МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Кафедра комп’ютеризованих систем управління

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри

_____Литвиненко О.Є.

«___» _____2021 р.

ДИПЛОМНИЙ ПРОЄКТ

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ
"БАКАЛАВР"

Тема: Програмний комплекс вимірювання трафіку в мережах мобільного зв’язку

Виконавець: Гусаков С.С.

Керівник: Марченко Н.Б.

Нормоконтролер: Тупота Є.В.

Київ 2021

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, комп'ютерної та програмної інженерії
Кафедра комп'ютеризованих систем управління
Спеціальність 123 "Комп'ютерна інженерія"
(шифр, найменування)

Освітньо професійна програма «Системне програмування»
Форма навчання заочна

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____Литвиненко О. Є.

«_____»_____2020 р.

ЗАВДАННЯ
на виконання дипломного проєкту

Гусакова Сергія Сергійовича
(прізвище, ім'я, по батькові)

1.Тема роботи: "Програмний комплекс вимірювання трафіку в мережах мобільного зв'язку"

затверджена наказом ректора від "21" грудня 2020 року № 2523 /ст.

2. Термін виконання роботи: з 11.01.2021 до 28.02.2021

3. Вихідні дані до проекту (роботи): постановка задачі до виконання роботи, мови програмування: *C++*, СУБД: *MySQL.*

4. Зміст пояснювальної записки (перелік питань, що підлягають розробці):

1) аналіз небезпек мережевих атак і необхідність контролю мережевого трафіку;
2) аналіз сучасних засобів моніторингу межевого трафіку;
3) описання програмної системи моніторингу та аналізу мережевого трафіку.

5. Перелік обов'язкового графічного матеріалу:

1) Принцип мережевої атаки;
2) Взаємодія компонентів програми;
3) Основні вікна програми;
4) Схема алгоритму виявлення мережевих атак;
5) Схема алгоритму модулю блокування атак.

6. Календарний план-графік

№ п/п	Етапи виконання дипломного проєкту	Термін виконання етапів	Примітка
1	Провести аналіз літератури за темою дипломнго проєкту та аналіз існуючих систем	11.01.21 12.01.21	
2	Зробити вибір компонентів системи	13.01.21- 14.01.21	
3	Розробити структуру програмних засобів системи	15.01.21- 16.01.21	
4	Розробити програмні засоби	17.01.21- 28.01.21	
5	Провести відладку програмних засобів на модельному зразку	29.01.21- 01.02.21	
6	Написати пояснювальну записку	02.02.21- 14.02.21	
7	Підготувати презентацію	15.02.21- 17.02.21	
8	Оформити супрводжувадьну документацію	18.02.21 19.02.21	

7. Дата видачі завдання «_11_» січня 2021 р.

Керівник дипломнго проєкту_____Марченко Н.Б.
(підпис)

Завдання прийняв до виконання_____Гусаков С.С.
(підпис студента)

РЕФЕРАТ

Пояснювальна записка до дипломного проєкту “Програмний комплекс вимірювання трафіку в мережах мобільного зв’язку”: 66 с., 27 рис., 23 літературних джерела, 1 додаток.

МОНІТОРИНГ, МЕРЕЖА, ТРАФІК, АРХІТЕКТУРА ДОДАТКІВ, ОБРОБКА ДАНИХ, ЗАХИСТ, ГРАФІЧНИЙ ІНТЕРФЕЙС КОРИСТУВАЧА, *Qt, C++*

Під час роботи над дипломним проєктом було розроблено програмний засіб для моніторингу стану мережі та мережевого трафіку.

Об’єкт дипломного дослідження – процес моніторингу та аналізу стану мережі.

Предмет дипломного дослідження – програмний засіб пошуку та відображення трафіку мобільної мережі.

Мета дипломного дослідження – розгляд основних методів моніторингу та аналізу, а також проектування та розробка програмного засобу.

Прогнози припущення щодо розвитку об’єкта дослідження – створення робочого зразка програми та використання його в настільних та серверних мережних комп’ютерних системах за умови можливості встановлення програмного забезпечення.

