Interrupt Moderation, Interrupt Coalescing, Interrupt Blanking, Interrupt Throttling, що дозволяють управляти затримкою доставки переривань за рахунок настроюваного таймера і обробляти отримання/відправку безлічі пакетів за одне переривання.

- MSI-X-розподіл I / O переривань по декількох процесорах і ядрах.
- New API (NAPI) - інтерфейс рівня ядра ОС Linux, що дозволяє застосовувати техніку зменшення кількості переривань (interrupt mitigation) з боку мережевих пристроїв.
- Receive-side Scaling (RSS) — технологія, що надає можливість динамічного балансування навантаження вхідних мережевих пакетів по декількох ядрах і процесорам (переривання надходять на різні процесори). Існують реалізації для масштабування на випадки більше 64 процесорів. Дана технологія підтримується в сімействі ОС Windows з появою Scalable Networking Pack. В ОС Linux аналог цієї технології називається Linux Scalable I / O.

Існує також ряд апаратних технологій від різних виробників процесорів, призначених для прискорення вводу/виводу.

- Intel Integrated I / O-технологія прямого підключення шини PCI Express 3.0 до процесора (без окремого PCI-контролера), реалізована в сімействі Intel Xeon E5
- Direct cache Access (DCA) - надання пристроям введення/виведення, таким як мережеві адаптери, можливості приміщення даних безпосередньо в кеш процесора Intel.

2.3.2 Класифікація мережевого трафіку

Тема класифікації мережевого трафіку сама по собі є дуже великою. Перш ніж переходити до методів, якими вона здійснюється, перерахуємо варіанти класифікації за її результатами, тобто об'єктам, які виходять на виході даного алгоритму, їх властивостям і можливостям їх подальшої обробки. За цим критерієм, можна виділити три основні варіанти класифікації. Далі вони перераховані в порядку збільшення» точності " класифікації:

- Тип трафіку не є достатньо змістовним способом класифікації і, як правило, або не піддається подальшому аналізу, або піддається досить простий додаткової уточнюючої класифікації. Залежно від сфери застосування, типи можуть бути різними. Серед прикладів, можна вказати:

- P2p, відео-стрімінг, веб-трафік - у випадку систем збору статистики та моніторингу,
- трафік мережевої атаки / нормальний трафік-у разі систем захисту від мережевих атак,
- трафік, що містить / не містить об'єкти копірайту, у разі систем контролю копірайту.
- Використовується протокол прикладного рівня (protocol identification) є досить змістовним і може, як використовуватися безпосередньо — наприклад, в системах збору статистики та моніторингу для підвищення рівня точності. Основним способом подальшої обробки є розбір протоколу, що включає два основних функції — збірка сесії прикладного рівня, в разі необхідності Витяг даних протоколу з окремих його полів (метаінформація рівня протоколу).
- Додаток, що передає дані (application identification), дає максимально деталізований рівень класифікації. На цьому рівні можуть здійснюватися ті ж види обробки, що і на рівні протоколу прикладного рівня, а також витягуватися і інтерпретуватися дані (метаінформація) конкретного додатка, що відповідає більш високому рівню їх подання. Наприклад, поле типу "рядок«, визначене на рівні протоколу, може відповідати» імені користувача" на рівні програми.

У різних прикладних завданнях результати ідентифікації протоколів і додатків можуть інтерпретуватися і, відповідно піддаватися різній подальшій обробці (як і в разі ідентифікації типу трафіку).

Наприклад, в разі системи захисту від шкідливого коду, під протоколом може розумітися командний (command-and-control, c&c) протокол ботнету, а під додатком — конкретний вірус. Відповідно, витягується метаінформація-команди ботнету, що передаються їм дані, а мета аналізу — з'ясування його функціоналу, оцінка поширеності і дослідження можливостей його деактивації.

У разі системи складання профілю користувача для подальшої демонстрації таргетованої реклами (наприклад, iMarker) в ролі протоколу може виступати HTTP, в ролі додатка — браузер, а об'єктом аналізу є запит

Користувача до пошукової системи, який піддається подальшому текстовому аналізу для вилучення ключових слів. Вибір конкретної прикладної задачі може значно впливати як на вибір алгоритму класифікації, так і на його параметри і продуктивність. Як приклад можна розглянути наступне порівняння. У разі системи статистики, алгоритм класифікації зазвичай працює послідовно на пакетах кожного потоку»до першого спрацьовування". Схема такої класифікації наведена на рис. 8.

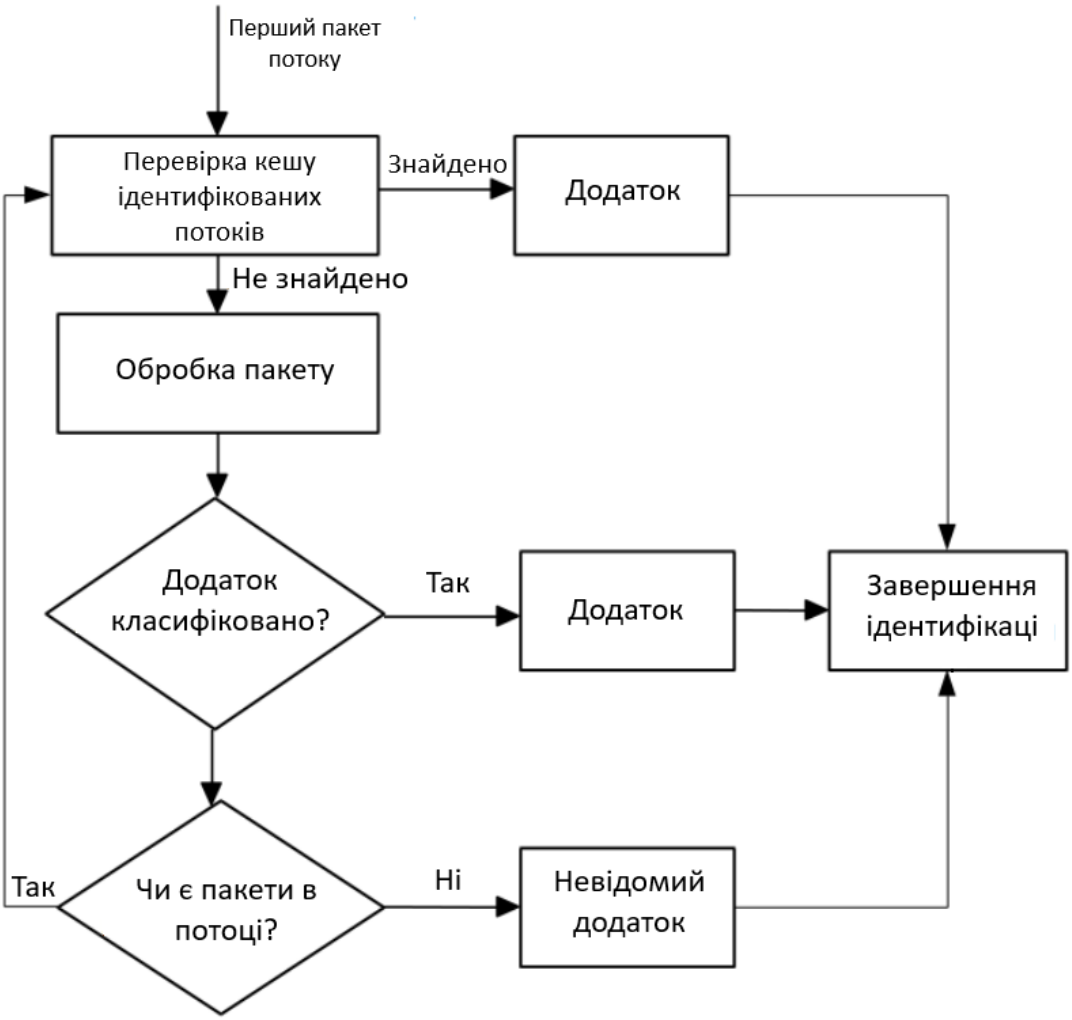


Рис. 8 - Схема класифікації «до першого спрацьовування».

У разі систем фільтрації за ключовими словами такий метод не підходить, так як в одному і тому ж мережевому потоці, в різних пакетах можуть зустрітися різні слова і, з точки зору системи класифікації, в цьому випадку даний потік потрапить відразу в кілька класів. У загальному випадку, очевидно, що перший підхід набагато продуктивніше, так як доводиться аналізувати значно менші обсяги даних. Крім того, в ряді підходів, для додаткового прискорення, аналізують не весь вміст пакета, а тільки деякий його префікс (за аналогією зі слайсингом). Наприклад, в роботі, для ідентифікації потоків, що містять шифровані і стислі дані, використовуються тільки перші 16 байт пакетів. У роботі проведено оцінку впливу розміру аналізованого префікса пакета на точність класифікації за протоколами та швидкість роботи класифікатора на трьох знятих мережевих трасах Unibs-GT, Polito, Polito-GT. Результати наведені на рис. 9, де на лівому графіку помилки класифікації позначені як misclassified, а трафік, який не вдалося класифікувати, як unknown.

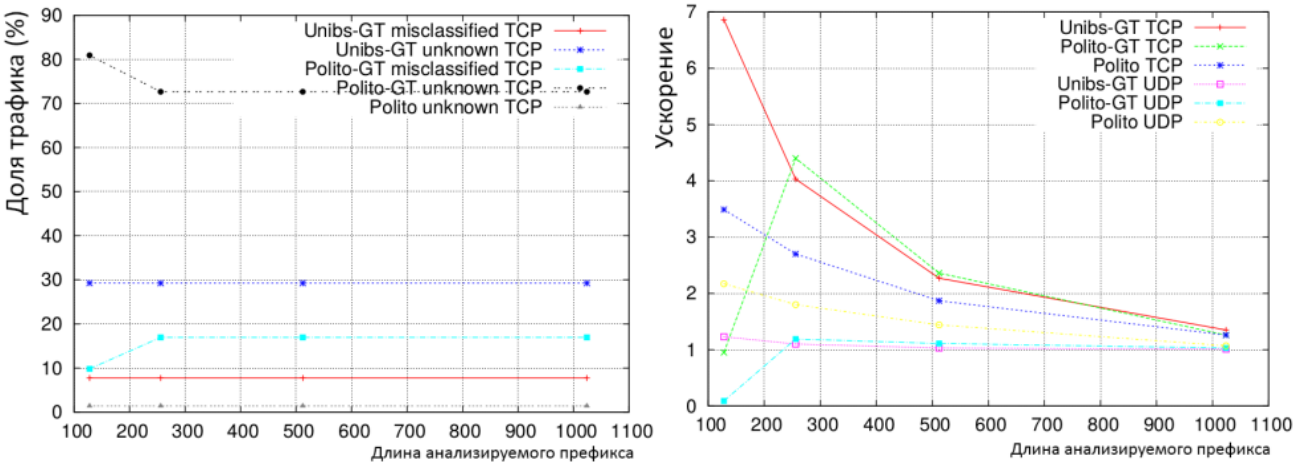


Рис. 9 – Оцінка впливу довжини на точність класифікації (ліворуч) та швидкість (праворуч).

На основі цих досліджень, зокрема робиться висновок про Надмірність проведення IP-дефрагментації і TCP-нормалізації при вирішенні даного завдання, так як дані алгоритми (особливо другий) досить ресурсоємні і практично не впливають на точність. Це відбувається через те, що для класифікації, як правило, використовується не більше 256 байт пакетів, а мінімальний розмір фрагмента зазвичай не менше 576 байт. Тобто, для даного завдання PBFS підхід більш кращий, ніж підхід MBFS (див.рис. 4).

Розглянувши види класифікації за одержуваними результатами і підходи в різних прикладних завданнях, перейдемо до розгляду конкретних алгоритмів класифікації.

Класичним підходом до класифікації є аналіз вмісту пакетів (payload-based). При цьому, як правило, виконується пошук т.зв. «сигнатур» (signature-based підходи) - характерних ознак, які заздалегідь створюються для кожного додатка або їх груп. Класифікація може виконуватися як на рівні окремих пакетів (stateless аналіз), або може враховуватися стан потоку (statefull аналіз). Для підвищення точності розпізнавання частина підходів використовує уточнені "сигнатури" на основі автоматів станів протоколів (див. 5). При такому підході, одержувані повідомлення, після їх класифікації, зіставляються з переходами в різних автоматах протоколів, і оцінюється коректність послідовностей таких переходів. Ця група підходів називається stateful Protocol Analysis Detection [1-2].

Як було показано на рис. 9, класифікація є найбільш навантаженим алгоритмом аналізу мережевих пакетів. Історично, через брак обчислювальних потужностей, робилися спроби досягнення збільшення продуктивності алгоритму за рахунок вибору джерела даних, використовуваних алгоритмом в процесі класифікації, таким чином, щоб оброблювані дані, будучи не менш інформативними, ніж вміст пакетів, були б більш компактні. Ця група підходів (на відміну від «сигнатурного») відноситься до класу «заснованих на виведенні» (inference based).

Висновки з науково-дослідної частини

З наведеного огляду можна зробити ряд висновків. Можна констатувати як кількісне (у зв'язку з ростами обсягів трафіку і ширини каналів зв'язку), так і якісне зростання (у зв'язку з новими прикладними завданнями) потреб у засобах аналізу трафіку. При цьому, незважаючи на величезне різноманіття конкретних рішень, що реалізують різні види аналізу, в основі більшості цих рішень лежить приблизно однакова схема, що добре видно на прикладі впровадження концепції «dpi як сервіс». У питаннях реалізації систем аналізу можна відзначити наступний ряд тенденцій:

- Розвиток спеціалізованих апаратних засобів (NP, tcam і т.д.), що дозволяють обробляти потоки даних в каналах максимально досяжною пропускної здатності.
- Розвиток програмних і програмно-апаратних технологій (NAPI, RSS, MSI-X, DCA і т.д.), що дозволяють обробляти потоки даних близько 10 Гбіт/з на стандартних серверних платформах.
- Перенесення (offloading) значної частини мережевої обробки безпосередньо на мережеві карти.
- Поява спеціалізованих мережевих стеків (в тому числі з відкритим вихідним кодом), що дозволяють здійснювати перехоплення без втрат на каналах 10 і більше Гбіт/с.це, в свою чергу, дозволяє реалізовувати досить потужні мережеві аналізатори на базі стандартних компонент, без використання дорогих спеціалізованих мережевих карт на базі FPGA і ASIC.
- Активні дослідження в області перенесення завдання класифікації мережевого трафіку на GPU в зв'язку з її високою ресурсоємністю і обмеженою кількістю обчислювальних ресурсів центрального процесора, навіть на багатопроцесорних системах.

Також можна бачити, що для кожного елемента загальної схеми аналізу в науковому, технічному та IT-спільнотах здійснюється пошук оптимальних рішень під конкретні прикладні завдання. У науковому співтоваристві вирішуються завдання пошуку оптимальних алгоритмів, наприклад для вирішення завдання класифікації, що є найбільш ресурсномісткою частиною будь-якої системи аналізу. У технічному співтоваристві здійснюється розробка апаратних засобів, що дозволяють забезпечити можливість вирішення прикладних завдань на каналах з постійно зростаючою пропускною здатністю. В IT-співтоваристві здійснюється синтез рішень, пропонованих двома іншими спільнотами, підбір конкретних параметрів алгоритмів для окремих прикладних завдань і пошук балансу, який зводиться, по суті, до вирішення оптимізаційних завдань в умовах великого числа змінних, серед яких можна згадати:

- необхідну повноту аналізу;
- необхідну точність аналізу;
- необхідну глибину аналізу;

- ціна конкретного програмно-апаратного рішення;
- продуктивність цього рішення;
- гнучкість рішення і його можливості по масштабуванню і нарощуванню функціоналу.

3 РОЗРОБКА АЛГОРИТМУ РОБОТИ ТА СТРУКТУРНОЇ СХЕМИ ПРИСТРОЮ

При виникненні тих чи інших неполадок адміністратор повинен мати у своєму розпорядженні всім необхідним, щоб оперативно визначити їх справжню причину. Програмне забезпечення аналізу мережевих протоколів, інстальоване на портативному комп'ютері, послужить непогану службу на стадії виявлення мережевих проблем, але виявиться абсолютно марною, коли ви намірилися встановити факт обриву одного з проводів в багатожильної кабелі, місце пошкодження ізоляції або дізнатися з чим пов'язано позаштатне поведінка якого-небудь мережевого адаптера.

За результатами аналізів і досліджень було проаналізовано безліч ідей, варіантів реалізації та обрано більш основні, що взаємо схожі з нашим проектом пристрою системи моніторингу мережевого трафіку на основі модуля Arduino наведеного на рис. 2.1. Модуль має низку переваг і приваблює наявністю мікроконтролера серії ATmega 328p, що дозволяє підключити рідкокристалічний дисплей. Однак, в залежності від типу корпусу модуль має 10-канальний або 6-канальний аналого-цифровий перетворювач. Дослідження використання зовнішнього АЦП показали, що частота дискретизації становить всього 8 кГц і тому дещо недоцільно використовувати його для цілей економії часу. Правда, при роботі перериваннями можливо частота 77 кГц.

