МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КІБЕРБЕЗПЕКИ, КОМП'ЮТЕРНОЇ
ТА ПРОГРАМНОЇ ІНЖЕНЕРІЇ
КАФЕДРА КОМП'ЮТЕРНИХ СИСТЕМ ТА МЕРЕЖ

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач випускової кафедри

_____ І.А. Жуков
(підпис)

«___» _____202__ р.

ДИПЛОМНА РОБОТА
(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ МАГІСТР

ЗА СПЕЦІАЛЬНІСТЮ 123 «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ»

Тема: «Система моніторингу та управління мережевим трафіком користувачів»

Виконавець: ___студент групи КС-231М Котов Ярослав Віталійович_____
(студент, група, прізвище, ім'я, по батькові)

Керівник: _____к.т.н., доцент Дрововозов Володимир Іванович_____
(науковий ступінь, вчене звання, прізвище, ім'я по батькові)

Нормоконтролер: _____ Малярчук В.О.
(підпис) (ПІБ)

Засвідчую, що у дипломній роботі немає запозичень праць інших авторів без відповідних посилань
Студент _____ Котов Я.В.
(підпис) (ПІБ)

Київ 2020

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, комп'ютерної та програмної інженерії
Кафедра комп'ютерних систем та мереж
Спеціальність 123 «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ І.А. Жуков
(підпис)
«_____» _____ 2020 р.

ЗАВДАННЯ
на виконання дипломної роботи

Котова Ярослава Віталійовича
(прізвище, ім'я, по батькові)

1. Тема роботи: «Система моніторингу та управління мережевим трафіком користувачів»
затверджена наказом ректора від «25» вересня 2020 р., № 1793/ст
2. Термін виконання роботи: з «05» жовтня 2020 р. по «30» грудня 2020 р.
3. Вихідні дані до роботи:
   3.1. Система моніторингу та управління мережевим трафіком
   3.2. Програмний модуль для відправки повідомлень
4. Зміст пояснювальної записки (перелік питань, що підлягають розробці):
   4.1. Вступ
   4.2. Аналіз методів і засобів моніторингу та управління мережевим трафіком користувачів
   4.3. Аналіз та вибір системи моніторингу та управління мережевим трафіком користувачів
   4.4. Реалізація та впровадження модифікації системи моніторингу та управління
   4.5. Рекомендації з супроводу та тестування роботи системи з інтегрованим програмним модулем
5. Перелік обов'язкового графічного матеріалу:
   5.1. Презентаційні матеріали

6. Календарний план-графік

№ пор.	Завдання	Термін виконання	Відмітка про виконання
1	Проаналізувати джерела, по темі дипломного проекту	05.10.2020 – 15.10.2020	
2	Проаналізувати матеріали відповідно методів та засобів моніторингу та управління мережевим трафіком користувачів	16.10.2020 – 26.10.2020	
3	Визначити основні задачі систем моніторингу та управління мережевим трафіком користувачів	27.10.2020 – 06.11.2020	
4	Виконати порівняльний аналіз систем моніторингу та управління	07.11.2020 – 17.11.2020	
5	Синтезувати рішення та пропозиції щодо вдосконалення системи та реалізувати інтеграцію рішення з системою	18.11.2020 – 26.11.2020	
6	Визначити основі вимоги та інструкції з експлуатації, та провести тестування роботи системи з інтегрованим програмним модулем	27.11.2020 – 30.11.2020	
7	Розробити структуру та зміст пояснювальної записки	01.12.2020 – 04.12.2020	
8	Оформити пояснювальну записку, розробити та оформити презентаційні матеріали до пояснювальної записки	05.12.2020 – 20.12.2020	
9	Захист дипломної роботи	22.12.2020	

Дата отримання завдання: «_05_» _____жовтня_____ 2020 р.

Керівник дипломної роботи: _____ _Дрововозов В.І._____
(підпис керівника) (ПІБ.)

Завдання прийняв до виконання: _____ _____Котов Я. В._____
(підпис випускника) (ПІБ.)

РЕФЕРАТ

Пояснювальна записка до дипломної роботи «Система моніторингу та управління мережевим трафіком користувачів», 90 сторінок, 46 рисунків, 6 таблиць, 30 літературних джерел.

КОМП'ЮТЕР, ОБ'ЄКТ, МЕРЕЖА, СЕРВЕР, ПРОГРАМА, СИСТЕМА.

Мета дипломної роботи – дослідити системи моніторингу та управління мережевим трафіком користувачів, та можливості з автоматизації процесів адміністрування мережі.

Завдання дипломної роботи: провести аналіз систем моніторингу та управління мережевим трафіком користувачів, синтезувати рішення та пропозиції щодо вдосконалення системи з метою вирішення актуальних проблем, та впровадити модифікацію системи з метою вирішення зазначених проблем.

Об'єкт дослідження – система моніторингу та управління мережевим трафіком користувачів, комплексна система моніторингу мережевої інтенсивності в режимі реального часу, система оповіщення та передачі даних.

Практична значимість роботи – забезпечення безперебійної роботи мережі, модифікована система моніторингу та управління мережевим трафіком користувачів, яка дозволяє адміністратору мережі швидко реагувати на несправності та слабкі місця в організації мережі.

Рекомендації щодо використання результатів дипломної роботи – об'єкт та концепція системи моніторингу та управління мережевим трафіком можуть бути в подальшому використані для ефективного адміністрування мережі.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

HTTP – Hyper Text Transfer Protocol

IP – Internet Protocol

OSI – Open System Interconnection

SQL – Structured Query Language

TCP – Transfer Control Protocol

UDP – User Datagram Protocol

ЕОМ – електронно-обчислювальна машина

ІС – інформаційна система

КМ – комп’ютерна мережа

ОС – операційна система

ВСТУП

Інформаційна інфраструктура сучасного підприємства представляє собою складний конгломерат різномасштабних і різнорідних мереж і систем. Щоб забезпечити їх злагоджену і ефективну роботу, необхідна керуюча платформа корпоративного масштабу з інтегрованими інструментальними засобами. Однак до недавнього часу сама структура індустрії мережевого управління перешкоджала створенню таких систем - «гравці» цього ринку прагнули до лідерства, випускаючи продукти обмеженою області дії, що використовують засоби і технології, не сумісні з системами інших постачальників. З розвитком телекомунікаційних мереж, інформаційних сервісів і лавиноподібним зростанням числа їх користувачів, у операторів зв'язку виникає проблема забезпечення якості надаваних послуг і безперебійного доступу до ресурсів мережі. Вона вирішується за рахунок впровадження систем моніторингу та управління мережевим трафіком користувачів, які повинні інформувати адміністраторів мережі про проблемні ситуації (події), що виникають внаслідок різних помилок, пов'язаних з роботою засобів зв'язку, використанням різного роду інформаційних сервісів, що генерують велику кількість з'єднань, проблем в області інформаційної безпеки і т.п. зазначені системи повинні володіти інструментами для проведення якісного та кількісного аналізу трафіку мережі, визначення стану роботи компонентів інфраструктури.

Виявлення аномальних подій в роботі мережі дозволяє своєчасно усунути або мінімізувати проблему, тим самим забезпечити належний рівень сервісу, як для роботи кінцевих споживачів, що використовують стандартні для користувача програми, так і інформаційних систем, для роботи яких потрібен телекомунікаційний ресурс з гарантованою пропускною здатністю або особливими параметрами.

Сьогодні ситуація змінюється на краще - з'являються продукти, які претендують на універсальність управління усім розмаїттям корпоративних інформаційних ресурсів, від настільних систем до мейнфреймів і від локальних мереж до ресурсів мережі. Одночасно приходить усвідомлення того, що керуючі додатки повинні бути відкриті для рішень всім користувачам.

Адміністратору слід пам'ятати, що з точки зору користувачів якість роботи прикладного програмного забезпечення в мережі виявляється визначальним. Всі інші критерії, такі як число помилок передачі даних, ступінь завантаженості мережевих ресурсів, продуктивність обладнання вторинними. "Хороша мережа" - це така мережа, користувачі якої не помічають, як вона працює.

Таким чином в процесі діяльності та збільшення обсягів надання послуг виникла проблема попереджувального виявлення несправних і слабких місць в організації мережі, тобто ставилося завдання впровадження рішення, що дозволяє прогнозувати необхідність заміни або модернізації ділянок мережі до того, як несправності позначиться на роботі абонентських вузлів. В тому числі з ростом клієнтської бази, а як наслідок і числа активного обладнання, виникла необхідність оперативного відстеження стану мережі в цілому і окремих її елементів в подробицях.

У багатьох сучасних підприємствах існують локальні мережі, у складі яких входять: сервери, точки доступу, персональні робочі станції та інше мережеве обладнання, що виконує загальнодоступні та, тимчасово, дуже важливі функції. Як правило, місцева мереже підприємств має одну або більше точок виходу у зовнішні глобальні мережі, у тому числі в Інтернеті. Більшість компаній та підприємств, постійно розвиває свою мережеву інфраструктуру, додаються нові сервери та мережеве обладнання для створення додаткових інформаційних ресурсів.

З кожним днем з'являються більш нові, високопродуктивні технічні рішення, які дозволяють збільшити ефективність використання інфраструктури мереж. На підприємствах середнього та найбільшого масштабного застосування одне високотехнологічне рішення дозволяє значно зменшити затрати на вміст більш громіздкої та розрізненої архітектури, створеної на основі різних продуктів, що створюють проблеми з взаємодією між собою. Спільно з позитивними сторонами цієї системи можна виділити кілька негативних факторів:

- великі фінансові затрати на реорганізацію;
- складність діагностики аварій;
- необхідність додаткового навчання персоналу.

Маючи на увазі високу вартість таких рішень, необхідно створити умови для стабільної роботи обладнання та виключити можливість виходу його зі строю, що приводить до великих затрат на відновлення.

Виходячи з поставлених завдань доцільно використовувати системи моніторингу та управління мережевим трафіком користувачів. Існує безліч готових систем, як вільно розповсюджених, так і комерційних, але перед тим, як включити яку-небудь із них у виробничий процес, необхідно провести загальний аналіз та вивчення всіх ризиків, пов'язаних із застосуванням таких систем.

Для того, щоб включити систему в уже діючу інфраструктуру, необхідно вивчити кілька параметрів. Система повинна включати в себе модулі, що дозволяють здійснювати управління мережевим обладнанням, робочими станціями та серверами, як у ручному режимі, так і автоматично. Так що система повинна відповідати кільком вимогам:

- мінімальна кількість матеріальних заробітків на внесення;
- висока безпека;
- висока швидкість входження;
- підтримка сучасних мережевих протоколів та технологій;
- взаємодія з наявними програмними продуктами.

До появи систем моніторингу мережі адміністратору доводилося підключатися шляхом використання протоколів *telnet*, *http*, *snmp*, *ssh* і т.п. і користуватися вбудованими засобами моніторингу та діагностики. Крім цього перевантаження мережі і плаваючі несправності виявлялися тільки при виникненні серйозних проблем у користувачів, що не дозволяло складати плани з модернізації мережі.

Все це вело в першу чергу до постійного погіршення якості пропонованих послуг і підвищення навантаження на системних адміністраторів і службу технічної підтримки користувачів, що тягло за собою колосальні збитки.

Таким чином актуальність даної роботи обумовлена тим, що в зв'язку з поширенням персональних комп'ютерів і створенням на їх основі автоматизованих робочих місць зросло значення локальних обчислювальних мереж та необхідність їх ефективного адміністрування. У відповідності зі сформованою ситуацією, виникає

необхідність в впровадженні системи моніторингу та управління мережевим трафіком користувачів. Для здійснення централізованого управління і моніторингу мережевим трафік потрібно звернути увагу на те, щоб не було витрачено жодних коштів або були мінімізовані витрати, при цьому, щоб системи були розгорнуті на базі вільно розповсюджуваних продуктів, що б забезпечило максимально ефективне рішення, яке в результаті впровадження дозволило:

– підвищити відмовостійкість інформаційної інфраструктури за рахунок оперативного, аж до фактичного виникнення, реагування на виникаючі проблеми;

– підвищити ефективність праці співробітників інформаційних відділів за рахунок зниження трудовитрат на пошук проблем і їх причин, а також автоматизації ряду рутинних операцій;

– знизити експлуатаційні витрати за рахунок мінімізації збитку від виникаючих проблем, скорочення числа збоїв і відсутність ліцензійних платежів за право використання рішення;

– обґрунтовано виділяти кошти на модернізацію обладнання та програмного забезпечення з урахуванням наявної статистики відмов, а також завантаження ресурсів обладнання та результатів аналізу їх причин.

Таким чином наукова новизна даної роботи полягає в синтезі рішень та пропозицій щодо вдосконалення системи моніторингу та управління мережевим трафіком користувачів, та реалізації її модифікації, яка б автоматизувала та спростила процеси аналізу та адміністрування комп’ютерної мережі, та підвищила швидкість реагування на позаштатні ситуації.

Мета дипломної роботи – дослідити системи моніторингу та управління мережевим трафіком користувачів, та можливості з автоматизації процесів адміністрування мережі.

Завдання дипломної роботи: – провести аналіз систем моніторингу та управління мережевим трафіком користувачів, синтезувати рішення та пропозиції щодо вдосконалення системи з метою вирішення актуальних проблем, та впровадити модифікацію системи з метою вирішення зазначених проблем.

Об’єкт дослідження – система моніторингу та управління мережевим трафіком користувачів, комплексна система моніторингу мережевої інтенсивності в режимі реального часу, система оповіщення та передачі даних.

Практична значимість роботи – забезпечення безперебійної роботи мережі, модифікована система моніторингу та управління мережевим трафіком користувачів, яка дозволяє адміністратору мережі швидко реагувати на несправності та слабкі місця в організації мережі.

Рекомендації щодо використання результатів дипломної роботи – об’єкт та концепція системи моніторингу та управління мережевим трафіком можуть бути в подальшому використані для ефективного адміністрування мережі.

РОЗДІЛ 1
АНАЛІЗ МЕТОДІВ І ЗАСОБІВ МОНІТОРИНГУ ТА УПРАВЛІННЯ МЕРЕЖЕВИМ ТРАФІКОМ КОРИСТУВАЧІВ

1.1. Аналіз мережевої моделі *OSI* в контексті моніторингу та управління мережевим трафіком

Мережевий моніторинг стосується не лише моніторингу фізичної мережі та хост-пристроїв, таких як маршрутизатори, мости, концентратори та комп'ютери, але також займається моніторингом служб, які працюють на деяких з цих пристроїв. Ці послуги забезпечують зберігання даних, маніпулювання, презентацію, послугами зв'язку і вони працюють на мережевому рівні та вище. В тому числі потрібно пам’ятати про такі служби як: служба доменних імен (*DNS*), динамічний протокол керування хостом (*DHCP*), протокол простого передачі повідомлення (*SMTP*), протокол передачі гіпертексту (*HTTP*). Служби рівня додатків вони працюють на найважливіших серверах в комп'ютерній мережі, таких як веб-сервери, поштові сервери (рис. 1.1).

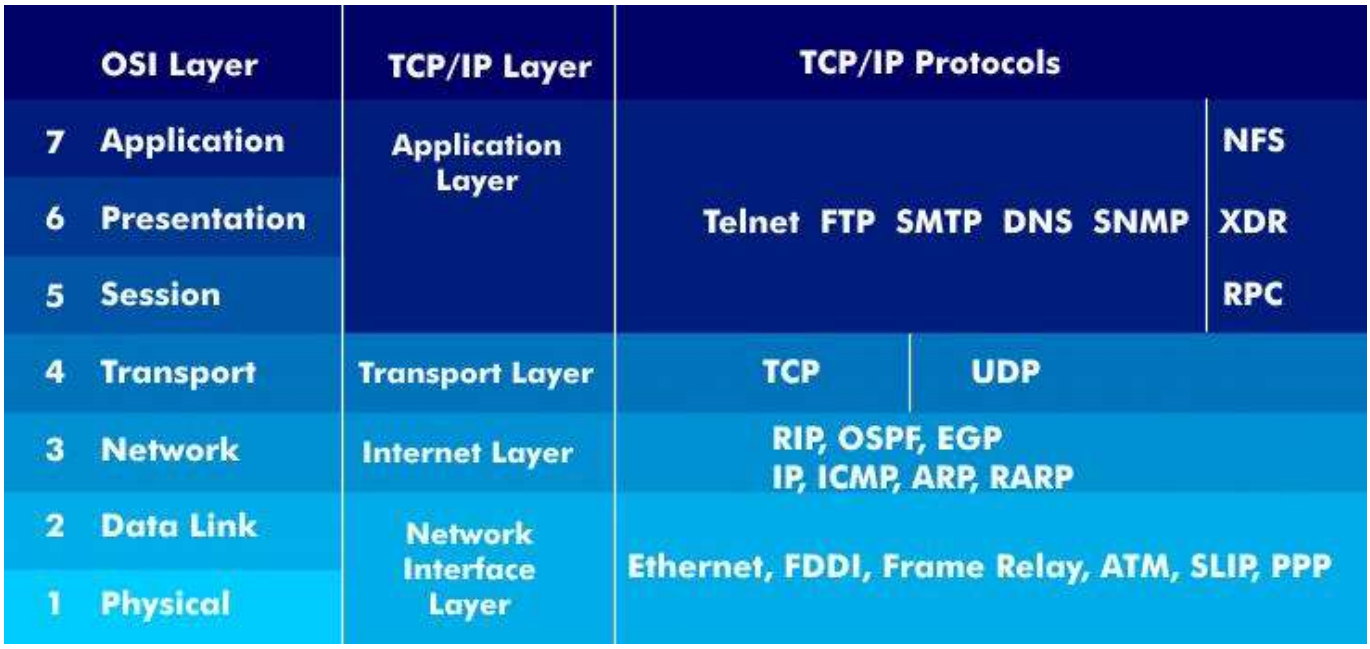

Рис. 1.1. Діаграма мережевої моделі *OSI* та її сервісів

Моніторинг доступності служби додатків передбачає сканування портів. Під час моніторингу мережевих пристроїв та хостів використовують концепції протоколів *SNMP* та *ICMP* відповідно.

Здебільшого рішення для моніторингу та діагностики продуктивності мережі розвинулися з більш традиційного і менш складного програмного забезпечення для моніторингу мережі. Ці інструменти моніторингу для отримання інформації про «здоров'я» мережі зазвичай використовують утиліту *Ping*, що працює на базі повідомлень протоколу *ICMP* (*Internet Control Message Protocol*, протокол міжмережевих керуючих повідомлень), що входить в стек протоколів *TCP* / *IP*, а також можливості щодо забезпечення синхронізації та проведення опитувань з центру моніторингу (комбінація *polling* / *traps*) на основі протоколу *SNMP* (*Simple Network Management Protocol*, простий протокол мережевого управління). Більш сучасні реалізації включають в себе можливості моніторингу, а також візуального представлення базового та інтелектуального аналізу стану всієї мережі аж до самих додатків. Більшість сучасних інструментів для моніторингу продуктивності мережі дозволяють виконувати п'ять наступних функціональних можливостей:

- моніторинг мережі і додатків;
- виявлення проблем з віртуалізацією і операційними системами;
- аналіз мережевих проблем;
- аналіз захоплених даних додатків і потоків;
- пошук кореневої причини інциденту або проблеми [1].

Зростаюча кількість організацій суттєво залежить від їхніх мереж та здатності приймати рішення на основі статусу їхніх мереж. Це вимагає від них засобів для управління та моніторингу таких мереж. Переривання обслуговування в мережі може коштувати сотні тисяч доларів для підприємства, яке є власником мережі. Те саме стосується супутникових мереж, які передають голос, дані та телевізійний вміст. Коли розраховується кожна секунда часу (3 мільйони доларів за 30 секунд у *Super Bowl*), система повинна постійно контролюватися.

Міжнародна організація зі стандартів (*OSI*) створила модель управління мережею. Модель складається з 5 функціональних областей:

- управління несправностями;
- управління конфігурацією;
- управління бухгалтерським обліком;
- управління продуктивністю;
- управління безпекою.

Несправність - ця функція є першою, що спадає на думку при управлінні мережею. Завдання полягає у виявленні, реєстрації та попередженні будь-яких проблем, які можуть вплинути на мережу та, зрештою, на послугу. Належна НМС повинна аналізувати тенденції, а за допомогою статистичних інструментів повинна мати можливість прогнозувати проблеми.

Коли виникає проблема з компонентом, підключеним до мережі, оператору над мережею надсилається повідомлення, часто використовуючи такий протокол, як *SNMP*. Таке повідомлення повинно викликати або ручну, або автоматичну дію. Наприклад, повідомлення з низьким рівнем *C* / *N* на віддаленому сайті супутникової мережі може спричинити збільшення вихідної потужності.

Журнали несправностей слід додатково аналізувати за допомогою статистичних інструментів, щоб визначити рівень обслуговування мережі або підмережі. Також корисно визначати продуктивність певних компонентів у мережі.

Конфігурація - метою функції конфігурації є відстеження та збереження версій апаратного та програмного забезпечення в мережі. Також відповідає за забезпечення ланцюгів у некомутованій мережі. Конфігурація різних компонентів може виконуватися локально або віддалено, але в кінцевому підсумку повинна відстежуватися та зберігатися в НМС.

Бухгалтерський облік - основною метою цієї функції є виставлення рахунків за пропоновані послуги, незалежно від того, є вони внутрішніми або зовнішніми клієнтами в організації. Він також збирає статистичні дані щодо використання мережевих ресурсів, тобто потужності, пропускної здатності, схем та інших ресурсів.

Продуктивність - це дуже важлива частина мережі, оскільки її метою є максимізація ресурсів у мережі, щоб вона працювала більш ефективно. Нові держави-члени повинні контролювати, оцінювати та коригувати наявні ресурси, щоб досягти

рівних домовленостей. Клієнти хочуть мінімізувати витрати, мінімізувати ручне втручання, підвищити надійність та ефективність.

Безпека - має справу з контролем доступу до ресурсів та попередженням, якщо доступ порушено. Деякі нові держави-члени можуть навіть зателефонувати органам влади, коли виникає проблема або коли виявляється вторгнення, навіть фізичне втручання [2].

1.2. Аналіз протоколів та стандартів моніторингу мережевого трафіку

З метою детального аналізу моніторингу та управління мережевим трафіком, розглянемо наявні протоколи та стандарти моніторингу.

SNMP (*Simple Network Management Protocol*) - це стандартизований протокол управління мережею. Він широко використовується для функцій моніторингу мережі, таких як збору помилок та статистика користувачів. У січні 1996 року він був оновлений до версії 2 і називається *SNMPv*2. *SNMPv*2 створюється в об'єктно-орієнтованому світі. Таким чином, *SNMPv*2 обговорюється з точки зору сутностей. Сутності *SNMPv*2 мають дві ролі: агент або менеджер. Агент - це процес, який використовує інструменти моніторингу мережі. Менеджер - це процес, який отримує інформацію, зібрану агентом. Це агент, коли він виконує операції управління у відповідь на сповіщення або коли надсилає сповіщення про пастку. Це менеджер, коли він ініціює сповіщення про управління або коли виконує запити на управління. Сутність *SNMPv*2 може діяти як в одній ролі, так і в обох ролях. *SNMPv*2 пропонує три типи методів доступу. Вони являють собою запит менеджера до відповіді менеджера, запит менеджера до відповіді агента або небажану відповідь агента. У першому методі два суб'єкти обох виступають менеджерами. Один суб’єкт надсилає запит іншому. Потім генерується відповідь. При другому методі одна сутність є менеджером, а інша - агентом. Менеджер видає запит, а агент діє відповідно до нього.

У третьому методі потрібні дві сутності. Один - агент, а другий - менеджер. Агент надсилає повідомлення менеджеру, не отримуючи жодного запиту. У цьому випадку менеджер не генерує жодної додаткової відповіді. Таким чином для

протоколу *SNMP* притаманні три ключові компоненти: керовані пристрої (*Managed Devices*), агенти (*Agents*) та системи управління мережею (*Network Management Systems - NMSs*), як це видно з рис. 1.2.

