УДК 004

В.Глобенко, магістр гр КН-21М-1,4,
Центральноукраїнський національний технічний університет

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ МОНІТОРИНГУ LAN МЕРЕЖ ІНФОРМАЦІЙНИХ ТА КОМП'ЮТЕРНИХ СИСТЕМ

У статті розроблено програмне забезпечення, яке призначено для системи моніторингу LAN мереж інформаційних та комп'ютерних систем. Метою розробки є дослідження та програмна реалізація системи моніторингу LAN мереж інформаційних та комп'ютерних систем. Об'єктом дослідження є процес моніторингу LAN мереж інформаційних та комп'ютерних систем. Предметом дослідження є методи моніторингу LAN мереж інформаційних та комп'ютерних систем. Методи дослідження базуються на методах побудови мереж інформаційних та комп'ютерних систем, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи моніторингу LAN мереж інформаційних та комп'ютерних систем. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.
комп'ютерні науки, моніторинг, LAN, мережі інформаційних та комп'ютерних систем

Постановка проблеми. Розвиток засобів обчислювальної техніки відбувається в багатьох напрямках, що розширюють сферу застосування ЕОМ і підвищують ефективність їхнього використання. Найбільше застосування ЕОМ відбувається при побудові різного виду комп'ютерних мереж, як локальних або корпоративних, так і глобальних. Сучасний світ неможливо представити без використання комп'ютерних мереж, які у тому або іншому вигляді застосовуються усюди, починаючи від елементарних хатніх комп'ютерних мереж і закінчуючи промисловими мережами, глобальними мережами, банківськими мережами, мережами для проведення наукових досліджень.

Актуальність теми магістерського дослідження обумовлена необхідністю підвищення ймовірності достовірного надання даних, які передаються по комп'ютерним мережам. Ці задачі є одними із ключових при розробці сучасних систем моніторингу локальних мереж.

Терміном **моніторинг мережі** називають роботу системи, що виконує постійне спостереження за комп'ютерною мережею в пошуках повільних або несправних систем і яка при виявленні збоїв повідомляє про їх мережному адміністраторові за допомогою пошти, телефону або інших засобів оповіщення. Ці задачі є підмножиною задач керування мережею.

У той час, як система виявлення вторгнень стежить за появою погроз ззовні, система моніторингу мережі виконує спостереження за мережею в пошуках проблем, викликаних перевантаженими й/або серверами, що відмовили, іншими пристроями або мережними з'єднаннями.

Наприклад, для того, щоб визначити стан веб-серверу, програма, що виконує моніторинг, може періодично відправляти запит HTTP на одержання сторінки; для поштових серверів можна відправити тестове повідомлення по SMTP і одержати по IMAP або POP3.

Невдалі запити (наприклад, у тому випадку, коли з'єднання не може бути встановлено, воно завершується по таймауту, або коли повідомлення не було доставлено) звичайно викликають реакцію з боку системи моніторингу.

Як реакція може бути:

–відправлено сигнал тривоги системному адміністраторові;

–автоматично активована система захисту від збоїв, що тимчасово виведе проблемний сервер з експлуатації, доти, поки проблема не буде вирішена, і так далі.

Проведений критичний аналіз існуючих методів обробки даних, дозволив структурувати область застосування таких технологій і виявити ряд обмежень. У зв'язку із цим виникла необхідність у розробці перспективного, однак ще недостатньо дослідженого теоретично й апробованого на практиці, динамічного багатопоточного методу обробки даних і програмного забезпечення для реалізації цього підходу.

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-10] було виявлено певні прогалини у забезпеченні системи моніторингу LAN мереж інформаційних та комп'ютерних систем.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи моніторингу LAN мереж інформаційних та комп'ютерних систем.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

– Огляд існуючих систем моніторингу LAN мереж інформаційних та комп'ютерних систем.

– Дослідження системи моніторингу LAN мереж інформаційних та комп'ютерних систем.

– Програмна реалізація системи моніторингу LAN мереж інформаційних та комп'ютерних систем.

Об'єктом дослідження є процес моніторингу LAN мереж інформаційних та комп'ютерних систем.

Предметом дослідження є методи моніторингу LAN мереж інформаційних та комп'ютерних систем.

Методи дослідження базуються на методах побудови мереж інформаційних та комп'ютерних систем, методах математичної статистики, методах розробки програмного забезпечення.

Виклад основного матеріалу. Розробка системи моніторингу мережі з використанням стандартних інструментів

Джерела телеметричних даних

Головне джерело телеметричної інформації про сервери Windows – журнал подій Security, а найважливіше джерело виробничої телеметрії – журнали подій System і Application. Досвідчені користувачі оснащення Event Viewer консолі Microsoft Management Console (MMC) знають, що у всіх журналах подій Windows застосовується один формат файлів (.evt). Запис про кожну подію містить стандартні поля (наприклад, дата, час, джерело події, категорія, ID події), за якими знаходиться поле опису з даними у вільній формі, унікальними для конкретної події. Будь-який додаток моніторингу, сумісний з журналами подій Windows, дозволяє генерувати попередження й звіти на основі джерела, категорії або ID події, але в ідеальному випадку потрібно мати можливість відфільтровувати записи за даними в описі події.

Мережні пристрої, такі як маршрутизатори, комутатори, бездротові AP і брандмауери, незмінно передають телеметричні дані через протокол SNMP або Syslog. SNMP був спроектований наприкінці 1980-х для керування множиною пристроїв у мережі Internet, що бурхливо розвивається,. Диспетчери SNMP збирають телеметричну інформацію від агентів через UDP-порт 162. Диспетчери можуть використовувати SNMP-команди Get для запиту конкретних телеметричних даних, називаних змінними (variable), або пасивно чекати звіту про важливі події від агентів через повідомлення Trap. Для моніторингу в сферах виробництва й безпеки досить збирати повідомлення Trap. Для поглибленого аналізу тенденцій і планування ресурсів варто опитувати агентів за допомогою команд Get.

Syslog

Syslog – стандарт протоколювання подій для UNIX. Перевага Syslog перед механізмом протоколювання подій Windows полягає в тому, що весь процес консолідації

потоків подій від численних систем – невід'ємна частина Syslog. У дійсності Syslog одночасно мережний протокол і формат журналу, і за замовчуванням він використовує UDP-порт 514. Кожне повідомлення Syslog містить поля дати, часу, пріоритету, ім'я хосту й тексту повідомлення. З технічної точки зору пріоритет – число від 0 до 191. Однак більшість додатків Syslog відображають пріоритет у вигляді двох складових: Facility і Level.

Facility. Спочатку Syslog проектувався для моніторингу BSD Unix, і величина Facility використовувалася для ідентифікації процесу Unix про який свідчить подія. Значення від 0 до 15 відповідають найважливішим процесам Unix, а значення від 16 до 23 (з іменами від Local0 до Local7) призначені для додатків і пристроїв. Більшість мережних пристроїв використовують значення від Local0 до Local7 (наприклад, пристрої Cisco задіють Local6 і Local7), але не всі. Маршрутизатор Xincom Twin Wan використовує майже всі низькі значення Facility.