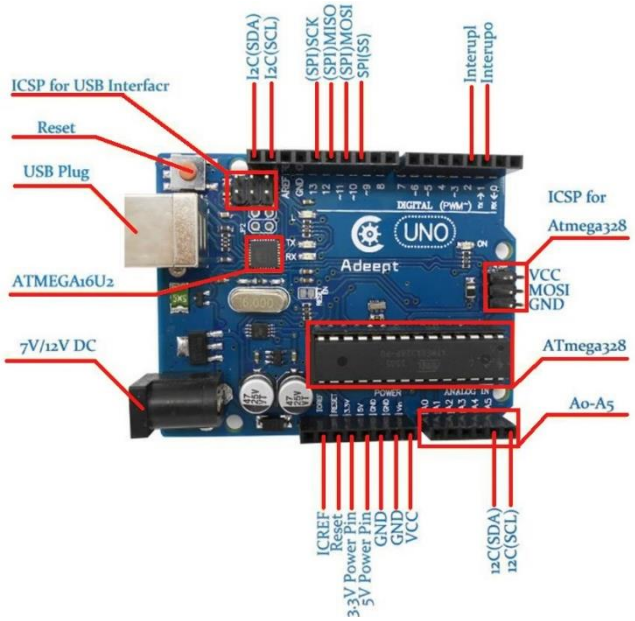


Рисунок 3.1 – Arduino модуль

3.1 Розробка алгоритму роботи приладу

У цьому розділі ми розробили блок-схему алгоритм якої показує роботу пристрою системи моніторингу мережевого трафіку.

Блок-схема по алгоритму роботи прибору представлена на малюнку 2.2.

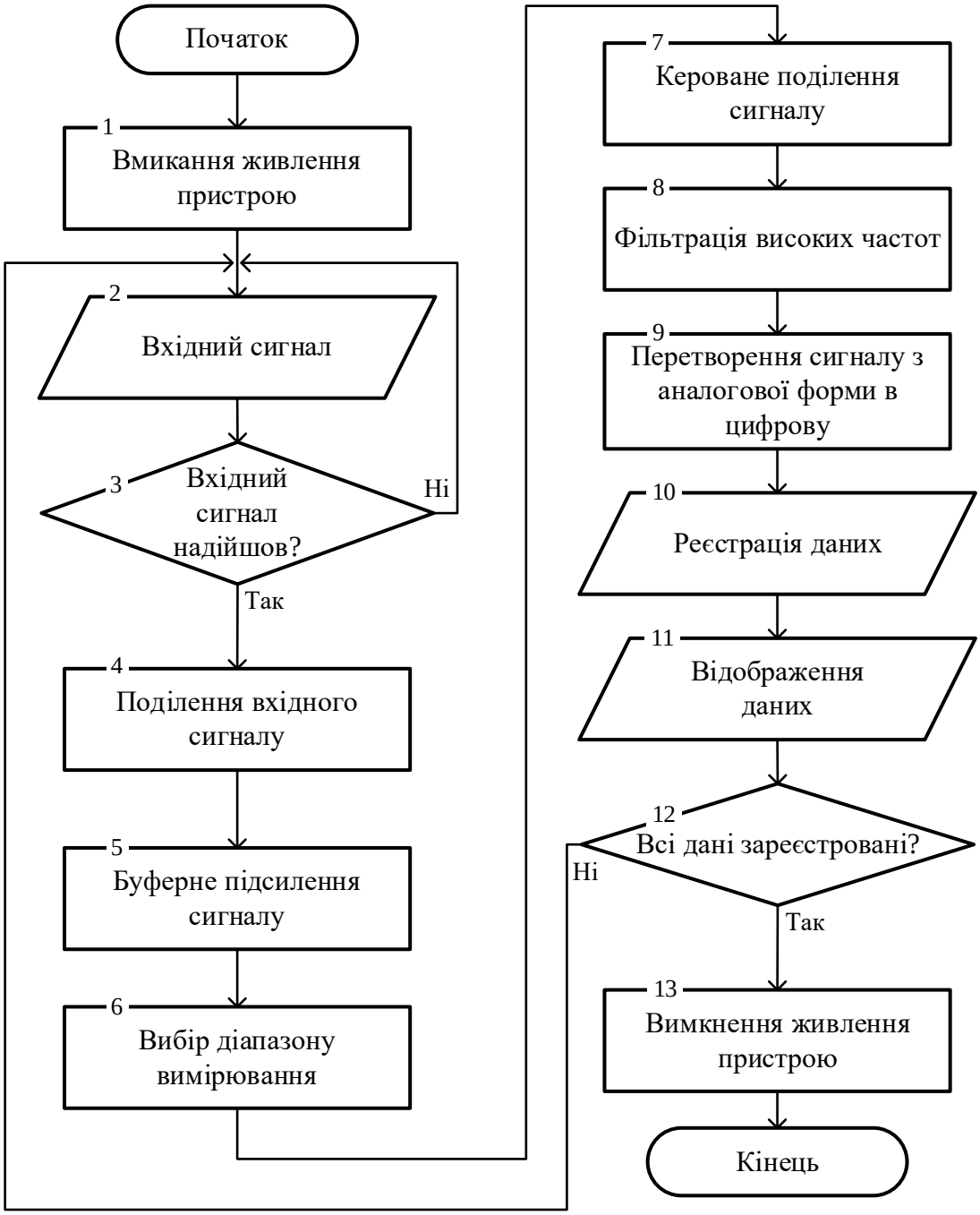


Рисунок 3.2 – Блок-схема алгоритм функціонування пристрою

Алгоритм роботи пристрою полягає у наступному:

Крок 1. Вмикаємо живлення пристрою.

Крок 2. На вхід пристрою подаємо деякий сигнал.

Крок 3. Перевіряємо, надійшов вхідний сигнал чи ні. Якщо ні, пристрій повертається до кроку 2, якщо так - переходить до роботи із сигналом.

Крок 4. Поділення вхідного сигналу напруги.

Крок 5. Буферне підсилення вхідного сигналу.

Крок 6. Вибір діапазону вимірювання сигналу

Крок 7. Кероване поділення сигналу.

Крок 8. Фільтрація високих частот.

Крок 9. Перетворення вхідної форми аналогового сигналу в цифоровий.

Крок 10. Реєстрація всіх даних.

Крок 11. Виведення даних на екран.

Крок 12. Перевірка, чи всі дані були зареєстровані. Якщо ні, пристрій повертається до кроку 2 та всі кроки знову повторюються, якщо так, переходить до наступного кроку.

Крок 13. Вимкнення живлення пристрою.

3.2 Розробка структурної схеми приладу

Аналізатори мережевих пакетів, або сніфери, призначені для вирішення проблем, які виникають у локальних мережах Ethernet. Вони повинні вміти перехоплювати, інтерпретувати та зберігати для подальшого аналізу пакети, що передаються по мережі. З одного боку, це дозволяє системним адміністраторам та інженерам служби технічної підтримки спостерігати за тим, як дані передаються по мережі, діагностувати та усувати проблеми, що виникають. У цьому сенсі пакетні сніфери є потужним інструментом діагностики мережевих проблем. З іншого боку, подібно до багатьох інших потужних засобів, що спочатку призначалися для адміністрування, з часом сніфери стали застосовуватися для забезпечення мережевої безпеки у локальних мережах.

Структурна схема системи моніторингу мережевого трафіку наведена на рисунку 3.3.

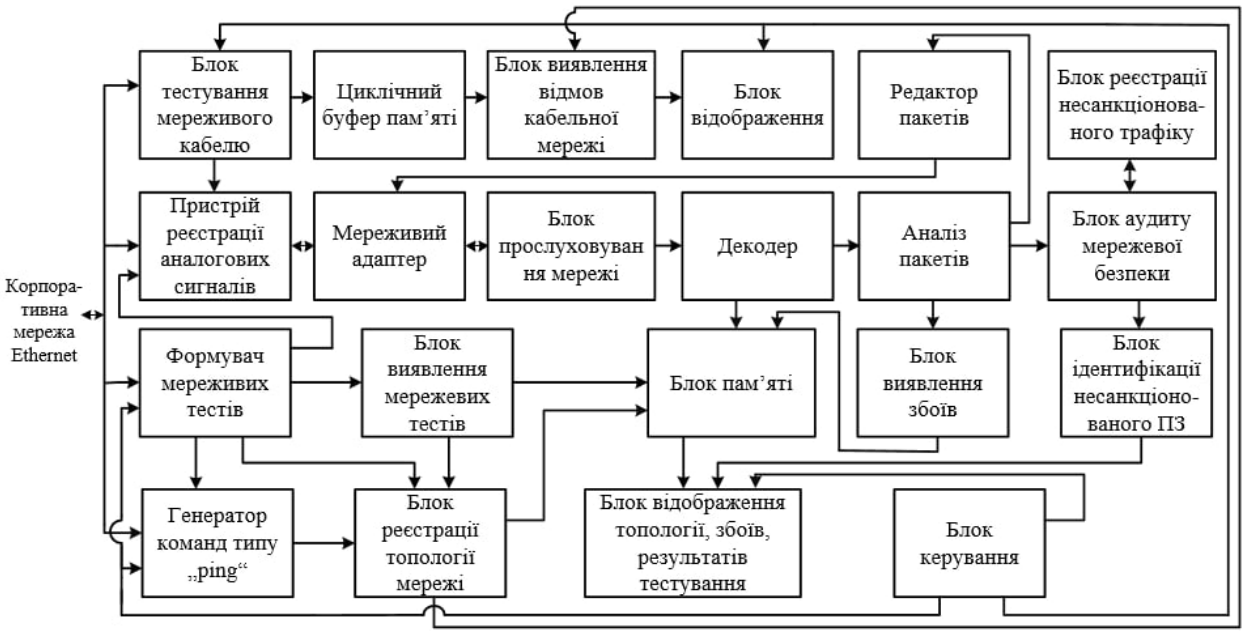


Рисунок 3.3 - Структурна схема системи моніторингу мережевого трафіку

Система моніторингу мережевого трафіку під'єднується до корпоративної локальної мережі Ethernet через пристрій реєстрації аналогових сигналів, який здійснює підсилення та нормування вхідного сигналу, аналого-цифрове перетворення та збереження значень сигналу, формування сигналів керування, формування зображення та передачу обробленої інформації до мережевого адаптеру.

Оскільки сніфери працюють на канальному рівні моделі OSI, вони не повинні діяти за правилами протоколів вищого рівня. Сніфери обходять механізми фільтрації (адреси, порти тощо), які драйвери Ethernet та стек TCP/IP використовують для інтерпретації даних. Пакетні сніфери захоплюють із дроту все, що по ньому приходить. Сніфери можуть зберігати кадри в двійковому форматі та пізніше розшифровувати їх, щоб розкрити інформацію вищого рівня, захована всередині (рис. 3.3).

Рисунок 3.4 - Схема роботи сніфера

Для того, щоб сніфер міг перехоплювати всі пакети, що проходять через мережевий адаптер, драйвер мережевого адаптера повинен підтримувати режим функціонування promiscuous mode («хаотичний» режим). Саме в цьому режимі роботи мережевого адаптера сніфер здатний перехоплювати всі пакети. Цей режим роботи мережевого адаптера автоматично активізується під час запуску сніферу або встановлюється вручну відповідними налаштуваннями сніфера. Весь перехоплений трафік передається декодеру пакетів, який ідентифікує та розщеплює пакети за відповідними рівнями ієрархії. Надана від декодеру інформація про пакети підлягає фільтрації та аналізу у відповідному аналізатору пакетів.

З точки зору мережевої безпеки, найбільшу небезпеку сніфери представляли тоді, коли інформація передавалася по мережі у відкритому вигляді (без шифрування), а локальні мережі будувалися на основі концентраторів (хабів). В даний час використання сніферів для отримання доступу до конфіденційної інформації завдання не з простих. Справа в тому, що при побудові локальних мереж на основі концентраторів існує певне загальне середовище передачі даних (мережевий кабель) і всі вузли мережі обмінюються пакетами, конкуруючи за доступ до цього середовища (рис. 3.4),

причому пакет, що посилається одним вузлом мережі, передається на всі порти концентратора і цей пакет прослуховують решту вузлів мережі, але приймає його тільки той вузол, якому він адресований. При цьому якщо на одному з вузлів мережі встановлено пакетний сніфер, то він може перехоплювати всі пакети мережі, що відносяться до даного сегменту мережі (мережі, утвореної концентратором).

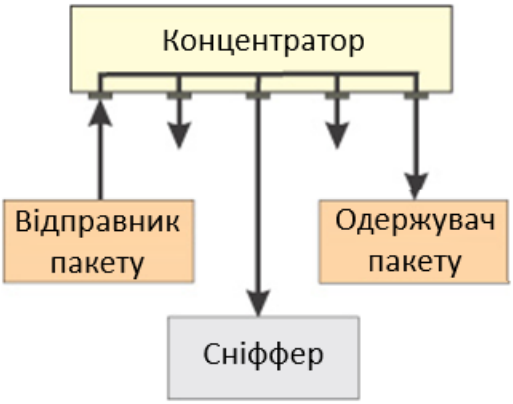


Рисунок 3.5 - При використанні концентраторів сніфер здатний перехоплювати всі пакети мережевого сегмента

Комутатори є більш інтелектуальними пристроями, ніж широкомовні концентратори, та ізолюють мережевий трафік. Комутатор знає адреси пристроїв, підключених до кожного порту, і передає пакети лише між потрібними портами. Це дозволяє розвантажити інші порти, не передаючи ними кожен пакет, як це робить концентратор. Таким чином, надісланий якимось вузлом мережі пакет передається тільки на той порт комутатора, до якого підключений одержувач пакета, а всі інші вузли мережі не мають можливості виявити цей пакет (рис. 3.6).

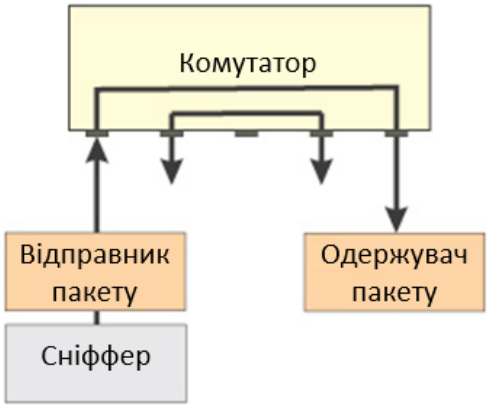


Рисунок 3.6 - При використанні комутаторів сніффер здатний перехоплювати лише вхідні та вихідні пакети одного вузла мережі

Тому якщо мережа побудована на основі комутатора, то сніфер, встановлений на одному з комп'ютерів мережі, здатний перехоплювати тільки ті пакети, якими обмінюється даний комп'ютер з іншими вузлами мережі. Незважаючи на те, що комутатори ізолюють мережевий трафік, всі керовані комутатори мають функцію перенаправлення або дзеркалювання портів. Тобто, за необхідності здійснення моніторингу всієї локальної мережі, вільний порт комутатора налаштовується таким чином, щоб на нього дублювалися всі пакети, що надходять до інших портів комутатора. Якщо в цьому випадку до такого порту підключити портативний мережевий аналізатор трафіку з пакетним сніфером, то він зможе перехоплювати всі пакети, якими обмінюються комп'ютери в даному мережевому сегменті.

Більше того, сніфери можуть успішно використовуватися не тільки для діагностики та локалізації мережевих проблем, але й для аудиту мережевої безпеки. Зокрема, застосування пакетних аналізаторів дозволяє виявити несанкціонований трафік за допомогою блоку реєстрації несанкціонованого трафіку, а за допомогою іншого блоку виявити та ідентифікувати несанкціоноване програмне забезпечення (ПЗ).

Аналізатор мережних пакетів працює на рівні мережного адаптера на канальному рівні та прихованим чином перехоплює весь трафік.

Основними складовими елементами системи моніторингу мережевого трафіку є:

- блок пам’яті для накопичення та обробки пакетів, у тому числі у вигляді найпростішого масиву із байтів;
- циклічний буфер пам’яті для збереження результатів тестування кабельної інфраструктури локальної мережі;
- аналізатор пакетів для керування фільтрацією зареєстрованих пакетів відповідно до заданого набору функцій та критеріїв;
- декодер-шифратор;
- редактор пакетів, який за командою вноситиме зміни в пакет і відправлятиме його назад до мережі.

Блок тестування мережевого кабелю, повинен надавати можливість щодо моніторингу стану кабельної інфраструктури локальної мережі Ethernet на рівні фізичної топології мережі та виконувати такі функції:

- визначення довжини мережного кабелю;

– виявлення обривів мережевого кабелю, замикання та порушення параметрів скручування кручених пар;
– визначення працездатності Ethernet-розеток та пристроїв, що знаходяться на протилежному кінці з'єднання;
– оцінка фізичної швидкості передачі лінії, її дуплексність, полярність, ідентифікатор мережного сегмента.