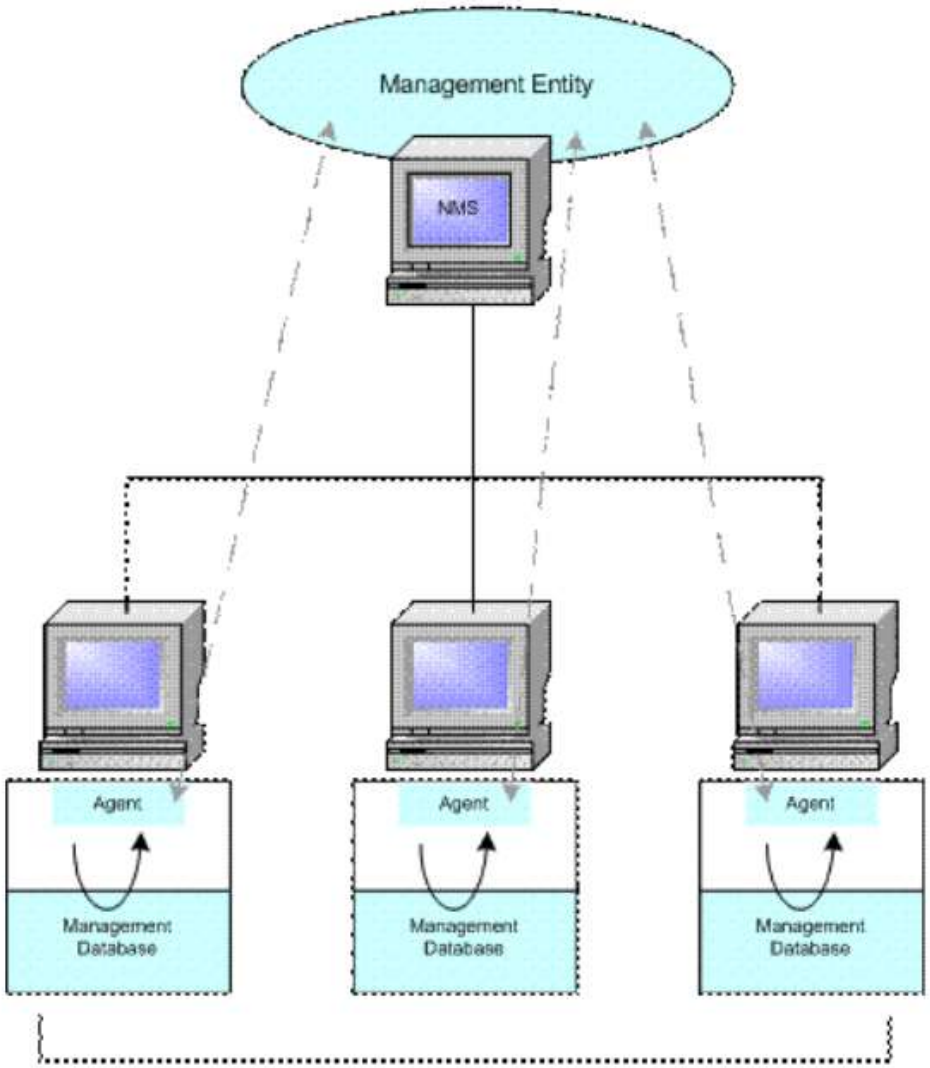

Рис. 1.2. Компоненти *SNMP*

Керовані пристрої включають в себе *SNMP*-агента і можуть складатися з маршрутизаторів, перемикачів, комутаторів, концентраторів, персональних комп'ютерів, принтерів і інших елементів, подібних до цих. Вони несуть відповідальність за збір інформації та роблять її доступною для системи управління мережею (*NMS*).

SNMP дозволяє збирати різні типи даних від мережевих пристроїв і серверів, що підтримують протокол. Для мережевих пристроїв це, звичайно, означає моніторинг конкретних станів інтерфейсу пристроїв і швидкості передачі даних. За

допомогою цього протоколу також можна стежити за станом апаратних засобів, включаючи блоки живлення, вентилятори, використання пам'яті і т. д.

Системи управління мережею (*NMS*) виконують додатки, які займаються моніторингом і контролем пристроїв управління. Ресурси процесора і пам'яті, які необхідні для управління мережею, надаються *NMS*. Для будь-якої керованої мережі повинна бути створена хоча б одна система управління. *SNMP* може діяти виключно як *NMS*, або агент, або може виконувати свої обов'язки або ін.

Агенти включають в себе програмне забезпечення, яке володіє інформацією з управління, і переводять цю інформацію в форму, сумісну з *SNMP*. Вони закриті для пристрою управління.

Деякі інструменти для моніторингу продуктивності мережі також здатні отримувати і відправляти різні повідомлення *Syslog* (системного журналу). Протокол *Syslog* є загальним стандартом для лог-повідомлень (інформації про різні події з прив'язкою до часу) всіх пристроїв мережевої інфраструктури. Ці повідомлення надсилаються централізованого інструментарію моніторингу мережі, який забезпечує їх зберігання, проведення аналізу, а також використовує для повідомлення інженерів зі служби технічної підтримки в разі порушення нормальної роботи системи.

Існує 4 основних команди, що використовуються *SNMP NMS* для моніторингу і контролю керованих пристроїв: читання, запис, переривання і операції перетину. Операція читання розглядає змінні, які зберігаються керованими пристроями. Команда записи змінює значення змінних, які зберігаються керованими пристроями. Операції перетину володіють інформацією про те, які змінні керованих пристроїв підтримують, і збирають інформацію з підтримуваних таблиць змінних. Операція переривання використовується керованими пристроями для того, щоб повідомити *NMS* про настання певних подій.

Як говорилося раніше, *SNMP* - протокол рівня додатків, який використовує пасивні сенсори, щоб допомогти адміністратору простежити за мережевим трафіком і продуктивністю мережі. Хоча, *SNMP* може бути корисним інструментом для мережевого адміністратора, він створює можливість для загрози безпеки, тому що він позбавлений можливості аутентифікації. Він відрізняється від віддаленого

моніторингу (*RMON*) тим, що *RMON* працює на мережевому рівні і нижче, а не на прикладному.

Віддалений моніторинг мережі (*RMON*) - це стандарт для контролю за Інтернет-трафіком. Це стандарт, який передбачається впроваджувати постачальниками пристроїв в Інтернеті, щоб мережа, що використовує сумісні з *RMON* пристрої, могла контролюватися за допомогою програмного забезпечення, сумісного з *RMON*. Загальна мета *RMON* - дозволити будувати пристрої моніторингу мережі, сумісні з *RMON*. Ці пристрої зазвичай називають моніторами або зондами, які вимірюють конкретні аспекти мережі, не заважаючи нормальній роботі. Ці пристрої, як правило, є окремими пристроями і розташовані у віддаленій частині мережі або навіть через межі мережі. Стандарт *RMON* дозволяє цим пристроям обмінюватися даними через мережу, яку вони контролюють. Зазвичай *RMON* визначається таким чином, щоб його можна було реалізувати в загальній мережі. Але деякі специфікації створені для моніторингу мереж *Ethernet*, оскільки це одна з найпопулярніших мереж, що використовуються в Інтернеті.

Хоча існує 3 ключові компоненти моніторингової середовища *RMON*, тут наводяться тільки два з них, як це видно на рис. 1.3. *RMON* є розширенням *SNMPv*2 і широко використовується для моніторингу мережі. *RMON* розширено завдяки створенню нового *MIB*, призначеного для збору інформації про мережу. Цей конкретний *MIB* зазвичай називають *RMON-MIB*. Усередині *RMON-MIB* визначаються об'єкти моніторингу мережі, і об'єкти поділяються на десять груп. Коли реалізовано один об’єкт у групі, усі об’єкти всередині однієї групи також повинні бути реалізовані.

Пристрої, сумісні з *RMON*, контролюються шляхом модифікації - *RMON-MIB*. Крім того, *RMON-MIB* містить статистичні дані, зібрані пристроєм. Об'єкти в *RMON-MIB* можуть бути представлені у вигляді таблиць. Людині легко зрозуміти та маніпулювати цією інформацією. Ці таблиці поділяються на контрольні таблиці та таблиці даних. Контрольні таблиці мають дозвіл на читання-запис. Таблиці керування оновлюються для встановлення нових команд для пристрою, сумісного з *RMON*. Таблиці даних мають дозвіл на читання. Ці таблиці містять статистичні дані.

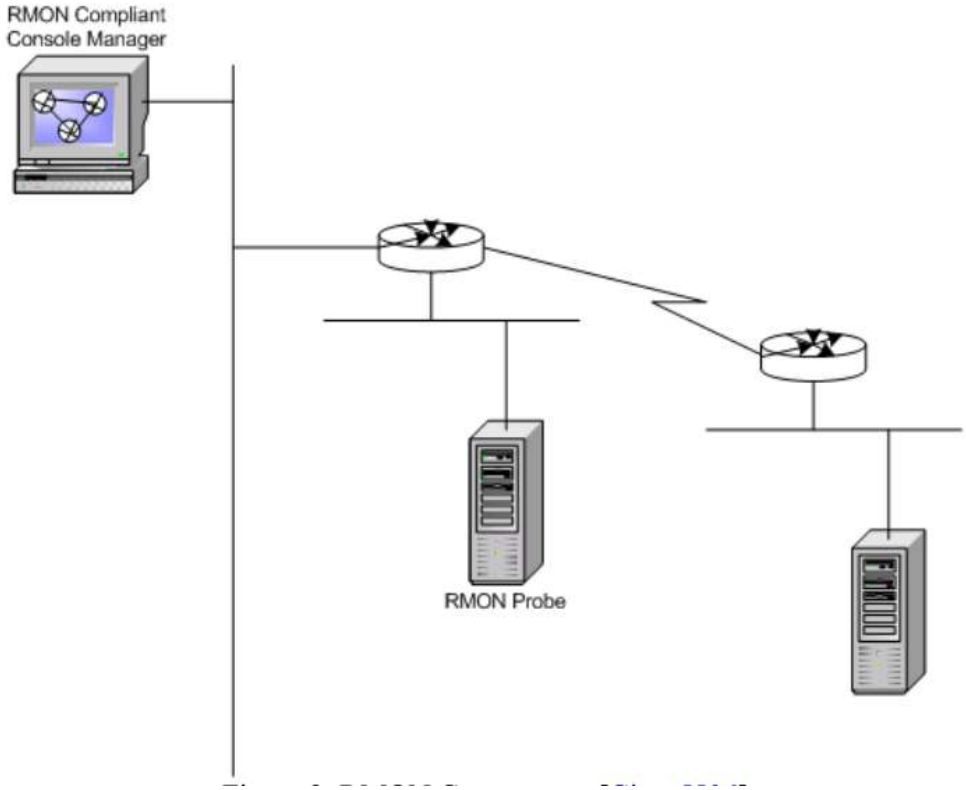

Рис. 1.3. Компоненти *RMON*

Два компонента *RMON* це датчик, також відомий як агент або монітор, і клієнт, також відомий як керуюча станція (станція управління). На відміну від *SNMP* датчик або агент *RMON* збирає і зберігає мережеву інформацію. Датчик - це вбудоване в мережеве пристрій (наприклад маршрутизатор або перемикач) програмне забезпечення. Датчик може запускатися також і на персональному комп'ютері. Датчик повинен поміщатися для кожного різного сегмента локальної або глобальної мережі, так як вони здатні бачити трафік, який проходить тільки через їхні канали, але вони не знають про трафік за їх межами. Клієнт - це зазвичай керуюча станція, яка пов'язана з датчиком, що використовують *SNMP* для отримання і корекції *RMON*-даних.

RMON використовує 9 різних груп моніторингу для отримання інформації про мережу:

– *statistics* (статистика виміряна датчиком для кожного інтерфейсу моніторингу для даного пристрою);

– *history* (облік періодичних статистичних вибірок з мережі і зберігання їх для пошуку);

– *alarm* (періодично бере статистичні зразки і порівнює їх з набором порогових значень для генерації події);

– *host* (містить статистичні дані, пов'язані з кожним хостом, виявленим в мережі);

– *hostTopN* (готує таблиці, які описують вершину хостів);

– *filters* (включає фільтрацію пакетів, грунтуючись на фільтровому рівнянні для захоплення подій);

– *packet capture* (захоплення пакетів після їх проходження через канал);

– *events* (контроль генерації і реєстрація подій від пристрою);

– *token ring* (підтримка кільцевих лексем).

*RMON*2 - це розширення *RMON*, яке фокусується на вищих рівнях трафіку над середнім рівнем контролю доступу (*MAC*). *RMON*2 робить акцент на *IP*-трафіку та на рівні додатків. *RMON*2 дозволяє програмам управління мережею контролювати пакети на всіх мережевих рівнях. Це відмінність від *RMON*, яка дозволяє контролювати мережу лише на рівні *MAC* або нижче. *RMON*2 призначений для використання програмами моніторингу мережі. Він не призначений для використання людиною. Кожен відстежуваний об'єкт повинен мати ім'я, синтаксис, рівень доступу та стан реалізації. Назва використовується для ідентифікації об'єкта, що контролюється. Ім'я має тип об’єкта та екземпляр об’єкта. Зазвичай ім’я - це текстовий рядок, який людина може прочитати. Синтаксис - це структура, визначена за допомогою позначення *ASN*.1. Ця абстрактна структура допомагає людині зрозуміти об'єкт, що контролюється. Рівень доступу означає, чи можна відстежуваний об'єкт читати, писати або обидва. Статус реалізації - це статус фактичного об'єкта. Існує чотири можливих значення: обов’язкове, необов’язкове, застаріле або застаріле. Об'єкти *RMON*2 поділяються на наступні 10 груп: каталог протоколів, розподіл протоколів, відображення адрес, хост мережевого рівня, матриця мережевого рівня, хост прикладного рівня, матриця рівня додатків, історія користувачів, *probeConfig*, *rmonConformance*. Ці групи є доповненням до існуючих груп у *RMON*. До *RMON* внесено 3 вдосконалення, щоб *RMON*2 і *RMON* добре працювали разом.

*RMON*2 може контролювати всі мережеві рівні. Це не обмежується *MAC* або мережевим рівнем. Він може розуміти та аналізувати пакети з набору типів протоколів. Каталог протоколів відстежує набір пакетів протоколів, що контролюються.

Традиційно *RMON*2 використовується для моніторингу мереж на основі кадрів, таких як *Ethernet*. По мірі того, як комутовані мережі, такі як банкомат та комутована локальна мережа, розширюються. *RMON*2 розширюється для моніторингу комутованих мереж. Це розширення називається *SMON*, моніторинг комутованої мережі. Ця назва ще не стандартизована.

У моніторингу комутованих мереж є кілька проблем, які відрізняються від моніторингу мереж на основі кадру. По-перше, дані в комутованих мережах орієнтовані на з'єднання, і один монітор не може захоплювати дані, слухаючи трансляції, як у мережах на основі кадру. По-друге, наскрізний моніторинг в комутованій мережі вимагає багато ресурсів. Повинні бути деякі способи агрегування даних, визначених програмами управління. По-третє, віртуальні комутовані мережі також повинні розглядатися, такі як *VLAN*. По-четверте, пріоритетність пакетів існує у взаємозміненій мережі. По-п'яте, *SMON* фокусується на моніторингу пакетів у високому рівні мережі, а не в клітинках нижнього шару. *SMON* бачить три різні типи джерел даних: джерело даних *RMON*, джерело даних *VLAN* та фізичне джерело даних. Джерелом даних *RMON* визначено сумісне з *RMON*, джерелом даних *VLAN* визначено включення віртуального джерела даних, створеного *VLAN*. Усі інші джерела даних групуються у фізичні джерела даних. У *SMON* також додана нова функція копіювання портів, яка дозволяє копіювати трафік з одного комутованого порту на інший порт. Це дозволяє контролювати рух на різних портах.

ATM-RMON - це стандарт дистанційного моніторингу, створений для мереж банкоматів. Мережа банкоматів є найновішою технологією, і її функція, орієнтована на з'єднання, вимагає розробки іншого набору стандартів. *ATM-RMON* розроблений форумом *ATM* і затверджений як стандарт у середині 1997 року. Це еквівалент *RMON* та *RMON*2, розроблених *IETF*. *ATM-RMON* розроблений на базі *RMON* з кількох причин. По-перше, він сумісний з *RMON*, так що існуючі додатки *RMON* можуть

працювати на банкоматі та зменшує витрати на розробку нових програм моніторингу. По-друге, багато протоколів, які в даний час контролюються *RMON*, емулюються через *ATM*, і *RMON* вже використовується для моніторингу цих даних. По-третє, *RMON*2 додано для оновлення *RMON*, і банкомат повинен передбачати проблеми сумісності з *RMON*2.

Незважаючи на те, що *ATM-RMON* базується на *RMON*, він включив багато нових функцій у *RMON*2, щоб залишатися сумісними в майбутньому. Наприклад, він включив нові функції *TopN*, такі як "час останнього створення"; замість розміру таблиці використовуються лічильники; і *ProtocolDirectory* вибирає, який протокол також слід відстежувати. *ATM-RMON* додав базову статистику, яка не існує в *RMON,* для моніторингу специфічних статистичних даних *ATM*, таких як кількість відправлених клітинок, кількість отриманих клітинок, кількість успішних налаштувань викликів, кількість спроб встановлення викликів та загальний час з'єднання. *ATM-RMON* щойно стандартизований. У цій версії ще багато недоліків. По-перше, мережа банкоматів призначена для передачі трафіку даних, голосу, відео та іншого трафіку. Але *ATM-RMON* контролює лише кадровий трафік. По-друге, в *ATM-RMON* не вказаний жоден фільтр комірок та група захоплення. Це дизайнерське рішення, щоб пришвидшити стандартизацію *ATM-RMON*. Не існує стандарту для моніторингу клітин. Таким чином, рекомендація щодо моніторингу була запропонована постачальникам, і вони повинні вибрати, який шлях застосовувати. Це полегшить майбутню стандартизацію. Є два варіанти, які можуть відрізнятися. Пристрій *RMON* може бути як внутрішнім, так і зовнішнім. Крім того, він може дозволити функції копіювання трафіку на інший порт або заборонити його [3-5].

1.3. Аналіз типів та методів моніторингу мережевого трафіку

З метою детального розуміння роботи функції моніторингу мережевого трафіку, розглянемо основні типи моніторингу та їх переваги та недоліки.

Моніторинг мережі може бути активним або пасивним. Пасивний моніторинг мережі зчитує дані з лінії, не впливаючи на трафік. Активний моніторинг мережі

додає можливість зміни даних на ліні. Пасивний моніторинг мережі існує у декількох формах.

Активний моніторинг звертається до мережі для збору вимірювань між принаймні двома кінцевими точками в мережі. Активні вимірювальні системи мають справу з такими показниками, як:

- доступність;
- маршрути;
- затримка пакетів;
- переупорядкування пакетів;
- втрата пакетів;
- вимірювання смуги пропускання (ємність, досяжна пропускна здатність).

Зазвичай використовувані інструменти, такі як *ping*, який вимірює затримку та втрату пакетів, і *traceroute*, який допомагає визначити топологію мережі, є прикладами основних активних засобів вимірювання. Вони обидва відправляють пакети (зонди) *ICMP* на призначений хост і чекають, щоб хост відповів відправнику. На рис. 1.4 можна побачити приклад команди *ping*, яка використовує активні вимірювання шляхом надсилання запит ехо-сигналу від вихідного хосту через мережу до вказаного пункту призначення. Потім адресат відправляє ехо-відповідь повернутися до джерела, з якого він отримав запит.

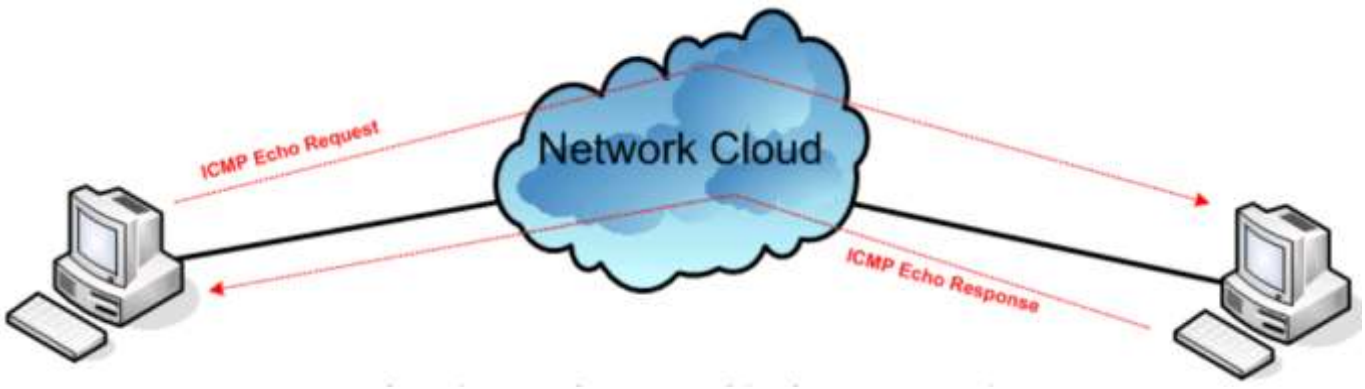

Рис. 1.4. *ICMP* команда *ping* (Активний моніторинг)

Людина може не тільки збирати наведені вище показники з активних вимірювань, але також може визначити топологію мережі. Інший типовим прикладом активного вимірювального інструменту є *iperf*. *Iperf* - це інструмент, що вимірює

продуктивність пропускної здатності *TCP* та *UDP*. Це повідомляє про пропускну здатність, затримку та втрати.

Проблема, яка існує при активному моніторингу, полягає в тому, що введення зондів у мережу може бути перешкодою для нормальної роботи трафіку в мережі. Часто активні зонди сприймаються інакше, ніж звичайний трафік, що і спричиняє не достовірність інформації.

В результаті наведеної вище інформації активний моніторинг дуже рідко реалізується як самостійний метод моніторингу, оскільки вводиться значна частина накладних витрат. З іншого боку, пасивний моніторинг не вносить багато, якщо взагалі не вносить накладні витрати на мережу.

Пасивний моніторинг на відміну від активного моніторингу не вводить трафік в мережу або не змінює вже наявний трафік у мережі. Також на відміну від активного моніторингу, пасивний моніторинг збирає інформацію лише про одну точку в мережі, яка вимірюється, а не між двома кінцевими точками, як активні заходи моніторингу. На рис 1.5 показано налаштування пасивної система моніторингу, де монітор розміщений на єдиному каналі між двома кінцевими точками і відстежує трафік, коли він проходить уздовж посилання.

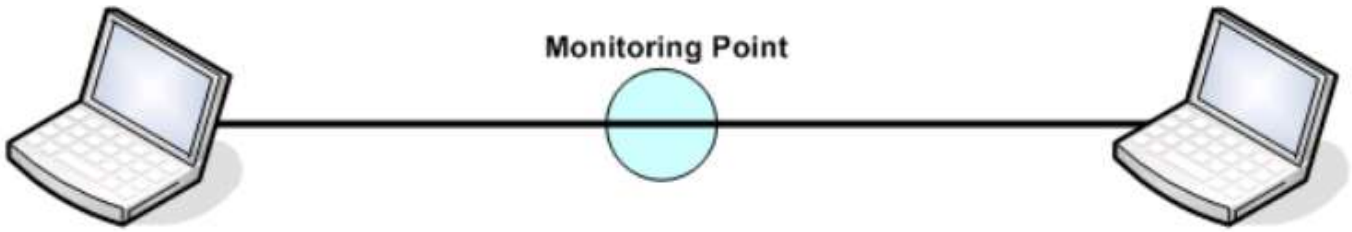

Рис. 1.5. Точка моніторингу між двома вузлами (Пасивний моніторинг)

Пасивний моніторинг має справу з такою інформацією, як:

- суміші трафіку та протоколів;
- точна швидкість передачі бітів або пакетів;
- час передачі пакетів та час прибуття.

Пасивного моніторингу можна досягти за допомогою будь-якої програми сніфінгу пакетів. Хоча пасивний моніторинг не має таких витрат, як активний моніторинг, він має свій власний набір недоліків. За допомогою пасивного

моніторингу вимірювання можна аналізувати лише в режимі офлайн. Це створює ще одну проблему з обробкою величезних наборів даних, які збираються.

Як бачимо, пасивний моніторинг може бути кращим за активний моніторинг, оскільки накладні дані не додаються в мережу, а час після обробки може зайняти велику кількість часу. Тому доцільно поєднання двох методів. Таким чином поєднання активного та пасивного моніторингу є краще, ніж користуватися тим чи іншим. Комбіновані методи використовують найкращі аспекти як пасивного, так і активного моніторингу мережі.

Простий моніторинг може бути простим для ручної оцінки, оскільки обсяг даних, що відстежуються та виробляються, невеликий. Моніторинг різного роду деталей про мережу та її трафік несе подібну перешкоду; інформація про несправності та зловмисників зібрана, але інформації стільки, що вона губиться в морі. Крім того, чим більше даних зафіксовано, тим більш технологічно вимогливим є збереження та обробка даних. Тому різні способи здійснення моніторингу мережі конкурують між собою, оскільки кожен має різні компроміси, орієнтовані на різні цілі, середовища та користувачів. На рисунку 1 представлена загальна архітектура моніторингу мережі. Процес моніторингу мережі складається з двох основних етапів, дублювання трафіку та аналізу трафіку.

Усі типи моніторингу мережевого трафіку мають одну спільну властивість, трафік з лінії дублюється, щоб копію можна було проаналізувати. Дублювання може відбуватися в одному з двох режимів: вбудований або дзеркальний. У дзеркальному режимі функція копіювання вже є вбудованою функцією маршрутизатора або комутатора.

Дзеркальне відображення портів - це функціональність, яка зазвичай доступна в мережевих комутаторах та маршрутизаторах, орієнтованих на підприємства. Трафік, що проходить через вибрані порти комутатора або маршрутизатора, дублюється на інший вибраний порт. Порт, що використовується для виведення продубльованого трафіку, зазвичай називається дзеркальним портом. На рис. 1.6 показано принцип дзеркального відображення портів. Обидва напрямки контрольованої лінії передаються в одному напрямку через дзеркальний порт. Є два

мінуси дзеркальних портів. По-перше, якщо сумарна пропускна здатність трафіку більша, ніж може передавати дзеркальний порт, дзеркальний порт стає перевантаженим і скидає пакети. Повнодуплексний трафік передається в одному напрямку через дзеркальний порт. Це до подвоєної пропускної здатності одного порту для двох портів, що обслуговуються комутатором, і навіть більше, якщо обслуговується більше двох портів. По-друге, у більшості комутаторів недостатньо обчислювальної потужності для обробки як перемикання, так і дзеркального відображення. Основна функція комутатора має пріоритет, і дзеркальне відображення може не працювати належним чином у періоди пікового трафіку.