Level. Інший елемент пріоритету повідомлень Syslog – Level, значення якого перебувають у межах від 0 до 7. Level характеризує ступінь важливості повідомлення.

Продуктивність і стан

Для повного функціонального моніторингу корисно контролювати об'єкти продуктивності (performance-object) і стан серверів з окремого комп'ютера або від провайдеру послуг. Адміністратори, не знайомі з об'єктами продуктивності, можуть досліджувати їх за допомогою оснащення Performance консолі MMC. Різниця між моніторингом журналу подій і об'єкта продуктивності наступна: з журналів подій можна витягти інформацію про будь-яку частину системи, у якій відбулися неполадки, а об'єкт продуктивності дозволяє переконатися, що конкретні параметри перебувають у припустимих межах. Наприклад, за допомогою об'єктів продуктивності можна стежити за простором жорсткого диска, так як системний журнал видає попередження, тільки коли тім заповнюється настільки, що користувач уже починає випробовувати незручності.

Ще одна типова перевірка із застосуванням об'єкта продуктивності – моніторинг коефіцієнта використання центрального процесора з відстеженням певних рівнів протягом тривалих періодів часу (наприклад, понад 90% протягом 10 хвилин). Однак при перевірках коефіцієнта використання центрального процесора варто проявляти обережність; неважко переплутати корисне навантаження з некерованим процесом і згенерувати помилкове повідомлення про проблему. Чудова властивість об'єктів продуктивності полягає в тому, що інші додатки можуть створювати власні об'єкти продуктивності й публікувати телеметричні дані, специфічні для даного додатка. Наприклад, Active Directory (AD), SQL Server і Exchange Server мають у своєму розпорядженні власні об'єкти продуктивності.

Відсутність повідомлень про помилки в журналі й показники продуктивності в межах припустимих порогів – гарні індикатори коректного функціонування. Однак деякі проблеми не знаходять відбиттів в індикаторах. Перевірки стану серверів – найефективніший спосіб переконатися в тому, що сервери й додатки працюють у мережі й успішно обробляють запити. Перевірки стану серверів надійні, так як вони виконують тестову транзакцію. Багато провайдерів додатків і служб в Internet дозволяють регулярно проводити тестові транзакції із сервером через обрані споживачем інтервали часу. Для Web-серверу можна періодично запитувати дану Web-сторінку й перевіряти, чи успішно вона передана. Для системи SQL Server можна періодично виконувати запит і перевіряти результати.

Однак навіть перевірки стану не розкривають всіх проблем. Наприклад, простий сигнал ping, переданий через кожні 5 мінут, дозволяє переконатися, що операційна система й набір протоколів активні, але не містять ніякої інформації про стан самого додатка. Мені доводилося зустрічати завислі сервери, які відповідали на сигнали ping. Аналогічно простий запит HTML-сторінки із сервера не доводить, що відповідний додаток електронної комерції на базі Active Server Pages (ASP) працює коректно.

Тому перевірки стану повинні бути як можна більш функціональними. Ще одне застереження: програму перевірки стану варто розміщати поза контрольованим виробничим середовищем. Якщо помилково розмістити програму перевірки стану на контрольованому

сервері, то, наприклад, не вдасться визначити відмову сервера або серверного з'єднання, так як додаток не зможе передати адміністраторові відповідне повідомлення. Але якщо додаток перевірки працює на окремому сервері (і якщо цей сервер доступний з Internet), то єдиний випадок, коли важливий додаток буде не готовий до роботи без відома адміністратора, – одночасна відмова виробничого й контролюючого середовища.

Необхідний інструментарій

Отже, що потрібно для моніторингу всіх пристроїв, серверів, журналів, пасток SNMP і подій Syslog? Очевидно, необхідні один-два інструмента за доступною ціною, що охоплюють всі елементи, які потрібно відслідковувати. Продукти моніторингу високого рівня, такі як Argent Guardian і Microsoft Operations Manager (MOM), дозволяють контролювати всі об'єкти продуктивності, журнали подій Windows, пастки SNMP, потоки подій Syslog і навіть виконувати різні перевірки стану. Деякі не настільки великі, менш дорогі пакети, такі як Sentry II компанії Engagent, EventTracker компанії Prism Microsystems і комплекс Event Log Management компанії Dorian, охоплюють підмножину телеметричних джерел і обмежений набір об'єктів продуктивності.

Збираючись придбати інструмент, варто скласти список всіх характеристик, які необхідно знати, і підібрати інструмент, що контролює їх всі. Якщо інструмент не забезпечує моніторинг важливого параметра, наприклад SNMP, то заповнити пробіл можна за допомогою безкоштовної або недорогої умовно безкоштовної утиліти. Далі буде розглянутий ряд таких інструментів, з яких можна скласти ефективний комплекс моніторингу.

Розглянемо три корисних інструменти, які можна додати до арсеналу мережного моніторингу: Log Parser, безкоштовний інструмент Microsoft; tail, відмінну утиліту з миру UNIX; утиліту Kiwi Syslog Daemon, що представлена безкоштовною й могутнішою, але проте недорогою версією. Інформація буде корисна навіть тим адміністраторам, які вже мають або мають намір придбати інструменти: більшість інструментів на ринку розташовують лише функціями попереджень і звітності, доповненими шаблоновими звітами й зразковими правилами розсилання попереджень. Методи проектування й аналізу, описані в розділі, будуть надзвичайно корисні навіть для власників розгорнутої на підприємстві програми моніторингу.

Моніторинг текстових журналів

Більшість серверних продуктів Microsoft і компонентів Windows протоколюють будь-які важливі події в журналах System або Application, але кожна служба або основний компонент операційної системи (наприклад, IIS, DHCP, SMTP, Internet Authentication Service, IAS) записують більш докладну інформацію у власний текстовий журнал у спеціальному форматі. Для моніторингу інформації, наявної тільки в цих текстових файлах, зручно використовувати Log Parser. Log Parser розпізнає будь-який формат текстового файлу з розмежувачами, наприклад із символами табуляції або комами (CSV), і дозволяє задіяти ту ж команду SQL Select для опитування текстових файлів журналів.

Таким чином, за допомогою Log Parser можна вирішити задачу підготовки звітів на основі текстових журналів. Але що робити із попередженнями про критичні події, що поступають у реальному часі, інформація про які зберігається в текстових файлах? Я рекомендую скористатися інструментом tail, запозиченим з UNIX. Tail відслідковує додавання нових рядків у зазначені користувачем текстові файли. Як тільки виявляються нові дані, tail посилає їх у стандартний вихідний потік (stdout). Вихідні дані tail можна направити в сценарій, що аналізує нові записи в міру їхнього протоколювання й при необхідності генерує попередження. Наприклад:

```
tail /f logfile.txt |LoopOnNewMessages.cmd
```

виявляє нові повідомлення, додані у файл logfile.txt, і направляє їх в LoopOnNewMessages.cmd. Loop OnNewMessages.cmd передає кожне повідомлення за адресою rsmith@ultimatewindowssecurity.com, але замість нього можна вказати будь-яку іншу поштову адресу. Щоб не пересилати не занадто істотні повідомлення, можна доповнити сценарій фільтруючою логікою.