Формувач мережевих тестів дозволяю оцінювати стан локальної мережі на рівні логічної топології мережи Ethernet, який:
– виявляє мережні пристрої та виводить на екран список мережевих станцій;
– видає команду «ping» на пристрої, відстежує шляхи трафіку до цих пристроїв і опитує їх за протоколом SNMP;
– надає статистичні дані щодо функціонування мережі, генерує трафік заданої інтенсивності, виконує тести на звернення MAC-пакетів;
– визначає порт концентратора або комутатора, до якого підключений кабель, запалюючи світлодіодний індикатор порту;
– відстежує мережні помилки, включаючи появу значної кількості коротких пакетів та надмірну активність мережевих адаптерів;
– у графічному вигляді подає дані про рівень завантаження ресурсів, у тому числі локальних/віддалених пристроїв та мережі в цілому;
– ідентифікує найбільш активні мережеві вузли (з точки зору звичайної, багатоадресної та широкомовної передачі);
– функція виявлення пристроїв дозволяє ідентифікувати хост-комп'ютери, комутатори, маршрутизатори, концентратори та агенти, що підтримують протокол SNMP, мережні принтери та робочі станції;
– генерує трафік та використовує протокол RMON2 для збору даних про мережу у процесі складання карти її топології;
– перехоплює та декодує пакети.

На основі технічних завдань і критеріїв побудови аналогічних пристроїв, буде розроблена структура пристрою електронної системи моніторингу мережевого трафіку.

Проаналізувавши критерії, технічні та наукові джерела у попередніх підрозділах, структурна схема пристрою повинна відображати та містити: швидкодіючий АЦП для прийому і обробки інформації з входів і вимірювання тимчасових діаграм для відображення сигналів на екрані управління пристроєм. Це вимірювання повинно дозволити оцінити рівень напруги і якість фронту імпульсу при модуляції сигналу коду Manchester II. На основі цієї інформації пристрій повинен сформувати двійкову послідовність зареєстрованої інформації, а потім декодувати всі поля IP-пакета. Блок-схема, що узагальнює ці умови та ТС, показана на малюнку 3.7.

Поділ сигналу вхідної напруги виконується двома послідовно з'єднаними резисторами в співвідношенні 1:10. Це робиться для можливості досягнення вимірювання вхідних сигналів в межах, встановлених ТП.

Буферне посилення реалізовано за допомогою ОУ, включеного в режимі повторювача напруги. Потрібно для виконання функцій погодження вихідних опорів і опорів навантаження. Буфер посилення виконує ідеально функцію, як генератор напруги з вихідним нульовим опором по напрузі, тим же самим зменшуючи вихідний опір джерела.

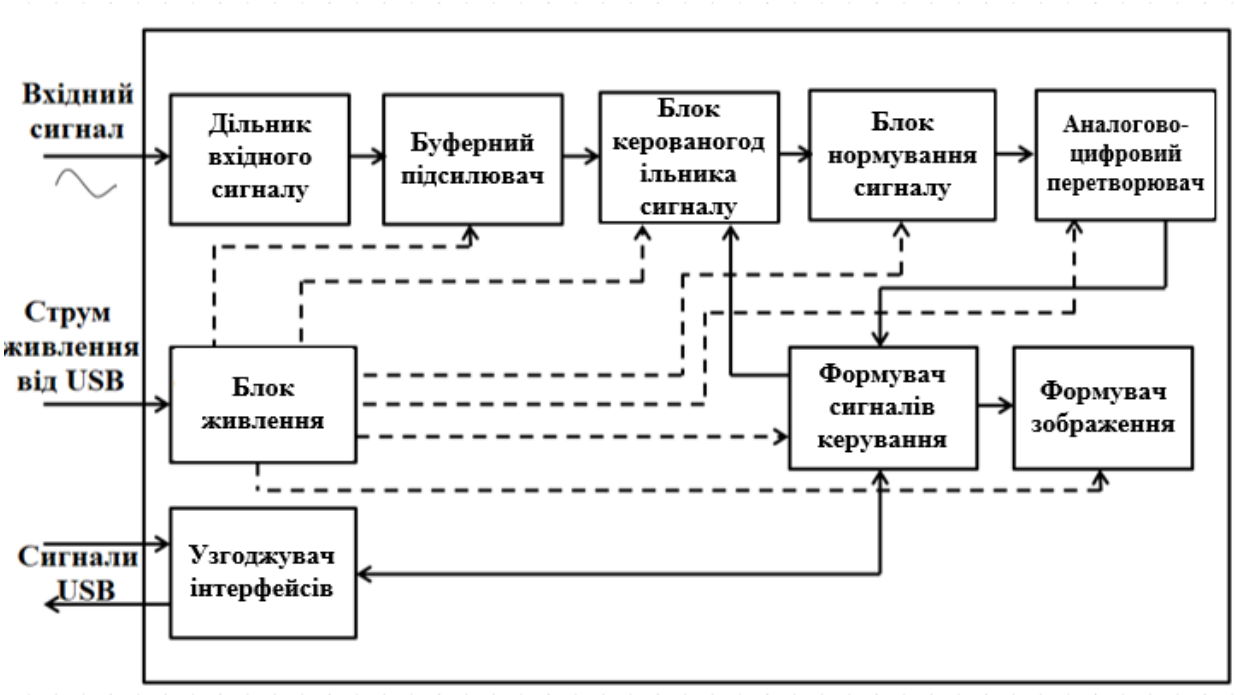


Рисунок 3.7 - Структурна схема приладу електронної системи моніторингу мережевого трафіку (блок реєстрації аналогових сигналів)

Кероване поділення сигналів здійснюється за допомогою аналогових мультиплексорів і дільника напруги з резисторів різного номіналу. Для розширення діапазону вимірювання і полегшення вибору скористаємося можливістю вибору коефіцієнта поділення напруг вхідних сигналів, керованого мікроконтролером. Поділ каналів здійснюється автоматично або за вибором користувача.[4].

Нормований коефіцієнт посилення досягається включенням операційного підсилювача в схему неінвертуючого підсилювача напруги. Він повинен виконувати високочастотну фільтрацію сигналу, обмежувати вироблений шум і збільшувати потужність посилення сигналів в 10 разів.

Аналого-цифрове перетворення здійснює вибір ряду вхідних сигналів і оцифровку сигналів перед подачею їх на плату управління.

Формування керуючих сигналів здійснюється за допомогою клавіатури і мікроконтролера, що виконує безліч функцій, серед яких: управління подільника, отримання даних і команд з клавіатури, їх обробка і виведення на екран з АЦП.

У Формуванні зображення використовуються екрани та світлодіоди для виконання функцій індикації. Екран управляється з МК і використовується для показу даних.

Живлення пристрою реалізує можливість забезпечити роботу зі стабілізатором рівня напруг під час роботи від зовнішнішніх джерел живлення і джерела живлення USB-порту, вибір джерел здійснюється за допомогою діода включеного між ними [3, 4].

4 РОЗРОБКА ФУНКЦІОНАЛЬНОЇ СХЕМИ ПРИСТРОЮ

Під час розробки прибору, було вирішено використовувати готовий мікропроцесорний модуль Arduino UNO R3 в поєднанні з мікроконтролером ATmega328, перетворювачем USB-UART і стабілізатором живлення в якості основи для його функціональності, універсальності і зручності пристрою.

Завдання полягало в збільшенні частоти дискретизації і динамічного діапазону вхідного сигналу. Ця проблема була вирішена застосуванням зовнішнього АЦП до вхідного блоку та використанням масштабування для управління вхідним сигналом.

Рішення всіх питань і відповідь на шляхи вирішення поставлених вище завдань дає функціональна схема пристрою електронної системи моніторингу мережевого трафіку, показана на малюнку 3.1.

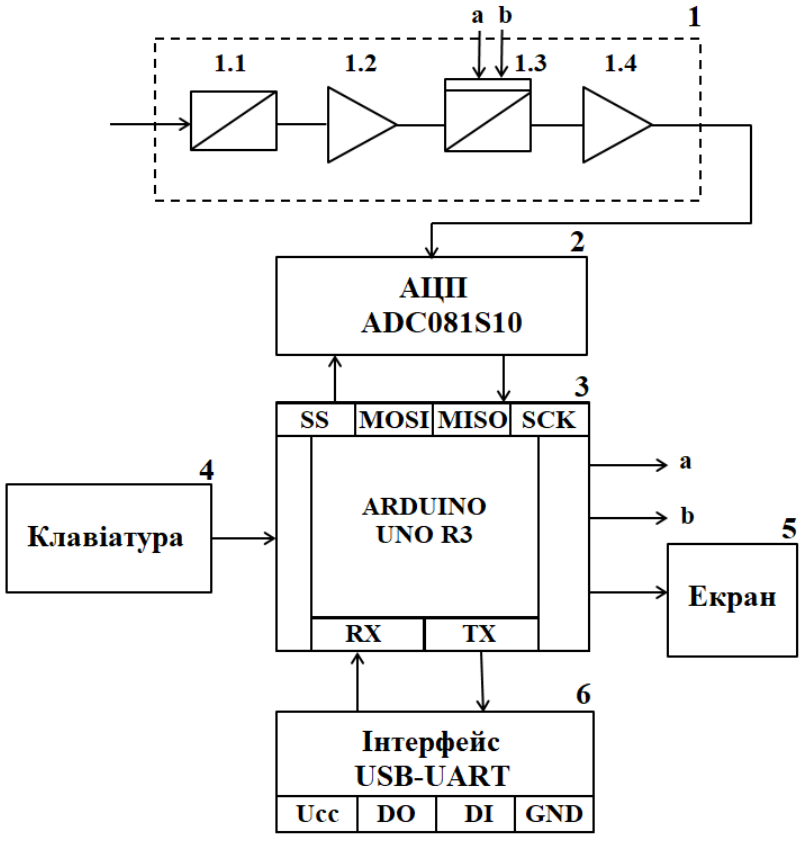


Рисунок 4.1 - Функціональна схема приладу електронної системи моніторингу мережевого трафіку

На рисунку 3.1 наведені наступні блоки:

1– блок попередньої обробки сигналу (1.1- подільник напруги, 1.2– буферний підсилювач, 1.3– керований подільник, 1.4– підсилювач нормування сигналу);

2– АЦП;

3– блок керування;

4– клавіатура;

5– екран;

6– перетворювач USB-UART.

Дана схема, що наведена на рис. 4.1 є неповним показом взаємодіючих блоків, оскільки модуль arduino не може покрити фактичні з'єднання між блоками, але це дозволяє нам програмувати процес роботи внутрішніх компонентів. Підсумовуючи результати і деталі дослідження даних елементів на здатність функціонувати в умовах заданих пристрою, в таблиці 4.1 були сформовані параметри приладу.

Таблиця 4.1 - Параметри приладу

Розрядність	8 біт
Частота дискретизації	1 МГц
Діапазон вимірювання напруги	від 0 до 50В
Напруга живлення	від 7 до 12В
Похибка вимірювання	± 5%
Максимальна частота вимірювання	200 кГц
Вхідний опір не меньше	100 кОм
Робота схеми пристрою	від -10 до +40°С

З рисунку 4.1 видно, що на вхід приладу надходить сигнал і обробляється блоком 1. Масштабування сигналів відповідає розрядній сітці АЦП. Тобто сильні сигнали напруги послаблюються дільником напруги з частотною корекцією, і навпаки, коли слабкі сигнали посилюються. Таким чином, блок 1 нормалізує вхідний сигнал відповідно до динамічного діапазону АЦП.

Дільник напруг 1.1 зменшує розмір вхідних сигналів у 10 разів, наближаючи його до вхідних діапазонів АЦП.

Наступний крок на шляху - буферний підсилювач 1.2. Він в собі має тільки одиничний коефіцієнт посилення і відправляє сигнали до керованого дільника напруг 1.3. Це дає можливість погодити кордони вхідних сигналів з діапазоном АЦП. Мікроконтролер допомагає здійснювати управління дільником через входи «a» і «b». Підсилювач нормування сигналів 1.4. виконує функцію підсилення сигналу та відфільтровування високих частот сигналу, щоб адекватно задовольнити вимоги перетворення.

На АЦП 2 покладено завдання оцифровувати і передавати сигнали на плату управління. Інтервали цифрових сигналів записуються в цифрову пам'ять. Пропускна здатність та швидкість запису вхідного сигналу визначає тактова частота АЦП.

Блок управління 3, що на рисунку 4.1 включає інтерфес та мікроконтролер. Призначений, для того щоб проводити налаштування приладу, аналізуючи параметри вхідного сигналу. Функція мікроконтролера полягає в управлінні блоком попередньої обробки сигналів 1, обробці команд з клавіатури 4, запису досліджуваних сигналів в оперативну пам'ять, обробці записаної інформації і відображенні її на екрані 5.

Панель управління необхідна для можливостей проведення налаштування приладу для проведення вимірів параметру сигналу. На панелі розташовані кнопки вибору режимів роботи, перемикач регулювання амплітуди вхідного сигналу і перемикач вибору частоти дискретизації при відображенні і запису на екран. Платформа Arduino зчитує, записує та надсилає дані, переглядаючи вхідні сигнали. На екрані відбувається показ даних і керування мікроконтролером.

Перетворювач інтерфейсу USB-UART забезпечує можливість оновлення програмного забезпечення мікроконтролера й синхронізацію при підключенні до ПК.

Обчислювальна апаратна платформа Arduino Uno містить в основі мікроконтролер ATmega328P із вбудованим завантажувачем. Що в свою чергу полегшує процес розробок апаратного забезпечення приладу і дає можливість вносити зміни в програмне забезпечення. З огляду на перераховані вище умови, можемо зробити висновки, що розрядність повинна бути 8 біт, а частота дискретизації АЦП – не менше 1 МГц. Також враховуються розміри корпусу мікросхем, наявність АЦП та ціна необхідної моделі. В результаті

обраний АЦП Texas Instruments ADC081S111. У таблиці 4.2 наведено характеристики АЦП.

Таблиця 4.2 - Основні характеристики ADC081S111

Напруга живлення	+2,7В ... 5,25В
Діапазон робочих температур	−40°С ...+85°С
Вхідна напруга	0В ... +5.25В
Розрядність	8 біт
Частота дискретизації	1 МГц

При виборі екрану ми враховували як його властивості, так і параметри, а саме:

- портативність;
- роздільна здатність;
- напруга живлення;
- низьке споживання струму;
- габаритні розміри екрану.
- швидкість відображення оновлення інформації;
- наявність вбудованих інтерфейсів (UART, SPI, I2C);
- доступність на ринку;
- ціна і можливісті швидкої заміни екранів.

Підводячи підсумки критеріїв, був обраний 1,3-дюймовий OLED-екранний модуль Waveshare, роздільна здатність якого 128×64, з подальшою заміною на більший розмір. Модуль екрану включає контролер SH1106, що полегшує розробку і дозволяє швидко і легко замінити модулі в разі несправності[9].

Ще одним критерієм вибору операційного підсилювача та мультиплексора було те, що смуга пропуску сигналу повинна у 10 разів перевищувати частоту дискретизації АЦП.

Оскільки мультиплексор є перемикачем вхідних діапазонів, то завдяки йому реалізований керований дільник напруги. Тому з урахуванням критеріїв були обрані аналоговий мультиплексор Analog Devices ADG704 і операційний підсилювач Analog Devices AD8033, які можуть виконувати функції повторювача для подачі сигналу на АЦП. Потім був обраний стабілізатор

напруги Texas Instruments UA78M05 за наступними критеріями: вихідна напруга має бути 5, що споживає блок пристрою.

Arduino Uno – це готова відкрита платформа, яка також поєднує мікроконтролер, стабілізатор живлення та перетворювач USB-UART. Це полегшує розробку, оновлення та налаштування пристроїв. Апаратне та програмне забезпечення електронної системи моніторингу інтернет трафіку, буде написано на мові програмування C++ в середовищі Arduino IDE [7]. Програма в свою чергу повинна мати можливість виконати задані функції приладу.

- отримання та оброблення інформації з входу АЦП;
- відображення сигналу на екрані;
- керування пристроєм.

Залежно від можливостей пристрою будуються програмні алгоритми, що описують основні можливості програмно-апаратного забезпечення.

5 РОЗРОБЛЕННЯ ЕЛЕКТРИЧНОЇ ПРИНЦИПОВОЇ СХЕМИ ПРИСТРОЮ

5.1 Вибір елементної бази

За функціональною схемою на рис. 4.1 і виходячи з умов приладу необхідно вибрати основну елементну базу і провести розробку електричної схеми пристрою електронної системи контролю мережевого трафіку за наступними параметрами:

- розрядність 8 біт;
- частота дискретизації 1 МГц;
- діапазон вимірювання напруги 0 … +50В;
- напруга живлення 7…12В;
- похибка вимірювання ± 5%;
- максимальна частота вимірювання 200 кГц;
- вхідний опір не менше ніж 100 кОм;
- робота схеми в температурному діапазоні від -10 до +40°С.