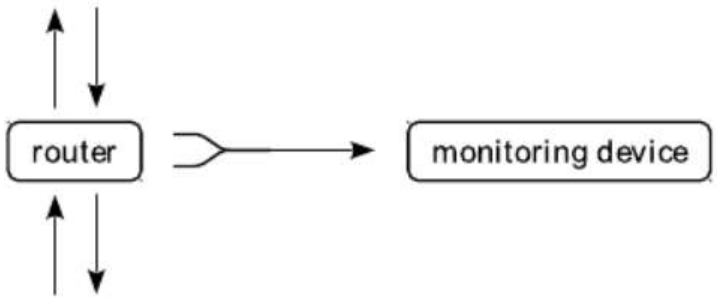

Рис. 1.6. Принцип дзеркального відображення портів

Порт тестового доступу (*TAP*) - це пристрій захоплення пакетів, який розміщений у вбудованому режимі, оскільки спостережувана лінія розділена. Пристрій *TAP* підключено між розділеними частинами лінії, і трафік дублюється. Одиничні *TAP* дублюють трафік до одного виходу, що складається з двох фізичних портів як для низхідної, так і для висхідної лінії повнодуплексного зв'язку. *TAP* регенерації продублюють трафік на кілька виходів. Агрегаційні *TAP* об'єднують обидва канали в один вихідний порт. Існує три типи *TAP*; мідні, волоконні та віртуальні. На рис. 1.7 показано дзеркальне відображення трафіку за допомогою тестового порту доступу. Обидва напрямки моніторингової лінії передаються окремо. Пасивні мідні TAP підключаються безпосередньо до лінії. Оскільки пасивні TAP не живляться, відключення електроенергії не може спричинити несправність на лінії. Той факт, що не може відбутися відключення живлення *TAP*, є перевагою.

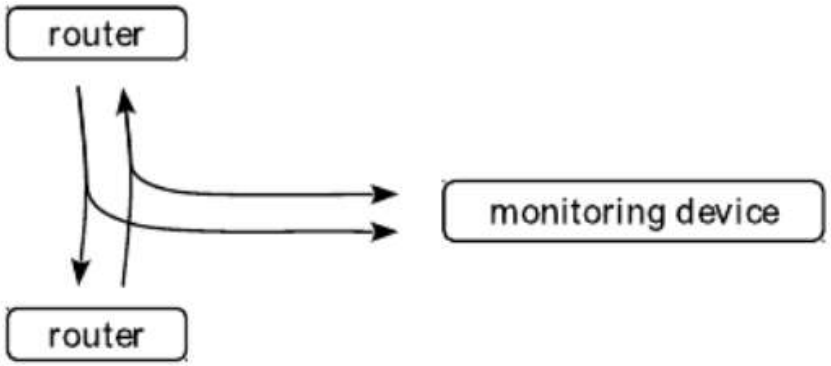

Рис. 1.7. Дзеркальне відображення порту за допомогою *TAP*

Недолік полягає в тому, що сигнал на лінії послаблюється *TAP*. Регенераційні оптичні *TAP* перенаправляють дуже малу частку вихідного сигналу на дзеркальний вихід і посилюють його на повну силу. Збої живлення просто відключають дзеркальне виведення і не викликають несправності на лінії [3-6].

1.4. Призначення систем моніторингу та управління мережевим трафіком

Термін “моніторинг мережі” описує систему, яка постійно відстежує всю топологію мережі щодо перешкод, уповільнення роботи або виходу з ладу компонентів і повідомляє відповідальну особу мережі електронною поштою, смс-повідомленням чи іншими сигналами тривоги у разі виникнення будь-якої проблеми. Моніторинг мережі, як правило, пов'язаний з функціями, що беруть участь в управлінні мережею. Управління мережею потрібно для забезпечення роботи мережі.

Ідеальна система моніторингу мережі повинна мати такі властивості:

– повинна автоматично і постійно контролювати мережі;

– повинна швидко інформувати адміністратора про проблему, як тільки вона виникає;

– повинна бути достатньо розумною, щоб вказати на проблему та її точне розташування в топології мережі. Також повинна бути можливість виявити наслідки проблеми для решти мережі та служб, які стануть недоступними;

– повинна вести облік змін у мережі, що полегшує пошук причини проблеми через зміну конфігурації;

– повинна забезпечувати віддалену автентифікацію та авторизацію для адміністратора для доступу до системи моніторингу звідусіль.

Відповідно до рекомендацій *ISO* можна виділити такі функцій засобів управління мережею:

– управління конфігурацією мережі (полягає в конфігурації компонентів мережі, включаючи їх місце розташування, мережні адреси і ідентифікатори, управління параметрами мережевих операційних систем, підтримку схеми мережі: також ці функції використовуються для іменування об'єктів);

– обробка помилок (це виявлення, визначення і усунення наслідків збоїв і відмов у роботі мережі);

– аналіз продуктивності (допомагає на основі накопиченої статистичної інформації оцінювати час відповіді системи і величину трафіка, а також планувати розвиток мережі);

– управління безпекою (містить у собі контроль доступу і збереження цілісності даних. У функції входить процедура аутентификации, перевірки привілеїв, підтримка ключів шифрування, управління повноваженнями);

– облік роботи мережі (включає реєстрацію і управління використовуваними в ресурсах і пристроями. Ця функція оперує такими поняттями як час використання й плата за ресурси).

Добре продумані системи конфігурації важливі для систем моніторингу мережі. Мережі часто змінюються, що може призвести до того, що модель мережі моніторингу мережі застаріла, що призведе до «чорних дір», які не контролюються. Якщо систему моніторингу мережі легко запустити, а її конфігурацію легко підтримувати, модель мережі повинна бути більш точною.

Опитування послуг – це тип мережевого тестування, коли система моніторингу мережі регулярно перевіряє, чи пристрій чи послуга доступні та працюють у межах нормальних параметрів. Це дозволяє відповісти на запитання про доступність, наприклад, чи доступний сервер, що розміщує наш веб-сайт у нашій мережі, чи ні. Більш конкретні тести опитування послуг можуть збирати додаткову інформацію про стан мережі. Одним із прикладів такої додаткової інформації є перевірка програмного

забезпечення веб-сервера; що він працює коректно, реагує та забезпечує правильний вміст без помилок.

Графіки часових рядів, що відображають дані про ефективність, отримані системою моніторингу мережі, можуть бути корисними для виявлення тенденцій та аномалій. Там, де дається велика кількість значень даних, такі графіки часто використовуються для ідентифікації змін стану мережі. Графіки повинні бути адаптовані до мережі, яка контролюється, і деякі дані корисніші для графіків, ніж інші. Графік часових рядів процесора корисний, оскільки він вказує, чи не перевантажені пристрої, і наскільки вони близькі до роботи на повну потужність. Графічне використання смуги пропускання на мережевих інтерфейсах показує, наскільки мережа також близька до роботи на повну потужність. Ця інформація корисна системним адміністраторам і може використовуватися для планування оновлення мережі та виявлення можливих вузьких місць.

Хороші чи погані зміни в мережі повинні якось дістатись до когось, особливо до відповідальних за моніторинг мережі. Системи моніторингу мережі досягають цієї функції сповіщення за допомогою систем подій. Системи подій можуть бути дуже простими системами, які перевіряють, чи є якесь значення в межах заданого порогу чи має певне значення. Наприклад, якщо хост стає недосяжним, подія може бути ініційована. Складні системи подій також можуть стежити за небажаними тенденціями або використовувати виявлення аномалій для виявлення подій, які можуть вплинути на мережу. Однак сама система сповіщення не дає повної картини, і може знадобитися певна експертиза, щоб визначити, що спричинило подію.

Інформаційна панель - це користувальницький інтерфейс, який забезпечує візуальне відображення інформації на одному екрані, таким чином, що всю цю інформацію можна відстежувати з одного погляду. Це корисно для систем моніторингу мережі, оскільки це дозволяє контролювати всю мережу з великого монітора з огляду на системного адміністратора і не вимагає активного пошуку на багатьох сторінках системи моніторингу мережі для ручного виявлення проблем. Багато інформаційних панелей містять графіки, які слід регулярно перевіряти для цієї мережі, наприклад графіки продуктивності. Він також включатиме зведені

статистичні дані, такі як поточна кількість несправностей або узагальнений показник стану мережі. Він повинен впорядкувати процес пошуку несправностей, забезпечивши легке виявлення першопричини події та вказуючи користувачеві в напрямку, де вони повинні розпочати подальші розслідування [7-11].

Висновки за розділом

В першому розділі проаналізовано мережеву модель *OSI* в контексті моніторингу та управління мережевим трафіком, в результаті аналізу детально розглянуто протоколи та стандарти моніторингу та їх переваги та недоліки між собою. З метою детального розуміння роботи функції моніторингу мережевого трафіку в даному розділі розглянуті загальні типи та методи моніторингу стану мережі. І в результаті детально розглянуто призначення та основні функції систем моніторингу та управління мережевим трафіком користувачів.

РОЗДІЛ 2
АНАЛІЗ ТА ВИБІР СИСТЕМИ МОНІТОРИНГУ ТА УПРАВЛІННЯ МЕРЕЖЕВИМ ТРАФІКОМ КОРИСТУВАЧІВ

2.1. Огляд та аналіз існуючих систем моніторингу та управління мережевим трафіком

Для того щоб забезпечити моніторинг та управління мережевим трафіком користувачів, проведемо огляд та аналіз існуючих систем.

Zabbix - це універсальний інструмент моніторингу, здатний відстежувати динаміку роботи серверів та мережевого обладнання, швидко реагувати на позаштатні ситуації і попереджати можливі проблеми з навантаженням. *Zabbix* може збирати статистику в зазначеної робочої середовищі і діяти в певних випадках заданим чином. Архітектура *Zabbix* зображена на рис. 2.1.

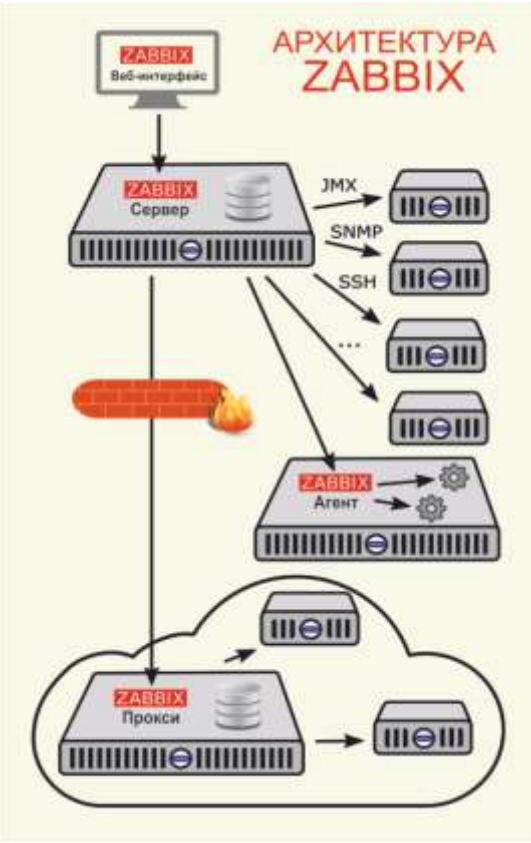

Рис. 2.1. Архітектура *Zabbix*

Основні можливості:

– розподілене моніторинг - до декількох тисяч вузлів. Конфігурація молодших вузлів повністю контролюється старшими вузлами, що перебувають на більш високому рівні ієрархії;

– сценарії на основі моніторингу;

– автоматичне виявлення;

– централізований моніторинг журналів;

– веб-інтерфейс для адміністрування і настройки (рис. 2.2);

– звітність і тенденції;

– *SLA*-моніторинг;

– підтримка високопродуктивних агентів (*zabbix-agent*) практично для всіх платформ;

– комплексна реакція на події;

– підтримка *SNMP v*1, 2, 3;

– підтримка *SNMP*-пасток;

– підтримка *IPMI*;

– підтримка моніторингу *JMX*-додатків;

– підтримка виконання запитів в різні бази даних без необхідності використання сценарної обв'язки;

– розширення за рахунок виконання зовнішніх скриптів;

– гнучка система шаблонів і груп;

– можливість створювати карти мереж;

– Інтеграція з зовнішніми системами за допомогою плагінів. Наприклад, *Zabbix* можна інтегрувати в *Grafana* для візуалізації даних, побудови графіків і дашборда.

Окремий блок можливостей пов'язаний з автоматичним виявленням: пристроїв за діапазоном *IP*-адрес, доступних на них сервісах, також реалізована *SNMP*-перевірка. Забезпечується автоматичний моніторинг виявлених пристроїв, автоматичне видалення відсутніх вузлів, розподіл по групам і шаблонами в

залежності від що повертається результату. Низькорівневе виявлення може бути використано для виявлення і для початку моніторингу файлових систем, мережевих інтерфейсів. Починаючи з *Zabbix* 2.0, підтримуються три вбудованих механізму низькорівневого виявлення:

– виявлення файлових систем;

– виявлення мережевих інтерфейсів;

– виявлення декількох *SNMP OID*.

Підтримувані платформи (сервер і агент): *AIX*, *FreeBSD*, *HP-UX*, *Linux*, *Mac OS*, *OpenBSD*, *SCO OpenServer*, *Solaris*, *Tru*64 / *OSF*; крім того, реалізовані агенти для *Novell Netware* і операційних систем сімейства *Windows* [12].

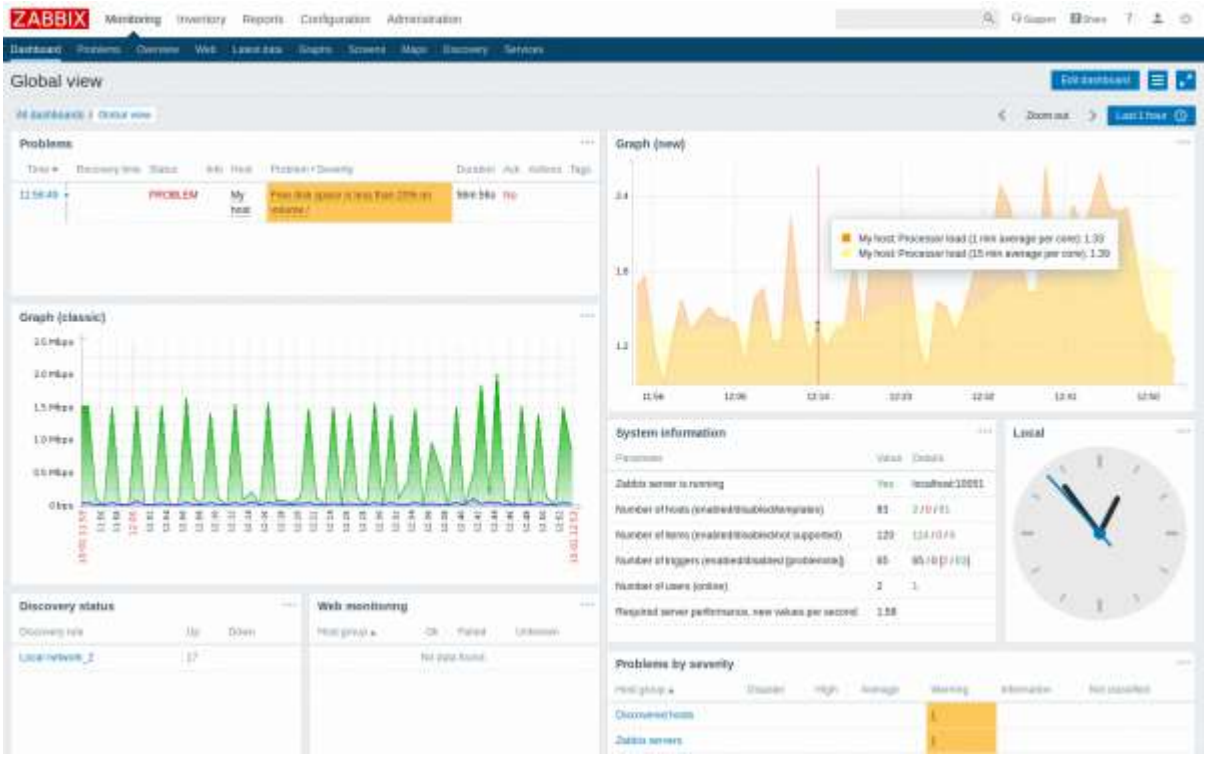

Рис. 2.2. Веб-інтерфейс *Zabbix*

Nagios - програма з відкритим кодом, призначена для моніторингу комп'ютерних систем і мереж: спостереження, контролю стану обчислювальних вузлів і служб, оповіщення адміністратора в тому випадку, якщо якісь із служб припиняють (або відновлюють) свою роботу (рис. 2.3).

Основні можливості:

– моніторинг мережевих служб (*SMTP*, *POP*3, *HTTP*, *NNTP*, *ICMP*, *SNMP*);

– моніторинг стану хостів (завантаження процесора, використання диска, системні логи) в більшості мережевих операційних систем;

– підтримка віддаленого моніторингу через шифровані тунелі *SSH* або *SSL*;

– проста архітектура модулів розширень (плагінів) дозволяє, використовуючи будь-яку мову програмування за вибором (*Shell, C ++, Perl, Python, PHP, C #* та інші), легко розробляти свої власні способи перевірки служб;

– паралельна перевірка служб;

– можливість визначати ієрархії хостів мережі за допомогою «батьківських» хостів, дозволяє виявляти і розрізняти хости, які вийшли з ладу, і ті, які недоступні;

– відправлення повідомлень в разі виникнення проблем зі службою або хостом (за допомогою пошти, пейджера, смс, або будь-яким іншим способом, визначеним користувачем через модуль системи);

– можливість визначати обробники подій, що відбулися зі службами або хостами для проактивного вирішення проблем;

– автоматична ротація лог-файлів;

– можливість організації спільної роботи декількох систем моніторингу з метою підвищення надійності і створення розподіленої системи моніторингу;

– включає в себе утиліту *nagiostats*, яка виводить загальне зведення по всім хостам, за якими ведеться моніторинг [13].

Zenoss – це система моніторингу *IT* інфраструктури поширювана під ліцензією *GPL*. Існує дві версії цього продукту - комерційний - *Zenoss Enterprise* і вільний - *Zenoss Core*, різниця яких полягає в наявності офіційної підтримки і додаткових модулів для *Enterprise* (рис. 2.4). Можливості системи:

– моніторинг мережевих пристроїв за допомогою *SNMP, SSH, WMI, JMX, Ping* / *ICMP, Syslog*;

– Моніторинг мережевих сервісів *HTTP, POP*3, *NNTP, SNMP, FTP*

– моніторинг системних ресурсів різних операційних систем (*Windows, Linux, FreeBSD, MacOS*);

– моніторинг продуктивності пристроїв;

– система оповіщення з налаштованим подіями, реакцією і виявленням взаємозв'язку;

– можливість розширення функціональності за рахунок плагінів *ZenPack* і плагінів системи моніторингу *Nagios* [14].

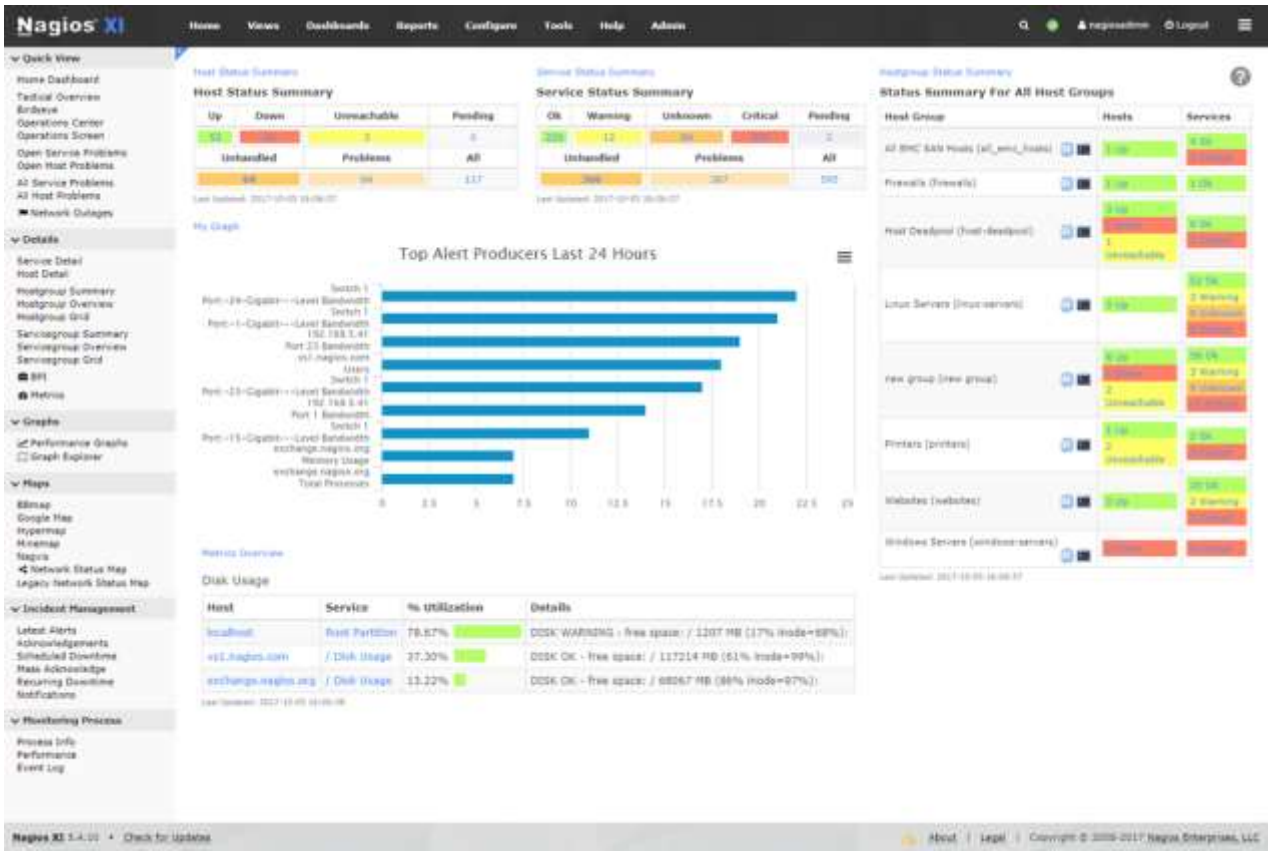

Рис. 2.3. Веб-інтерфейс *Nagios*

Icinga – комп'ютерна система з відкритим вихідним кодом, а також додаток для моніторингу мережі. Спочатку було створено як відгалуження від системи моніторингу *Nagios*. *Icinga* це спроба виправити недоліки в процесі розробки *Nagios*, додаючи нові можливості, такі як сучасний Веб 2.0 стиль користувальницького інтерфейсу, додаткові з'єднувачі для баз даних (*MySQL, Oracle Database*, *PostgreSQL*), *REST API* дозволяє адміністраторам додавати безліч розширень без внесення змін в ядро *Icinga*.

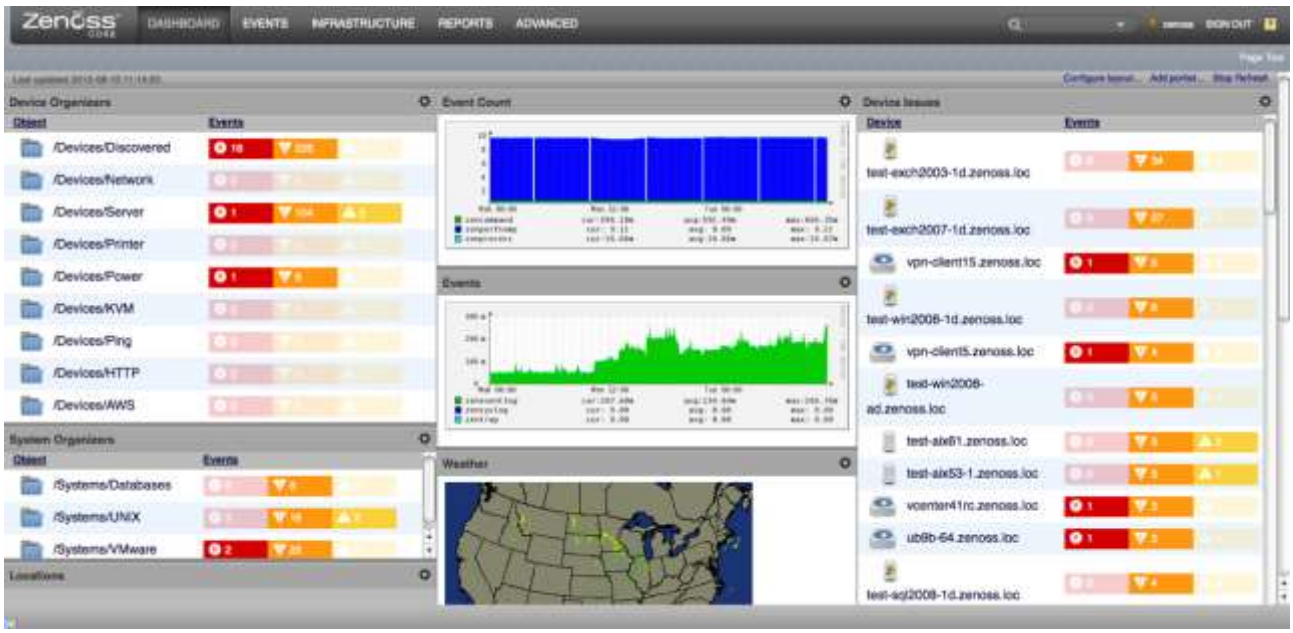

Рис. 2.4. Веб-інтерфейс *Nagios*

Так само розробники *Icinga* намагаються чіткіше задовольняти потреби громади і більш швидко інтегрувати виправлення. Оскільки *Icinga* це відгалуження *Nagios*, то *Icinga* пропонує такі ж функції, як і *Nagios*, з деякими доповненнями, такими як додатковий модулі звітності з поліпшеною точністю *SLA*, додаткові з'єднувачі для баз даних *Oracle* і *PostgreSQL* і розподілені системи обчислення для надлишкового моніторингу. Так само, для спрощення міграції між системами моніторингу, *Icinga* сумісна з плагінами *Nagios*. Огляд можливостей:

– моніторинг мережевих служб (*SMTP*, *POP*3, *HTTP*, *NNTP*, *Ping* і т. Д.);

– моніторинг ресурсів хоста (завантаження ЦПУ, використання дисків, використання оперативної пам'яті);

– моніторинг серверних компонентів (комутатори, маршрутизатори, сервери, датчики температури, вологості і т. д.);

– просте створення плагінів, що дозволяє користувачам розробляти власні типи перевірок служб;

– паралельна перевірка служб;

– створення ієрархії мережевих хостів, що дозволяє відрізняти неробочі хости від недоступних;

– можливість призначення обробників подій;

– можливість автоматичної відправки повідомлень по *E-Mail*, через систему миттєвого обміну повідомленнями, *SMS* і т. д;

– ескалація повідомлень.