Tail

Витягти інформацію з журнальних файлів або інших текстових файлів можна за допомогою широко розповсюдженої програми grep. Але є й інший корисний інструмент із миру UNIX, гідний зайняти місце в інструментальному наборі адміністратора Windows, – програма tail. По суті, tail показує останні кілька рядків текстового файлу – це особливо корисно при аналізі файлів журналів. Наприклад, якщо адміністратор становить нові правила для брандмауера, tail покаже, як правила відбиваються на файлі журналу.

Tail є в більшості систем UNIX, а версію Win32 можна завантажити в рамках прийняття умов ліцензії GNU з Web-вузла Sourceforge (*http://unxutils.sourceforge.net*). Спочатку потрібно завантажити файл UnxUpdates.zip, а потім витягти tail.exe у своєму комп'ютері.

Автономне використання Tail

При використанні поза комбінацією з іншими програмами, tail показує кілька останніх рядків текстового файлу. За допомогою декількох параметрів можна змінити подання інформації на екрані. Особливо корисний параметр follow (-f), що дозволяє безупинно відслідковувати й виводити на екран зміни в текстовому файлі. Наприклад, команда

tail -f ex050410.log

показує останні 10 рядків журнального файлу з ім'ям ex050410.log і буде відслідковувати й відображати нові записи в міру їхньої появи. Якщо файл являє собою журнал Web-служби Microsoft IIS і хто-небудь звертається до Web-вузла, IIS зробить у журналі новий запис. Нові додавання негайно відображаються на консолі, у якій працює tail. Цей параметр спрощує діагностику, дозволяючи негайно побачити нові записи.

Спільне застосування Tail і Grep

Як відомо, grep – програма, що веде пошук зазначених послідовностей символів у цільовому текстовому файлі. Наприклад, при діагностиці комп'ютера, що працює з Windows Firewall, потрібно відшукати в журналі брандмауера дії, зроблені в певний день. Журнал не розділений по датах і досить великий.

За допомогою grep можна витягти рядки даних від 7 березня 2009 року й записати їх у новий текстовий файл:

```
grep « 2009-03-07» p-firewall.log
> 030705 p-firewall.log
```

Як щодо tail? Команду можна використовувати для обробки журналів брандмауера в процесі діагностики або відстеження атак у реальному часі. Але можна застосувати tail разом з grep, щоб виводити на екран тільки певні дані.

Для початку варто настроїти брандмауер на запис журналів у текстовий файл. Всі системи UNIX використовують syslog для протоколювання подій; більшість комерційних брандмауерів також підтримують syslog. Якщо на системі UNIX використовуються grep і tail, то варто настроїти брандмауер на пересилання даних syslog у хост-машину syslog. Користувачі Windows можуть установити й працювати із сервером syslog на базі Windows. Я рекомендую Kiwi Syslog Daemon фірми Kiwi Enterprises, відмінний інструмент для збереження даних syslog у текстовому файлі.

Потім потрібно побудувати шаблон на основі синтаксису постачальника брандмауера. Наприклад, адміністратор використовує брандмауера Cisco PIX і хоче одержувати оповіщення щораз, коли хтось звертається до Web-служб через брандмауера. За допомогою tail і grep можна в реальному часі виявляти в журналах символи «/80» (представляють Web-трафік у журналі PIX), наприклад:

tail -f pix.log | grep «/80»

Більш вдалий підхід – використовувати метасимволи регулярних виражень, які забезпечують більше складну фільтрацію, чим звичайні текстові рядки:

tail -f pix.log | grep /80[[:space:]]

Освоєння регулярних виражень вимагає часу, але в нагороду ви одержуєте бібліотеку корисних і ефективних шаблонів, які можна використовувати для пошуку майже будь-яких даних, – безсумнівно, це виправдує витрачені зусилля.

Ускладнений Tail

Grep і tail – прості у використанні й дуже гнучкі програми. При роботі з консольними додатками обидва інструменти значно спрощують аналіз журнальних файлів і повсякденне адміністрування. Версія командного рядка tail – швидка й проста в експлуатації, і, імовірно, прихильники строгих правил нададуть їй перевагу завдяки простоті й можливості пересилати вихідні дані в інші програми, такі як grep. Але існують версії tail із графічним інтерфейсом Windows, причому деякі з них наділені більш складними функціями, наприклад кольоровим виділенням співпадаючих послідовностей. Таке форматування допомагає відзначати важливі файли.

Зразок безкоштовної графічної програми tail для Windows – BareTail компанії Bare Metal Software (її можна завантажити за адресою *http://www.baremetalsoft.com/baretail*). Як і tail, BareTail відображає текстовий файл і відслідковує доповнення до файлу, але оскільки BareTail працює із графічним інтерфейсом, вона має у своєму розпорядженні функції виділення.

Завдяки таким функціям простіше виявити певний текст (наприклад, конкретну IP-адресу або порт) «на ходу», спостерігаючи за журналом брандмауера. Можна також змінити шрифт, без праці скопіювати рядок тексту й відкрити недавно переглянуті файли журналів за допомогою списку недавно використаних файлів Windows.

Моніторинг SNMP і журналів Syslog

Контролювати телеметричні джерела SNMP і Syslog легко завдяки безкоштовній версії програми Kiwi Syslog Daemon компанії Kiwi Enterprises. Цей диспетчер серверів Windows Syslog і SNMP дозволяє зібрати всі телеметричні дані про мережні пристрої в одній програмі. Із графічного інтерфейсу програми можна настроїти фільтри для збору повідомлень, що відповідають певним критеріям, а потім указати одне або кілька дій, що вживаються у відповідь на повідомлення. Можна побудувати фільтри для видалення непотрібних повідомлень і вказати, що повідомлення, що залишилися, повинні генерувати попередження або зберігатися в базі даних для наступних звітів. За допомогою Kiwi Syslog Daemon можна фільтрувати повідомлення за часом дня, днем тижня, пристроям, рівню, IP-адресі звітного агента або рядкам у повідомленні. Крім того, інструмент може виконувати різноманітні дії – оповіщення по електронній пошті, збереження в базі даних по ODBC, запуск програми й інші – у відповідь на зазначені події.

Безкоштовна версія Kiwi Syslog Daemon інтерактивно працює в настільному комп'ютері, тому адміністратор повинен зареєструватися, щоб контролювати пристрої. Але розширена версія продукту функціонує як служба, а її вартість – усього 100 дол. для одного сервера. Якщо активізувати моніторинг пасток SNMP, необхідно також указати поля Facility і Level, які використовуються інструментом при перетворенні пастки в повідомлення Syslog. Наприклад, можна вказати пастки SNMP як Facility Local4 і Level 3 Error. Потім можна скласти правила розсилання попереджень спеціально для пасток SNMP шляхом фільтрації повідомлень Local4.