Результати дипломного проєкту рекомендується використовувати при розробці нових програмних засобів, які надають можливість аналізу трафіку мережевих комплексів.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

API – *Application Programming Interface* (прикладний програмний інтерфейс)

CGI – *Common Gateway Interface* (загальний інтерфейс об’єднання)

CRM – *Customer Relationship Management*

DNS – *Domain Name Service* (сервер доменних імен)

FTP – *File Transfer Protocol* (протокол передачі файлів)

GUI – *Graphical User Interface* (графічний інтерфейс користувача)

LAN – *Local Area Network* (локальная мережа)

TCP/IP – *Transmission Control Protocol/Internet Protocol* (протокол управління передачею/міжмережевий протокол)

ВСТУП

Актуальність. За час карантину обсяг мобільного інтернет-трафіку в операторів виріс на 10-30% в залежності від регіону – здебільшого за рахунок відеочатів. Збільшилася і споживання голосових послуг – в середньому на 20% [13].

Обсяг трафіку вже навряд чи повернеться на докарантинний рівень, так як режим вимушеної самоізоляції прискорив цифрову освіту населення. При цьому додаткові витрати українських операторів на модернізацію мереж через збільшення навантаження можуть досягти 900 млн. доларів [5].

З початку режиму самоізоляції не тільки провайдери домашнього інтернету зіткнулися зі зростанням трафіку на своїх мережах. Мобільні оператори також відчувають додаткові навантаження. Обсяг голосового трафіку виріс на 12-23% в залежності від регіону. Трафік збільшився насамперед за рахунок відеочатів і меседжерів. Так, в "*Vodafone*" [13] повідомили про те, що за перші два тижні карантину (*с* 16 по 29 березня 2020 року) абоненти стали вдвічі більше користуватися відеочатами в популярних месенджерах. Число постійних користувачів сервісу *Zoom* виросло в 50 разів. При цьому найпопулярнішим засобом для відеозв'язку у абонентів оператора залишається *WhatsApp* – обсяг відеотрафіка збільшився на 80%. Також зростання трафіку показали всі сервіси з підтримкою відеочатів: *FaceBook* (+ 94%), *Skype* (+ 233%), *Viber* (+ 89%).

Все це вимагає додаткової ємності і пропускних спроможностей як у фіксованій мережі, так і в мобільній. І якщо звернути увагу на мобільні мережі і розглянути *LTE*, то фізичні межі радіоінтерфейсу в цій технології були практично досягнуто. Подальше хоч якесь помітне збільшення пропускної здатності в цьому і наступних поколіннях технологій бездротового зв'язку можливо за рахунок виділення нових, додаткових смуг частот. У операторів в кожному регіоні своя специфіка з точки зору як конкуренції, так і технічного оснащення. Тому проблему з приростом трафіку необхідно

розглядати локально – в деяких суб'єктах не знадобиться нічого робити, а в інших необхідно буде добудовувати і оптимізувати мережу, наприклад ставити нові базові станції та перенаправляти трафік, пояснив аналітик.

Якщо до карантину в середньому абонент, активно користується мобільним інтернетом, споживав 5-6 ГБ трафіку на місяць, то зараз йому потрібно вже 10 ГБ або більше [13].

Існують кілька основних факторів, які і в світі, так і в Україні, що змушують мобільних операторів впроваджувати різні технології розвантаження мобільного трафіку. Це, перш за все, його різке зростання, який в чималому ступені обумовлений бумом продажів смартфонів і планшетних ПК. За оцінкою *J'son & Partners Consulting,* в 2011 році сумарний український трафік мобільної передачі даних досяг позначки в 247 петабайт (ПБ), збільшившись більш ніж в 3 рази в порівнянні з 2010 році, а за період з 2011 р 2019 роки в 12,8 разів – до 3 160 ПБ. При цьому частка мобільного трафіку в сукупному трафіку мобільної передачі даних складе 2-3%.