Візуальне оформлення та звіти (рис. 2.5):

– можливість настройки через веб;

– інтерфейс *Icinga Web* 2 для відображення статусу служб і пристроїв;

– модуль звітів, заснований на *JasperReports* для двох призначених для користувача інтерфейсів;

– шаблони звітів;

– база звітів з різним рівнем доступу і автоматичним створенням звітів;

– різні доповнення для *SLA*;

– звіти про використання потужностей;

– графіки стану і продуктивності (через плагіни *PNP*4*Nagios*, *NagiosGrapher*, *InGraph*) [15].

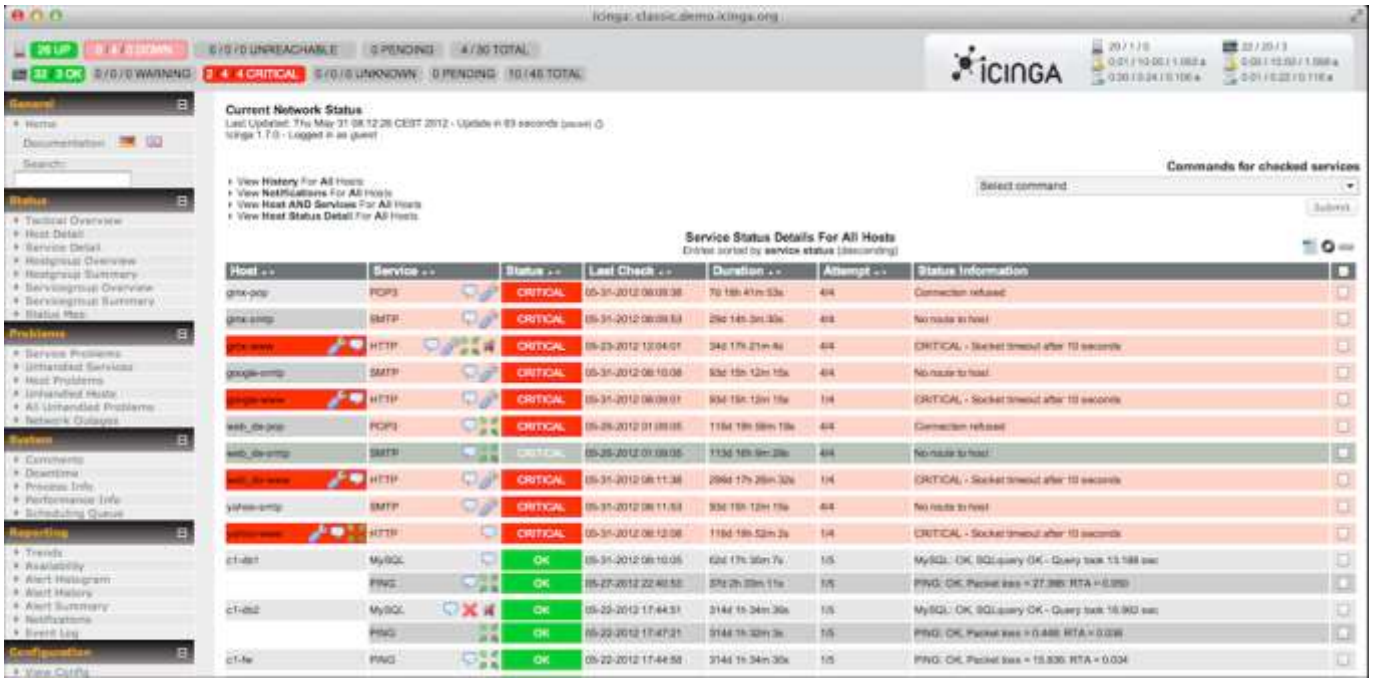

Рис. 2.5. Веб-інтерфейс *Icinga*

Cacti – це безкоштовна програма, що входить в *LAMP*-набір серверного програмного забезпечення, яке надає стандартизовану програмну платформу для побудови графіків на основі практично будь-яких статистичних даних. Якщо будь-

який пристрій або сервіс повертає числові дані, то вони, швидше за все, можуть бути інтегровані в *Cacti*. Існують шаблони для моніторингу широкого спектру обладнання - від *Linux*- і *Windows*-серверів до маршрутизаторів і комутаторів *Cisco,* - в основному все, що спілкується на *SNMP* (*Simple Network Management Protocol*, простий протокол мережевого управління). Існують також колекції шаблонів від сторонніх розробників, які ще більше розширюють і без того величезний список сумісних з *Cacti* апаратних засобів і програмного забезпечення (рис. 2.6).

Незважаючи на те, що стандартним методом збору даних *Cacti* є протокол *SNMP*, також для цього можуть бути використані сценарії на *Perl* або *PHP*. Фреймворк програмної системи вміло розділяє на дискретні екземпляри збір даних і їх графічне відображення, що дозволяє з легкістю повторно обробляти і реорганізовувати існуючі дані для різних візуальних уявлень. Крім того, ви можете вибрати певні часові рамки і окремі частини графіків просто клікнувши на них і перетягнувши.

Так, наприклад, можна швидко переглянути дані за кілька минулих років, щоб зрозуміти, чи є поточна поведінка мережевого обладнання або сервера аномальним, або подібні показники з'являються регулярно. А використовуючи *Network Weathermap, PHP*-плагін для *Cacti*, ви без надмірних зусиль зможете створювати карти вашої мережі в реальному часі, що показують завантаженість каналів зв'язку між мережевими пристроями, що реалізуються за допомогою графіків, які з'являються при наведенні покажчика миші на зображення мережевого каналу. Багато організацій, що використовують *Cacti*, виводять ці карти в цілодобовому режимі на 42-дюймові РК-монітори, встановлені на стіні, дозволяючи IT-фахівцям миттєво відстежувати інформацію про завантаженість мережі і стан каналу.

Таким чином, *Cacti* - це інструментарій з великими можливостями для графічного відображення та аналізу тенденцій продуктивності мережі, який можна використовувати для моніторингу практично будь-який контрольованої метрики, що подається у вигляді графіка. Дане рішення також підтримує практично безмежні можливості для настройки, що може зробити його занадто складним при певних застосуваннях.

NeDi - це безкоштовне програмне забезпечення, відносить до *LAMP*, яке регулярно переглядає *MAC*-адреси і таблиці *ARP* в комутаторах вашої мережі, каталогізуючи кожне виявлене пристрій в локальній базі даних. Даний проект не є настільки добре відомим, як деякі інші, але він може стати дуже зручним інструментом при роботі з корпоративними мережами, де пристрої постійно змінюються і переміщаються.

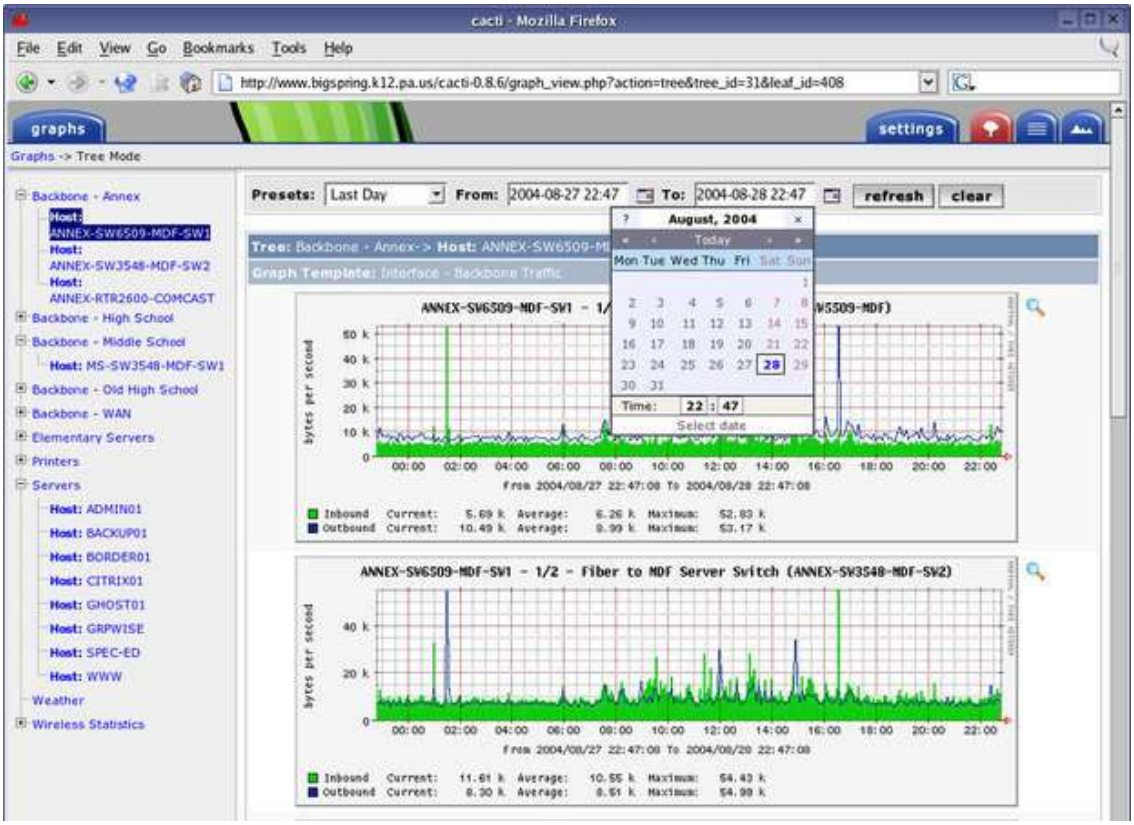

Рис. 2.6. Веб-інтерфейс *Cacti*

Через веб-інтерфейс *NeDi* задати пошук для визначення комутатора, порту комутатора, точки доступу або будь-якого іншого пристрою по *MAC*-адресу, *IP*-адресою або *DNS*-імені (рис. 2.7). *NeDi* збирає всю інформацію, яку тільки може, з кожного мережевого пристрою, з яким стикається, витягаючи з них серійні номери, версії прошивки і програмного забезпечення, поточні тимчасові параметри, конфігурації модулів і т. Д. Ви навіть можете використовувати *NeDi* для відзначення *MAC*- адрес пристроїв, які були втрачені або вкрадені. Якщо вони знову з'являться в мережі, *NeDi* повідомить вам про це.

Виявлення запускається процесом *cron* з заданими інтервалами. Конфігурація проста, з єдиним конфігураційним файлом, який дозволяє значно підвищити кількість налаштувань, в тому числі можливість пропускати пристрої на основі регулярних виразів або заданих меж мережі. *NeDi*, зазвичай, використовує протоколи *Cisco Discovery Protocol* або *Link Layer Discovery Protocol* для виявлення нових комутаторів і маршрутизаторів, а потім підключається до них для збору їхньою інформацією. Як тільки початкова конфігурація буде встановлена, виявлення пристроїв буде відбуватися досить швидко.

До певного рівня *NeDi* може інтегруватися з *Cacti*, тому існує можливість зв'язати виявлення пристроїв з відповідними графіками *Cacti*.

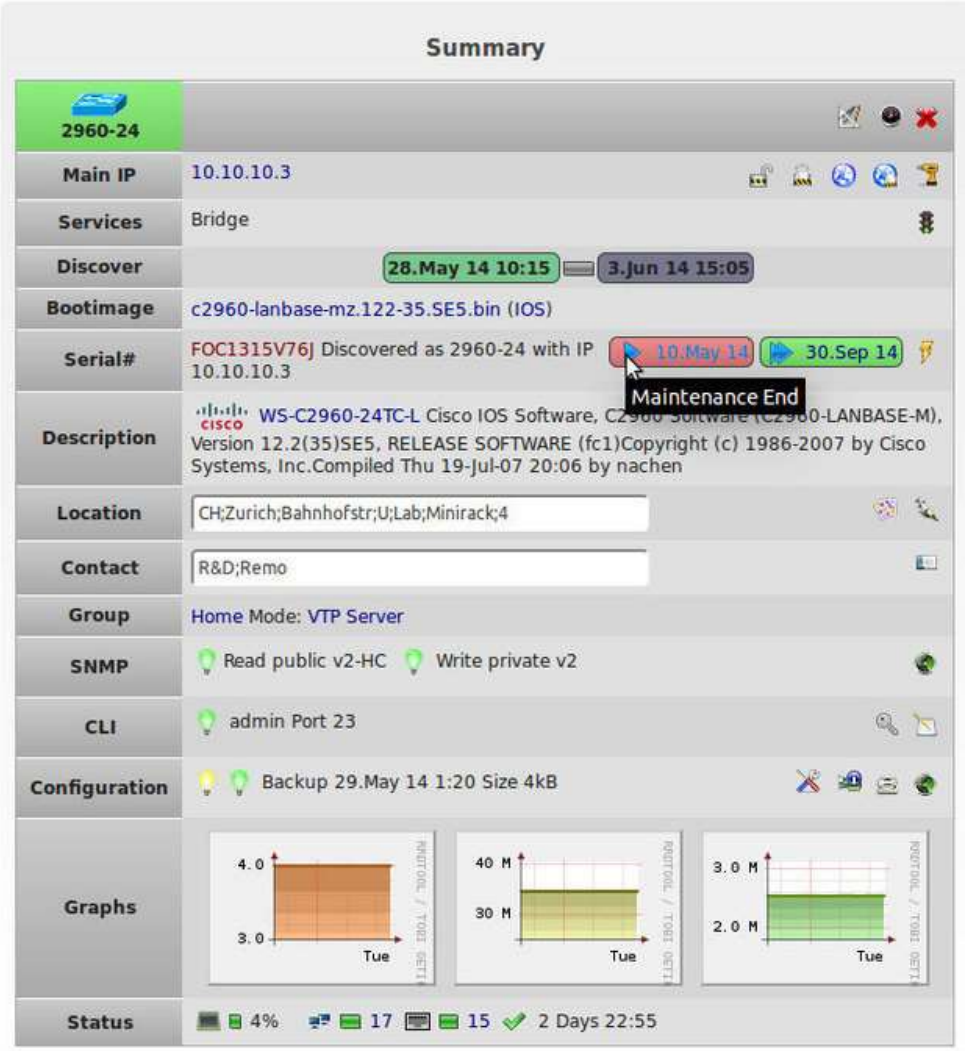

Рис. 2.7. Інтерфейс *NeDi*

Observium - це програма для моніторингу мережевого обладнання та серверів, яке має величезний список підтримуваних пристроїв, що використовують протокол *SNMP*. Як програмне забезпечення, що відноситься до *LAMP*, *Observium* відносно легко встановлюється і налаштовується, вимагаючи звичайних установок *Apache*, *PHP* і *MySQL*, створення бази даних, конфігурації *Apache* і тому подібного. Він встановлюється як власний сервер з виділеним *URL*-адресою (рис. 2.8).

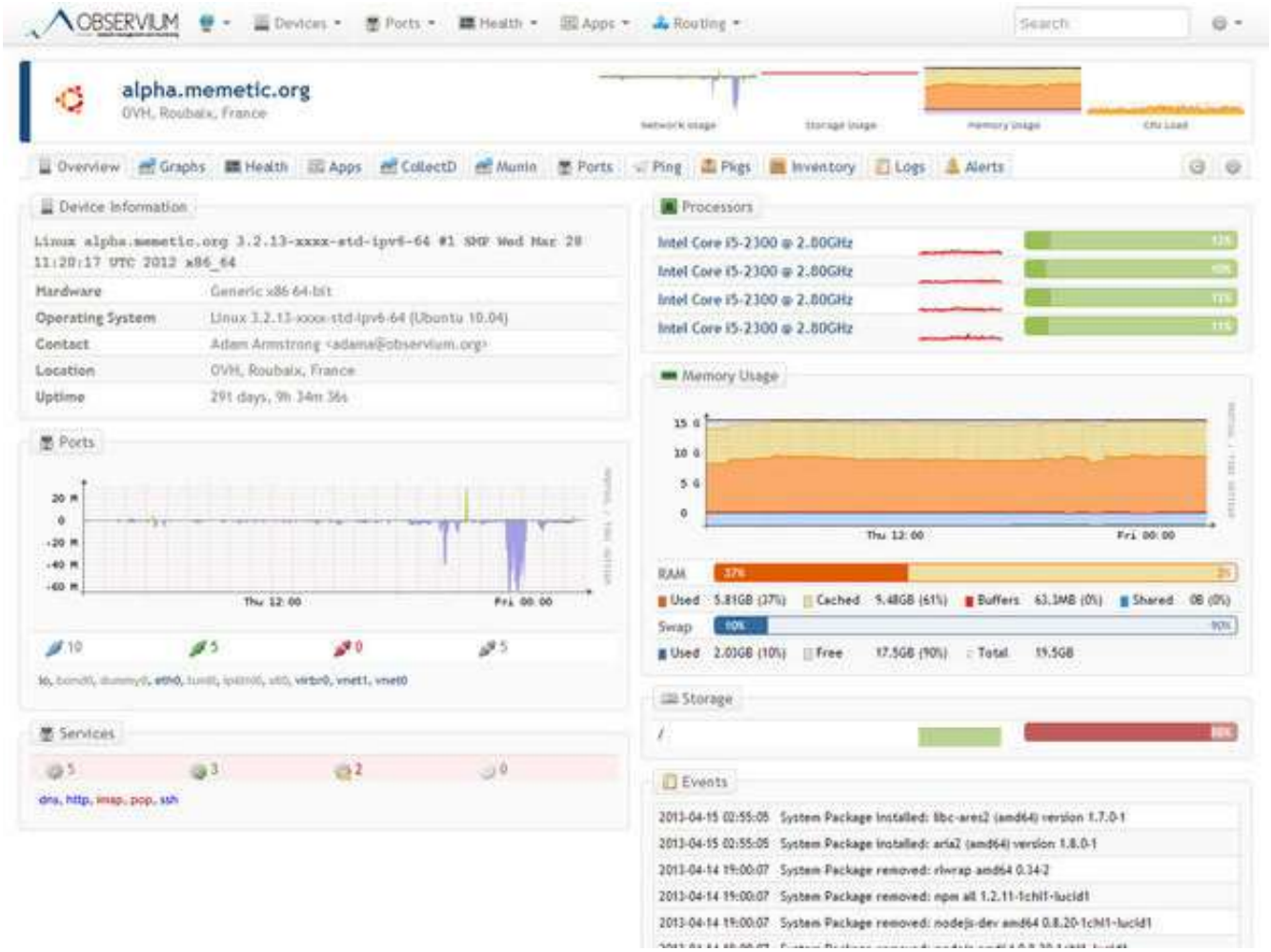

Рис. 2.8. Інтерфейс *Observium*

Opsview - система моніторингу мереж, заснована на *Nagios* 3, *Net-SNMP*, *RRDtool* та *Nagvis* (рис. 2.9).

Особливості *Opsview*:

– можливість розширення архітектури моніторингу, що підтримує контроль працездатності тисяч об'єктів;

– має подієвий механізм відображення стану об'єктів моніторингу, наочно відображає зв'язок часу настання подій і контрольованих елементів;

– присутність набору утиліт для міграції даних з інших систем моніторингу;

– сумісність з плагінами, агентами і розширеннями, підготовленими для *Nagios*;

– відображення інформації про динаміку зміни продуктивності і доступності систем;

– архітектура системи спочатку розрахована на можливість переробки під свої потреби і легкість інтеграції з іншими системами;

– підтримка моніторингу широкого спектру різних мережевих пристроїв, операційних систем і додатків [11].

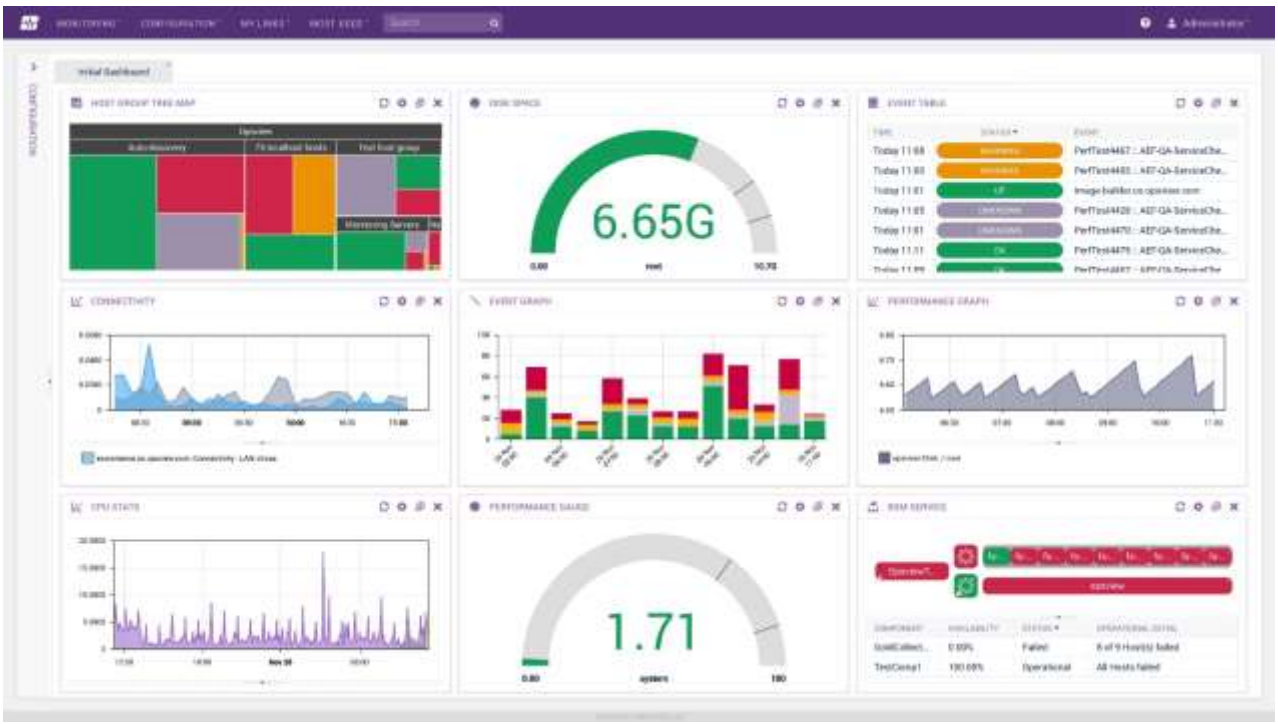

Рис. 2.9. Інтерфейс *Opsview*

Високорівнева програмна платформа для моніторингу мереж і мережевих пристроїв *OpenNMS* (рис. 2.10) дозволить створити рішення мережевого моніторингу для будь-якої IT-інфраструктури промислового масштабу. Можна збирати системні показники за допомогою *JMX*, *WMI*, *SNMP*, *NRPE*, *XML HTTP*, *JDBC*, *XML*, *JSON* і т. Д.

За допомогою *OpenNMS* можна у вашій мережі, як виявляти зв'язку мережевих топологій на другому рівні моделі *OSI*, так і відстежувати неполадки в маршрутизації на рівні 3. Ця система моніторингу не використовує агентів, а побудована на подієво-орієнтованій архітектурі, а також підтримує роботу в зв'язці з системою агрегації даних і відображення графіків в реальному часі *Grafana*.

penNMS має вбудовані модулі формування звітності, а це означає, що ви можете переглядати звіти у вигляді красивих дашборда (*dashboard*, аналітичних інформаційних панелей) і діаграм. В цілому, *OpenNMS* отримав прекрасний користувальницький інтерфейс.

Рис. 2.10. Інтерфейс *OpenNMS*

Можна встановити *OpenNMS* в *Docker* - програмний інструментарій для управління ізольованими *Linux*-контейнерами.

Короткий перелік доступних можливостей:

– *openNMS* спеціально розроблявся для *Linux*, але також є реалізована підтримка *Windows*, *Solaris* і *OSX*;

– моніторинг температури пристроїв;

– налаштовуючи інформаційна панель адміністратора;

– моніторинг електропостачання;

– підтримка *IPv*4 і *IPv*6;

– налаштування формування повідомлень про події та їх відправка по електронній пошті, СМС, *XMPP* (розширюваний протокол обміну повідомленнями та інформацією про присутність, раніше відомий як *Jabber*) і іншими способами;

– географічна карта мережевих вузлів для відображення місця розташування «проблемних» вузлів і перебоїв у наданні послуг з використанням карт таких картографічних порталів, як *Open Street Map, Google Maps* або *Mapquest*.

PRTG Network Monitor (рис. 2.11) - цей інструмент призначення для моніторингу локальних мереж будь-которого розміру. ВІН працює только з *Windows*. Продукт пошірюється по платній Ліцензії, заощаділі можна только на Першому місяці использование з метою Знайомство з можливіть програми. Інструмент добро впорається з усіма завдання моніторингу, просканує технічний стан всех підключеніх до мережі прістроїв, виявило кібератакі.

Серед функцій, які зацікавлять адміністратора корпоратівної мережі:

– перевірка трафіку;

– можливість збереження даних статистики моніторингу мережі в базу для подальшого аналізу;

– карта стану мережі, Вивчення якої можливо в режімі реального часу;

– збір даних про стан и допустимому навантаженні на всі елементи мережі.