Отже, існують ресурси для моніторингу різноманітних джерел телеметричних даних. Перш ніж почати проектувати власне рішення для моніторингу, корисно познайомитися з інструментами, які є на ринку. Вони доступні й повнофункціональні. Однак не можна одержати повне рішення, просто здобуваючи інструмент. Потрібно визначити критерії для звітів і попереджень, щоб не одержувати занадто багато повідомлень про незначні події, але не слід упадати в іншу крайність і задавати настільки строгі критерії, що рішення моніторингу може перешкодити виконанню тої самої задачі, для якої воно призначалося. Для досягнення балансу варто становити критерії, відтинаючи незначні, а не вибираючи важливі події. Єдине виключення із цього правила – журнал Security, що набагато коротше, а крім того, краще документовано. Повна база даних подій журналу Security і їхніх значень

опублікована в Security Log Encyclopedia на сайті Ultimate Windows Security (*www.ultimatewindowssecurity.com*).

Для підготовки ефективної й вичерпної процедури моніторингу потрібно прикласти певні зусилля, але вони не пропадуть впусту. Немає нічого гірше, ніж довідатися про проблему від користувачів і після перегляду журналів виявити, що попередження надходили трьома днями раніше.

Вимоги бізнесу й законодавства щодо інформаційної безпеки й звітності дуже високі. Порушення безпеки можливі, але адміністратор і його компанія набагато успішніше переборють труднощі, якщо добре підготуються до критичної ситуації. Ефективному моніторингу немає повноцінної заміни.

Розробка удосконаленого методу моніторингу мережі

Як було відмічено вище, однією з основних проблем, що стоять зараз перед розроблювачами систем керування комп'ютерними мережами є проблема достовірного надання даних про їхній стан. Потреба в надійно працюючих великих комп'ютерних мережах усе вища з кожним днем. Тому при проектуванні сучасних систем керування дуже важливу роль відводять розробкам оптимізованих за часом алгоритмам збору й обробки даних.

Імовірність збереження актуальності інформації на момент її використання чисельно дорівнює:

$$P = \frac{c^2}{(c+b)(c+q)}, \quad (1)$$

де c – середній час значимої зміни реальної інформації щодо інформації, збереженої в БД;

b – середній час підготовки, передачі й уведення інформації для відновлення БД;

q – середній час між двома послідовними опитуваннями того самого пристрою.

Ідеальним випадком є ситуація, коли інформація про будь-яку зміну стану досягає БД у момент зміни. З використанням механізму багатопоточності можна «скоротити» час очікування інформації від джерела, використавши його для обробки інформації від іншого джерела.

На представленому нижче графіку (рисунок 1) видна залежність імовірності збереження актуальності даних від відношення часу очікування відповіді до часу обробки даних.

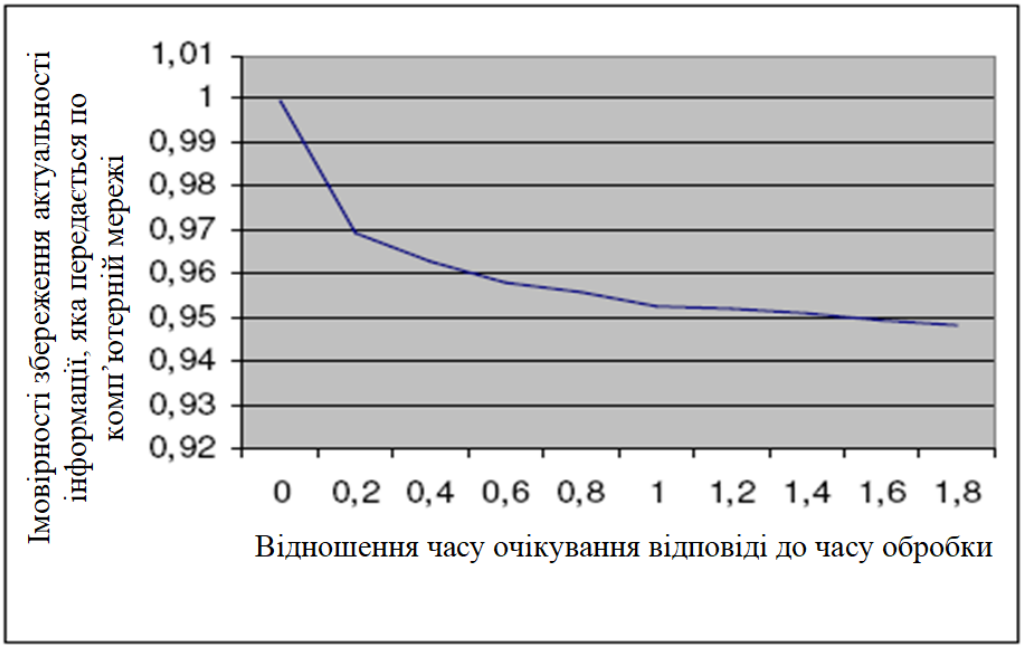

Рисунок 1 – Залежність імовірності збереження актуальності від відношення часу очікування відповіді до часу обробки

Із графіка видно, що навіть незначне скорочення часу очікування даних може привести до істотного збільшення ймовірності збереження актуальності даних, якщо час очікування приблизно дорівнює часу обробки. Можна зробити висновок, що для мереж, у яких при роботі системи моніторингу час обробки інформації приблизно дорівнює часу очікування після запиту, їсти можливість істотно підвищити актуальність інформації при використанні алгоритмів, що дозволяють використовувати час простою для обробки інформації.

Сучасні операційні системи дозволяють використовувати багатопоточну схему роботи додатків. Це досягається за рахунок розподілу робочого часу процесора між різними додатками, що дозволяє декільком програмам працювати «одночасно».

Весь процесорний час персонального комп'ютера (сервера) можна розділити на періоди. В один такий період всі додатки одержують по «шматочку» процесорного часу для своїх потреб. Їхні розміри залежать від пріоритетів додатків в операційній системі. Чим вище пріоритет – тим довший часовий інтервал, у якому додаток використовує процесор. У рамках даного інтервалу додаток також може й не використовувати процесор (перебуває в стані очікування). Чим більше потоків «всередині» у додатка, тим більше додаток в цілому одержує цього процесорного часу.

Для складних математичних задач введення механізму паралельного розрахунку даремно, тому що час роботи додатка приблизно дорівнює часу процесора для розрахунку задачі. Для задач моніторингу дана схема навпроти є досить вигідною. Вона дозволяє одержати виграш за часом за рахунок того, що в процесі опитування робочої станції є часові інтервали, у яких не використовуються ресурси процесора, пам'яті, мережі. Це інтервали очікування відповіді від віддаленого пристрою. Тим самим, якщо використовувати ці інтервали для роботи інших потоків, то можна буде зменшити час «простою» процесора. А за рахунок цього відбувається оптимізація за часом. Очевидно, що якщо змусити процесор працювати без «простою» у рамках відведеного для додатка процесорного часу й при цьому не допускати переповнення пам'яті або надлишкового мережного трафіку, така система буде максимально ефективна.