Пристрій в електронній системі моніторингу мережевого трафіку також має забезпечити, згідно з відповідними блоками на функціональній схемі рис. 2, функції:

- функцію відображення сигналу у діапазоні обраному користувачем;
- точність зображення у діапазоні обраному користувачем;
- обраний час індикації;
- відображення результатів на екрані;
- можливість керування.

Для побудови пристрою електронної системи моніторингу інтернет трафіка необхідно обрати набір мікросхем, відповідно на яких реалізуємо усі блоки пристрою. Велику кількість мікросхем було відібрано шляхом аналізу перерахованих вище умов і можливостей пристрою.

- мікроконтролер - ATmega328P-Atmel;
- АЦП - ADC081S101-TI;
- операційний підсилювач - AD8033-Analog Devices;
- мультиплексор - ADG704-Analog Devices;

- регулятор напруги - UA78m05-TI;
- перетворювач USB-UART - FT232RL-FTDIChip.

5.2 Мікроконтролер - ATmega328P-Atmel

Мікроконтролер ATmega328P є кращим із варіантів за параметрами для приладу. Мікроконтролер необхідний для роботи функцій управління блоком обробки попередньої сигналів, обробки з клавіатури команд, запису до оперативної пам'яті сигналів, які досліджуються й обробки і відображення інформації на нашому екрані.

Після аналізу ринку на наявність доступних мікросхем цієї серії, була обрана апаратно-обчислювальна платформа Arduino Uno R3. Ця платформа заснована саме на цьому мікроконтролері і включає в себе перетворювач USB-UART, бутлоадер (завантажувач) і стабілізатор живлення. Так що ніякий інший зовнішній програматор нам уже не буде потрібен. Це значно полегшить процеси розробки програмного забезпечення приладу та збільшує швидкість заміни деяких модулів, у разі необхідності. Виходячи з вищевикладеного, нам вигідно використати платформу на базі Arduino UNO R3 та створити пристрій на базі МК ATmega328. Його особливості та переваги:

- кількість RAM - 2048 байт ;
- максимальна швидкодія, 20 MIPS;
- кількість інтерфейсів (апаратна підтримка) – 4;
- струм спожвання (max), мА – 14;
- низька ціна.

Arduino Uno включає все в себе необхідне для кращої та необхідної роботи з мікроконтролером. Роз'єм для програмування внутрішньої схеми (ICSP) і кнопка скидання. 14 цифрових входів та виходів (6 з яких можуть бути задіяні як виходи ШІМ), 6 аналогових входів, роз'єм USB, кварцовий резонатор 16 МГц, роз'єм живлення, кнопка скидання, роз'єм внутрішньосхемного програмування (ICSP).

Мікроконтролер ATMega328P має 32 контакти, які показані на малюнку 5.1.

Мікроконтролер ATMega328P має 32 контакти в корпусі TQFP, як показано на рис. 5.2.

Напруга живлення, що подається на контакти VCC і ND мікроконтролера, не повинна перевищувати значення, зазначеного в технічному документі ATmega328P [5,6] - 5,5 В.

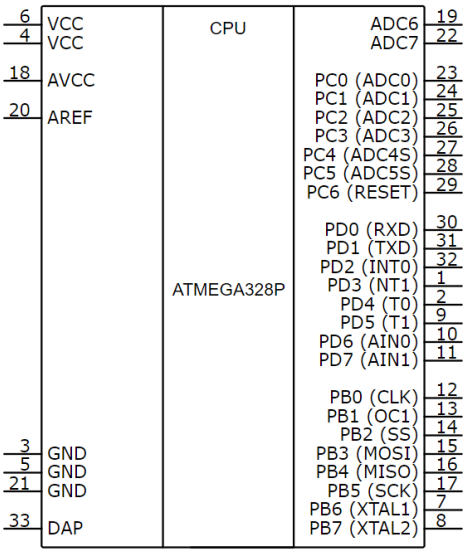


Рисунок 5.1 – Схема призначення контактів мікроконтролера ATMega328P

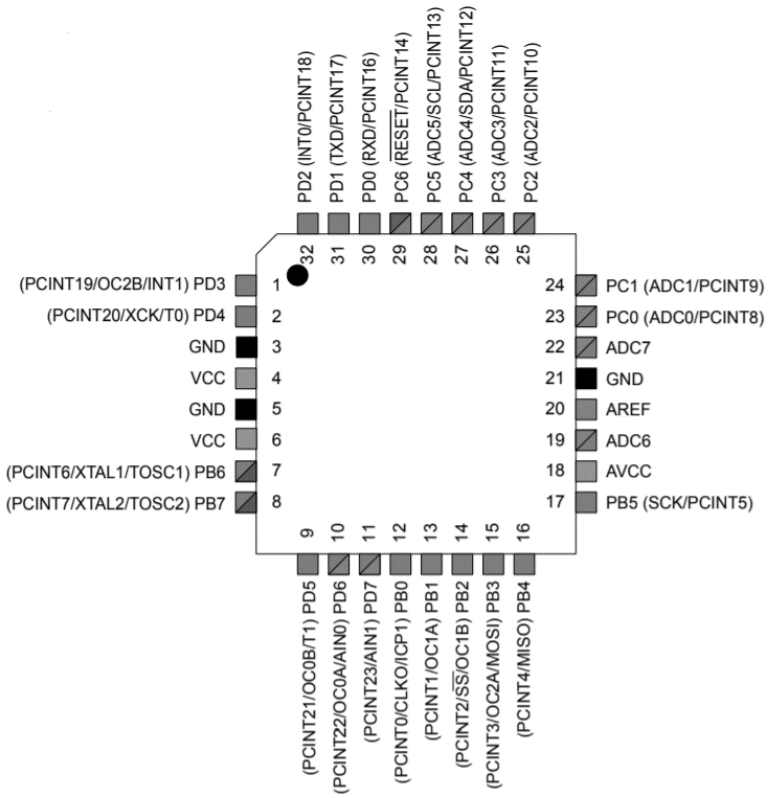


Рисунок 5.2 – Схема призначення контактів мікроконтролера ATMega328P у корпусі TQFP

Таблиця 1 – Призначення контактів ATMega328P

VCC і GND	живлення цифрових схем мікроконтролера;
AVCC і GND	живлення аналого-цифрового перетворювача;
Reset	генерація сигналу скидання мікроконтролера;
Порт B	має 8-розрядний двонаправлений порт і вміщує в собі навантажувальні резистори;
Порт C	має 8-розрядний вихідний порт. Порт C використовується для шини адреси;
Порт D	має 8-розрядний двонаправлений порт, який має вбудовані навантажувальні резистори;
XTAL1, XTAL2	вхід та вихід інвертуючого підсилювача тактової частоти генератора;
AGND	даний вивід повинен бути під’єднаний до окремого заземлення.
AREF	вхід опорної напруги для аналого-цифрового перетворювача. На даний вивід подається напруга у межах між AGND и AVCC;
TOSC1, TOSC2	вхід та вихід інвертуючого підсилювача;

Arduino Uno має відновлювальний запобіжник для захисту порту USB від замикань коротких та перевантажень. У випадку коли через USB-порт надходить струм більше 500 мА, запобіжник автоматично розриває з’єднання до тих пір, поки не буде усунена причина короткого замикання або перевантаження.

Фізичні характеристики

Максимальна ширина та довжина друкованої плати встановлена на 6,9 см і 5,4 см відповідно, при цьому роз'єми USB та живлення виходять з плати. Чотири монтажних отвори дозволяють монтувати плату на поверхню чи до корпусу [6].

Нижче наведені основні важливі розрахунки для створення електричної схеми.

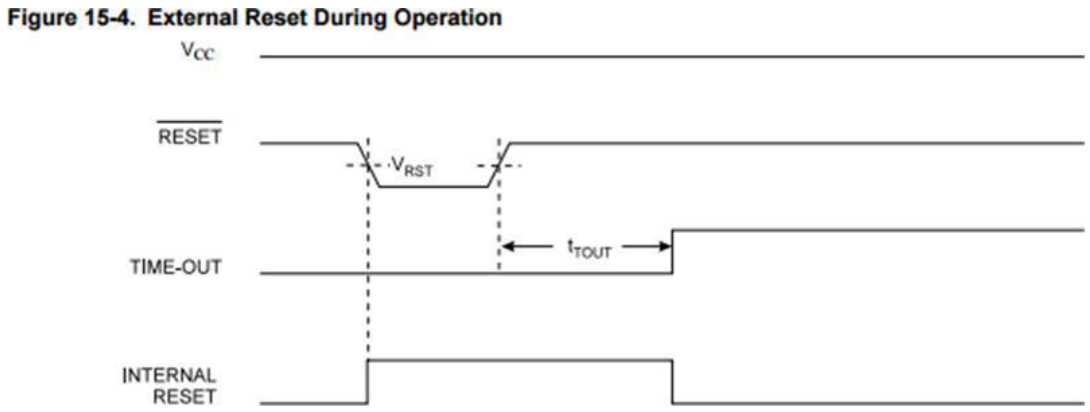


Рисунок 5.3 - Час потрібний мікроконтролеру для скидання

З рис. 5.3 видно, що час необхідний для скидання мікроконтролера, становить два цикли. Це пов'язано з тим, що частота, на якій працює мікроконтролер, становить f_{mk}= 16 МГц [6].

На рис. 5.4 показана схема скидання МК.

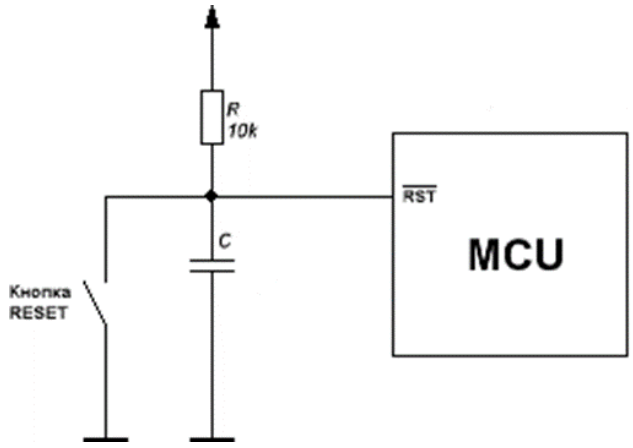


Рисунок 5.4 – Електрична схема скидання мікроконтролера

5.3 АЦП - ADC081S101-TI

Аналого-цифрові перетворювачі призначені для перетворення в цифрові сигнали аналогових. Такі перетворення вимагають квантування аналогових сигналів. Іншими словами, нам потрібно обмежити миттєве значення аналогового сигналу певним рівнем, який називається рівнем квантування. Ключові характеристики АЦП включають кількість бітів, час перетворення та нелінійність. ADC081S101-ti був обраний тому, що частота дискретизації АЦП становить 1 МГц або вище, а розрядність становить 8 біт.

Texas Instruments ADC081S111 - це однокристальна економічна мікросхема з аналогово-цифровим перетворенням (АЦП) з 8-бітовою роздільною здатністю і частотою дискретизації до 1 Млн/с. АЦП використовує архітектуру регістрів з послідовним наближенням з вбудованою схемою вибірки і зберігання. Послідовний інтерфейс сумісний зі стандартами, такими як SPI™, QSPI™, MICROWIRE™ і багатьма популярними стандартами DSP.

ADC081S101 використовує джерело живлення як джерело опорної напруги. В результаті допустима напруга на вході АЦП може змінюватися від 0 до +V сердечника. Частота перетворення задається послідовною тактовою частотою (SCLK). Перетворювач також має режим низького енергоспоживання в режимі очікування. Основні технічні характеристики представлені в таблиці 5.1. Це повністю задовольняє нашим умовам [9].

Таблиця 5.1 - Основні характеристики ADC081S10

Напруга живлення	+2,7В ... +5,25В
Діапазон робочих температур	−40°С ...+85°С
Межі вимірюваної напруги на вході АЦП (Верхня границя залежить від максимальної напруги живлення)	0В ... +5.25В
Розрядність	8 біт
Частота дискретизації	1 МГц

На рис. 5.5 показані умовне позначення і призначення контактів аналого-цифрового перетворювача ADC081S051.

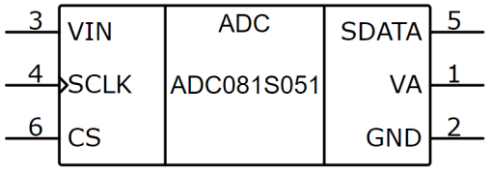


Рисунок 5.5 – Позначення та призначення контактів ADC081S051

Таблиця 1 - Призначення контактів на схемі рис. 5.5 ADC081S051

VIN	Аналоговий вхід. Цей сигнал може варіюватися від 0 В до VA;
SCLK	Вхід цифрового годинника. Цей годинник безпосередньо управляє процесами перетворення і зчитування;
SDATA	Виведення цифрових даних;
CS	Вибір чіпа. З нижнього краю CS починається процес перетворення;
VA	Позитивний вивід живлення. Цей вивіл повинен бути підключений до джерела від +2,7 В до +5,25 В і підключений до GND з конденсатором ємністю 1 мкФ і монолітним конденсатором ємністю 0,1 мкФ;
GND	Заземлення.

5.4 Операційний підсилювач - AD8033-Analog Devices

Операційний підсилювач Analog Devices AD8033 було обрано, оскільки він діє як підсилювач зворотного зв'язку напруги та буфер (ретранслятор).

AD8033 - це операційний одиночний підсилювач із зворотнім зв'язком по напрузі і високоефективним вхідним каскадом на польових транзисторах. Цей операційний підсилювач має набагато кращі характеристики, ніж аналогічні дешеві операційні підсилювачі на польових транзисторах: низький рівень власних шумів, що не перевищують 11 нВ/Гц-2 та 0,6 фА/Гц-2 й високу швидкодію, смуга пропускання якої 80 МГц і швидкість наростання вихідного сигналу 80) Вт/мкс. Широкий діапазон живлення від 5 В до 24 В крім того, AD8033 має вхідний динамічний діапазон, рівний напрузі живлення, що підвищує його універсальність [10].

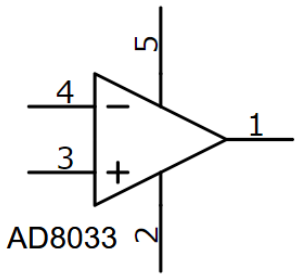


Рисунок 5.6 - Позначення операційного підсилювача Analog Devices AD8033 з призначення контактів

- Ширина смуги за рівнем -3 дБ: 80 МГц;
- Швидкість наростання: 80 В / мкс;
- Широкий діапазон напруг живлення від 5 В до 24 В;
- Низька напруга зміщення: 1 мВ;
- Високий коефіцієнт ослаблення синфазного сигналу: -100 дБ;
- Споживаний струм 3.3 мА / канал.

5.5 Мультиплексор - AG704-Analog Devices

Аналогові мультиплексори Analog Devics ADG704 призначені для виконання функції комутації аналогових сигналів.

ADG704 - це Аналоговий мультиплексор CMOS, що складається з 4-х незалежних каналів. Розроблений на основі передового субмікронного процесу, цей мультиплексор забезпечує швидке перемикання, низький опір, низький струм витоку і широку смугу пропускання при низькому енергоспоживанні. Забезпечує чудову лінійність і низький рівень комутаційних спотворень. ADG704 може працювати від одного джерела живлення в діапазоні від 1,8 В до +5,5 В, що робить його ідеальним для використання в обладнанні з батарейним живленням і ЦАП і АЦП нового покоління. ADG704 перемикає один з чотирьох входів на загальний вихід Wb, визначений 3-бітовим рядком двійкової адреси A0, A1 sat. Логічний " 0 " на виводі EN відключає пристрій. Кожен перемикач ADG704 працює однаково добре в обох напрямках при включенні [8].

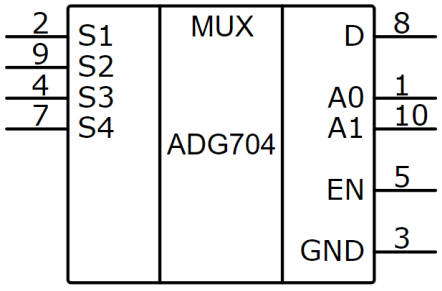


Рисунок 5.7 - Призначення контактів мультиплексора ADG704

Таблиця 1 - Призначення контактів ADG704

GND	Заземлення
S	Вивід джерела. Може бути вхід або вихід
D	Зливний вивід. Може бути вхід або вихід
A0, A1	Входи логічного управління
EN	Вхід логічного управління

5.6 Регулятор напруги - UA78m05-TI

Регулятор напруги Texas Instruments UA78m05 був обраний для регулювання і усунення проблем з шумом і розподілом, пов'язаних з регулюванням точки опори. Регулятор може забезпечити вихідний струм до 500 мА. Він за своєю природою стійкі до перевантажень з внутрішнім обмеженням струму і відключенням при перегріванні [7].