Рис. 2.11. Інтерфейс *PRTG Network Monitor*

Оцінюючі переваги програми, потрібно відзначити наступне:

– опція настройки панелей під спожи адміністратора;

– можливість вести моніторинг в гнучкий форматі;

– карта мережі в реальному часі.

Недоліки:

– відносно висока вартість;

– відсутність групових сенсорів, що опитують одночасно кілька датчиків;

– немає окремої бази даних для зберігання статистичної інформації.

Total Network Monitor (рис. 2.12) - в цьому продукті вдало поєднуються доступність і дієвість. Він відрізняється великою кількістю функціональних можливостей, але слід звернути увагу на те, що майже всі з версій, які розповсюджуються безкоштовно, не мають графічного інтерфейсу. Спілкуватися з програмою доведеться за допомогою текстового інтерфейсу, а не іконок або піктограм. Програма дозволяє з заданою періодичністю перевіряти будь-який параметр роботи мережі, наприклад, доступність і якість роботи серверів. Важливо, що вона має великий обсяг самостійних повноважень: автоматично запускає антивірус, вносить дані в журнал подій, при необхідності перезавантажує процеси. Всі результати перевірок доступні у вигляді звітів про роботу системи.

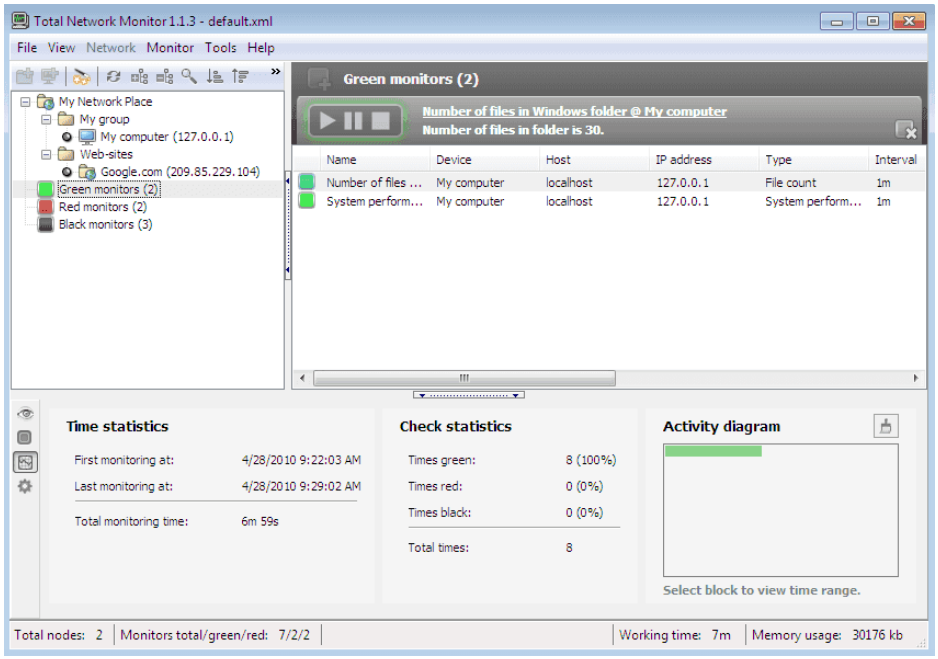

Рис. 2.12. Інтерфейс *Total Network Monitor*

До плюсів цього інструменту моніторингу мережевої безпеки експерти відносять:

– невисоку ціну;

– простоту установки;

– легкість управління.

Серед мінусів відзначають наступне:

– відсутність дашборда в інтерфейсі;

– відсутність мобільної версії;

– розробники не оновлюють програмний продукт, тому він відстає від аналогічних засобів забезпечення безпеки;

– неможливість роботи в режимі багатопоточності.

Kismet (рис. 2.13) – невеликий *open-source*-додаток для системних адміністраторів компанії будь-якого рівня. Система дозволяє:

– аналізувати трафік в локальній мережі;

– знаходити несправності або відхилення;

– передбачати і усувати збої.

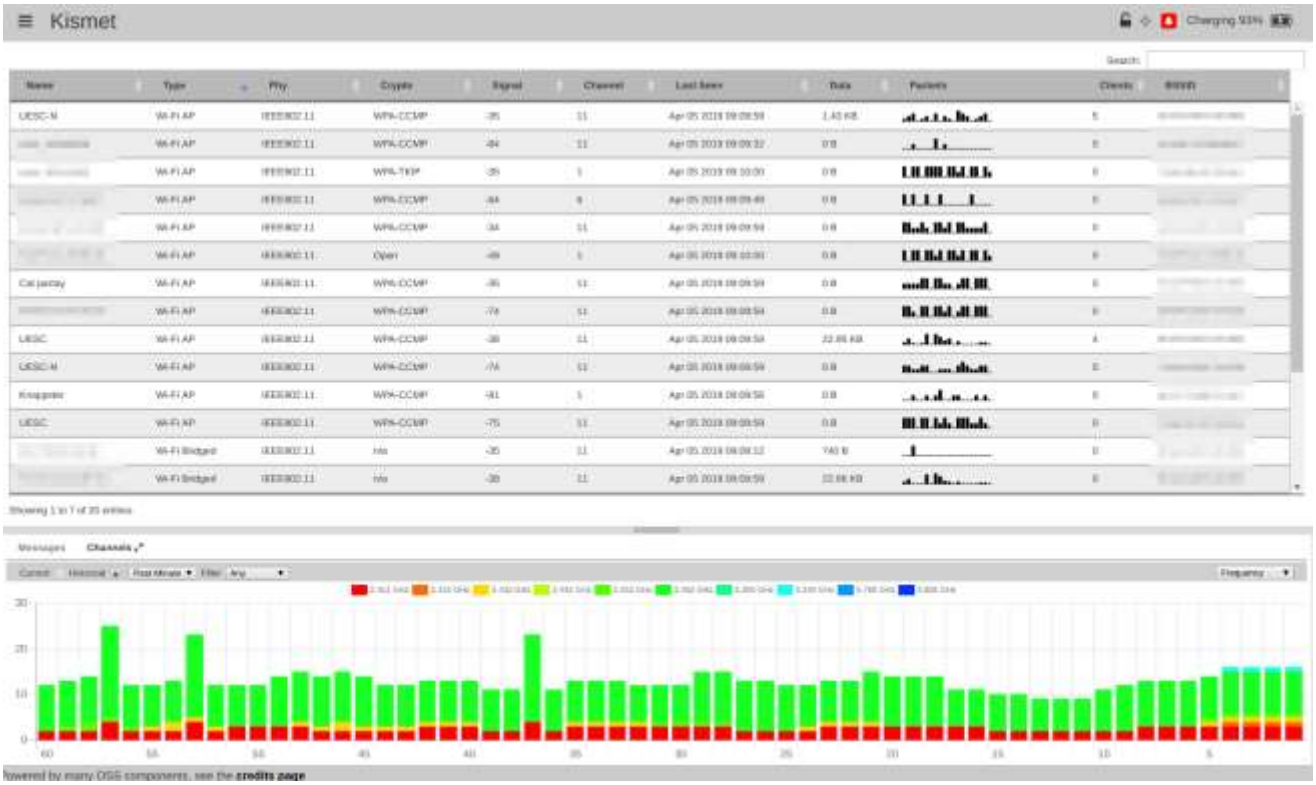

Рис. 2.13. Інтерфейс *Kismet*

Важливо, що воно з рівним успіхом працює з більшістю операційних систем. Додаток зручний для аналізу локальних мереж, що працюють на базі стандарту 802.11 *b*. Його допустимо використовувати для мереж з прихованим *SSID*.

При проведенні моніторингу вийде виявити:

- невірно організовані точки доступу до мережі;
- спроби несанкціонованого проникнення;
- приховані пристрої, які можуть завдати шкоди мережі.

Зовнішні атаки з легкістю діагностуються як на рівні локальної мережі, так і на рівні каналів зв'язку. Повідомлення про атаку налаштовується зручним для адміністратора способом.

Переваги програми:

- безкоштовність;
- має простий інтерфейс;
- реалізований принцип пакетного сніфферу.

Сніфер – це утиліта, яка використовує карту мережі, що працює в режимі *promiscuous mode* (адаптер в цьому режимі відправляє всі пакети, отримані по фізичних каналах, з додатком для обробки і допомагає аналізувати трафік за багатьма параметрами).

Недоліки:

- вимагає уваги до управління;
- сканер неполадок працює повільно [16-18].

2.2. Порівняльний аналіз систем моніторингу та управління

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 устаткування, від надійності роботи якого залежить робота всієї мережі. Розподілений характер великої корпоративної мережі унеможливлює підтримання її роботи без централізованої системи управління, яка в автоматичному режимі збирає інформацію про стан

кожного концентратора, комутатора, мультиплексора і маршрутизатора і надає цю інформацію до оператора мобільного зв'язку.

Система моніторингу та управління працює зазвичай в автоматизованому режимі: система виконує найбільш прості дії з управління мережею автоматично, а складні рішення, на основі підготовленої системою інформації, приймає людина.

У зв'язку з тим, що самі системи управління являють собою складні програмно-апаратні комплекси, існує межа доцільності застосування системи управління, яка визначається складністю мережі, різноманітністю застосовуваного комунікаційного обладнання і ступенем його распределенности по території. Однак при зростанні мережі може виникнути необхідність об'єднання розрізнених програм управління пристроями в єдину систему управління, в зв'язку з чим, можливо, доведеться відмовитися від цих програм і замінити їх інтегрованою системою управління. Для пошуку оптимальної системи управління проведемо порівняння систем моніторингу за параметрами наведеними в табл. 2.1.

Таблиця 2.1

Порівняльний аналіз систем моніторингу та управління мережевим трафіком

Система моніторингу	Параметри								
	1	2	3	4	5	6	7	8	9
Argus	+	-	-	-	+	+	+	+	+
Intellipool Network Monitor	+	-	-	+	-	+	+	-	+
AdRem NetCrunch	-	+	-	+	-	+	+	+	-
IPHost Network Monitor	+	+	-	+	-	+	-	+	-
NetMRI	-	+	-	+	-	+	+	+	+
NetQoS Performance Center	+	+	+	+	-	+	+	+	+
OPNET ACE Live	+	+	+	+	-	+	-	+	+
Opsview	+	+	-	+	+	+	+	+	+
Performance Co-Pilot	-	+	-	-	+	-	-	+	+
Scrutinizer	+	+	-	-	-	+	+	+	+
Orion	+	+	+	+	+	+	+	+	+
Zenoss	+	+	+	+	-	+	+	+	+
Nagios	+	+	-	-	+	+	+	+	+
Zabbix	+	+	+	-	+	+	+	+	+

1. Формування звітів *SLA* (*Service Level Agreement*). Контроль гарантованих параметрів якості обслуговування *SLA*, що визначають міжмережеві взаємодії.

2. Формування трендів. Виявлення основних тенденцій динаміки показників якості роботи мережі.

3. Прогнозування трендів. Прогнозування зміни динаміки показників якості роботи мережі.

4. Аналіз топології мережі. Збір інформації про елементи мережі.

5. Використання агентної моделі моніторингу. Наявність пристроїв, які здійснюють збір і передачу інформації про роботу мережі.

6. Підтримка *SNMP* (*Simple Network Management Protocol*). Використання протоколу *SMNP* для обміну інформацією про стан об'єктів спостереження в режимі реального часу.

7. Протоколювання подій. Формування докладних записів про стан елементів мережі.

8. Датчики позаштатних ситуацій. Наявність пристроїв для оповіщення про виникнення критичних ситуацій, негативної тенденції до зміни показників якості роботи телекомунікаційної мережі.

9. Розподілений моніторинг. Моніторинг сигнального обміну на предмет відповідності роботи обладнання певним специфікаціям протоколів [19].

На сьогоднішній момент існує безліч систем моніторингу та управління, які дозволяють вирішити багато проблем в ході експлуатації мережевого обладнання. Скорочення розходів на впровадження таких систем бажано (а в деяких випадках необхідно) тому розглянемо програмні продукти, що працюють під управлінням операційних системних сімейств *Linux*. Для подальшого аналізу відображено ряд систем, представлений в табл. 2.2. Після проведення порівняльного аналізу систем моніторингу та управління видно, що такі системи, як *Nagios*, *Zennos*, *Zabbix* відповідають всім вимогам. Крім того, ці системи є найбільш зручними для створення на їх основі власного рішення, так як вони:

- дозволяють створювати власні перевірки;
- дозволяють створювати власні компоненти;

– дозволяють легко інтегрувати інші системи;
– є основою ряду систем моніторингу;
– дозволяють створити власний метод конфігурування [20].

Таблиця 2.2

Порівняльний аналіз вільнорозповсюджуваних систем моніторингу та управління мережевим трафіком

Система	*Nagios*	*Icinga*	*Zabbix*	*GroundWork*	*Zenoss*
Діаграми	+	-	+	+	+
Прогнозування подій	-	-	+	-	+
Автоматичне виявлення	Через плагін	Через плагін	+	Через плагін	+
Агент	+	+	+	+	-
SNMP	Через плагін	Через плагін	+	Через плагін	+
Syslog	Через плагін	-	+	Через плагін	+
Групування подій	+	+	+	+	+
Зовнішні скрипти	+	+	+	+	+
Плагіни	+	+	+	+	+
Складність створення плагінів	Легко	Середньо	Легко	Середньо	Середньо

Продовження таблиці 2.2

Система	*Nagios*	*Icinga*	*Zabbix*	*GroundWork*	*Zenoss*
Метод збереження даних	*SQL*, плоска БД	*SQL*, плоска БД	*SQLite*, *MySQL*, *PostgreSQL*, *Oracle*	*SQL*, плоска БД	*MySQL* для подій, *Zope* для всього іншого
Управління доступом	+	+	+	+	+
Тригери	+	+	+	+	+
Доступ через *Web*	Перегляд, Звіти, Управління	Перегляд, Звіти, Управління	Повний доступ	Перегляд, Звіти, Управління	Повний доступ
Розподілений моніторинг	+	+	+	+	+

2.3. Вибір та обгрунтування системи

Аналіз з попереднього підрозділу показав, що системи моніторингу, пропоновані на світовому ринку, подібні за виконуваних функцій. Всі вони надають майже однаковий 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.

Всі розглянуті системи моніторингу в основному 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 входити такі компоненти: формування звітів, модуль управління *SNMP*, архів і 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 та організувати подальшу роботу з нею. Консоль управління реалізує функції конфігурації і управління системою.

Для вирішення поставленого завдання вибір зроблений на користь системи моніторингу *Zabbix*, оскільки вона має ліцензію на вільне програмне забезпечення і поширюється безкоштовно, що забезпечить здійснення централізованого управління і моніторингу мережі без витрачання жодних коштів на саму систему. *Zabbix* по праву вважається одним з найбільш просунутих інструментів для віддаленого моніторингу апаратних і програмних ресурсів. Система з успіхом дозволяє вирішувати завдання по відстеженню мережевої активності і працездатності серверів, а також попереджати про потенційно небезпечні ситуації. Завдяки вбудованим механізмам аналізу і прогнозування, *Zabbix* може стати основою для створення повноцінної стратегії ефективного використання *IT*-інфраструктури в компаніях будь-якого масштабу. Крім того, *Zabbix* має добре розвинену документацію на програмне забезпечення. І в результаті порівняльного аналізу ця система має найбільшу кількість переваг [20].

Висновки за розділом

В другому розділі детально розглянуті та проаналізовані основі системи моніторингу та управління мережевим трафіком користувачів. Для подальшого вибору на основі таких параметрів як: аналіз топології мережі, збір інформації про елементи мережі, тощо – сформовано порівняльну характеристику. На основі проведеного аналізу обрано систему, яка покриває найбільшу кількість показників. В результаті вибір зроблений на користь системи моніторингу *Zabbix*, оскільки вона має ліцензію на вільне програмне забезпечення, поширюється безкоштовно, має багато інструментів для віддаленого моніторингу апаратних і програмних ресурсів, дозволяє вирішувати завдання по відстеженню мережевої активності і працездатності серверів, а також попереджати про потенційно небезпечні ситуації.

РОЗДІЛ 3
РЕАЛІЗАЦІЯ ТА ВПРОВАДЖЕННЯ МОДИФІКАЦІЇ СИСТЕМИ МОНІТОРИНГУ ТА УПРАВЛІННЯ

3.1. Опис та вибір компонентів системи

Для реалізації та налаштування системи моніторингу та управління мережевим трафіком користувачів опишемо та виберемо компоненти системи.

Zabbix складається з декількох основних компонентів програмного забезпечення, призначення яких викладені нижче.

Zabbix Сервер. Це ядро програмного забезпечення *Zabbix*. Сервер може віддалено перевіряти мережеві сервіси (такі як веб-сервери і поштові сервери), використовуючи прості перевірки сервісів, але він також є центральним компонентом, яким агенти повідомляють інформацію про доступність, цілісності і статистику. Сервер є сховищем, в якому зберігаються всі конфігураційні, статистичні та оперативні дані, і він є тим компонентом в програмному забезпеченні *Zabbix*, який сповістить адміністраторів у разі виникнення проблем з будь-яким контрольованим обладнанням. *Zabbix* може також виконувати моніторинг без агентів, а також моніторинг мережевих пристроїв за допомогою *SNMP* агентів.

Zabbix проксі. Проксі це необов'язковий компонент розгортання *Zabbix*. Проксі збирає дані про продуктивність і доступність для *Zabbix* сервера. Всі зібрані дані заносяться в буфер на локальному рівні та передаються *Zabbix* сервера, до якого належить проксі. *Zabbix* проксі є ідеальним рішенням для централізованого моніторингу віддалених місць, філій, мереж, які не мають місцевих адміністраторів. *Zabbix* проксі може бути також використаний для розподілу і зняття навантаження з одного *Zabbix* сервера. В цьому випадку, проксі тільки збирає дані, що забезпечує менше навантаження на ЦПУ і на введення / виведення диска на самому сервері.

Zabbix агент. Для активного моніторингу локальних ресурсів і додатків (таких як жорсткі диски, пам'ять, статистика процесора і т.д.) на системах в мережі, повинні

бути запущені *Zabbix* агенти. Агент буде збирати інформацію про роботу системи, на якій він працює, і надавати ці дані *Zabbix* сервера для подальшої обробки. У разі виникнення проблем (наприклад, жорсткий диск заповнився або аварійно завершився певний процес), *Zabbix* сервер може попереджати адміністраторів конкретного обладнання, від якого і виникла проблема. *Zabbix* агенти є надзвичайно ефективними бо вони використовують рідні системні виклики для збору статистичної інформації.

Веб-інтерфейс. Веб-інтерфейс надано для забезпечення легкого доступу до даних моніторингу і конфігурації системи *Zabbix* звідки завгодно і з будь-якої платформи. Інтерфейс є частиною *Zabbix* сервера, і зазвичай (але не обов'язково), запущений на тому ж фізичному сервері що і *Zabbix* сервер.

Таким чином вся система складається з декількох компонентів: *Zabbix* сервер, бази даних, веб-інтерфейс, агент і проксі. Основні компоненти системи зображені на рис. 3.1.

Рис. 3.1. Основні компоненти системи *Zabbix*

Zabbix-сервер. Це основна частина програми. Сервер запитує дані, обробляє і аналізує їх.

Бази даних. Результати аналізу сервера зберігаються в базах даних протягом встановленого часу.

Веб-інтерфейс. З його допомогою зручно працювати з настройками *Zabbix*.

Агент. *Zabbix agent* - це програма, яка встановлюється на хост, збирає необхідні дані і відправляє їх на *Zabbix*-сервер. Агент працює в двох режимах: в активному (агент запитує список параметрів, які потрібні серверу) і в пасивному (отримує запити від сервера). Це необов'язковий компонент системи. Сервер може збирати інформацію з пристрою за допомогою інших інструментів, про які докладніше можна дізнатися на офіційному сайті *Zabbix*.

Проксі. Проксі управляє агентами, що дозволяє знизити навантаження на *Zabbix*-сервер. Як і агент, проксі - необов'язковий інструмент.

Zabbix-сервер запитує дані з пристрою або програми і аналізує їх. Далі всі свої спостереження сервер поміщає в базу даних користувача. Готово, тепер адміністратор може проводити свій аналіз.

Zabbix працює з такими базами даних, як:

- *MySQL*;
- *PostgreSQL*;
- *SQLite*;
- *Oracle*.

Таким чином для реалізації поставленої задачі були обрані такі компоненти системи:

- *zabbix*-сервер - для обробки даних на сервері;
- *zabbix*-агент - для отримання даних с хостів;
- база даних - для збереження даних;
- веб-інтерфейс - для зручного управління та моніторингу [21, 22].

3.2. Схема мережі та налаштування системи

Для підготовки інфраструктури необхідно підготовити стенди на яких буде працювати система для цього будемо використовувати наступну схему локальної мережі зображену на рис. 3.2.

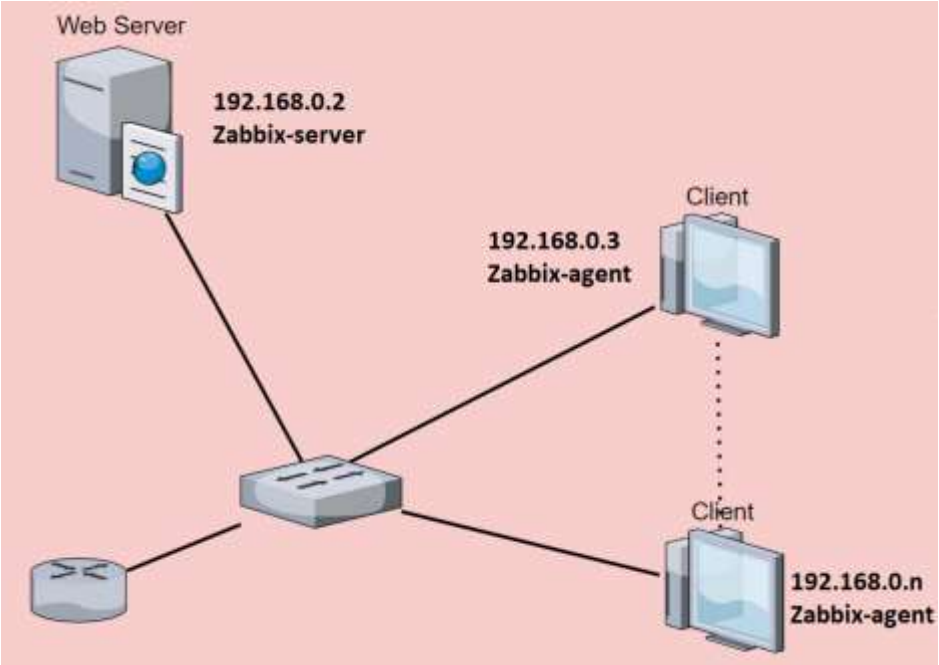

Рис. 3.2. Схема мережі

Проведемо налаштування системи: для цього необхідно підготовити сервер та виконати підготовчі процедури. Для отримання актуальної інформації необхідно, щоб на сервері був правильний час. Для цього спочатку задаємо правильну тимчасову зону:

timedatectl set-timezone Europe/Kyiv

Потім встановлюємо і запускаємо сервіс синхронізації часу:

apt-get install chrony

systemctl enable chrony

systemctl start chrony

Налаштування брандмауера. Для роботи сервера, відкриваємо такі порти:

ufw allow 80,443,10050,10051/*tcp*

ufw allow 10050,10051/*udp*

Де:

- 80 (порт для *http* запитів (веб-інтерфейс));
- 443 (для *https* запитів (веб інтерфейс));
- 10050 (порти для отримання інформації від *zabbix* агентів).

Управління сервером *Zabbix* буде здійснюватися за допомогою веб-інтерфейсу. Для цього необхідно встановити і налаштувати веб-сервер, СУБД і *PHP*. Для цього

будемо використовувати сервер баз даних *mariadb*, веб-сервер *NGINX*, для установки вводимо:

apt-get install mariadb-server

Дозволяємо автозапуск сервера баз даних і запускаємо *mariadb*:

systemctl enable mariadb

systemctl start mariadb

Задаємо пароль для суперкористувача СУБД:

mysqladmin -u root password

Для установки веб-серверу вводимо команду:

apt-get install nginx

Запускаємо *nginx* і дозволяємо його автозапуск:

systemctl enable nginx

systemctl start nginx

Відкриваємо веб-браузер і переходимо по посиланню *http*: // <*IP*-адреса сервера> / - можна побачити вікно вітання зображене на рис. 3.3.

Рис. 3.3. Вікно вітання веб-серверу *Nginx*

Інтерфейс *zabbix* розроблений на *PHP* - наш веб-сервер повинен обробляти скрипти, написані на ньому. Встановлюємо *php* і необхідні компоненти:

apt-get install php php-fpm php-mysql php-pear php-cgi php-common php-ldap php-mbstring php-snmp php-gd php-xml php-gettext php-bcmath

Для настройки *php*, відкриваємо файл:

vi /etc/php/7.2/fpm/php.ini

Редагуємо наступні параметри:

date.timezone = "*Europe/Kyiv*"

...

max_execution_time = 300

...

post_max_size = 16*M*

...

max_input_time = 300

...

max_input_vars = 10000

Дозволимо запуск *php-fpm* і перезапустити його:

*systemctl enable php*7.2*-fpm*

*systemctl restart php*7.2*-fpm*

Для того, щоб *NGINX* обробляв *PHP*, відкриваємо конфігураційний файл:

vi /etc/nginx/sites-enabled/default

У секції *location* додаємо параметр *index*:

```
    location / {
        index  index.php;
        ...
    }
```

Всередині секції *server* додамо наступне:

```
location ~ \.php$ {
        set $root_path /var/www/html;
        fastcgi_buffer_size 32k;
        fastcgi_buffers 4 32k;
        fastcgi_pass unix:/run/php/php7.2-fpm.sock;
        fastcgi_index index.php;
        fastcgi_param SCRIPT_FILENAME $root_path$fastcgi_script_name;
        include fastcgi_params;
        fastcgi_param DOCUMENT_ROOT $root_path;
```

}

Де / *var* / *www* / *html* - кореневий шлях зберігання скриптів, /*run*/*php*/*php*7.2 *fpm*.*sock* - шлях до сокетного файлу *php-fpm* (точне розташування файлу можна подивитися в файлі конфігурації /*etc*/*php*/7.2/*fpm*/*pool*.*d*/*www*.*conf*).