Як уже було визначено раніше, збільшення кількості паралельно опитуваних робочих станцій спочатку збільшує продуктивність системи, а після створення n+1 потоку, навпаки, сповільнюють її. Це пов'язане з закінченям обчислювальних ресурсів.

На продуктивність системи впливають три основних фактори: завантаженість центрального мікропроцесора, оперативний пам'яті, а також мережний трафік.

Очевидно, що поки всі три ресурси будуть використані не повністю, додавання нового потоку в систему буде збільшувати її продуктивність. Але як тільки один з ресурсів буде повністю вичерпаний, продуктивність системи або впаде, або втратить динаміку росту. Приміром, якщо при неповному завантаженні центрального процесора буде повністю зайнята доступна оперативна пам'ять, при додаванні нового потоку в систему, частину процесорного часу буде витрачатися на керування задачами по перевантаженню даних з оперативної пам'яті на жорсткий диск. Таким чином, одна з основних задач, яку необхідно вирішити при створенні систем моніторингу формулюється в такий спосіб: необхідно визначити максимальне число потоків, при якому система опитування робочих станцій буде працювати з максимальною ефективністю.

Щоб формально описати дану задачу необхідно визначити, як залежить продуктивність системи від значення вищеописаних факторів.

При запуску опитування мережі відбувається формування запиту і його відправлення віддаленому пристрою. Після цього потік переходить у стан очікування й до відповіді робітник станції майже не займає ресурсів процесора, пам'яті й не створює мережного трафіку. Природно, що в момент очікування відповіді одним потоком, використовувати ресурси процесора може інший потік.

Виходячи із усього вищесказаного й за умови необмежених ресурсів оперативної пам'яті й мережного трафіку, були визначені наступне співвідношення для визначення оптимальної кількості потоків, що додаються в систему моніторингу:

$$N = \frac{t_t}{t_p}, \tag{2}$$

де tt – тривалість потоку (від запиту даних до закінчення їхньої обробки);

tp – тривалість використання потоком ресурсів центрального процесора.

Дане співвідношення справедливо для ідеального випадку. У реальній ситуації процесор не може надати всі свої ресурси для системи моніторингу. Частина його ресурсів іде на керування операційною системою й іншими, а також використовується іншими додатками.

У зв'язку із цим протягом часу, за яке працює потік, до ресурсів процесора звертаються не тільки потоки системи моніторингу, але й інші додатки.

У такий спосіб вищенаведене співвідношення можна обмежити цією умовою:

$$N = \frac{t_t(1-P)}{t_p}, \tag{3}$$

де tt – тривалість потоку (від запиту даних до закінчення їхньої обробки);

tp – тривалість використання потоком ресурсів центрального процесора;

P – коефіцієнт завантаженості процесора іншими додатками (0 < P < 1).

Таке уточнення співвідношення дозволяє в будь-який момент часу визначити має сенс чи ні в цей момент часу додати додатковий потік у систему.

Але існують і інші фактори, що впливають на продуктивність системи моніторингу. Другим по значимості є завантаженість оперативної пам'яті. Усе раніше наведені міркування дійсні за умови, що оперативна пам'ять не повна, тобто система не використовує файл підкачування. У випадку ж його використання існують додаткові часові витрати за часом на перевантаження даних з файлу до пам’яті й обернено.

При цьому варто пам'ятати, що перевантаження даних відбувається тільки в тому випадку, коли потік готовий до виконання. А це відбувається не на кожному циклі ітерації в керуючому потоці.

Тому для випадку, коли всі потоки не будуть міститися в оперативній пам'яті, то попереднє співвідношення не може бути використана. У випадку недостачі оперативної пам'яті необхідно використовувати наступну формулу:

$$N = \frac{t_t(1-P)}{t_p + M \cdot t_s}, \tag{4}$$

де tt – тривалість потоку (від запиту даних до закінчення їхньої обробки);

tp – тривалість використання потоком ресурсів центрального процесора;

P – поточна завантаженість процесора (0 < P < 1);

ts – час перезавантаження даних з оперативної пам'яті у файл;

M – число таких перезавантажень.

Визначення значення коефіцієнта M не представляє особливої праці. Перевантаження відбудеться тільки тоді, коли потік перебуває в стані роботи, а не очікування. У зв'язку із цим значення M можна визначити за формулою:

$$M = \frac{t_p}{t_i}, \tag{5}$$

де tp – тривалість використання потоком ресурсів центрального процесора;

ti – час, на який надається доступ до ресурсу процесора потоку, при передачі йому керування в одній ітерації

Немаловажним фактором є завантаженість мережі. Очевидно, що при повному завантаженні мережі про паралельність також безглуздо говорити. Потоки будуть формуватися в послідовні черги, і при цьому мережа буде практично непрацездатна для інших додатків і користувачів. У двох попередніх випадках, перевантаження параметра вело лише до істотного вповільнення роботи одного комп'ютера, а для мережного трафіку може паралізувати роботу всієї мережі. Через це для кожної конкретної мережі встановлюється гранично припустимий мережний трафик, що може створювати система моніторингу. Його розрахунок ведеться з розрахунку розмірів мережі, її швидкості, кількості мережних додатків, часу доби й т.д.

Поєднуючи все вищесказане в єдину задачу одержуємо, що:

якщо оперативна пам'ять не переповнена:

$$N = \frac{t_t(1-P)}{t_p}, \tag{6}$$

−якщо оперативна пам'ять переповнена:

$$N = \frac{t_t(1-P)}{t_p + \frac{t_p}{t_i} \cdot t_s}, \tag{7}$$

−Обидва співвідношення обмежені умовою, що не перевищена установлена межа мережного трафіку, створюваного системою моніторингу.

−Виконання цієї умови, а також визначення переповнення оперативної пам'яті відбувається відповідно до раніше певних співвідношень.

−Всі параметри, необхідні для розрахунку оптимальної кількості потоків надаються операційною системою.

−Визначивши оптимальну кількість потоків у будь-який момент часу, була вирішена тільки половина поставленої задачі. У класичній багатопоточній схемі опитування мережі, нам потрібно розподілити весь діапазон IP-адрес на N груп, і провести опитування. Але класична схема не враховує того, що в сучасному житті під один додаток у мережі не виділяється сервер. Завантаженість сервера, на якому встановлена система моніторингу, постійно міняється через використання інших додатків, розташованих на ньому. А зі зміною завантаженості сервера міняється оптимальна кількість потоків, у рамках яких проходить опитування кінцевого або мережного устаткування. Друга проблема класичної схеми полягає в тому, що час опитування одного пристрою залежить від його типу, об'єму збирається информации, що, часу, необхідного на її обробку, а також швидкості каналу зв'язку. При «класичному» розподілі IP-адрес на групи всі ці моменти не враховуються, а це приводить до того, що в деякий момент часу одна група повністю оброблена, а інша ні.