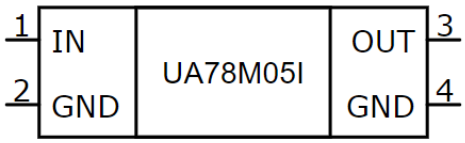


Рисунок 5.8 - Позначення регулятора напруги Texas Instruments UA78m05 з призначення контактів

5.7 Перетворювач USB-UART - FT232RL-FTDIChip

Мікросхема FT232RL, що на рис. 5.9, представляє універсальний асинхронний приймач, що має високо інтегрований послідовний інтерфейс USB-COM.

FT232RL дозволяє легко обмінюватися даними між будь-яким зовнішнім пристроєм на мікроконтролері і комп'ютером через USB-порт останнього, використовуючи мінімум додаткових компонентів (потрібно тільки USB-роз'єм і кілька резисторів). може бути організована в мікросхема вже містить тактовий генератор, пам'ять EEPROM і може працювати в режимах послідовного обміну даними і бітового вибуху [12].

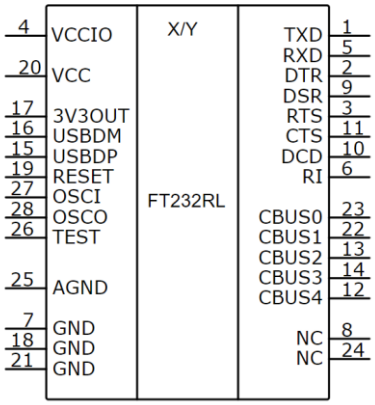


Рисунок 5.9 – Схема виводів універсального приймача FT232RL

Таблиця 1 - Призначення контактів FT232RL

USBDP	USB- сигнал передачі даних плюс, що включає внутрішній послідовний резистор і підтягуючий;
USBDM	USB-сигнал передачі даних мінус, що включає внутрішній послідовний резистор;
VCCIO	живлення інтерфейсу UART від + 1,8 В до +5,25 в
GND	заземлення;
3V3OUT	вихід + 3,3 В від вбудованого регулятора LDO;
VCC	живлення ядра пристрою від +3,3 В до + 5,25 В;
AGND	пристрій аналогового заземлення для внутрішнього тактового помножувача;
NC	немає внутрішнього підключення;
RESET	вхід скидання;
TEST	перехід пристрою в режим тестування IC;
OSCI	вхідний осередок генератора 12 МГц;
OSCO	вихідний осередок генератора 12 МГц
TXD	передача асинхронного виведення даних;
DTR	термінал виходу управління / сигналу;
RTS	запит на відправку керуючого вихідного сигналу;
RXD	прийом асинхронного введення даних;
RI	вхід управління кільцевим індикатором;
DSR	вхід управління;
DCD	вхід управління виявленням носія даних;
CTS	очищення для відправлення сигнал керуючого входу;
CBUS	настроюваний контакт вводу-виводу CBUS.

5.8 Вибір екрану

Екранний модуль 1,3 IPS OLED (B) був обраний виходячи з портативності пристрою і декількох параметрів. Частота оновлення зображення, ціна і можливість швидкої заміни екрану в разі виходу з ладу.
1,3-дюймовий OLED-екранний модуль Waveshare (B)-це доступний за ціною OLED-дисплей з діагоналлю 1,3 дюйма і роздільною здатністю 128x64, розроблений Waveshare. Дисплейний модуль має два своїх інтерфейсу: SPI і S2Сю. У цьому випадку ми рекомендуємо використовувати 4-провідний варіант SPI. Дисплей управляється драйвером SH1106, розробленим SINO WEALTH. Колір дисплея синій. Шилд підключається до джерела живлення +5B і землі мікросхеми Arduino UNO, а також до вихідних портів мікроконтролера ATmega328P, а саме PC5 і PC4 [11].

Таблиця 5.2 - Основні характеристики екранного модуля 1.3inch OLED (B)

Напруга живлення	+3,3B ... +5B
Інтерфейси 3-wire	3-wire SPI, 4-wire SPI, I2C
Діагональ екрану	1,3 дюйми
Роздільна здатність екрану	128 на 64 пікселів
Габаритні розміри	40.50 мм на 37.50 мм
Діапазон робочих температур	−30°C ...+70°C

5.9 Вибір типу резисторів

При виборі резисторів враховуємо наступні параметри:

- номінальний опір;
- розсіювана потужність;
- допуск;
- робоча температура;
- тип резистора;
- корпус, тип монтажу резистора;
- максимальна робоча напруга.

Виходячи з наших вимог, що поставлені вище, у пристрої використаємо корпуси типу SMD 0805 (мікросхемні резистори). Розміри таких резисторів менше, ніж у інших резисторів, що дозволяє зменшити вагу і габарити, а також зберегти максимальну потужність, що скаладає 0,125 Вт, достатню для роботи приладу. Для того, щоб досягти високої точності вимірювань в пристрої використовуються резистори серії Е-96 з точністю ±1%, робочою температурою від -55°С до +125°С і максимальною робочою напругою 150в.

5.10 Вибір типу конденсаторів

Під час підбору конденсаторів були враховані параметри для оптимальної роботи, а саме:

- номінальна ємність конденсатора;
- робоча напруга конденсатора;
- тип конденсатора;
- робоча температура;
- допуск.

Виходячи з параметрів, вимог та характеристик, для використання були обрані мікросхеми конденсаторів SMD 0603. Цей тип конденсатора має невеликі розміри і вагу, параметри підходять для портативних пристроїв. Максимальна напруга при роботі для цих складських керамічних конденсаторів становить 50 В, що є більш ніж достатньо для роботи пристрою. Коефіцієнт температурний ємності обраний зі значенням x7r, що має хороші температурні параметри (підходить для портативних пристроїв, так як його властивості істотно не змінюються з температурою). Робоча температура в межах від -55 ° С до + 125 °С та не значна похибка в розмірі ±10%. Всі ці параметри відповідає вимогам ТЗ.

5.11 Вибір діодів та світлодіодів

У дисплеї для індикації використаємо світлодіод KLS9-t0805ugc з малим споживанням струму і необхідною яскравістю. На схемі підключення в додатку Б світлодіоди позначені як VD1…VD4. Опишемо призначення кожного окремо світлодіода. VD1- вмикається якщо присрій заряджається або

працює від мережі; VD2- вмикається при роботі пристрою; VD3-VD4 – по черзі працюють, якщо пристрій отримує сигнал та виконується аналіз.

Діоди вибрані згідно функції, що виконують в пристрої, і їх функції по захисту від перевищення допустимого рівня напруг. При цьому основними критеріями були швидкість відкриття каналу, напруга, напруга зворотного пробою діода і ємність діода. Підібрано діод Шотткі BAT 30 із забезпеченням умов і зазначених критеріїв.

5.12 Вибір кнопок

Залежно від умов була обрана кнопка DTSM-32 з робочою напругою 12 В в корпусі для поверхневого монтажу. Кнопки мають механічне навантаження і були обрані через їх надійності при навантаженні. Пояснимо функцію кожної окремо кнопки. SB1 – включення і виключення пристрою, SB2 – кнопка вибору (selest); SB3 – SB4 – кнопки вгору і вниз для вибору режиму роботи; SB 5 – SB 6 – кнопки для збільшення або зменшення розгортки модуляції сигналу.

6 ТЕХНІКО-ЕКОНОМІЧНА ЧАСТИНА

6.1 Розрахунок повної собівартості пристрою

Собівартість пристрою - це поточні витрати компанії на його виробництво і продаж, виражені в грошах. Витрати на виготовлення пристрою формують собівартість, а витрати на виробництво і маркетинг формують загальну собівартість. Калькуляція - розрахунок вартості обладнання за витратами.

Витрати, що пов'язані з виготовленням і продажем пристрою, згруповані за наступними статтями:

- комплектуючі та матеріали;
- заробітна платня;
- витрати на експлуатацію та технічне обслуговування обладнання;
- маркетингові витрати.

Вартість матеріалів і комплектуючих розраховується з ціни за одиницю матеріалів/комплектуючих і їх потрібної кількості. Вартість цін на комплектуючі та матеріали слід взяти з даних (сайтів, каталогів, прайс-листів) від виробників та постачальників деталей, матеріалів та послуг

Розрахунок вартості компонентів показаний в таблиці 6.1, а витрати на матеріали та сировину - в таблиці 6.2.

З урахуванням заготівельно-транспортних витрат (kt-з=12%), витрати на матеріали та комплектуючі складають:

КМ=(К+М)·(100+kт-з)/100

КМ=(1795,5+182)·(100+12)/100=2215(грн.)

Таблиця 6.1 – Вартість затрат на комплектуючі

№ з\п	Назва комплектуючого	Кількість, шт	Ціна за од., грн	Вартість, грн
Мікросхеми				
1	ATmega328P-Atmel	1	600	600
2	ADG704-Analog Devices	1	250	250
3	AD8033-Analog Devices	2	78	156
4	UA78m05-TI	1	65	65
5	ADC081S101-TI	1	140	140
6	FT232RL-FTDIChip	1	260	260
Резонатори				
7	Кварцовий резонатор HC49	1	7	7
Конденсатори				
9	Murata-TZB4P400BA10R01-8,5-40пФ	1	30	30
10	Murata-SMD0603-X7R-2200пФ-50В±10%	1	1	1
11	Murata-SMD0603-X7R-0,1мкФ-50В±10%	9	1,5	13,5
12	Murata-SMD0603-X7R-300пФ-50В±10%	1	1	1
13	Murata-SMD0603-X7R-1мкФ-50В±10%	1	1	1
14	Murata-SMD0603-X7R-1000пФ-50В±10%	1	1	1
15	Murata-SMD0603-X7R-4700пФ-50В±10%	2	1	1
Діоди				
	Світлодіод KLS9-T0805UGC	4	3	12
	Діоди BAT30	3	20	60
Резистори				
	ROYALOHM-SMD0805-499Ом-0,125Вт±1%	1	10	10
	ROYALOHM-SMD0805-300Ом-0,125Вт±1%	1	2	2
	ROYALOHM-SMD0805-100Ом-0,125Вт±1%	2	2	4
	ROYALOHM-SMD0805-511кОм-0,125Вт±1%	1	2	2
	ROYALOHM-SMD0805-487кОм-0,125Вт±1%	1	2	2
	ROYALOHM-SMD0805-100кОм-0,125Вт±1%	1	2	2
	ROYALOHM-SMD0805-11кОм-0,125Вт±1%	1	2	2

	ROYALOHM-SMD0805-1кОм-0,125Вт±1%	3	2	6
	ROYALOHM-SMD0805-4,02кОм-0,125Вт±1%	1	2	2
	ROYALOHM-SMD0805-4,99кОм-0,125Вт±1%	1	2	2
	ROYALOHM-SMD0805-93,1Ом-0,125Вт±1%	1	2	2
	ROYALOHM-SMD0805-56Ом-0,125Вт±1%	4	2	8
	ROYALOHM-SMD0805-10кОм-0,125Вт±1%	6	2	12
Роз'єми				
	XH 2,54 2Y	1	5	5
	XH 2,54 8Y	1	9	9
	XH 2,54 2Y	1	5	5
	KLS1-229-5FB-B	1	14	14
Кнопки				
	Кнопка DTSM-32	6	18	108

Загальна ціна комплектуючих складає: 1795,5 грн.

Таблиця 6.2 – Вартість затрат на матеріали

Матеріал, сировина	Одиниця виміру	Норма витрати	Ціна за одиницю, грн	Вартість, грн
Сировина для корпусу	кг	0,4	300	120
Припій	кг	0,05	100	5
Флюс	кг	0,07	500	35
Провід монтажний	м	0,4	25	10
Лак	кг	0,1	120	12
Сумарні витрати				182

6.2 Вартість затрат на заробітну плату (ЗП):

$$ЗП = \sum_{i=1}^{n} Тг_i \cdot Нч_i,$$

де Тгі – погодинна ставка для спеціалістів (електронщиків, лаборантів та ін.), які задіяні у виготовленні пристрою, грн/год;

Нчі – час витрачений працівником на виготовлення та налаштування пристрою, год. n - кількість робітників, приймаючих участь у виготовленні пристрою;

$$ЗП = \sum^{2} 40 * 24 = 1920 \text{ (грн)}$$

Погодинна ставка, яка розраховується, на основі розміру місячної заробітної плати спеціаліста:

$$Тг_i = \frac{Тм_i}{Вф_i \cdot 10},$$

де $Тм_i$ – місячна ставка (оплата) спеціаліста, грн;

$Вф_i$ – фактичний час, який відпрацьований за розрахунковий місяць (період). 8- кількість годин, відпрацьованих за зміну.

Тг=7040/22*8=40(грн/год)

6.3 Витрати на утримання і експлуатацію устаткування.

У разі, якщо устаткування перебуває на балансі підприємства витрати на утримання і експлуатацію устаткування(ВУЕУ) = основна зарплата, приймаємо 50%.

ВУЕУ =1920 · 0,5 = 960 (грн.)

6.4 Відрахування на соціальні заходи

Відрахування на соціальні заходи містять відрахування від суми основної і додаткової зарплати за встановленими ставками:

- на обов'язкове державне пенсійне страхування;
- на державне страхування від нещасних випадків;
- на обов'язкове державне соціальне страхування на випадок безробіття;
- у зв'язку з тимчасовою втратою працездатності і витратами, зумовленими народженням дитини і похованням

$$В_{соц} = З_о \cdot \frac{38,52}{100} \qquad (6.5)$$

Всоц=1920*38,58/100=740(грн.)

6.5 Витрати на збут

Включають витрати на рекламу та передреалізаційну підготовку пристрою. Орієнтовно ці витрати визначаються в розмірі 5-10% від виробничої собівартості.

5095*0,07 = 356,65(грн.)

Рс=5095+356,65=5451,65 (грн.)

де, Рс – повна собівартість.

Таблиця 6.3 – Калькуляція собівартості пристрою (установки)

№	Найменування статей калькуляції	Проектний варіант
1.	Основна заробітня плата	1920
2.	Відрахування на соціальні заходи	740
3.	Витрати на утримання і експлуатацію устаткування:	960
4.	Матеріали та комплектуючі	2215
Виробнича собівартість		5835
5.	Витрати на збут	356,65
Повна собівартість		6191,65

6.6 Висновки з техніко-економічної частини

Здійснено ряд організаційно-технологічних заходів щодо зниження трудомісткості продукції та підвищення продуктивності праці за рахунок скорочення чисельності працюючих на підприємствах. Використання нової

технології, точні розпили, заміна морально і фізично застарілого обладнання на більш технічне і високопродуктивне. Пропоновані заходи дозволять зробити даний вид продукції більш конкурентоспроможним за рахунок зниження собівартості і відпускної ціни товару.

З огляду на розраховані параметри, можна зробити висновок, що розробка і впровадження даного пристрою має техніко-економічний сенс. Перевага системи перед маршрутизаторами полягає у використанні таблиць динамічної маршрутизації в розподілених мережах. Що стосується ціни системи, то переведення її на автоматизоване виробництво може знизити її на порядок.

7. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПРИСТРОЮ ЕЛЕКТРОННОЇ СИСТЕМИ МОНІТОРИНГУ МЕРЕЖЕВОГО ТРАФІКУ

7.1 Розробка програмного забезпечення на мові С++

Основне завдання-розробка програмного забезпечення пристрою і демонстрація його основного функціоналу.

Програмне забезпечення для пристрою розроблено на мові програмування С++ в середовищі програмування Arduino IDE. Програма повинна реалізувати функціонал пристрою. Це:

- індикація сигналів;
- управління пристроєм;
- приймання і обробка інформації з входів АЦП.
- час відображення.

Для полегшення концепції роботи програми за заданими умовами були розроблені конкретні програмні алгоритми, що описують основні завдання програмного забезпечення.

Алгоритми та описи додатків пристроїв:

1) Підключити бібліотеки і застосувати можливості мови.

2) Підключення до бібліотеки (SPI.h) для використання інтерфейсу SPI та бібліотеку екранної графіки (U8glib.h).

3) Ініціалізація змінних і констант.

4) Ініціалізація порту (для введення/виведення).

5) Ініціалізувати інтерфейси SPI, I2C та UART.

Основні цикли програми включають читання та обробку даних з АЦП, відображення даних на екрані та перевірку надходження даних.

6) Читання даних з АЦП.