Перевіряємо налаштування *nginx*:

nginx -t

І перезавантажуємо його:

systemctl restart nginx

Створюємо *index*.*php* з наступним змістом:

vi /*var*/*www*/*html*/*index*.*php*

<?*php phpinfo*(); ?>

Відкриваємо веб-браузер і переходимо по посиланню *http*: // <*IP*-адреса сервера> / - тепер можна побачити зведену інформацію по *PHP* і його налаштувань на табл. 3.1.

Таблиця 3.1

Зведена інформацію по *PHP* і його налаштувань

PHP Version 7.2.17-0ubuntu0.18.04.1

System	Linux zabbix 4.15.0-50-generic #54-Ubuntu
Build Date	Apr 18 2019 14:12:38
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.2/fpm
Loaded Configuration File	/etc/php/7.2/fpm/php.ini

Веб-сервер готовий для роботи з *Zabbix Web*. Переходимо до установки самого *Zabbix* сервера.

Спочатку встановимо репозиторій версії *Zabbix*. Для цього

переходимо на сторінку *https://repo.zabbix.com/zabbix/* і переходимо в розділ з самою останньою версією пакета - потім переходимо в *ubuntu* / *pool* / *main* / *z* / *zabbix-release* / - копіюємо посилання на останню версію релізу (рис. 3.4).

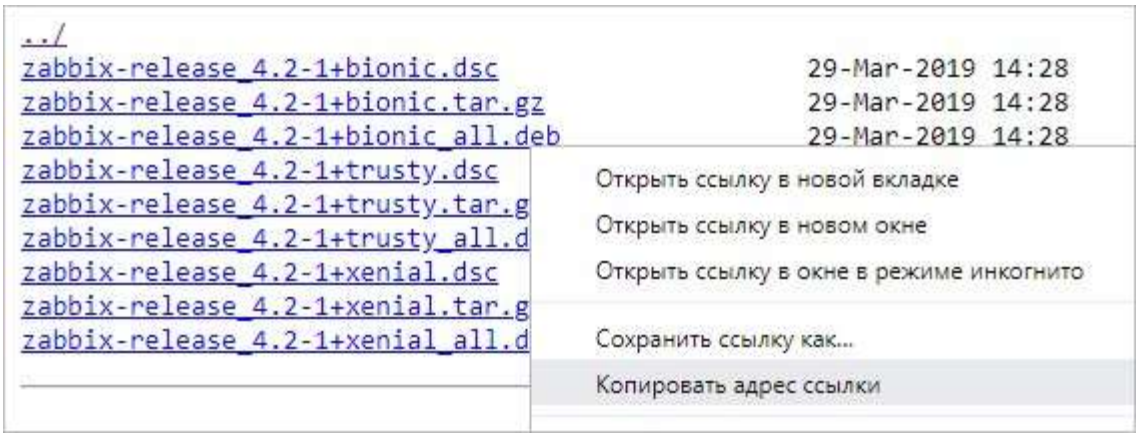

Рис. 3.4. Репозиторій версії *Zabbix*

Щоб зрозуміти, яку кодову назву нашої системи, вводимо команду:

cat/etc/lsb-release | grep DISTRIB_CODENAME.

Викачуємо файл сховища командою:

*wget https://repo.zabbix.com/zabbix/*4.2/*ubuntu/pool/main/z/zabbix-release/zabbix-release_*4.2-1%2B*bionic_all.deb*

Встановлюємо його:

*dpkg -i zabbix-release_*4.2-1+*bionic_all.deb*

Оновлюємо списки пакетів:

apt-get update

Встановлюємо сервер, вводячи команду:

apt-get install zabbix-server-mysql zabbix-frontend-php zabbix-get

Перейдемо до налаштування бази даних, входимо в оболонку ведення *sql*-команд:

mysql -uroot –p

Створюємо базу даних:

*CREATE DATABASE zabbix DEFAULT CHARACTER SET utf*8 *DEFAULT COLLATE utf*8*_bin*;

Створюємо користувача для підключення і роботи зі створеною базою:

GRANT ALL PRIVILEGES ON zabbix. TO zabbix@localhost IDENTIFIED BY 'zabbixpassword';*

В даному створено користувача *zabbix* з доступом до бази *zabbix* і паролем *zabbixpassword*.

Виходимо з *sql*-оболонки:

\q

У складі *zabbix* йде готова схема для СУБД *MySQL* / *MariaDB* або *postgreSQL*. У нашому випадку, нам потрібен *MySQL*. Для застосування схеми переходимо в каталог:

cd /usr/share/doc/zabbix-server-mysql/

Розпаковуємо архів з дампом бази:

gunzip create.sql.gz

Відновлюємо базу їх дампа:

mysql -v -u root -p zabbix < create.sql

Після введення команди система запросить пароль. Необхідно ввести пароль, який був встановлен після установки *mariadb*.

Переходимо до налаштування *zabbix*, відкриваємо конфігураційний файл:

vi /etc/zabbix/zabbix_server.conf

Додаємо строку:

DBPassword=zabbixpassword

Оскільки налаштовуємо портал на підключення до бази з паролем *zabbixpassword*, який був задан при створенні бази для *zabbix*.

І перевіряємо наступні рядки:

DBName=zabbix

DBUser=zabbix

Ім'я бази і користувача повинні бути *zabbix* (як і було створено в *mariadb*).

Створюємо каталог для включення конфігураційних файлів (з якоїсь причини, він може бути не створений при установці):

mkdir /etc/zabbix/zabbix_server.conf.d

Також створюємо каталог для логів і задаємо власника:

mkdir /var/log/zabbix-server

chown zabbix:*zabbix /var/log/zabbix-server*

Дозволяємо автозапуск сервера моніторингу:

systemctl enable zabbix-server

После запускаем сам сервер *zabbix*:

systemctl start zabbix-server

Налаштування *nginx*, при установці *zabbix-web* файли порталу копіюються в каталог /*usr*/*share*/*zabbix*. Наш веб-сервер працює з каталогом /*var*/*www*/*html*. Міняємо це - відкриваємо конфігураційний файл *nginx*:

vi /etc/nginx/sites-enabled/default

Редагуємо параметри *root* і *set* $ *root_path*:

root /usr/share/zabbix;

...

set $*root_path /usr/share/zabbix*;

Перезапускаємо *nginx*:

systemctl restart nginx

Останнє, що лишилось це налаштування веб-інтерфейсу для управління *Zabbix*: Відкриваємо браузер і переходимо за адресою *http*: // <*IP*-адреса сервера> / - відкриється сторінка установки *Zabbix Web*. Клікаємо за посиланням *Next Step*:

У наступному вікні уважно дивимося на результати перевірки нашого веб-сервера - справа ми повинні побачити всі *OK*. Якщо це не так, перевіряємо настройки і виправляємо попередження і помилки, після перезапускаємо сторінку *F*5 для повторної перевірки налаштувань.

Коли всі результати будуть *OK*, натискаємо по *Next Step*.

У наступному вікні ми залишаємо налаштування підключення до бази як є - додатково прописуємо пароль, який задали при створенні користувача *zabbix*, в нашому випадку, пароль був *zabbixpassword*. Після натискаємо *Next Step* (рис. 3.5).

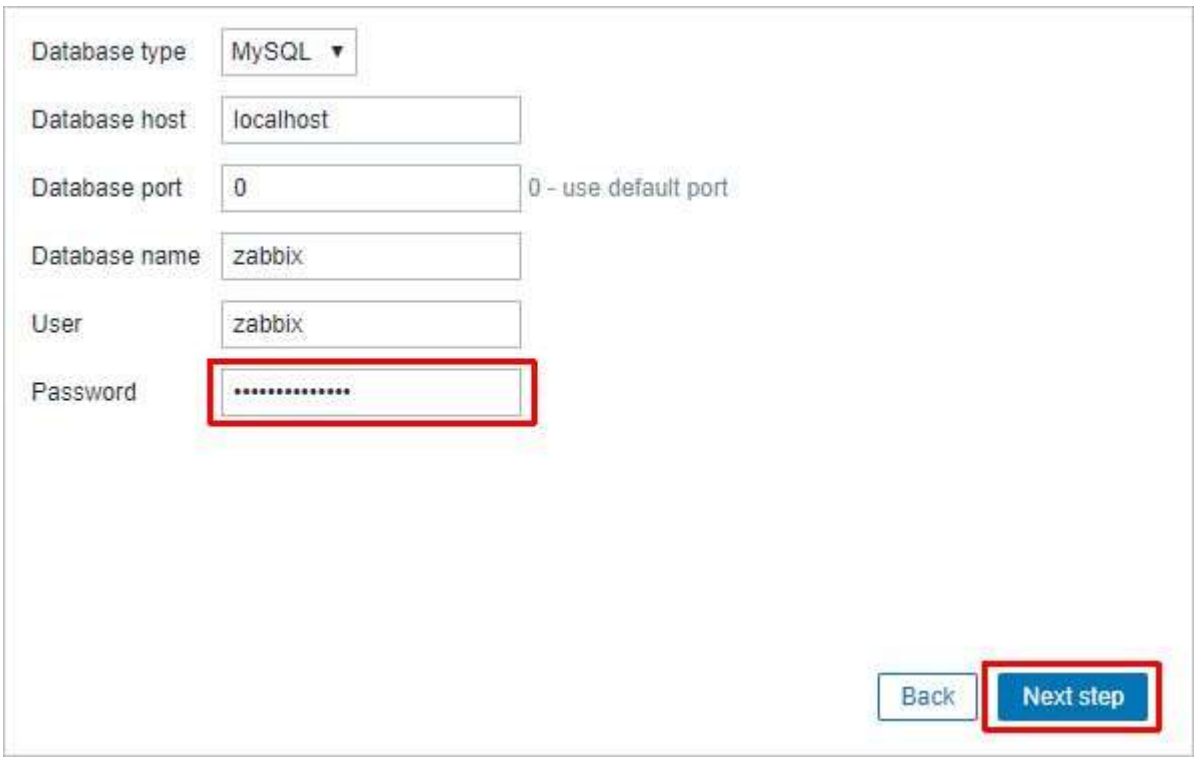

Рис. 3.5. Налаштування веб-інтерфейсу

В наступного вікні залишаємо все як є і натискаємо *Next step*, в останньому вікні перевіряємо настройки і натискаємо *Next Step* (рис. 3.6).

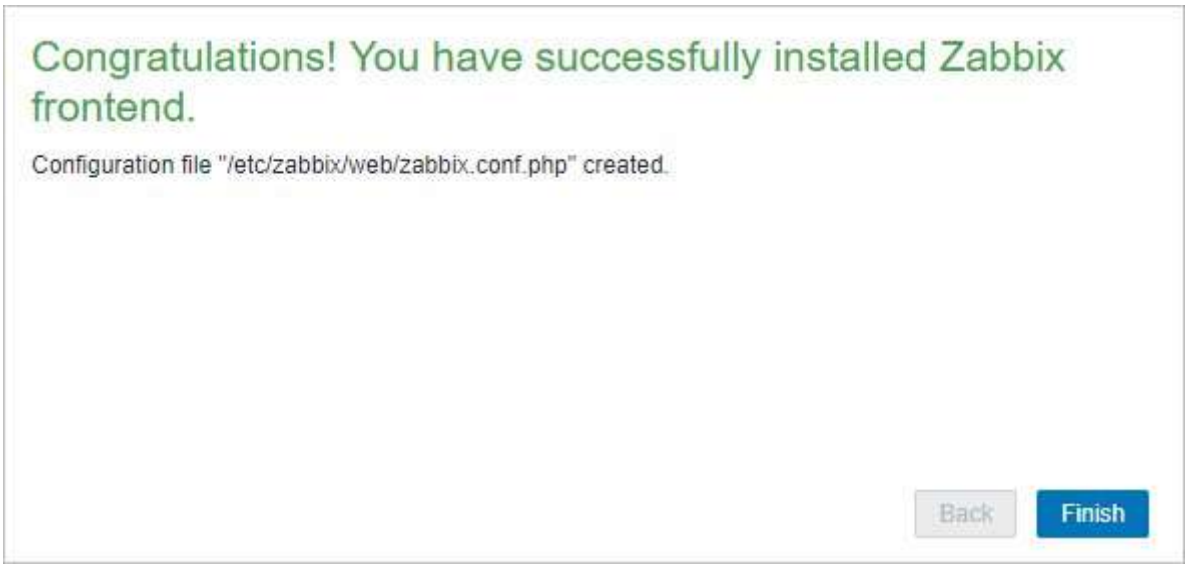

Рис. 3.6. Вікно успішного налаштування веб-інтерфейсу *Zabbix*

Установка завершена - натискаємо *Finish*.

Далі у вікні вводимо логін *Admin* і пароль *zabbix* (за замовчуванням) - відкриється вікно зі зведеною інформацією з моніторингу (рис. 3.7).

System information

Parameter	Value	Details
Zabbix server is running	Yes	localhost:10051
Number of hosts (enabled/disabled/templates)	84	1 / 0 / 83
Number of items (enabled/disabled/not supported)	79	73 / 0 / 6
Number of triggers (enabled/disabled [problem/ok])	48	48 / 0 [1 / 47]
Number of users (online)	2	1
Required server performance, new values per second	1.12	

Рис. 3.7. Вікно зі зведеною інформацію з моніторингу

Проведемо налаштування агенту, для цього на комп’ютері користувача потрібно встановити репозиторій та агент *zabbix*:

apt-get install zabbix-agent

Відкриємо конфігураційний файл:

vi /etc/zabbix/zabbix_agentd.conf

І для параметру *Server* потрібно встановити *ip*-адрес *zabbix* серверу, дозволяємо автозапуск агента і запускаємо його:

systemctl enable zabbix-agent

systemctl start zabbix-agent

Додаємо вузол на якому встановлено агент до системи для подальшого відстежування. Вузол мережі в *Zabbix* - це об'єкт мережі (фізичний, віртуальний), який потрібно відстежувати. Визначення того, що може бути "вузлом мережі" в *Zabbix*, дуже гнучке. Це може бути фізичний сервер, мережевий комутатор, комп’ютер користувача, тощо. Для додавання нового вузла мережі потрібно натиснути “Створити”. Ця дія покаже нам діалог настройки вузла мережі (рис. 3.8).

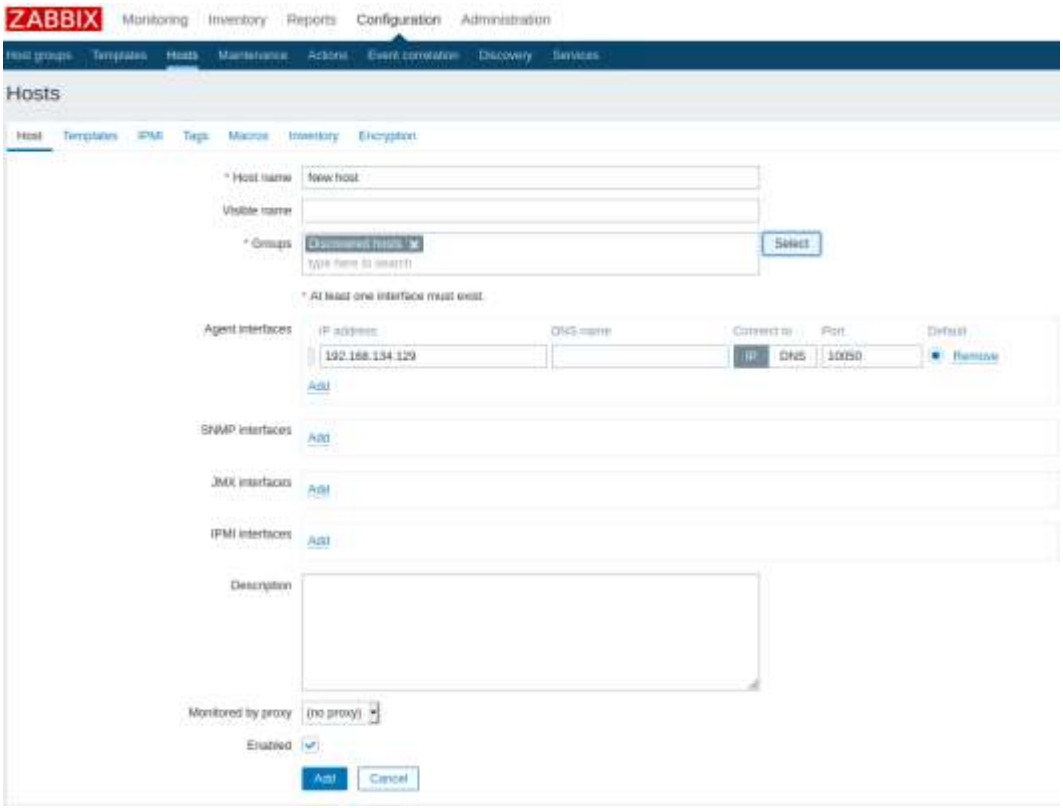

Рис. 3.8. Діалог додавання нового вузла мережі

Додаємо шаблон вузла, для того щоб співвіднести його до відповідної категорії (рис. 3.9).

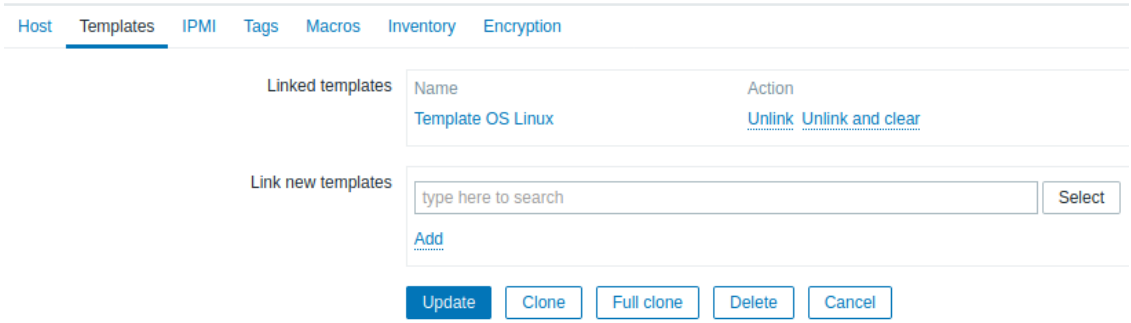

Рис. 3.9. Додавання категорії

Повертаємося до вкладки *All Hosts* і біля доданого вузла індикатор *zbx* повинен відображати зелений колір (рис. 3.10), що свідчить про успішне налаштування [23, 24].

Рис. 3.10. Успішне додавання вузла до системи

3.3. Синтез рішень та пропозицій щодо вдосконалення обраної системи

Для вирішення актуальних проблем існує декілька напрямів та пропозицій щодо вдосконалення системи моніторингу та управління мережевим трафіком. Спочатку необхідно визначити, яка інформація є найбільш важливою для моніторингу та збору. Для багатьох організацій втрата пристроїв посередині, як стек основних комутаторів, не призведе до відсутності електронної пошти та доступу до Інтернету для користувачів. Дуже важливо, знати, які пристрої можуть спричинити серйозні вузькі місця у ваших користувачів, якщо вони вийдуть з ладу. Також потрібно знати, які показники слід дивитись, що буде сигналом, коли пристрій виходить з мережі. Для впевненості, що є можливість отримувати ці показники зі своїх критично важливих пристроїв. Більшість програм для моніторингу мережі використовує *SNMP* або інші протоколи для збору загальних показників пристрою, але також надає можливість створювати власні зонди пристроїв, щоб отримувати тільки критичну інформацію.

Налаштування сповіщення. Повідомлення корисні лише в тому випадку, якщо вони надходять у потрібний час, з потрібної причини, до потрібної людини. Важливо знати, коли є потенційна мережева проблема, але також не хочеться, щоб поштові скриньки заповнювали помилкові тривоги чи помилкові спрацьовування. Спрощення моніторингу мережі, шляхом налаштувавши сповіщення на потрібну інформацію важливий критерій для ефективного моніторингу стану мережі.

Визначення та контроль базових показників ефективності мережевої інфраструктури. Перш ніж вживати заходів для поліпшення стану мережі, слід знати, як вона поводиться зараз, а отже, як контролювати продуктивність мережі. Поточна ефективність послужить базовою лінією. Тоді можна встановити відповідні порогові діапазони для прийнятної продуктивності пристрою. Таким чином, коли пороги перевищені, це каже про те, що насправді є справу з низькою продуктивністю, а не лише з помилковим сигналом тривоги.

Впровадити постійний моніторинг (цілодобово та без вихідних). Моніторинг під час перебування в офісі - це перший крок до оптимальної роботи мережі, але

ввімкнення постійного моніторингу допоможе підтримувати здорову роботу мережі в будь-який час доби - особливо в таких галузях, як фінанси, де потрібна цілодобова доступність товарів та послуг. Постійний моніторинг мережі не вимагає більше робочого часу чи часу в офісі. Рішення для моніторингу мережі буде стежити за станом пристрою та надсилати сповіщення в режимі реального часу електронною поштою або текстом у будь-який час, коли виникають проблеми з мережею [25, 26].

Проведений аналіз показав, що на даний час існує необхідність впровадження програмного модулю чи модифікації системи моніторингу та управління мережевим трафіком користувачів, який би дозволив в будь-який час за допомогою пошти чи мобільного додатка:

1. Отримувати необхідну інформацію про стан мережі;
2. Дозволив би відстежувати мережевий трафік користувачів в реальному часі;
3. Отримувати оповіщення про показники та стан мережі;
4. Постійно відстежувати стан мережі;
5. Можливість отримувати оповіщення адміністратором в будь-який час та незалежно від місця знаходження.

Таким чином система забезпечить не тільки велику функціональність моніторингу та управління, але й мобільність моніторингу та управління цією системою.

3.4. Реалізація модифікації системи шляхом інтеграції програмного модулю

В зв’язку з необхідністю отримання оповіщень з системи в будь-який час про стан мережі та ключових вузлів було вирішено реалізувати та інтегрувати програмний модуль, який би дозволяв:

- отримувати актуальні дані про важливі події в мережі;
- графіки з інформацією про стан вузлів та мережі.

Для отримання оповіщень на мобільний пристрій зручніше всього

використовувати месенджери бо вони мають оповіщення про повідомлення та швидкий доступ до них, підтримку медіа-файлів, та можливості до інтеграції з іншими програмами чи системами. Також месенджери мають такі переваги як:

- миттєве отримання повідомлень;
- обмін будь-якими файлами;
- можливості з використання *API*;
- шифрування повідомлень;
- автоматична синхронізація між пристроями.

Аналіз найпопулярніших сьогодні месенджерів з точки зору інтеграції та отримання повідомлень з системи представлений в табл. 3.2.

Таблиця 3.2

Деталізоване порівняння Месенджерів

Особливість	*Telegram*	*Viber*	*WhatsApp*
Документація	Відкрита та деталізована	Відкрита та деталізована (розробка нових додатків неможлива)	Практично відсутня
Відмовостійкість та Надійність	Висока	Середня	Середня
Конфігурації	Широкий спектр налаштувань	Обмеження на конфігурації	Обмеження на конфігурації
Чат-боти	Необмежені можливості на створення	Існують певні вимоги та обмеження на створення	Не підтримуються

Продовження таблиці 3.2

Особливість	*Telegram*	*Viber*	*WhatsApp*
Швидкість Відправки Повідомлень	150 мілісекунд	900 мілісекунд	550 мілісекунд
Підтримуваний вміст	Будь-який з можливістю форматування	Підтримуються майже любий медіа контент	Існують обмеження на медіа

Виходячи з проведеного аналізу та порівняльної характеристики найбільшу кількість переваг має месенджер *Telegram*.

Для того, щоб отримувати оповіщення від системи в *Telegram*, в першу чергу необхідно створити *Telegram* бота, з яким буде взаємодіяти модуль та система *Zabbix*. Для цього додаємо собі в контакти *@BotFather* і пишемо йому спочатку /*start*, потім /*newbot* (рис. 3.11).