−З обліком цих двох причин для рішення поставленої задачі дана модель була дороблена в такий спосіб:

−Визначаємо оптимальне число потоків у цей момент

−Запускаємо п потоків у яких відбувається опитування перших п адрес діапазону.

−Як тільки в якому-небудь потоці опитування закінчиться (або буде встановлена його неможливість) відбувається визначення оптимального числа потоків у цей момент.

−Якщо оптимальне число потоків менше поточного, то потік (у якому закінчилася робота) знищується. Якщо більше, то створюється ще к потоків (к = оптимальне число потоків – існуюче число потоків).

−Така схема дозволяє відслідковувати зміну стану системи в часі й дозволяє рівномірно розподіляти пристрою між потоками. На рисунку 2 представлений описаний вище алгоритм у графічній формі. Для того щоб оцінити запропонований вище алгоритм системи моніторингу, було проведено його моделювання, а також моделювання класичних

алгоритмів, у системі GPSS WORLD. Всі необхідні для такого моделювання параметри були визначені на реальній мережі. На їхній підставі були виведені наступні базові значення:

–Середнє число потоків для опитування мережі дорівнює 4.

–Середній час опитування 1 пристрою становить 80 секунд.

–Кількість перевантажень процесора або пам'яті серверів, на яких установлені засоби керування мережею рівнялося в середньому 6 разів за годину.

–На підставі цих даних було зроблене моделювання опитування великої локальної мережі (більше 250 комп'ютерів) для трьох реалізацій моніторингу (однопотоковий, багатопоточний і динамічний багатопоточний). За отриманим даними були побудовані графіки продуктивності різних реалізацій моніторингу (рисунок 3).

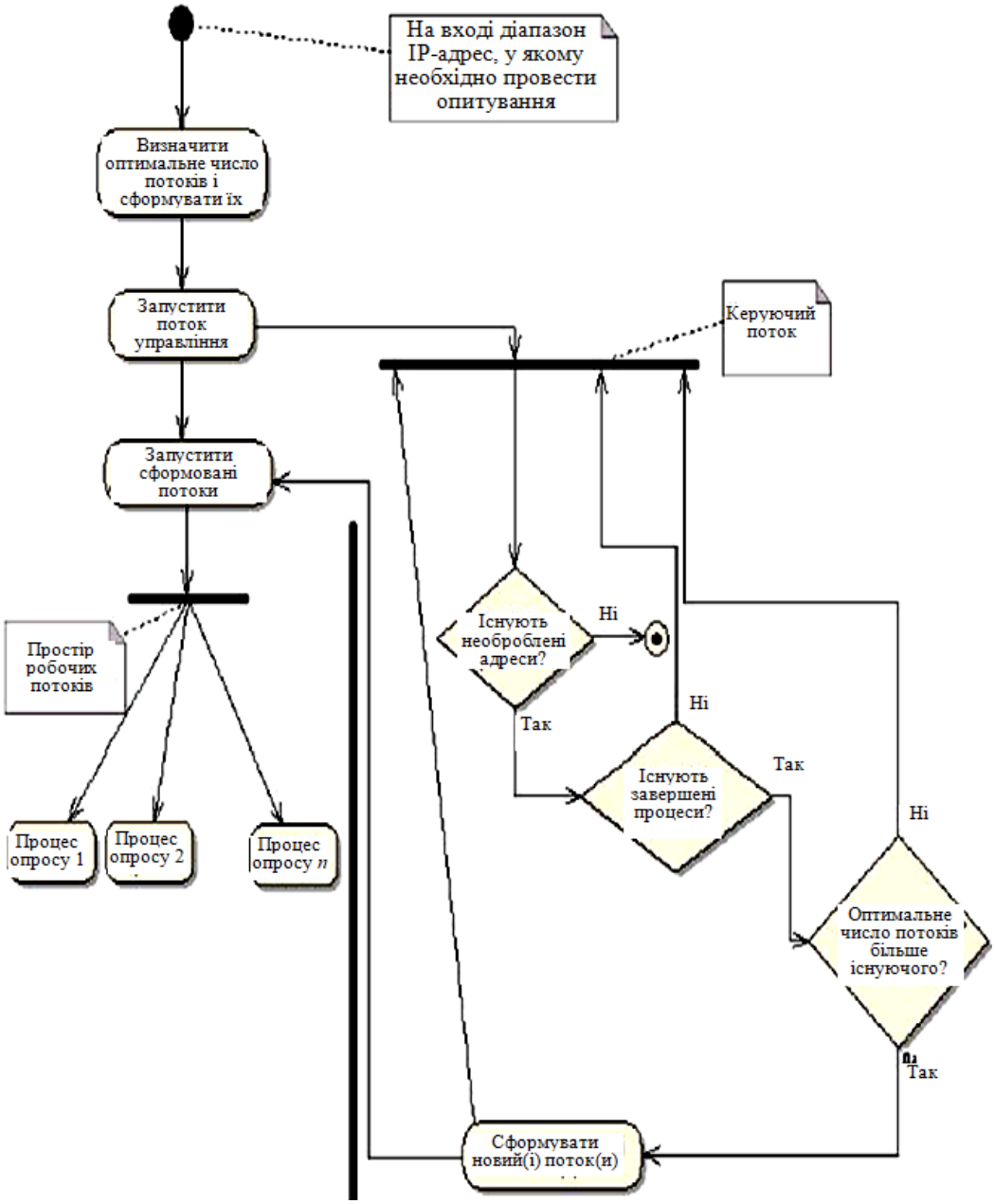

Рисунок 2 – Схематичне представлення удосконаленого алгоритму моніторингу мережі

Результати моделювання підтвердили раніше висунуті припущення. Модернізований багатопоточний алгоритм був реалізований у системі моніторингу. Після проведення тестових випробувань системи на мережі отримані дані підтвердили результати моделювання. Порівняння результатів отриманих при моделюванні й випробуванні системи

моніторингу, побудованої на модернізованому багатопоточному алгоритмі, представлені на рисунку 4.

На підставі аналізу наведених вище даних установлено, що час збору й обробки інформації для мережних або кінцевих пристроїв було скорочено більш ніж на 20%. Це означає, що час між повторними опитуваннями однієї й тієї ж робочої станції при круговому безперервному опитуванні змінилося, і стало становити 80% від того, котре забезпечує система моніторингу, побудована по класичній моделі. Таким чином, імовірність збереження актуальності даних для системи моніторингу, побудованої на алгоритмах розроблених у даній роботі для мереж збільшився з 92% до 95,5%.

Даний результат перевершує 95%, що у цей момент вважається мінімальним рівнем для вимог по актуальності даних.