АЦП підключається до мікроконтролера через інтерфейс SPI. Для прийому даних в АЦП надсилаються два байти нулів, і у відповідь на кожен АЦП відправляє два байти даних, які зсуваються для отримання одного байта даних. Також, залежно від вашого виборукористувачем функцій (стабілізація,

авторозгортка та ін.) іде визначення відповідних даних для відображення на екрані.

7) Виведення даних на екран.

Для управління екраном використовується бібліотека " U8glib.h", що містить основні функції, необхідні для відображення даних. Основні характеристики:

- u8g.drawStr() - Показати текст.
- u8g.drawPixel() – Намалювати піксель.
- u8g.drawLine() – Намалювати лінію.

Відображення поступове. Спочатку малюється статичний текст (імена, значення), потім відображаються розраховані динамічні компоненти (сигнальні і частотні дані).

8) Обробка даних.

Коли дані надходять з порту, він переходить до функції, яка обробляє ці дані, змінюючи відповідне значення, встановлене Користувачем.

Дотримуючись програмного алгоритму, було розроблено та виготовлено програмне забезпечення, представлене в додатку А.

7.2 Розробка програмного забезпечення

Розробка програмного забезпечення пристрою здійснюється в середовищі Atmel Studio мовою асемблера для виконання функцій, що демонструють індикацію мікроконтролера.

Програмне забезпечення:

```
.def temp = r16 ; задаємо ім'я нашому регістру загального призначення
        .org 0x0000 ; починаємо програму з reset
        rjmp reset
reset:
                ldi temp, LOW(RAMEND) ; Покажчик стека вказує
                out SPL,  temp        ; на останню адресу ОЗУ
                ldi temp, HIGH(RAMEND)
                out SPH,  temp
        ldi temp, 0b00000001 ; порт B на вихід
```

```
                out DDRB, temp   ;

// Основне тіло програми
main:
                sbi PORTB,5
                rcall delay
                cbi PORTB,5
                rcall delay
rjmp main
// Підпрограма затримки
delay:
                clr r20
                clr r21
m_1:
                inc r20
                brne m_1
                inc r21
                brne m_1
                ret
```

ВИСНОВКИ

Розроблено електронну систему моніторингу мережевого трафіку при виконанні магістерської кваліфікаційної роботи згідно з поставленими завданнями.

1. Одним з ефективних засобів моніторингу якості обміну інформацією є перевірка обміну даними через послідовні порти шляхом підключення до фізичних послідовних портів серверів і робочих станцій для аналізу та документування обміну даними. Це аналізатор зв'язку в реальному часі, який може аналізувати інтернет-трафік.

2. Пристрій був реалізований за допомогою Arduino UNO на базі мікроконтролера ATmega328P, що значно скоротило час розробки та вартість пристрою.

3. Пристрої електронної системи моніторингу мережевого трафіку, представлені в магістерській кваліфікаційній роботі, мають обмежені амплітудно-частотні характеристики, але одним з основних завдань на даному етапі було мінімізувати габарити пристроїв. Обмеження-це перевага. Надалі при розробці сімейств пристроїв частотна і амплітудна характеристики можуть бути підігнані під інші необхідні параметри.

5. Створіть електричну схему вашого пристрою.

6. На основі аналізу, розрахунку і розробки схем були створені алгоритми роботи Програм і програмних кодів, перевірені функціональність і працездатність пристрою. Розроблена система також пропонує можливість розширення системи шляхом підключення різних нових компонентів. Це дозволить надалі виконувати додаткові дослідницькі та аналітичні функції, а також контроль і контроль над їх параметрами.

СПИСОК ЛІТЕРАТУРИ

1. Пятібратов А.П. обчислювальні системи, мережі та телекомунікації: Підручник для вузів / А. П. Пятібратов, Л. П. Гудино, А. а. Кириченко; під ред. А. П. Пятібратова. М.: 2017. 736 с.
2. Костенко О.Ю. дослідження систем моніторингу корпоративних мереж передачі даних / О. Ю. Костенко, О. О. Барабанова // Наука, Освіта, інновації: шляхи розвитку: матеріали шостий Всерос. наук.- практ. конф. (Петропавловськ-Камчатський, 21-24 квітня 2015 р.). Петропавловськ-Камчатський: КамчатГТУ, 2015. С. 80– 84.
3. Система мережевого моніторингу на базі NETFLOW [Електронний ресурс]. – 2022. − Режим доступу: https://www.opennet.ru/docs/RUS/netflow_bsd/
4. Засоби моніторингу та аналізу мережі [Електронний ресурс]. − Режим доступу:https://wiki.cuspu.edu.ua/index.php/Засоби_моніторингу_та_аналіз у_мережі
5. Технічні дані Arduino UNO [Електронний ресурс] − Режим доступу: https://is.gd/uc3WPu
6. Технічні дані ATmega328P [Електронний ресурс] − Режим доступу до ресурсу: https://tinyurl.com/mr2nyw8f .
7. Технічні дані UA78m05 [Електронний ресурс] − Режим доступу: https://goo.su/GwlNAde.
8. Технічні дані ADG704 [Електронний ресурс] − Режим доступу до ресурсу: https://cuti.cc/BzG5r
9. Технічні дані ADC081S101 [Електронний ресурс] − Режим доступу до ресурсу: https://www.ti.com/lit/ds/symlink/adc081s101.pdf.
10. Технічні дані AD8033 [Електронний ресурс] − Режим доступу до ресурсу: https://doc.softelectronics.ru/docs/op1/AD8033.pdf
11. Технічні дані екран 1.3inch OLED (B) [Електронний ресурс] − Режим доступу до ресурсу: https://www.waveshare.com/wiki/1.3inch_OLED_(B)
12. Технічні дані FT232RL [Електронний ресурс] − Режим доступу до ресурсу: https://www.rcscomponents.kiev.ua/product/ft232rl_22404.html

13. U8glib Arduino [Електронний ресурс] − Режим доступу: https://goo.su/Z2OFK.
14. Мова програмування С++ [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: https://maup.com.ua/assets/files/lib/book/c_plisplus.pdf
15. Редактор графічний Easyeda [Електронний ресурс] − Режим доступу: https://easyeda.com/editor
16. Microsoft Visual Studio [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: https://clck.ru/336Wq9
17. Системи моніторингу та аналізу мережевого трафіку [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: http://eprints.library.odeku.edu.ua/id/eprint/5971/1/Krylov_Roxrobka_system_analizy_merejevogo_trafiky.pdf
18. Моніторинг і аналіз мережевої інфраструктури [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: https://otherreferats.allbest.ru/programming/00177628_0.html
19. Васильєв В.Р. Покроковий перетворювач двійкових чисел в двійково-десяткові/ Борисенко О.А., Бережна О.В., Горішняк А.О., Сердюк В.В., Васильєв В.Р., студент// фізика, електроніка, електротехніка (ФЕЕ-2021). Матеріали та програма науково-технічної конференції. – Суми: СумДУ, 2021. - С.145.
20. Васильєв В.Р. Перетворення двійкових чисел в фібоначчієві/ Борисенко О.А., Васильєв В.Р., Литвиненко А.М., // фізика, електроніка, електротехніка (ФЕЕ-2021). Матеріали та програма науково-технічної конференції. – Суми: СумДУ, 2020. - С.160.
21. Васильєв В.Р. Розробка методів тривимірного (3D) біопринтінгу для друку гелевими біополімерами/ Колесник М.М., Знаменщиков Я.В., Дейнека В.М., Васильєв В.Р. // фізика, електроніка, електротехніка (ФЕЕ-2019). Матеріали та програма науково-технічної конференції. – Суми: СумДУ, 2019. - С.138.
22. Васильєв В.Р. Плівки CZTS, отримані методом струменевого друку чорнилами на основі поліольно-синтезованих нанокристалів // Васильєв В.Р., Доброжан О.А., Опанасюк Н.М., Опанасюк А.С., // фізика,

електроніка, електротехніка (ФЕЕ-2019). Матеріали та програма науково-технічної конференції. – Суми: СумДУ, 2019. - С.138.

23. Borysenko Oleksiy Нероздільні коди в системах обробки інформації / Oleksiy Borysenko, Olga Berezhna, Svitlana Matsenko, Viktor Serdiuk, Andrii Horishniak, Vitaly Vasilyev // Системи управління, навігації та зв'язку. Збірник наукових праць. – Полтава: ПНТУ, 2021. – Т. 2 (64). – С. 58-62. – doi:https://doi.org/10.26906/SUNZ.2021.2.058.

ДОДАТОК А

Програмне забезпечення приладу електронної системи моніторингу мережевого трафіку:

#include "U8glib.h" #include <SPI.h>

// Внутрішній АЦП

const int slaveSelectPin = 10;

byte useThreshold = 1; // 0 = Off, 1 = Rising, 2 = Falling byte theThreshold = 128; // 0-255, Multiplied by voltageConst unsigned int timePeriod = 200; // 0-65535, us or ms per measurement (max 0.065s or 65.535s)

byte voltageRange = 1; // 1 = 0-5V, 2 = 0-2.5V, 3 = 0-1.25V

boolean autoHScale = true; // Automatic horizontal (time) scaling boolean linesNotDots = true; // Draw lines between data points

const byte high_speedADC_time_us = 10; //Час роботи АЦП 13 мкс + 3 мкс.

// Variables that can probably be left alone

const byte vTextShift = 1; // Vertical text shift (to vertically align info)

// Leave at 100 for 128x64 pixel display (Кількість вибірок)

const byte numOfSamples = 100;

//unsigned int HQadcReadings[numOfSamples]; // Читання даних з АЦП (оскільки 10 біт) byte adcReadings[numOfSamples]; // Читання даних з АЦП (8 біт)

byte thresLocation = 0; // Threshold bar location

float voltageConst = 0.07936511; // Scaling factor for converting 0-63 to V

// Ініціалізація змінних

float avgV = 0.0; // Для середньої напруги float maxV = 0.0; // Максимальна напруга

float minV = 0.0; // Мінімальна напруга float ptopV = 0.0; //

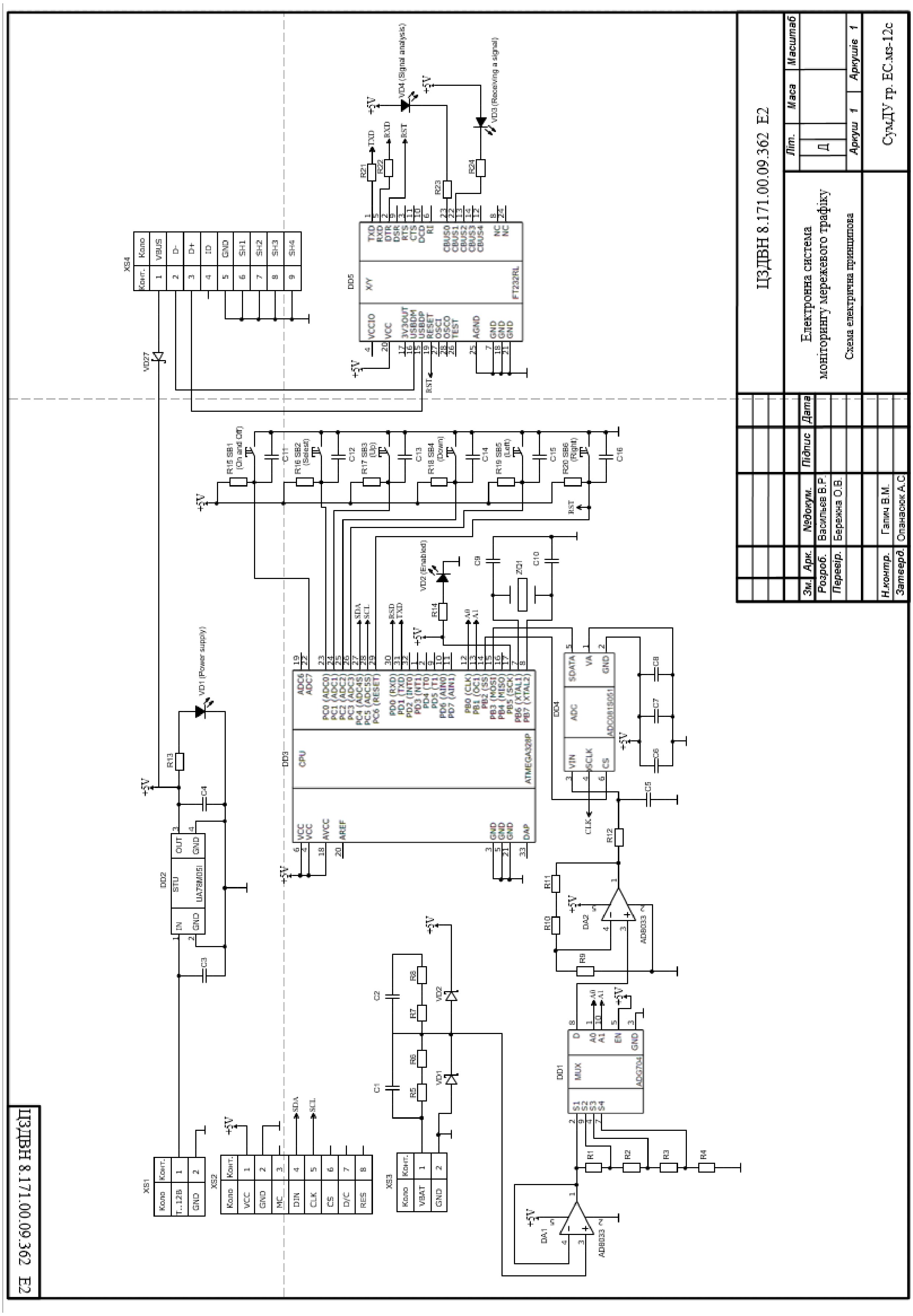
```
float theFreq = 0; // Частота (можна додати період)
const byte theAnalogPin = A0; // Data read pin
// HW SPI/I2C:
//U8GLIB_SSD1306_128X32 u8g(U8G_I2C_OPT_NONE); // I2C / TWI
//U8GLIB_SSD1306_128X64 u8g(U8G_I2C_OPT_NONE|U8G_I2C_OPT_DEV_0); // I2C / TWI U8GLIB_SSD1306_128X64 u8g(U8G_I2C_OPT_FAST); // Fast I2C / TWI
//U8GLIB_SSD1306_128X64 u8g(U8G_I2C_OPT_NO_ACK);
// Display which does not send AC
// defines for setting and clearing register bits #ifndef cbi
#define cbi(sfr, bit) (_SFR_BYTE(sfr) &= ~_BV(bit)) #endif
#ifndef sbi
#define sbi(sfr, bit) (_SFR_BYTE(sfr) |= _BV(bit)) #endif
int adc(){
byte data = 0;
//float data_f = 0; digitalWrite(slaveSelectPin, LOW);
data = (SPI.transfer(0)<<4)|(SPI.transfer(0)>>4); digitalWrite(slaveSelectPin, HIGH);
return data;
void collectData(void) //collect data from adc (main part)!!!
{
byte tempThres = 0; // TEMP byte i = 0;
if (autoHScale == true)// Автопідстроювання часу{
// With automatic horizontal (time) scaling enabled,
// scale quickly if the threshold location is far, then slow down if (thresLocation > 5*numOfSamples/8) {
timePeriod = timePeriod + 5;
} else if (thresLocation < 3*numOfSamples/8) { timePeriod = timePeriod - 5;
```

```
} else if (thresLocation > numOfSamples/2) { timePeriod = timePeriod + 1;
} else if (thresLocation < numOfSamples/2) { timePeriod = timePeriod - 1;
}
}
// Enforce minimum time periods
if (timePeriod < high_speedADC_time_us) { timePeriod = high_speedADC_time_us;
}
// Adjust voltage constant to fit the voltage range if (voltageRange == 1) {
//voltageConst = 1;
voltageConst = 0.079365151; // 0-5V was//0.0523810; // 0-3.30V //
} else if (voltageRange == 2) {
voltageConst = 0.0261905; // 0-2.5V was//0.0261905; // 0-1.65V //
} else if (voltageRange == 3) {
voltageConst = 0.01309526; // 0-1,25V was//0.0130952; //0-0.825V //
}
// If using threshold, wait until it has been reached if (voltageRange == 1) tempThres = theThreshold;
else if (voltageRange == 2) tempThres = theThreshold << 1; else if (voltageRange == 3) tempThres = theThreshold << 2; if (useThreshold == 1) {
i = 0; while ((adc()>tempThres) && (i<256)) i++; i = 0; while ((adc()>tempThres) && (i<256)) i++;
}
else if (useThreshold == 2) {
i = 0; while ((adc()>tempThres) && (i<256)) i++; i = 0; while ((adc()>tempThres) && (i<256)) i++;
// Collect ADC readings
for (i=0; i<numOfSamples; i++) { adcReadings[i] = adc();
//adcReadings[i] = analogRead(theAnalogPin)/4;
```

```
if (timePeriod > high_speedADC_time_us) //was -35
{
delayMicroseconds(timePeriod - high_speedADC_time_us); //was -35
}
for (i=0; i<numOfSamples; i++) {
// Scale the readings to 0-63 and clip to 63 if they are out of range. if (voltageRange == 1) {
//adcReadings[i]<<4 & 0b11111111;
if (adcReadings[i]>>2 < 0b111111) adcReadings[i] = adcReadings[i]>>2 & 0b111111; else adcReadings[i] = 0b111111;
} else if (voltageRange == 2) {
if (adcReadings[i]>>3 < 0b11111) adcReadings[i] = adcReadings[i]>>3 & 0b111111; else adcReadings[i] = 0b111111;
} else if (voltageRange == 3) {
if (adcReadings[i]>>4 < 0b1111) adcReadings[i] = adcReadings[i]>>4 & 0b111111; else adcReadings[i] = 0b111111;
}
// Invert for display
adcReadings[i] = 63-adcReadings[i];
}
// Calculate and display frequency of signal using zero crossing if (useThreshold != 0) {
if (useThreshold == 1) { thresLocation = 1;
while ((adcReadings[thresLocation]<(63-(theThreshold>>2))) && (thresLocation<numOfSamples- 1)) (thresLocation++);
thresLocation++;
while ((adcReadings[thresLocation]>(63-(theThreshold>>2))) && (thresLocation<numOfSamples- 1)) (thresLocation++);
}
```

```
        else if (useThreshold == 2) { thresLocation = 1;
        while       ((adcReadings[thresLocation]>(63-(theThreshold>>2)))       &&
(thresLocation<numOfSamples- 1)) (thresLocation++);
        thresLocation++;
        while       ((adcReadings[thresLocation]<(63-(theThreshold>>2)))       &&
(thresLocation<numOfSamples- 1)) (thresLocation++);
        }
```