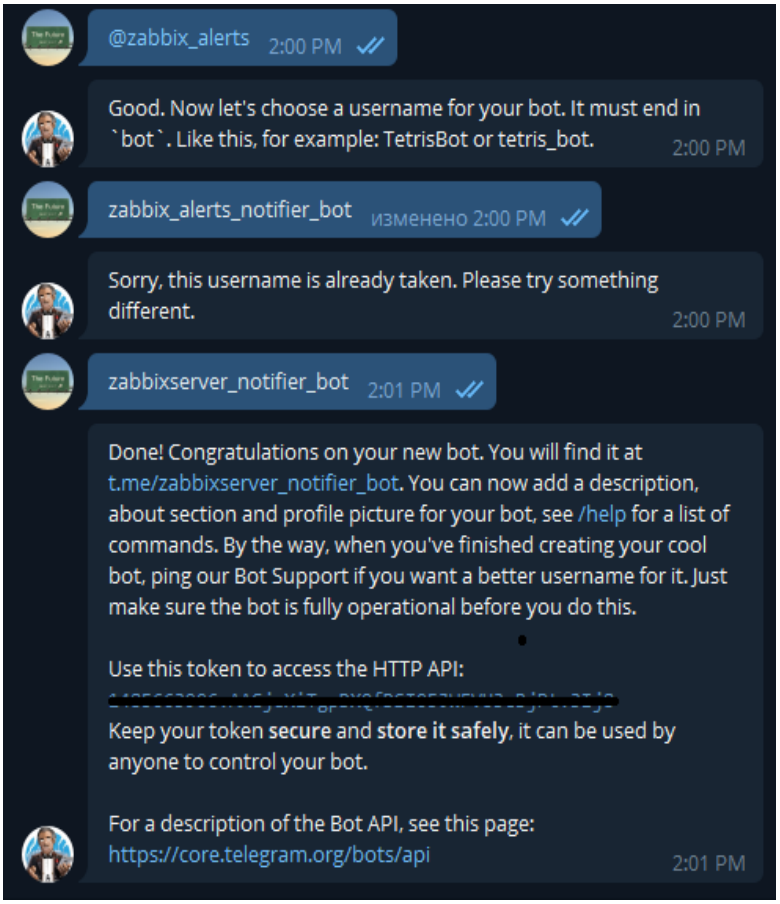

Рис. 3.11. Створення бота

Бот для сповіщень створений і отримали для нього *token*, який знадобиться далі. Тепер потрібно додати в свій список контактів створеного бота. Для цього знайдемо його по імені. В цьому випадку ім'я *@zabbixserver_notifier_bot*.

Тепер перевіримо, як працює відправка повідомлень через бота. Для цього в консолі сервера набираємо таку команду:

curl --header 'Content-Type: application/json' --request 'POST' --data '{"chat_id":"210806260","text":"Test message"}' "https://api.telegram.org/bot1393668911:AAHDfgghTHgfhdfyX28R-wxKfvH1WR6-vdNw/sendMessage"

Результат виконання команди можна побачити на рис. 3.12.

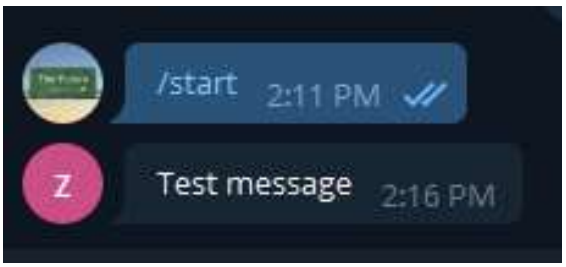

Рис. 3.12. Успішне налаштування бота

Переходимо до налаштування відправки графіків з інформацією про вузол від системи *Zabbix*, для цього використаємо кастомний відкритий плагін *Zabbix-in-Telegram*, виконуємо команду на сервері:

git clone https://github.com/ableev/Zabbix-in-Telegram

Далі потрібно буде встановити кілька модулів *python*, перераховані в *requirements.txt*. Для цього знадобиться *pip*. Встановлюємо його і модулі:

apt install python3-pip

pip3 install -r requirements.txt

Копіюємо в директорію */usr/lib/zabbix/alertscripts* файли *zbxtg.py* і *zbxtg_settings.example.py* з завантаженого з *github* проекту. Останній перейменовуємо в *zbxtg_settings.py*. Призначаємо користувача *zabbix* власником цих файлів:

cp zbxtg.py /usr/lib/zabbix/alertscripts

cp zbxtg_settings.example.py /usr/lib/zabbix/alertscripts

cd /usr/lib/zabbix/alertscripts

mv zbxtg_settings.example.py zbxtg_settings.py

chown -R zabbix. /usr/lib/zabbix/alertscripts

Наводимо вміст *zbxtg_settings.py* до такого виду:

zbx_server = "*http*://127.0.0.1/"

zbx_api_user = "*Admin*"

zbx_api_pass = "*zabbix*"

zbx_server_version = 4

zbx_db_host = "*localhost*"

zbx_db_database = "*zabbix*"

zbx_db_user = "*zabbix*"

zbx_db_password = "*zabbixpassword*"

Зберігаємо конфіг і перевіряємо роботу скрипта:

/usr/lib/zabbix/alertscripts/zbxtg.py "*@some_name*" "*Test*" "*Test message from module*"

В разі успішного налаштування отримуємо наступне повідомлення від бота (рис. 3. 13).

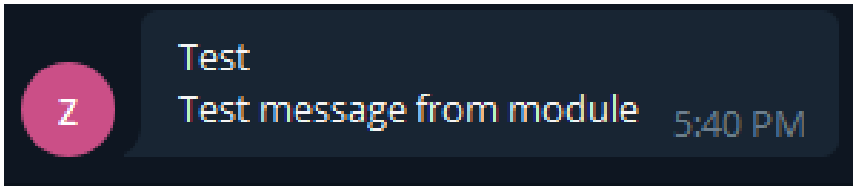

Рис. 3.13. Повідомлення від бота

Для інтеграції модулю в систему *Zabbix* переходимо в *web* інтерфейс і додаємо новий спосіб сповіщень. Для цього переходимо на вкладку *Administration – Media types* та налаштовуємо наступним чином (рис. 3.14):

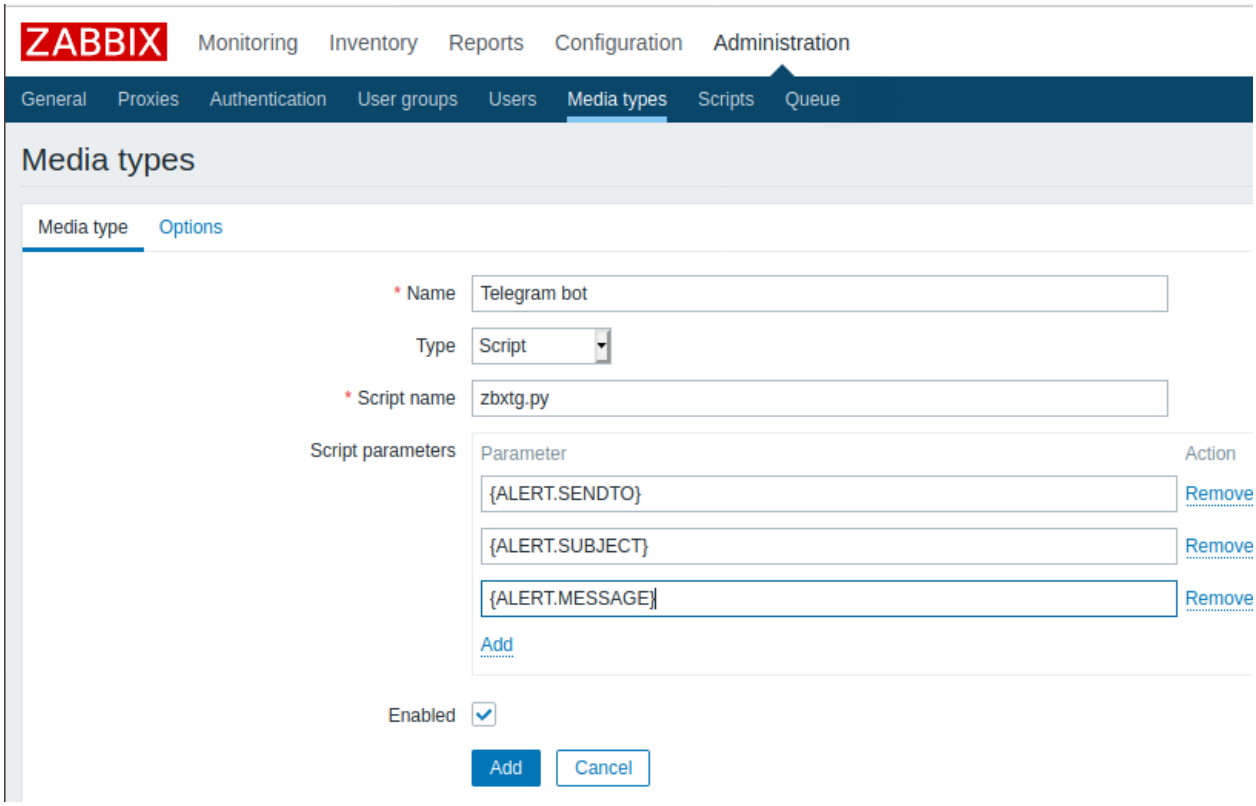

Рис. 3.14. Додавання нового типу оповіщення

Після додавання нового типу оповіщення перевіримо правильність налаштування, для цього потрібно натиснути кнопку “*Test*” в колонці *Action* навпроти назви доданого типу оповіщення *Telegram bot*. Далі відкриється вікно з налаштуванням тестового повідомлення, яке зображене на рис. 3.15.

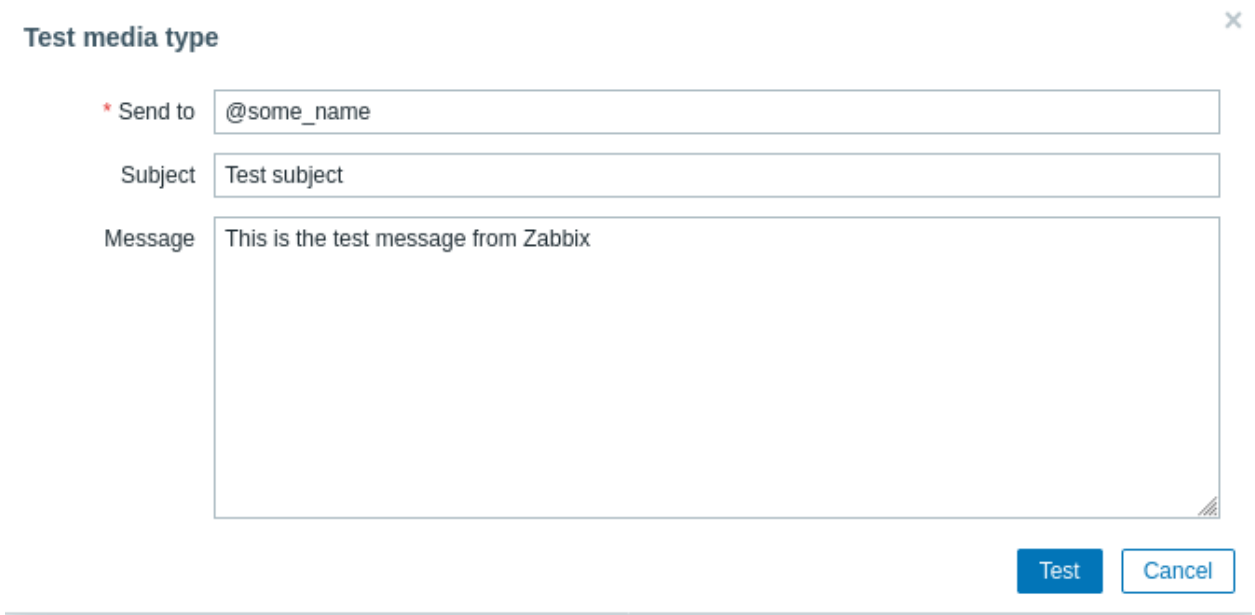

Рис. 3.15. Вікно з тестовим повідомленням

Після чого натискаємо кнопку “*Test*” і система *Zabbix* використовуючи інтегрований модуль повинна відправити повідомлення на вказану адресу, отримане повідомлення можна побачити на рис. 3.16:

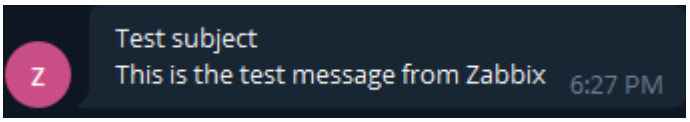

Рис. 3.16. Отримане повідомлення від системи *Zabbix*

Далі обраний тип оповіщення потрібно додати адміністратору для цього перейдемо *Administration – Users – Admin*, вибираємо метод *Telegram bot* та адресу на яку потрібно слати (рис. 3.17).

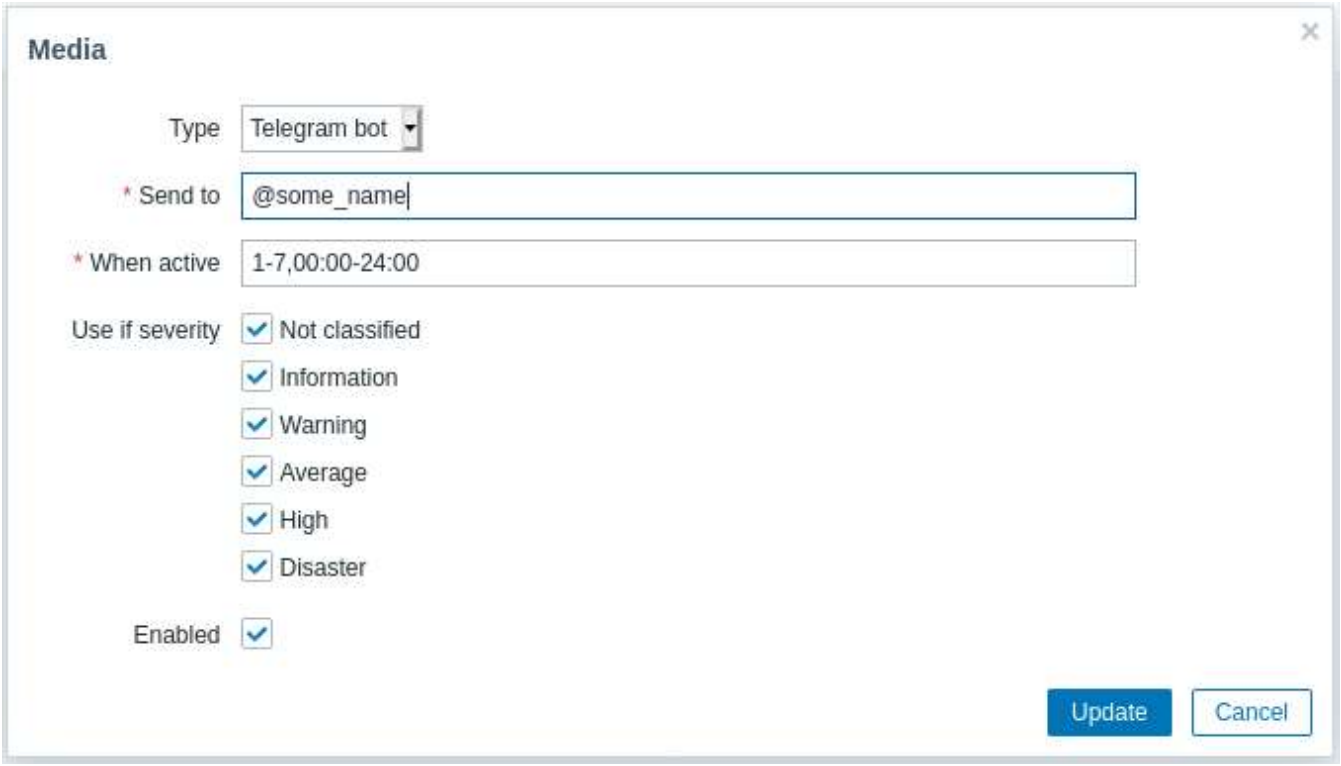

Рис. 3.17. Вікно з доданим методом оповіщення адміністратору

Далі лишається додати тип тригеру та дію, для цього перейдемо на вкладку *Configuration – Actions*, та натискаємо *Create action*, після цього відкриється вікно з чотирма вкладками для налаштування:

– опис дії та тригерів;
– операція при спрацюванні тригеру;
– операція при відновленні стану чи вирішенню проблеми;

– операція при оновлені статусу.

Дію та тригери налаштовуємо відповідно до рис. 3.18.

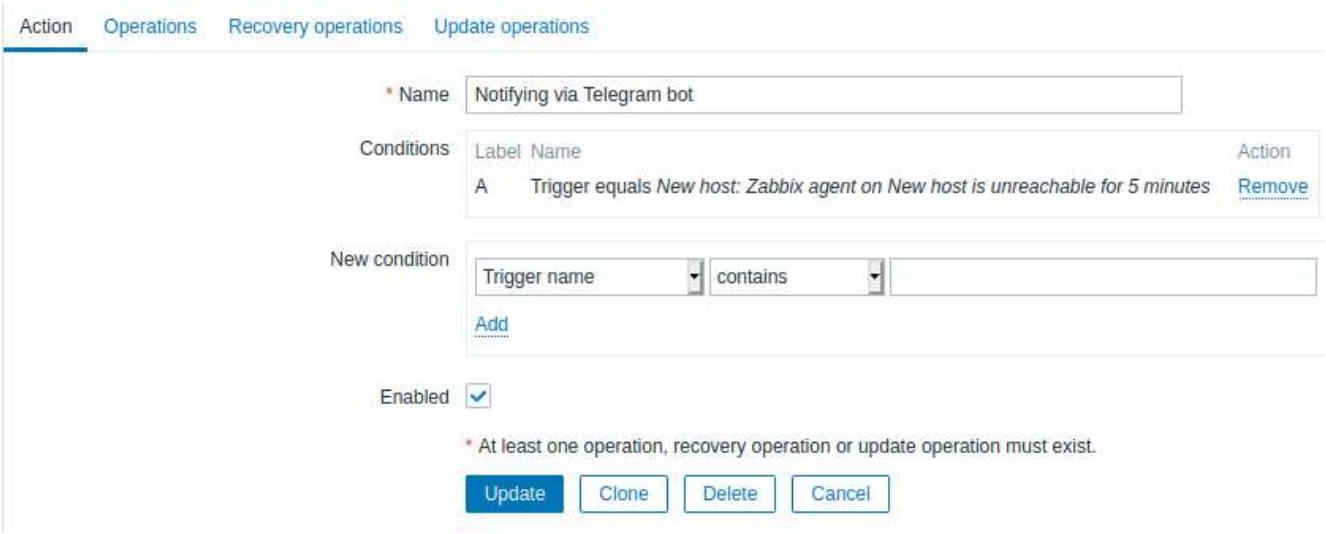

Рис. 3.18. Вікно налаштувань дій та тригерів

В якості операції потрібно вказати відповідні операції, які надаються програмним модулем (рис. 3.19).

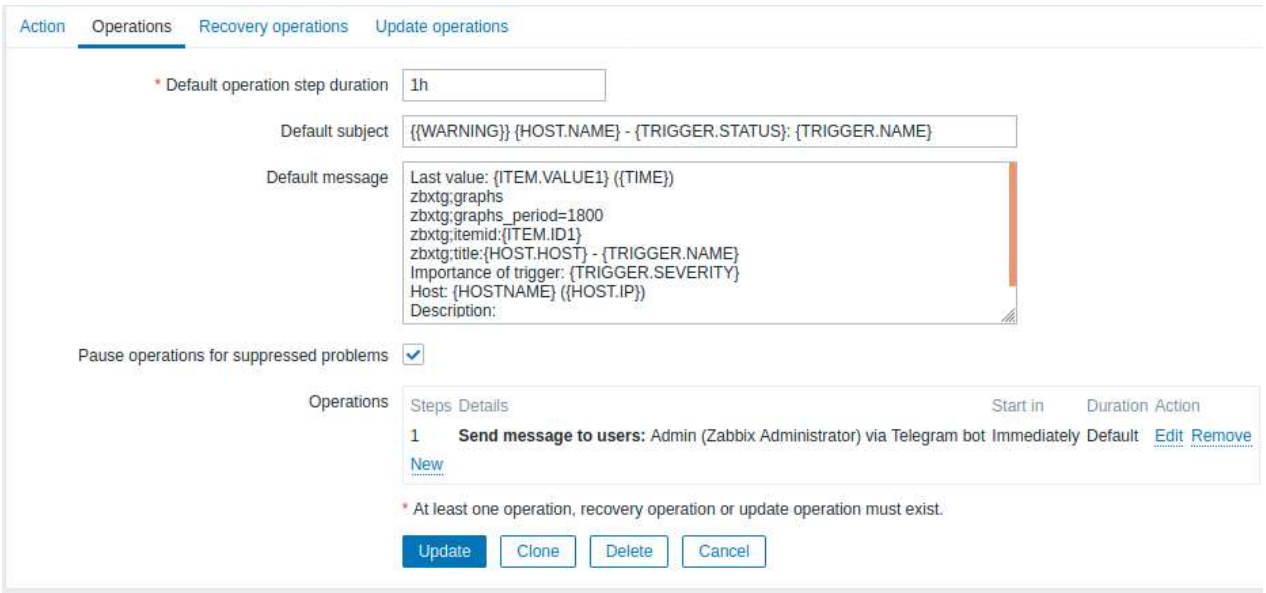

Рис. 3.19. Вікно з налаштуваннями операції

Де:

1. {{*WARNING*}} – макрос для виставлення іконки *emoji* зі знаком оклику;
2. *zbxtg*; *graphs* – включає відправку графіків;
3. *zbxtg*; *graphs_period* = 10800 – період за який будується графік;

4. *zbxtg*; *itemid*: {*ITEM.ID*1} – вибір *itemid* для графіка на основі тригера;

5. *zbxtg*; *title*: {*HOST.HOST*} – {*TRIGGER.NAME*} заголовок графіка.

В випадку вирішення проблеми вказуємо також відповідні операції зображені на рис. 3.20.

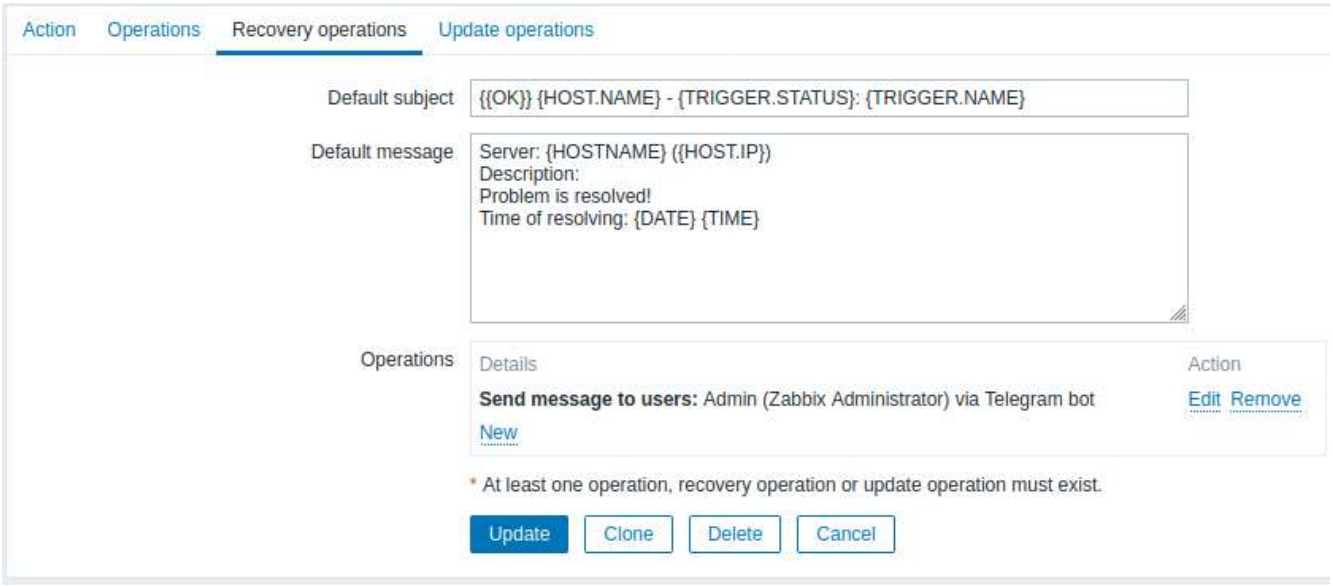

Рис. 3.20. Вікно з операціями в випадку вирішення проблеми

Після налаштування дій та тригерів – модуль повністю є інтегрованим та працездатним, лишається протестувати його роботу [27, 28].

Висновки за розділом

В третьому розділі описано та вибрано компоненти системи, та підготовано тестову інфраструктуру системи моніторингу та управління мережевим трафіком *Zabbix*. Далі проведено аналіз системи, і синтезовано рішення та пропозиції щодо її вдосконалення. В результаті реалізовано модифікацію системи шляхом інтеграції програмного модуля, який дозволяє отримувати оповіщення від системи про важливі події в мережі за допомогою *Telegram* боту, що забезпечує високу мобільність та швидке реагування на позаштатні ситуації.

РОЗДІЛ 4
РЕКОМЕНДАЦІЇ З СУПРОВОДУ ТА ТЕСТУВАННЯ РОБОТИ СИСТЕМИ З ІНТЕГРОВАНИМ ПРОГРАМНИМ МОДУЛЕМ

4.1. Основні вимоги та інструкції з експлуатації

Розглянемо основні вимоги та інструкції з використання системи, зазвичай це вимоги до апаратного та програмне забезпечення, підтримувані платформи та ін.

Zabbix потрібна і оперативна пам'ять, і фізична пам'ять на жорсткому диску. Відправною точкою можуть бути 128 МБ оперативної пам'яті і 256 МБ вільного місця на жорсткому диску. Втім, очевидно, що обсяг необхідної дискової пам'яті залежить від кількості спостережуваних вузлів мережі і спостережуваних параметрів. Якщо планується тривалий час зберігати історію спостережуваних параметрів, то буде потрібно принаймні кілька гігабайт для зберігання даних історії в базі даних. Кожен процес демона *Zabbix* вимагає кілька підключень до бази даних. Обсяг пам'яті необхідний кожним підключенням до бази даних залежить від налаштувань бази даних.

Zabbix і особливо база даних може вимагати значних процесорних ресурсів в залежності від кількості спостережуваних параметрів і обраної бази даних.

У зв'язку з вимогами безпеки і критично важливим характером роботи системи моніторингу, єдиною операційною системою, яка може забезпечити необхідну продуктивність, відмовостійкість і гнучкість є операційна система *UNIX*. *Zabbix* працює на всіх провідних версіях ОС.