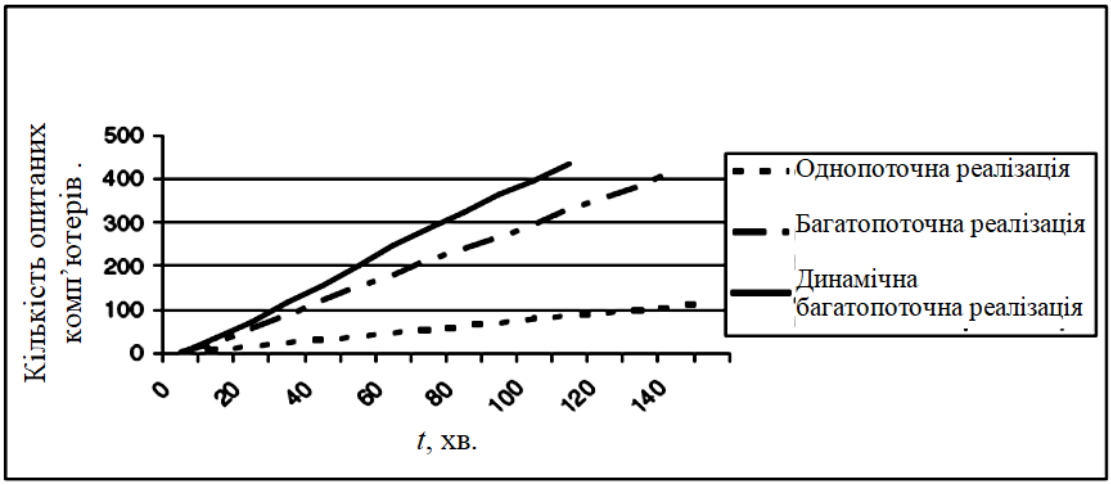

Рисунок 3 – Продуктивність різних реалізацій моніторингу при моделюванні

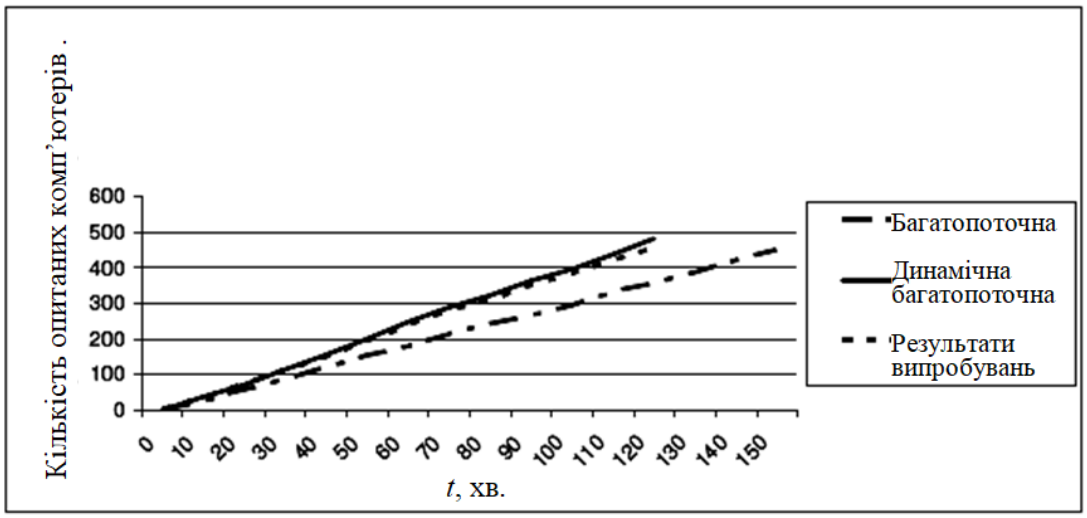

Рисунок 4 – Продуктивність різних реалізацій моніторингу при реальному випробуванні

Розробка структурної схеми

Структурна схема системи зображена на рисунку 5. З рисунку видно, що моніторинг локальної мережі здійснюється по трьох напрямках: Моніторинг обладнання; Моніторинг ресурсів; Моніторинг трафіку.

Моніторинг обладнання включає в себе побудову списку наявного обладнання та здійснення його контролю. До мережного обладнання, що підлягає моніторингу, відносяться: персональні комп'ютери, ноутбуки, сервери, принтери, ip-телефони.

Моніторинг ресурсів дозволяє переглядати та завантажувати наявні в мережі ресурси, а також розміщювати чи приховувати для загального доступу свої ресурси. До ресурсів

локальної мережі відносяться: файли, мультимедіа, бази даних, сервіси інформаційної безпеки, список користувачів.

Моніторинг трафіку використовується для контролю вхідного та вихідноготрафіку. Він включає у себе контроль підключених інтерфейсів, статистику подій по основним мережним протоколам: TCP, UDP, IP та ICMP.

TCP – один з основних мережних протоколів Інтернету, призначений для управління передачею даних в мережах і підмережах TCP/IP.

UDP – один із протоколів в стеку TCP/IP. Від протоколу TCP він відрізняється тим, що працює без встановлення з'єднання. UDP – це один з найпростіших протоколів транспортного рівня моделі OSI, котрий виконує обмін дейтаграмами без підтвердження та гарантії доставки.

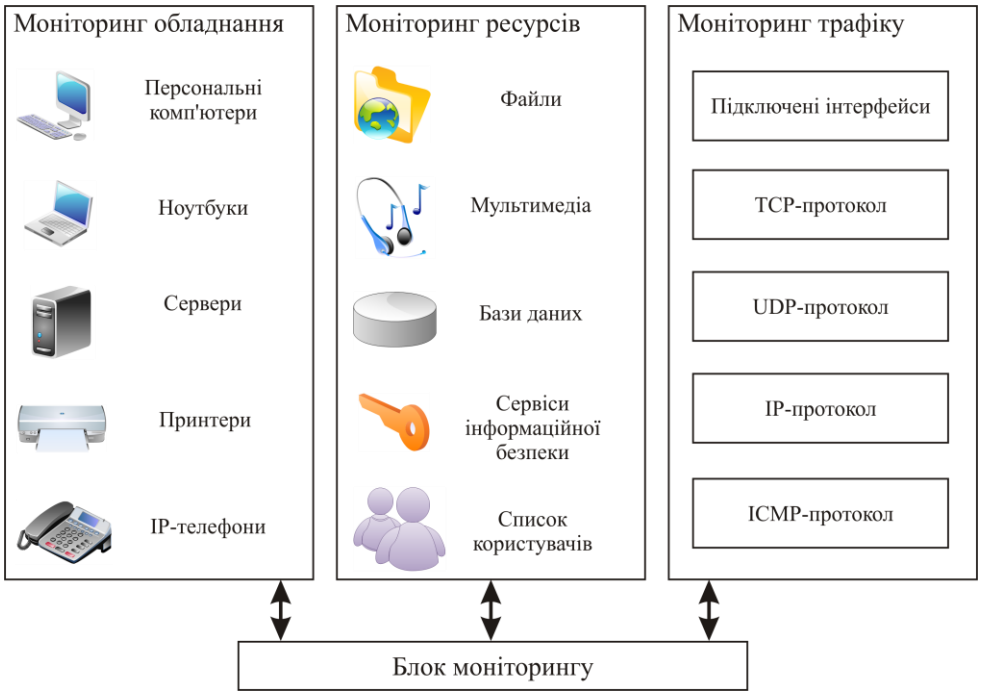

Рисунок 5 – Структурна схема системи

IP – найбільш широко розповсюджена реалізація ієрархічної схеми мережної адресації. Використовуваний в мережі Інтернет, протокол відповідає за адресацію пакетів, але не відповідає за встановлення з'єднань, не є надійним і дозволяє реалізувати тільки негарантовану доставку даних.