ДОДАТОК Б


ЦЗДВН 8.171.00.09.362 Е2
Електронна система моніторингу мережевого трафіку
Схема електрична принципова
СумДУ гр. ЕС.м3-12с
Розроб. Васильєв В.Р.
Перевір. Бережна О.В.
Н.контр. Гапич В.М.
Затверд. Опанасюк А.С.
ATMEGA328P
FT232RL
ADC081S051
ADG704
AD8033
UA78M05I
VD1 (Power supply)
VD2 (Enabled)
VD3 (Receiving a signal)
VD4 (Signal analysis)

ДОДАТОК В

Розробка методів тривимірного (3D) біопринтінгу для друку гелевими біополімерами

Колесник М.М., *доцент*; Знаменщиков Я.В., *асистент*;
Дейнека В.М., *аспірант*; Васильєв В.Р., *студент*
Сумський державний університет, м. Суми, Україна

Людське тіло має здатність до регенерації, проте вона обмежена багатьма чинниками, зокрема розміром дефекту тканини. Будь-яке пошкодження тканини, що перевищує критичний розмір, потребує хірургічних методів лікування. Найбільш поширеним при цьому є метод, що використовує власні тканини чи біоматеріали у вигляді тканинноінженерних скаффолдів. Тривимірні скаффолди відіграють роль штучного міжклітинного каркасу, стимулюють ріст та диференціацію клітин під час формування нової, власної тканини організму. Використання тривимірних скаффолдів обумовлюється критичними факторами: вибором біоматеріалу та методу створення. Серед методів створення скаффолдів найбільш перспективний є їх 3D друк, зокрема струменевий друк біоматеріалів.. Під час струменевого друку біоматеріал наноситься шар за шаром аж до одержання необхідної конструкції. Саме метод 3D біопринтінгу дозволяє створити скаффолд завантажений плюріпотентними клітинами, який після трансплантації зможе відтворити повноцінну тканину.

Нами проведена модифікація 3D принтера та розроблена відповідна головка, що дозволить використовувати для друку біополімери у вигляді гелю (рисунок 1). В подальшому експериментальним шляхом будуть підібрані вид полімеру та визначені його характеристики для оптимальної швидкості полімеризації друкованих шарів.

В результаті цього буде розроблено оптимальну тривимірну будову скаффолду для можливого заселення його культурою клітин та проведена характеризація поверхні та внутрішньої структури зразка за допомогою скануючої електронної мікроскопії. Для визначення можливості застосування матеріалів для тканинної інженерії буде визначена швидкість біодеградації з використання розчину SBF (simulated body fluid) та отримані дані щодо типу тканини, в якій може бути використаний метод тканинної інженерії.

ДОДАТОК Г

Плівки CZTS, отримані методом струменевого друку чорнилами на основі поліольно-синтезованих нанокристалів

Васильєв В.Р., *студент*; Доброжан О.А., *асистент*; Опанасюк Н.М., *доцент*; Опанасюк А.С., *професор*

Сумський державний університет, м. Суми, Україна

Напівпровідникова сполука Cu2ZnSnS4 (CZTS), що має *p*-тип провідності, високий коефіцієнт поглинання видимого світла (104-105 см-1), оптичну ширину забороненої зони, що відповідає максимуму Шоклі-Квайзера (*Eg*~1,5 еВ) є екологічно чистою, дешевою альтернативою традиційним поглинальним матеріалам (Si, CIGS, CdTe) тонкоплівкових сонячних елементів (СЕ). Одним з перспективних шляхів зниження собівартості таких приладів є застосування гнучких підкладок для нанесення тонких плівок CZTS та використання для їх одержання 3D чи 2D друку.

В роботі досліджено вплив післяростового відпалу на розмір і форму зерен, товщину, кристалічну структуру, фазовий і хімічний склад тонких плівок CZTS, нанесених на поліамідні підкладки.

Наночастинки (НЧ) синтезували за методикою описаною в [1]. Після цього їх виділяли з суспензії центрифугуванням при 4000 об/хв протягом 10 хв та двічі промивали етанолом для видалення побічних продуктів реакції. Чорнило формували диспергуванням 1 г вакуумно висушених нанокристалів в суміші 8 г дистильованої води і 1 г EG. Утворені чорнила були використані для друку плівок на гнучких підкладках Kapton. Під час процесу нанесення застосувалось 5 циклів друку з короткою стадією попереднього відпалу на гарячій пластині принтеру 30 сек при температурі 150 0С. Після осадження плівки відпалювали при трьох температурах (*Ta* = 150, 175, 200 0С) протягом трьох різних часів (*ta* = 30, 60, 90 хв).

Встановлено, що синтезовані НЧ Cu2ZnSnS4 мали розмір (7 ± 4) нм. Методами SAED і XRD підтверджено, що вони мали кристалічну структуру кестериту з незначною кількістю вторинної фази Cu2S. Відпалювання плівок при температурі попереднього відпалу 150 0С і наступних післяростових відпалах при (150-200) 0С протягом (30-120) хв дозволили незначно зменшити вміст фази Cu2S, а також збільшити розмір їх зерен від 5,1 нм до 7,6 нм, що свідчить про поліпшення якості кристалічної структури шарів.

ДОДАТОК Ґ

Перетворення двійкових чисел в фібоначчієві

Борисенко О.А., *професор*; Васильєв В.Р., *студент*;
Литвиненко А.М., *студент*
Сумський державний університет, м. Суми, Україна

Сьогодні в телекомунікаційних системах використовується велика кількість різних завадостійких кодів, серед яких в силу своєї відносної простоти і можливості наскрізного контролю набув код Фібоначчі, який складається з фібоначчієвих чисел. При цьому важливим завданням, необхідним для застосування коду Фібоначчі, є необхідність перетворення в нього двійкового коду. Це пов'язано з тим, що код Фібоначчі в багатьох випадках зв'язується з двійковими цифровими системами. Саме це завдання зв'язку двійкового коду з кодом Фібоначчі і вирішується в даній роботі.

Для вирішення цієї задачі пропонується використати Фібоначчі-восьмеричний код, який складається з послідовності 4-розрядних кодів - сегментів, що містять по 8 фібоначчієвих чисел. В результаті вони відносно легко переходять в 3-розрядні сегменти двійкових чисел. Це спрощує перетворення двійкових чисел в фібоначчієві числа і робить його більш швидкодіючим і надійним.

Пропонований спосіб перетворення двійкового числа у фібоначчієве число полягає в наступному:

1. Двійкове число ділиться на сегменти, що складаються з 3 бітів, починаючи з нульового біта молодшого розряду і закінчуючи бітом старшого розряду. В результаті буде отримано двійково-вісімкове число.

2. Кожен двійковий сегмент перетворюється у відповідний сегмент фібоначчієвого числа.

3. З фібоначчієвих сегментів формується Фібоначчі-восьмеричне число і процедура перетворення двійкового числа у Фібоначчі- восьмеричне число закінчується.

Отримана послідовність сегментів Фібоначчі-восьмеричного числа послідовно зберігається в буферному пристрою і після цього передається по каналу зв'язку.

ДОДАТОК Д

Покроковий перетворювач двійкових чисел в двійково-десяткові

Борисенко О.А., *професор*, Бережна О.В., *доцент*,
Горішняк А.О., *аспірант*, Сердюк В.В., *аспірант*, Васильєв В.Р., *студент*
Сумський державний університет, м. Суми, Україна

Двійково-десяткові числа застосовуються під час передачі та відображення інформації, яка знімається з датчиків тепла, води, електрики та інших подібних пристроїв. Вони також застосовуються в вимірювальних цифрових схемах, які, наприклад, вимірюють частоту, час, напругу. Тому постає задача перетворення двійкових чисел в двійково-десяткові числа. Є алгоритми побудови такого перетворення діленням на число 10, поданому в двійковому вигляді [1].

Однак, цей шлях є досить складним, особливо при схемній реалізації перетворювача, тому що потребує ділення на 10. В даній роботі пропонується зменшити складність перетворення замінивши ділення двійкового числа покроковим відніманням з нього 1, поки в кінцевому підсумку не буде отриманий 0, а паралельно при цьому проводиться покрокове підсумовування 1 в двійково-десяткових лічильниках з початковим встановленням їх в 0. Тоді в кінцевому підсумку отримана сума в цих лічильниках буде виражати вхідне двійкове число в двійково-десятковому вигляді.

Очевидним недоліком цієї схеми є те, що швидкість перетворення двійкового числа в двійково-десяткове значно знижується, тому що кількість тактів при перетворенні двійкового числа в двійково- десяткове число буде дорівнювати величині двійкового числа. З ростом кількості розрядів в ньому число тактів перетворення збільшується за експоненціальним законом. Однак, в багатьох випадках цей недолік не є вирішальним. Зате простота алгоритму, особливо при апаратній реалізації, зменшує складність програми і схеми, що дає йому важливу перевагу. Ця перевага складається з того, що збільшується надійність відповідних схем та спрощується технологія їх виробництва. Надійність збільшується тому, що зменшується кількість елементів перетворювача двійкового коду в двійково-десятковий код.

ДОДАТОК Е

О. А. Борисенко, О. В. Бережна, С. М. Маценко,
В. В. Сердюк, А. О. Горішняк, В. Р. Васильєв
Сумський державний університет, Суми, Україна

НЕРОЗДІЛЬНІ КОДИ В СИСТЕМАХ ОБРОБКИ ІНФОРМАЦІЇ

А н о т а ц і я . У зв'язку з необхідністю збільшення ефективності цифрових систем обробки та передачі даних зростають вимоги до забезпечення їх завадостійкості. Її необхідність виникає, як правило, при оперативному зчитуванні інформації з датчиків, які використовуються в системах обр обки інформації. При цьому бажано використовувати завадостійкі коди, які одночасно дозволяють як обробляти, так і передавати інформацію. Такі коди здійснюють її наскрізний контроль. Це дозволяє підвищувати швидкість обробки та передачі інформації і при цьому економити апаратуру систем. Кодів наскрізного контролю відомо небагато, тому що найбільш вживані на практиці роздільні коди, наприклад, циклічні та подібні до них, використовуються, як правило, для передачі інформації і не можуть ефективно контролювати її обробку. Вирішують задачу наскрізного контролю нероздільні коди, а серед них на сьогодні найбільш перспективними кодами можна вважати коди Фібоначчі. Також досить ефективні в цьому плані є рівноважні і біноміальні коди. У даній роботі проводиться обґрунтування використання нероздільних кодів в завадостійких системах обробки і передачі інформації. Серед нероздільних кодів особливе місце займають коди Фібоначчі, які складаються з чисел Фібоначчі. Ці числа можна додавати, віднімати, множити та ділити. На їх основі будуються автомати Фібоначчі з широким спектром можливостей обробки інформації. Однією з її задач є фібоначчієва лічба. Фібоначчієві числа можуть бути за формою мінімальними та максимальними. Особливістю чисел Фібоначчі є те, що вони мінімальні, і тому лічба та лічильники на їх основі будуть більш простими та надійними в порівнянні з іншими способами фібоначчієвої лічби. Крім того, в них більш легко виявляються і частково виправляються поодинокі помилки. Але головне в

них є те, що інформація фібоначчієвого лічильника може безпосередньо без кодуючого пристрою направлятися в канал зв'язку, де будуть виявлятися і при необхідності виправлятися деякі помилки, що виникають в ньому. Недоліком такого кодування буде необхідність перетворювати фібоначчієву інформацію в двійкову. Однак, це перетворення потрібно робити далеко не завжди, тому що нерідко ця інформація є керуючою і відображається на відповідних пристроях відображення.

К л ю ч о в і с л о в а : телекомунікаційна система, нероздільні коди, помилки, завадостійкий код.

Вступ

Постановка проблеми. На сьогодні. як і раніше, одним з основних завдань системи обробки та передачі інформації залишається їх захист від перешкод та збурень. При цьому потребується підвищення ефективності кодів, що в них використовуються, як з точки зору швидкодії і завадостійкості систем, так і їх апаратурних витрат. Однак, пошук таких кодів виявився досить складним. Він становить проблему, яка частково вирішується в даній роботі.
Аналіз останніх досліджень і публікацій. Спроби застосування потужних роздільних завадос- тійких кодів, які використовуються в телекомуніка-ційних системах для організації наскрізного контро- лю, не принесли поки що особливого успіху, почасти тому, що ці коди спрямовані на забезпечення завадо- стійкої передачі інформації, а не її обробки [1-6]. Правильність роботи джерела інформації такі коди залишають поза контролем. Джерелом інформації в даному випадку може бути цифровий автомат або обчислювальна система в цілому. Однак, це не означає, що деякі із завадостійких кодів, що застосовуються в телекомунікаційних сис- темах, не можуть бути задіяні для контролю цифрових автоматів та обчислювальних систем. Тим більше, що на сьогодні телекомунікаційні системи забезпечені великою різноманітністю завадостійких кодів від простих й до досить потужних, здатних виявляти та виправляти пакети помилок високої кратності [3, 5, 6]. Серед завадостійких кодів слід шукати коди, які б одночасно контролювали як передачу інформації безпосередньо в телекомунікаційній системі, так й її джерело. Тим самим здійснювався б наскрізний контроль єдиним кодом системи обробки та передачі інформації,

що дає їй суттєві переваги щодо швидкодії, вартості та надійності в порівнянні з системами, в яких обробка та передача інформації контролюється окремими кодами.

Метою статті є пошук завадостійких кодів, які одночасно контролюють обробку і передачу інфор- мації та обґрунтування їх застосування для контролю роботи комплексу обчислювальних і телекомунікаційних засобів. Виклад основного матеріалу

Робота цифрової телекомунікаційної системи відбувається наступним чином. Стан цифрового джерела інформації перетворюється в комбінацію завадостійкого коду, яка передається по каналу зв'язку. На приймальному кінці ця комбінація перевіряється декодувальним пристроєм на наявність помилки й при її відсутності передається приймачу інформації. При наявності помилки вона може бути виправлена або за допомогою повторної передачі, або, при наявності достатньої надмірності інформації, безпосередньо приймачем.

Надмірність вводиться пристроєм, що кодує, який розташований після джерела інформації, під час перетворення кодової комбінації, яка генерується джерелом інформації, в завадостійку комбінацію. У результаті до вхідної двійкової кодової комбінації або додаються додаткові контрольні розряди або вона перетворюється в комбінацію іншого коду з іншою кількістю розрядів. У першому випадку надлишковий код буде роздільним, а в другому – не- роздільним [1-2]. При цьому важливою особливістю телекомунікаційних систем є те, що в них обов'язково існують пристрої, що кодують та декодують. Перші з них вводять в передані повідомлення надлишкову інформацію, а другі за її допомогою визначають правильність отриманих повідомлень. Особливістю нероздільних кодів, є те, що в них до розрядів вхідних кодових комбінацій, які кодують стани цифрового автомата, не додаються контрольні розряди, як в роздільних кодах, а вони перетворюються за певними правилами в інші комбінації, які після цього мають більше розрядів. При цьому як в роздільних, так і в нероздільних кодах з'являється надлишкова інформація, яка виділяє в завадостійкому коді дозволені комбінації. Поява забороненої комбінації є ознакою її помилковості. Завдяки цьому визначається правильність переданої комбінації на приймальному кінці.