Zabbix протестований на наступних платформах:

- *Linux*;
- *IBM AIX*;
- *FreeBSD*;
- *NetBSD*;
- *OpenBSD*;

– *HP-UX*;

– *Mac OS X*;

– *Solaris*;

– *Windows*: всі версії починаючи з *XP* (тільки *Zabbix* агент).

Також *Zabbix* може працювати і на інших *Unix*-подібних операційних системах. *Zabbix* відключає дампи пам'яті, якщо скомпільовано з шифруванням і не запуститься, якщо система не дозволяє відключення дампов пам'яті.

Відносно програмного забезпечення *Zabbix* побудований на сучасному веб-сервері *Apache*, провідних СУБД, і мовою сценаріїв *PHP*.

Підтримувані СУБД представлені в табл. 4.1.

Таблиця 4.1

Підтримувані СУБД

Програма	Версія	Коментар
MySQL	5.0.3 - 8.0.*x*	Потрібно, якщо *MySQL* використовується як основна база даних *Zabbix*'а. Потрібно *InnoDB engine*. *MariaDB* також працює з *Zabbix*.
Oracle	10*g* та більш нова	Потрібно, якщо *Oracle* використовується як основна база даних *Zabbix*'а.
PostgreSQL	8.1 та більш нова	Потрібно, якщо *PostgreSQL* використовується як основна база даних *Zabbix*'а. Пропонуємо використовувати *PostgreSQL* принаймні версії 8.3, який показує дуже хорошу продуктивність *VACUUM*.
TimescaleDB	1.0 чи більш нова	Потрібно, якщо *TimescaleDB* використовується як основна база даних *Zabbix*'а.
IBM DB2	9.7 чи більш нова	Потрібно, якщо *IBM DB2* використовується як основна база даних *Zabbix*'а.
SQLite	3.3.5 чи більш нова	Підтримується тільки на стороні *Zabbix* проксі. Потрібно, якщо *SQLite* використовується базою даних *Zabbix* проксі.

Основні вимоги до серверу можна побачити в табл. 4.2. Обов'язкові вимоги потрібні завжди. Тоді як опціональні вимоги потрібно тільки для підтримки певних функцій.

Таблиця 4.2

Основні вимоги до Серверу

Вимога	Статус	Опис
libpcre	Обов’язково	*PCRE* бібліотека потрібна для підтримки *PCRE* сумісних регулярних виразів. Найменування може відрізнятися в залежності від *GNU / Linux* дистрибутива, наприклад '*libpcre3*' або '*libpcre1*'. Зверніть увагу, що необхідна саме *PCRE* (*v8.X*), тоді як *PCRE2* (*v10.X*) бібліотека не використовується.
libevent		Потрібно для масового збору метрик і *IPMI* моніторингу. Версія 1.4 або новіша. Зверніть увагу, що для *Zabbix* проксі це вимога опциональное, і потрібно тільки для *IPMI* моніторингу.
libpthread		Потрібно для підтримки м'ютексів (*mutex*) і блокувань читання-запису (*read-write*).
zlib		Потрібно для підтримки стиснення.
OpenIPMI	Опціонально	Потрібно для підтримки *IPMI.*
libssh2		Потрібно для підтримки *SSH.* Версія 1.0 або новіша.
fping		Потрібно для елементів даних *ICMP* пінг.

Продовження таблиці 4.2

Наступне програмне забезпечення буде потрібно для роботи веб-інтерфейсу *Zabbix*:

– *Apache* версії 1.3.12 чи більш пізніша;

– *PHP* версії 5.4.0 чи більш пізніша.

Вимога	Статус	Опис
libiksemel	Опціонально	Потрібно для підтримки *Jabber*.
libxml2		Потрібно для моніторингу *VMware* і *XML XPath* предобработки.
net-snmp		Потрібно для підтримки *SNMP*.

Cookies та *Java Script* повинні бути включені. Підтримуються останні версії *Google Chrome*, *Mozilla Firefox*, *Microsoft Internet Explorer* і *Opera*. Також і інші браузери (*Apple Safari*, *Konqueror*) можуть працювати з *Zabbix* [29].

4.2. Тестування роботи системи з інтегрованим програмним модулем

Протестуємо роботу системи з інтегрованим програмним модулем, для цього використаємо встановлений сервер та агент *Zabbix*. Відкриваємо браузер і переходимо за адресою *http*: // <*IP*-адреса сервера> /. Після чого відкриється екран "Привітання" в *Zabbix*. Вводимо ім'я користувача *Admin* з паролем *zabbix* для входу під Супер-Адміністратором *Zabbix* (рис. 4.1).

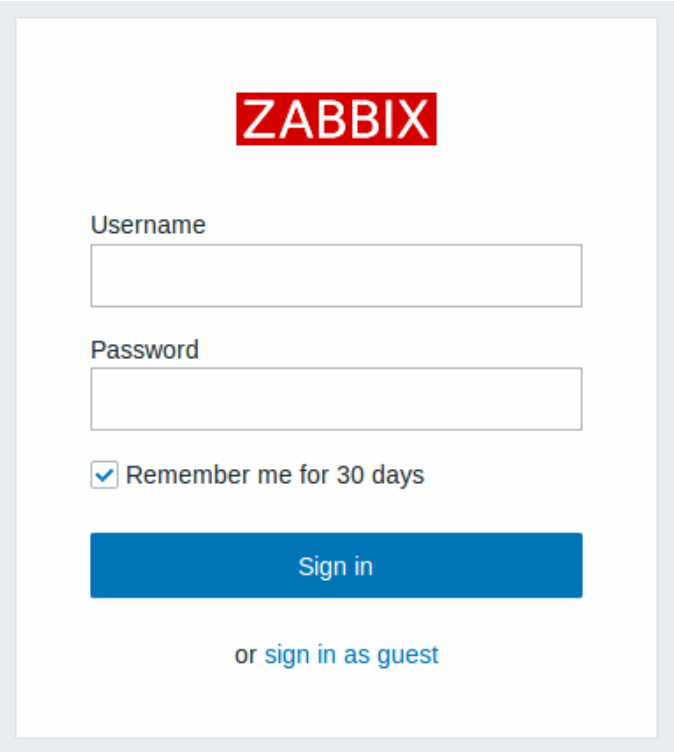

Рис. 4.1. Скріншот вікна привітання в *Zabbix*

Далі потрібно додати новий вузол мережі з агентом *zabbix*. Для додавання нового вузла мережі потрібно натиснути Створити. Ця дія покаже діалог настройки вузла мережі (рис. 4.2).

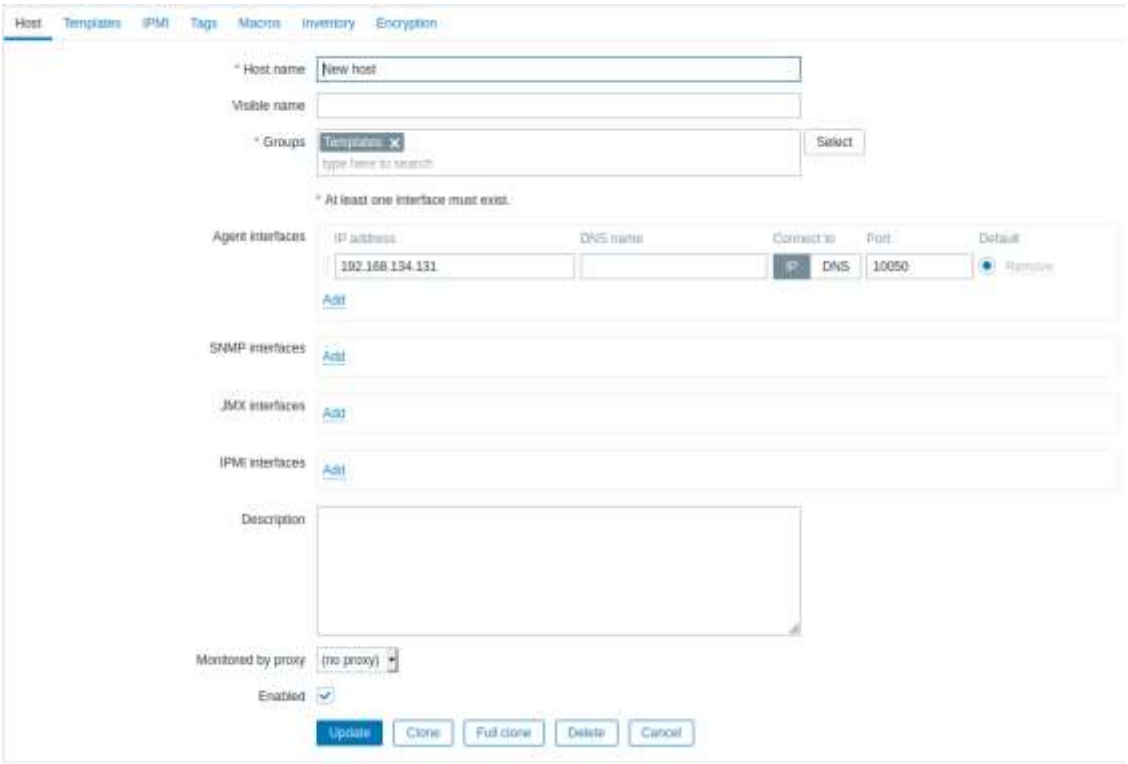

Рис. 4.2. Вікно з опціями для додавання нового вузла мережі

Після додавання вузла він буде відстежуватись на основній сторінці веб-інтерфейсу (рис. 4.3).

Рис. 4.3. Вузли та статус системи

Оскільки модуль вже є інтегрованим та налаштованим з попередніх розділів, для відправки оповіщень адміністратору, лишається відключити від мережі доданий вузол, після чого отримуємо оповіщення від *Telegram* боту, що заданий вузол недоступний вже 5 хвилин (рис. 4.4).

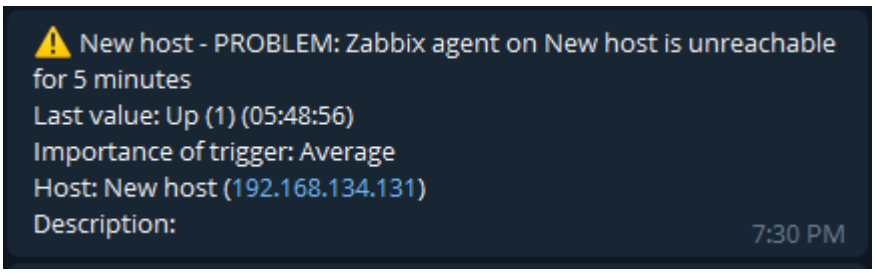

Рис. 4.4. Оповіщення від *Telegram*-боту про недоступність вузла

Та наступним повідомленням телеграм бот відправляє детальну інформації та графік проблеми (рис. 4.5).

Рис. 4.5. Детальний графік проблеми

Відновимо підключення до мережі раніше відключеного вузла, після чого отримуємо повідомлення про вирішення проблеми (рис 4.6).

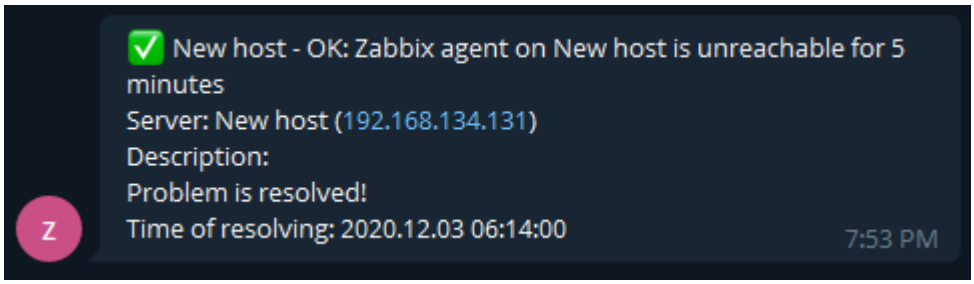

Рис. 4.6. Оповіщення про вирішення проблеми з підключенням вузла

Таким чином, система *Zabbix* з інтегрованим програмним модулем, що забезпечує оповіщення адміністратора мережі шляхом використання *Telegram* бота була налаштована правильно і в результаті спрацювала вірно [30].

Висновки за розділом

В четвертому розділі дано рекомендації з супроводу системи з інтегрованим програмним модулем, а саме розглянуто основні вимоги та інструкції з використання системи, та приведено вимоги до апаратного та програмного забезпечення. Далі протестовано систему *Zabbix* з інтегрованим програмним модулем для оповіщення про події використовуючи *Telegram* бота. В процесі тестування системи створену новий вузол мережі та відключено його від мережі, в результаті чого система відстежила цю подію, та відреагувала відповідним чином, а саме відправила оповіщення з детальною інформацією про подію. Відновивши підключення вузла до мережі – система відстежила цю подію та відіслала оповіщення про те, що вузол став доступним. Що засвідчило те, що система *Zabbix* з інтегрованим програмним модулем налаштована та працює вірно.

ВИСНОВКИ

Результатом дипломної роботи стала модифікована система моніторингу та управління мережевим трафіком користувачів *Zabbix* з інтегрованим програмним модулем, що використовує *Telegram* бота, як засіб швидкого оповіщення адміністратора мережі про позаштатні ситуації. В результаті, адміністратор може швидко відреагувати та усунути відповідну проблему, та відновити працездатний стан вузла, пристрою чи мережі.

Відповідно до теми та завдання дипломної роботи в першому розділі проаналізовано поточний стан і можливості моніторингу та управління мережевим трафіком користувачів. В першу чергу проаналізовано мережеву модель *OSI* в контексті моніторингу та управління мережевим трафіком, а саме протоколи моніторингу такі як: *SNMP* та *RMON*. Досліджено призначення систем моніторингу та управління. Оскільки інформаційна інфраструктура сучасного підприємства складається із різнорідних мереж і систем, щоб забезпечити їх злагоджену і ефективну роботу, доцільно використовувати системи моніторингу та управління мережевим трафіком користувачів.

В роботі проведено огляд та порівняльний аналіз існуючих систем моніторингу та управління мережевим трафіком користувачів. Результатом чого стало зрозуміло, що на підприємствах середнього та найбільшого масштабного застосування одного вільного високотехнологічного рішення дозволяє значно зменшити затрати на впровадження та супроводження системи моніторингу та управління мережевим трафіком. Аналіз з попереднього підрозділу показав, що системи моніторингу, пропоновані на світовому ринку, подібні за виконуваних функцій. Всі вони надають майже однаковий 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. Всі розглянуті системи моніторингу в основному 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 входити такі компоненти: формування звітів, модуль управління *SNMP*, архів і 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 та організувати подальшу роботу з нею. Консоль управління реалізує функції конфігурації і управління системою.

Для вирішення поставленого завдання в результаті аналізу вибір зроблений на користь системи моніторингу *Zabbix*, оскільки вона має ліцензію на вільне програмне забезпечення і поширюється безкоштовно, що забезпечить здійснення централізованого управління і моніторингу мережі без витрачання жодних коштів на саму систему. *Zabbix* по праву вважається одним з найбільш просунутих інструментів для віддаленого моніторингу апаратних і програмних ресурсів. Система з успіхом дозволяє вирішувати завдання по відстеженню мережевої активності і працездатності серверів, а також попереджати про потенційно небезпечні ситуації. Завдяки вбудованим механізмам аналізу і прогнозування, *Zabbix* може стати основою для створення повноцінної стратегії ефективного використання *IT*-інфраструктури в компаніях будь-якого масштабу. Крім того, *Zabbix* має добре розвинену документацію на програмне забезпечення. І в результаті порівняльного аналізу ця система має найбільшу кількість переваг.

В роботі описано та обрано компоненти для реалізації системи моніторингу та управління мережевим трафіком користувачів *Zabbix* такі як: *Zabbix*-сервер, *Zabbix*-агент, *Zabbix*-проксі, веб-інтерфейс. Для реалізації системи було підготовано віртуальне середовище з сервером та вузлом мережі. Далі проведено встановлення та налаштування системи.

Досліджено та синтезовано рішення та пропозиції щодо вдосконалення та модифікації обраної системи моніторингу та управління мережевим трафіком користувачів *Zabbix*. Проведений аналіз показав, що на даний час існує необхідність впровадження програмного модулю чи модифікації системи моніторингу та

управління мережевим трафіком користувачів, який би дозволив в будь-який час за допомогою пошти чи мобільного додатка: отримувати необхідну інформацію про стан мережі, дозволив би відстежувати мережевий трафік користувачів в реальному часі, отримувати оповіщення про показники та стан мережі, постійно відстежувати стан мережі, можливість отримувати оповіщення адміністратором в будь-який час та незалежно від місця знаходження. Таким чином система забезпечить не тільки велику функціональність моніторингу та управління, але й мобільність моніторингу та управління цією системою.

Для реалізації поставленої задачі в роботі проведено аналіз існуючих методів отримання повідомлень і в результаті обрано месенджер, як засіб миттєвого отримання повідомлень, оскільки ця технологія має такий ряд переваг як: миттєве отримання повідомлень, обмін будь-якими файлами та медіа контентом, можливості з використання *API*, шифрування повідомлень, автоматична синхронізація між пристроями. Далі проведено аналіз основних на даний час месенджерів, в результаті порівняльна характеристика свідчить, що *Telegram* має багато можливостей для інтеграції з іншими системами та такі переваги як: відкрита і деталізована документація, висока відмовостійкість та надійність, широкий спектр налаштувань, необмежені можливості на створення чат-ботів, найвищу швидкість відправки повідомлень, підтримка будь-якого вмісту та медіа контенту. В результаті створено чат-бота, який відсилає повідомлення з системи. Для взаємодії *Telegram* чат-бота з системою моніторингу та управління мережевим трафіком *Zabbix* використано та інтегровано програмний модуль, який приймає повідомлення від системи, та відсилає відповідне повідомлення адміністратору, використовуючи створеного бота. Проведено налаштування системи на генерацію повідомлень про позаштатні ситуації, які відбуваються з вузлами в обраному сегменті мережі, який відслідковується.

В роботі також розглянуто основні вимоги та інструкції з експлуатації, детально проаналізовано вимоги до апаратного та програмного забезпечення. Основними з яких є те, що обсяг необхідної дискової пам'яті залежить від кількості спостережуваних вузлів мережі і спостережуваних параметрів. Якщо планується тривалий час зберігати історію спостережуваних параметрів, то буде потрібно

принаймні кілька гігабайт для зберігання даних історії в базі даних. Кожен процес вимагає кілька підключень до бази даних. Обсяг пам'яті необхідний кожному підключенню до бази даних залежить від налаштувань бази даних. *Zabbix* і особливо база даних може вимагати значних процесорних ресурсів в залежності від кількості спостережуваних параметрів і обраної бази даних. Відносно програмного система *Zabbix* має доволі широкий спектр для налаштувань.

Проведено тестування роботи системи моніторингу та управління мережевим трафіком *Zabbix* з інтегрованим програмним модулем для використання *Telegram* чат-бота, як метод оповіщення адміністратора про надзвичайні ситуації, в результаті проведеного тестування система виявила несправність та відправила відповідне повідомлення з описом події чи несправності використовуючи *Telegram*-бота.

СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Панов И. Часть 1. Пять ключевых функций систем мониторинга производительности сети! [Електронний ресурс] / Игорь Панов – Режим доступу до ресурсу: *https://networkguru.ru/piat-cliuchevykh-funktcii-sistem-monitoringa-proizvoditelnosti-seti/*.

2. *Velasco H. OSI Model for Network Management Systems* (*NMS*) [Електронний ресурс] / *Hector Velasco* – Режим доступу до ресурсу: *http://www.hector.velasco.me/images/NMS.pdf.*

3. *Cecil A. A Summary of Network Traffic Monitoring and Analysis Techniques* [Електронний ресурс] / *Alisha Cecil* – Режим доступу до ресурсу: *http://www.cse.wustl.edu/~jain/cse567-06/ftp/net_monitoring/index.html.*

4. *Cecil A.* Огляд методів аналізу та моніторингу мережевого трафіку [Електронний ресурс] / *Alisha Cecil* – Режим доступу до ресурсу: *https://mylektsii.su/11-47171.html.*

5. *Wong E. Network Monitoring Fundamentals and Standards* [Електронний ресурс] / *Edmund Wong* – Режим доступу до ресурсу: *https://www.cse.wustl.edu/~jain/cis788-97/ftp/net_monitoring.pdf.*

6. *Svoboda J. Network Monitoring Approaches: An Overview* [Електронний ресурс] / *J. Svoboda, I. Ghafir, V. Prenosil* – Режим доступу до ресурсу: *https://www.researchgate.net/publication/324966958_Network_Monitoring_Approaches_An_Overview.*

7. *An Efficient Network Monitoring and Management System* [Електронний ресурс] – Режим доступу до ресурсу: *https://core.ac.uk/download/pdf/207663439.pdf.*

8. *Network Monitoring System (Net-Mon)* [Електронний ресурс] – Режим доступу до ресурсу: *http://www.multiresearch.net/cms/publications/CFP1272017.pdf.*

9. Засоби моніторингу та аналізу мережі [Електронний ресурс] – Режим доступу до ресурсу: *https://wiki.cuspu.edu.ua/index.php/*Засоби_моніторингу_та_аналізу_мережі.

10. Мониторинг компьютерных сетей [Електронний ресурс] – Режим доступу до ресурсу: *https://alp-itsm.ru/interesting/monitoring_kompyuternyix_setej/.*

11. Класифікація засобів моніторингу та аналізу [Електронний ресурс] – Режим доступу до ресурсу: *https://wiki.cuspu.edu.ua/index.php*/Класифікація_засобів_моніторингу_та_аналізу.

12. *Zabbix* [Електронний ресурс] – Режим доступу до ресурсу: *https://ru.wikipedia.org/wiki/Zabbix.*

13. *Nagios* [Електронний ресурс] – Режим доступу до ресурсу: *https://ru.wikipedia.org/wiki/Nagios.*

14. Zennos [Електронний ресурс] – Режим доступу до ресурсу: *https://ru.wikipedia.org/wiki/Zennos.*

15. Icing*a* [Електронний ресурс] – Режим доступу до ресурсу: *https://ru.wikipedia.org/wiki/Icinga.*

16. 7 бесплатных программ для мониторинга сети и серверов [Електронний ресурс] – Режим доступу до ресурсу: *https://networkguru.ru/monitoring-seti-setevogo-oborudovaniia-serverov/.*

17. 5 лучших бесплатных систем мониторинга ИТ-инфраструктуры [Електронний ресурс] – Режим доступу до ресурсу: h*ttps://networkguru.ru/5-besplatnykh-sistem-monitoringa-it-infrastruktury/.*

18. МОНИТОРИНГ СЕТЕВОЙ ИНФРАСТРУКТУРЫ [Електронний ресурс] – Режим доступу до ресурсу: *https://searchinform.ru/services/outsource-ib/zaschita-informatsii/monitoring-setevoj-infrastruktury/.*

19. Костенко Е. Ю. СИСТЕМА МОНИТОРИНГА ДЛЯ КОНТРОЛЯ ТРАФИКА ТЕХНОЛОГИЧЕСКИХ СЕТЕЙ ПЕРЕДЧИ ДАННЫХ [Електронний ресурс] / Е. Ю. Костенко, Р. Р. Дуйсенгалиев, Е. А. Барабанова – Режим доступу до ресурсу: *https://cyberleninka.ru/article/n/sistema-monitoringa-dlya-kontrolya-trafika-tehnologicheskih-setey-peredachi-dannyh.*

20. Кузнецов А. С. МОНИТОРИНГ И УПРАВЛЕНИЕ СЕТЬЮ ПЕРЕДАЧИ ДАННЫХ [Електронний ресурс] / А. С. Кузнецов, И. А. Дрейман, А. П. Капуста – Режим доступу до ресурсу: *http://elib.sfu-kras.ru/handle/2311/6919.*

21. КОМПОНЕНТЫ *ZABBIX* [Електронний ресурс] – Режим доступу до ресурсу: *https://www.zabbix.com/documentation/1.8/ru/manual/installation/components.*

22. Что такое *Zabbix* и как его использовать [Електронний ресурс] – Режим доступу до ресурсу: *https://2domains.ru/support/hosting/chto-takoye-zabbix-i-kak-eto-ispolzovat.*

23. Установка и начальная настройка сервера мониторинга *Zabbix* на *Ubuntu* Server [Електронний ресурс] – Режим доступу до ресурсу: *https://www.dmosk.ru/miniinstruktions.php?mini=zabbix-server-ubuntu*

24. НОВЫЙ УЗЕЛ СЕТИ [Електронний ресурс] – Режим доступу до ресурсу: *https://www.zabbix.com/documentation/3.0/ru/manual/quickstart/host.*

25. *Steps to Improve Your Network Infrastructure Monitoring* [Електронний ресурс] – Режим доступу до ресурсу: *https://www.helpsystems.com/intermapper/resources/articles/10-steps-improve-network-monitoring.*

26. *Network Monitoring Best Practices* [Електронний ресурс] – Режим доступу до ресурсу: *https://www.whatsupgold.com/best-practices/network-monitoring.*

27. Деталізоване Порівняння Месенджерів [Електронний ресурс] – Режим доступу до ресурсу: *https://messenger-comparison.azurewebsites.net/uk/.*

28. Отправка уведомлений и графиков из *zabbix* в *telegram* [Електронний ресурс] – Режим доступу до ресурсу: *https://serveradmin.ru/nastroyka-opoveshheniy-zabbix-v-telegram/.*

29. ТРЕБОВАНИЯ [Електронний ресурс] – Режим доступу до ресурсу: *https://www.zabbix.com/documentation/4.2/ru/manual/installation/requirements.*

30. ВХОД И НАСТРОЙКА ПОЛЬЗОВАТЕЛЯ [Електронний ресурс] – Режим доступу до ресурсу: *https://www.zabbix.com/documentation/4.2/ru/manual/quickstart/login.*