ICMP – мережний протокол, що входить в стек протоколів TCP/IP. В основному ICMP використовується для передачі повідомлень про помилки й інші виняткові ситуації, що виникли при передачі даних. Також на ICMP покладають деякі сервісні функції, зокрема на основі цього протоколу заснована дія таких загальновідомих утиліт як ping та traceroute.

Висновки. У статті наведені теоретичне узагальнення й рішення наукового завдання дослідження методів моніторингу LAN мереж інформаційних та комп’ютерних систем. Рішення даного завдання полягало у вирішенні наступних задач: Був проведений огляд існуючих систем моніторингу LAN мереж інформаційних та комп’ютерних систем; Досліджена система моніторингу LAN мереж інформаційних та комп’ютерних систем; На основі отриманих результатів досліджень створена програмна реалізація системи моніторингу LAN мереж інформаційних та комп’ютерних систем; Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання моніторингу LAN мереж інформаційних та комп’ютерних систем. Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної

діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Список літератури

1. О.А. Смірнов, П.С. Усік, «дослідження перспектив використання технологічних рішень в мережах 5g» у Кібербезпека та інформаційні технології: монографія. – Х. : ТОВ «ДІСА ПЛЮС», 2020.С. 122-135.
2. Смірнов О.А., Дрєєва Г.М., «Метод генерування фрактального трафіку за допомогою моделі генератора на графі» у Інформаційна безпека та інформаційні технології: монографія / за заг. ред. В. С. Пономаренка. – Х. : Вид. Рожко С.Г. 2019. С. 123-139.
3. Смирнов А.А., Коваленко А.В. Комплекс математических моделей технологии тестирования web-приложений. Інформаційні технології: сучасний стан та перспективи: монографія / За загальною редакцією В.С. Пономаренка. – Х.: ТОВ «ДІСА ПЛЮС», 2018. – 461 с.
4. Смирнов А.А., Коваленко А.В. Разработка метода управления рисками разработки программного обеспечения. Інформаційні технології: проблеми та перспективи: монографія / За загальною редакцією В.С. Пономаренка. – Х.: Видавець Рожко С.Г., 2017. – 447 с.
5. Смірнов О.А., Смірнова Т.В., Буравченко К.О., Кравченко С.С., Горбов В.О., «Хмарна система підтримки прийняття рішень технологічного процесу відновлення поверхонь конструкцій і деталей машин». Сучасні інформаційні системи. 2021. Т. 5, № 4. С. 79-95
6. Смірнов, О.А., Полігенько О.О., Одарченко Р.С., Терещенко Л.Ю.Усік П.С., «Інформаційна технологія та програмне забезпечення для підвищення ефективності планування підсистеми базових станцій стільникового зв'язку». Проблеми телекомунікацій. № 1(26). С. 83-96. 2020.
7. Смірнов О.А., Усік П.С., Миронець І.В., Буравченко К.О., Якименко Н.М. «Метод підвищення ефективності розподіленої обробки даних у комп'ютерних системах операторів стільникового зв'язку» Вісник Черкаського державного технологічного університету. Технічні науки. №4. С. 103-110. 2020.
8. О.А.Смірнов, Т.В.Смірнова, Л.І. Поліщук, К.О. Буравченко, А.О.Макевнін, «Дослідження хмарних технологій як сервісів», Кібербезпека: освіта, наука, техніка. № 3(7). С. 43-62. 2020.
9. Smirnov, O., Neskorodieva, T., Fedorov, E., Rudakov, K., Neskorodieva, A. «Method Detection Audit Data Anomalies on Basis Restricted Cauchy Machine» CEUR Workshop Proceedings, Volume 3187, 2022, pp. 1-12. (Scopus).
10. Smirnov O., Smirnova T., Anas M. Al-Oraiqat, Drieiev O., Polishchuk L., Sheroz Khan, Yassin M. Y. Hasan, Aladdein M. Amro, Hazim S. AlRawashdeh «Method for Determining Treated Metal Surface Quality Using Computer Vision Technology». Sensors (Basel, Switzerland) Volume 22, Issue 16, 6223, 2022. (Scopus).
11. Smirnov, O., Lakhno, V., Akhmetov, B., Chubaievskyi, V., Khorolska, K., Bebeshko, B. «Selection of a Rational Composition of İnformation Protection Means Using a Genetic Algorithm». In: Rajakumar, G., Du, KL., Vuppalapati, C., Beligiannis, G.N. (eds) Intelligent Communication Technologies and Virtual Mobile Networks. Lecture Notes on Data Engineering and Communications Technologies, vol 131. 2023. Springer, Singapore. pp. 21-34. (Scopus).
12. Smirnov O.A., Al-Oraiqat A.M., Ulichev O.S., Meleshko Ye.V., Al-Rawashdeh H.S., Polishchuk L.I. «Modeling strategies for information influence dissemination in social networks». Journal of Ambient Intelligence and Humanized Computing Volume 13, Issue 5. Springer, Cham. 2022, pp. 2463-2477. (Scopus).
13. Smirnov O., Kuznetsov A., Kryvinska N., Kiian A., Kuznetsova K. «Full Non-Binary Constant-Weight Codes». SN Computer Science, Vol 2, 337, 2021. https://doi.org/10.1007/s42979-021-00739-w (Scopus).
14. Smirnov O., Kovalenko O., Kovalenko A., Kavun S. «Quantitative Risk Assessment Method Development in the Context of the SDLC-model». 2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T), 2021, pp. 203-208, doi: 10.1109/PICST54195.2021.9772143 (Scopus).
15. Smirnov O., Neskorodieva T., Fedorov E., Rymar P. «Neural Network Modeling Method of Transformations Data of Audit Production with Returnable Waste». CEUR Workshop Proceedings Volume 3101, 2021, Pages 192-207. (Scopus).
16. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova K. «Data hiding scheme based on spread sequence addressing». CEUR Workshop Proceedings Volume 2805, 2020, Pages 44-58. (Scopus).
17. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». International Journal of Computing; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256. (Scopus).
18. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». CEUR Workshop Proceedings. Volume 2740, 2020, Pages 102-114. (Scopus).
19. Smirnov O.A., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». Journal of theoretical and applied information technology Vol.98. No 21, 2020, P. 3334-3346. (Scopus).
20. Smirnov O., Kuznetsov A., Arischenko A., Chepurko I., Onikiychuk A., Kuznetsova T. «Pseudorandom sequences for spread spectrum image steganography». CEUR Workshop Proceedings Volume 2654, 2020, Pages 122-131. (Scopus).