Роздільні завадостійкі коди в телекомунікаційних системах на практиці на сьогодні застосовуються частіше ніж нероздільні коди в силу їх більшої потужності і можливості збереження інформаційних розрядів вихідної кодової комбінації, яка надходить від джерела інформації при її передачі. Наприклад, широко відомий роздільний код з перевіркою на парність або на непарність використовує для отримання контрольного розряду згортку інформаційних розрядів за модулем 2. Інформаційні розряди при цьому залишаються без змін [1-3].

З іншого боку, відкритість інформаційних розрядів роздільних кодів робить їх незахищеними від зовнішнього доступу. Однак, цей недолік несуттєвий, тому що захист роздільних кодів здійснюється при необхідності за допомогою відповідних методів захисту інформації.

Нероздільні коди на відміну від роздільних кодів не мають інформаційної та контрольної частини в явному вигляді. Вони мають загальну ознаку для всіх переданих комбінацій, яка відрізняє заборонені кодові комбінації від дозволених, наприклад, коли дозволеними комбінаціями будуть комбінації, що містять постійну кількість одиниць.

Однак, нероздільні коди зустрічаються в телекомунікаційних системах значно рідше, тому що вони, як правило, менш потужні ніж роздільні коди. Крім того, виникає необхідність перетворювати вхідні комбінації одного коду в вихідні комбінації іншого коду. При цьому, як й в роздільних кодах, виникає інформаційна надмірність, але вона прихована. Тому її використання для виявлення та виправлення помилок в комбінаціях після кодування в повідомленнях може бути більш складним, ніж в роздільних кодах. Значить, тоді більш складними будуть кодуючі та декодуючі пристрої телекомунікаційної системи. Так, наприклад, рівноважний код, який є нероздільним, для виявлення помилки вимагає підрахунок кількості одиниць в кодової комбінації та порівняння результату з контрольним числом, й тому буде більш складним у порівнянні з кодом з перевіркою на парність або на непарність, де відбуваються тільки операції додавання одиниць за модулем 2. Крім того, ще потрібно перетворювати вхідні комбінації в комбінації рівноважного коду.

реальних задач є недостатньою і на практиці потребує збільшення. Однак, є перспективні розробки, які дозволяють говорити про переваги нероздільних кодів, тому що в них одночасно відбувається захист інформації, як від помилок, так й від несанкціонованого доступу. Так що питання про те, які коди будуть більш перспективними для передачі та зберігання інформації в майбутньому відкрите.

Однак, головна перевага нероздільних кодів в порівнянні з роздільними кодами полягає в тому, що нероздільні коди можуть бути ефективно застосовані для підвищення завадостійкості цифрових автоматів. Їх використання по суті є єдиним способом збільшення їх завадостійкості без резервування.Роздільні коди, які використовуються в телекомунікаційних системах, в силу їх природи не дозволяють ефективно організувати завадостійку обробку інформації цифровими автоматами. А от нероздільні коди, в силу того, що використовують при синтезі тільки дозволені комбінації, дозволяють будувати завадостійкі цифрові автомати. Перехід автомата в стан, який не є дозволеною комбінацією, помилковий.

Неефективність використання роздільних кодів для завадостійкого кодування цифрових автоматів пов’язана з тим, що цифровий автомат на відміну від системи передачі інформації, яка передає її за один такт без змін, обробляє інформацію впродовж декількох тактів, на кожному з яких в загальному випадку з’являється нова інформація. Визначити правильність цієї інформації за допомогою роздільного кодування важко, тому що треба заздалегідь передбачити якою буде комбінація на виході авто- мата на наступному такті та порівняти її з комбіна- цією, яка фактично з'явиться. Зазвичай для виявлення помилок в автоматах, які використовують роздільні коди, необхідне дублювати апаратуру, а для їх виправлення – потроєння, що значно здорожує відповідні цифрові пристрої та системи, роблячи їх громіздкими та складними при експлуатації [2, 3].
На відміну від роздільних кодів нероздільні завадостійкі коди дозволяють знаходити помилки при обробці інформації цифровими автоматами за рахунок своєї надмірності, в тому числі й за рахунок природної надмірності цифрових автоматів. До того ж такі коди дають можливість завадостійкої передачі інформації безпосередньо з обчислювальних пристроїв без додаткового кодування в каналах зв'язку.

Тим самим реалізується наскрізний контроль як обчислювального пристрою, який виступає в ролі джерела інформації, так й каналу зв'язку. У ньому виключаються кодуючі пристрої в каналі зв'язку, а в ряді випадків й декодуючі. Це здешевлює всю сис- тему обробки та передачі інформації, підвищуючи при цьому надійність та швидкість її роботи.

Поява в процесі обробки інформації цифровим автоматом забороненої комбінації буде вказувати на помилку. Її виявить структура автомату або відповідний простий декодуючий пристрій. Тим самим цифровий автомат самостійно вирішує, чи є його стан правильним чи помилковим. Тому стає непотрібним пристрій завадостійкого кодування при подальшій передачі стану автомата по каналу зв'язку. Це з одного боку заощаджує апаратурні витрати, а з іншого – підвищує швидкодію роботи та надійність системи обробки і передачі інформації. Хоча в окремих випадках при каналах зв'язку з високим рівнем шуму можна поставити на вході телекомунікаційної системи пристрій кодування для додаткового роздільного коду, й тоді буде отриманий код, який поєднує завадозахисні властивості роздільного та нероздільного кодів, що збільшить надійність передачі кодових комбінацій.

Природно, що такий обчислювальний пристрій із забороненими кодовими комбінаціями ускладнюється в порівнянні з пристроєм, в якому вони відсутні. Але можливість виявлення та в деяких випадках виправлення помилок перекриває цей недолік. В майбутньому саме такі пристрої, що працюють в із забороненими комбінаціями, повинні прийти на зміну двійковим цифровим автоматам без заборонених комбінацій. Питання при цьому буде стояти тільки в тому, який нероздільний код виявиться найкращим для того чи іншого завдання обробки інформації.

На сьогодні поки немає остаточної відповіді на це питання. Тут визначальну роль будуть грати питання швидкодії та надійності роботи обчислювальних пристроїв, що використовують нероздільні коди. З цієї точки зору особливий інтерес повинні викликати нероздільні коди, що представляють завадостійкі системи числення: фібоначчієві, біноміальні, факторіальні та їм подібні. Характерною властивістю цих систем числення є те, що їх кодові комбінації являють собою відповідні числа, над якими можна виконувати різні арифметичні та логічні дії.

Нероздільний код для завадостійкої обробки та передачі даних. Виходячи з вищесказаного, потрібно знайти та дослідити конкретний завадостійкий нероздільний код, який би здійснював наскрізний контроль як при обробці інформації, так й при її передачі. При цьому він повинен давати можливість відносно просто здійснювати арифметико-логічні операції у відповідному обчислювальному пристрої.

В якості такого коду в даній роботі пропонується використовувати код Фібоначчі. Відповідно обчислювальний пристрій, який вирішує це завдання в запропонованому коді, буде представлятися «автоматом Фібоначчі».

У даній роботі під «автоматом Фібоначчі» розуміється будь-який пристрій, який виконує навіть в обмеженому вигляді арифметико-логічні операції над числами Фібоначчі, введення даних, їх зберіган ня та формування сигналів керування. Такий пристрій може бути як вузькоспеціалізованим, наприклад, таким, що виконує за допомогою лічильника тільки операції підрахунку, зберігання та виведення керуючих даних, так й універсальним, тобто таким, що має можливість програмування, додаткову зовнішню та оперативну пам'ять і виконує логічні та арифметичні операції [4-10].

Вибір коду Фібоначчі для дослідження в даній роботі не є випадковим, тому що він досить широко досліджений в роботах [11-13]. На сьогодні існують методи та алгоритми фібоначчієвого підсумовування та лічби і на їх основі відповідні пристрої, а також цифро-аналогові та аналого-цифрові перетворювачі. Крім того, код Фібоначчі досить простий для схемної реалізації і при цьому він здатний виявляти помилки і деякі з них виправляти.

Особливості коду Фібоначчі. Код Фібоначчі складається з фібоначчієвих чисел і є на сьогодні досить широко відомим [5-13]. Його особливістю, як й всіх інших завадостійких кодів, є наявність для нього дозволених та заборонених комбінацій.

Для коду Фібоначчі, який за своєю природою використовує одиниці та нулі, ознакою помилки є наявність в них двох та більше одиниць поспіль. Причому поява трьох одиниць, що стоять поруч, дозволяє виправляти одиночну помилку, що особливо важливо в задачах обробки інформації, де

новий запит інформації неможливий. Як бачимо, алгоритм виявлення та виправлення помилок є досить простим.

Однак, цей код, крім завадостійкої обробки ін- формації, здатний також досить ефективно передавати інформацію з виявленням та виправленням деяких помилок. Це поряд з можливістю надійної обробки інформації на його основі робить його універсальним для багатьох застосувань, тому що зазвичай пристрої та системи обробки інформації не тільки формують дані, а й оперативно їх передають на різні вихідні пристрої.

Такими пристроями можуть бути пристрої відображення інформації, вимірювальні пристрої, такі як частотоміри, таймери та інші їм подібні.

Існує дві модифікації коду Фібоначчі – мінімальний (нормальний) та максимальний код, одержуваний при розгортці фібоначчієвих чисел. Реалізація арифметичних операцій над ними відбувається в процесі переходу від мінімальної форми до максимальної форми та зворотно.

Однак, існує й можливість виконання цих операцій в мінімальній формі без всяких переходів, що дає певні переваги у швидкодії та апаратурних витратах відповідних цифрових автоматів Фібоначчі[7, 13].

Мінімальна форма коду Фібоначчі покладена воснову даної роботи та відповідних автоматів Фібоначчі.

Фібоначчієві числа. Фібоначчієва система числення генерує фібоначчієві числа в нормальній (мінімальній) формі, ваги яких являють собою послідовність чисел Фібоначчі 1, 1, 2, 3, 5, 8, Fn. Кожне число з цієї послідовності

$$F_n = F_{n-1} + F_{n-2.}.$$

Номер фібоначчієвого числа задається нумераційної функцією, ваги якої представляються числами Фібоначчі:

$$N = a_n F_n + a_{n-1} F_{n-1} + \ldots + a_i F_i + \ldots a_1 F_1,$$

де a_i - двійкове значення і-го розряду фібоначчієвого числа; n - довжина числа; F_i - вага і-го розряду.

В скороченому вигляді фібоначчієве число записується так:

$$N_a = a_n a_{n-1} \ldots a_i \ldots a_1..$$

Нульовий розряд в ньому відсутній. Наприклад, числа 11, 17, 23, 41 і 52 в мінімальній формі коду Фібоначчі представлені в таблиці 1.

Таблиця 1. – Фібоначчієві числа

Номер розряда n	8	7	6	5	4	3	2	1
Вага розряда	34	21	13	8	5	3	2	1
N=11	0	0	0	1	0	1	0	0
N=17	0	0	1	0	0	1	0	1
N=23	0	1	0	0	0	0	1	0
N=41	1	0	0	0	1	0	1	0
N=52	1	0	1	0	1	0	0	0

Джерело: розроблено авторами.

Діапазон фібоначчієвих чисел

$$P = F_n + F_{n-1}.$$

В кодах Фібоначчі ознакою помилки є поява двох та більше одиниць поруч. При наявності трьох одиниць, що розташовані поруч, одиниця всередині повинна бути замінена нулем. В результаті помилка виправляється. Так, якщо з'являється на вході приймача фібоначчієве число 0111010101, то це означає, що в ньому сталася помилка в 8 розряді. Для її виправлення достатньо одиницю, що розташована в цьому розряді, інвертувати в 0. Тоді правильним буде число 0101010101 = 33.

Оцінка завадостійкості фібоначчієвих чисел. Фібоначчієві числа є нероздільними. Тому їх оцінка може проводитися відповідно до методики, запропонованої Харкевичем. Суть цієї методики зводить- ся до того, що в коді виділяються підмножини дозволених та заборонених кодових комбінацій. Після цього знаходиться відношення кількості заборонених комбінацій до їх загальної кількості, і це число віднімається від 1. Отриманий результат показує ймовірність виявлення помилки кодом, що розгля- дається. Ця ймовірність змінюється від 0, коли заборонених комбінацій немає, й до 1, коли всі комбінації відносяться до заборонених комбінацій. Цей критерій показує, що ймовірності виявлення помилок зі збільшенням довжини фібоначчієвих чисел збільшуються, й при необмеженому зростанні довжини чисел прагнуть до 1. Звідси випливає висновок, що автомат Фібоначчі, який обробляє більш довгі числа, є більш надійним, ніж автомат, що працює з числами меншої довжини.

Висновки

Серед завадостійких кодів особливими властивостями виділяються завадостійкі нероздільні коди. Вони дозволяють одночасно контролювати збір, обробку та передачу інформації. Це дозволяє здійснювати одним й тим же кодом наскрізний контроль систем обробки та передачі інформації, що спрощує таздешевлює їх контроль, а також збільшує достовірність обробки і передачі інформації.

У якості одного з перспективних завадостійких нероздільних кодів пропонується використовувати код Фібоначчі, який відрізняється простотою технічної реалізації та здатністю обробляти інформацію. Він же ефективно може використовуватися і для по- дальшої передачі інформації за допомогою телекомунікаційної системи.

Аналогічно, як і фібоначчієві числа, можна використовувати біноміальні числа біноміальних систем числення та числа інших подібних систем, наприклад, факторіальних. Для них також існують відповідні лічильники, а інформація з них здатна здійснювати наскрізний самоконтроль. Можна отримати й інші нові, поки ще невідомі, самоконтрольовані нероздільні коди, які можуть здійснювати наскрізний контроль при обробці та передачі інформації. Однак, вони потребують дослідження їх ефективності.

СПИСОК ЛІТЕРАТУРИ

1. Error-Correction Coding and Decoding / F.M. Tomlinson, C.J. Tjhai, M.A. Ambroze, M. Ahmed, M. Jibril. – Cham, Switzerland: Springer Open, 2017. – 520 p.

2. The art of error correcting coding / R.H. Morelos-Zaragoza. – John Wiley, 2016. – 263 p.

3. The theory of Error-Correcting Codes / F. MacWilliams, N. Sloane. – North Holland, 1977. – 762 p.

4. Кулик И.А. Метод оценки границ применения сжатия на основе двоичных биномиальных чисел / И.А. Кулик, А.И. Новгородцев, М.С. Шевченко // Системи обробки інформації. – 2019. – № 2(157). – С. 57-62.

5. Borysenko O. Description and applications of binomial numeral systems / O. Borysenko, V. Kalashnikov, N. Kalashnykova // Computer Science and Cyber Security. – 2016. – Vol. 2(2). – P. 13–21.

6. Кулик І.А., Шевченко М.С. Розробка інформаційно-керуючих систем на основі двійкової біноміальної системи числення. Системи обробки інформації. 2020. № 2(161). С. 78-85. https://doi.org/10.30748/soi.2020.161.09.

7. Borysenko O. Development of the Fibonacci-Octal Error Detection Code for Telecommunication Systems / O. Borysenko, S. Matsenko, S. Spolitis, V. Bobrovs // 24th International Conference Electronics. – 2020. – P. 1–5 // https://doi.org/10.1109/IEEECONF49502.2020.9141620.

8. Fibonacci and Lucas Numbers / V. Hoggatt. – MA: Houghton Mifflin, 1969. – 92 p.

9. Fibonacci & Lucas Numbers and the Golden Section: Theory and Applications / S. Vajda. – Chichester: Ellis Horwood Ltd, 1989. – 189 p.

10. The Fibonacci Numbers / N. Vorobyov. – DC Heath, Boston, 1966, 47 p.

11. Stakhov A. Fibonacci p-codes and Codes of the “Golden” p proportions / A. Stakhov // New Informational and Arithmetical Foundations of Computer Science and Digital Metrology for Mission-Critical Applications. British Journal of Mathematics & Computer Science. – 2016. – Vol. 17. – No. 1. – P. 1–49 // https://doi.org/10.9734/BJMCS/2016/25969.

12. Ávila T. Bruno. Meta-Fibonacci Codes: Efficient Universal Coding of Natural Numbers / T. Bruno Ávila, Ricardo M. Campello de Souza // IEEE Transactions on Information Theory. – 2017. – Vol. 63. – No. 4. – P. 2357–2375 // https://doi.org/10.1109/TIT.2017.2663433.

13. Cui X. An Enhancement of Crosstalk Avoidance Code Based on Fibonacci Numeral System for Through Silicon Vias / X. Cui, X. Cui, Y. Ni, M. Miao, J. Yufeng // IEEE Transactions on Very Large Scale Integration (VLSI) Systems. – 2017. Vol. 25. – No. 5. – P. 1601–1610 // https://doi.org/10.1109/TVLSI.2017.2651141.