Проблема особливо актуальна для великих міст, в яких вже зараз, як показали результати польових досліджень незалежних випробувачів, спостерігається незадовільна якість зв'язку. Збільшення кількості смартфонів і планшетних ПК (в тому числі з вбудованим модемом 3*G* / 4*G*) і подальше за цим лавиноподібне наростання трафіку змушують операторів шукати спосіб розвантажити мережі, щоб голосові сервіси і сервіси передачі даних працювали з прийнятною якістю. Проблема особливо актуальна для великих міст, в яких вже зараз, як показали результати польових досліджень незалежних випробувачів, спостерігається незадовільна якість зв'язку. Збільшення кількості смартфонів і планшетних ПК (в тому числі з вбудованим модемом 3*G* / 4*G*) і подальше за цим лавиноподібне наростання трафіку змушують операторів шукати спосіб розвантажити мережі, щоб голосові сервіси і сервіси передачі даних працювали з прийнятною якістю. Проблема особливо актуальна для великих міст, в яких вже зараз, як показали результати польових досліджень незалежних випробувачів, спостерігається

незадовільна якість зв'язку. Збільшення кількості смартфонів і планшетних ПК (в тому числі з вбудованим модемом $3G$ / $4G$) і подальше за цим лавиноподібне наростання трафіку змушують операторів шукати спосіб розвантажити мережі, щоб голосові сервіси і сервіси передачі даних працювали з прийнятною якістю.

Зазвичай, метою зловмисників організуючих атаку, є блокування доступу до ресурсу. І хоча блокування ресурсу не так критичне, як втрата конфіденційної інформації, але вона зводить нанівець саму цінність ресурсу, оскільки жоден клієнт для якого цей ресурс створювався, не має можливості отримати до нього доступ. Як наслідок – втрата прибутку для власника ресурсу. Крім того *DoS* – атака може вивести ресурс з ладу (при використанні зловмисниками спеціальних програм або фрагментів програмного коду використовують уразливості в системі захисту) або ж бути тільки прикриттям, яке відверне службу інформаційної безпеки від атаки спрямованої на злом ресурсу.

Метою даного проєкту є розробка програмного комплексу, який буде виконувати облік та аналіз мережевого трафіку в мобільній мережі. Передбачається аналіз протоколів, окремих пакетів з можливістю декодування захоплених пакетів, відображення ієрархії вкладеноих пакетів і протоколів.

У відповідності до сформульованої мети у даному проєкті було поставлено наступні завдання:

– провести аналіз сучасного стану наукових досліджень у напрямку впровадження засобів обліку та аналізу трафіку мобільних мереж;

– розробити програмний комлекс з повним спекторм звітності для мережевих адміністраторів;

– розробити системний модуль перехвату, оцінки та аналізу трафіку мобільних мереж, який буде проводити аналіз протоколів передачі даних та розшифровувати пакети;

– розробити модулі графічного інтерфейсу користувачів.

Удосконалення методів оцінки та аналізу трафіку мобільних мереж передбачається за рахунок комбінування низьковрівневих програмних засобів і високошвидкісних інтерфесів користувачів, що дозволяє впроваджувати розроблений комплекс оцінки та моніторингу трафіку.

РОЗДІЛ 1
АНАЛІЗ МЕТОДІВ КОНТРОЛЮ МЕРЕЖЕВОГО МОБІЛЬНОГО ТРАФІКУ

1.1. Аналіз пакетного ядра *LTE* для реалізації мобільної мережі

Пакетне ядро *LTE* – це обладнання і програмне забезпечення, яке здійснює обробку та маршрутизацію трафіку всередині приватної мобільної мережі 4*G*. По суті, це центр управління мережею, через який взаємодіють всі базові станції. Ядро пов'язано з радіопідсистеми за допомогою стандартних інтерфейсів, тому може розглядатися як самостійний елемент мережі і самостійний продукт. Але, природно, при цьому воно повинно підтримувати рекомендації 3*GPP*.

В основу ядра приватної мережі 4*G* лягли напрацювання для загальнодоступних мереж (рис. 1.1).

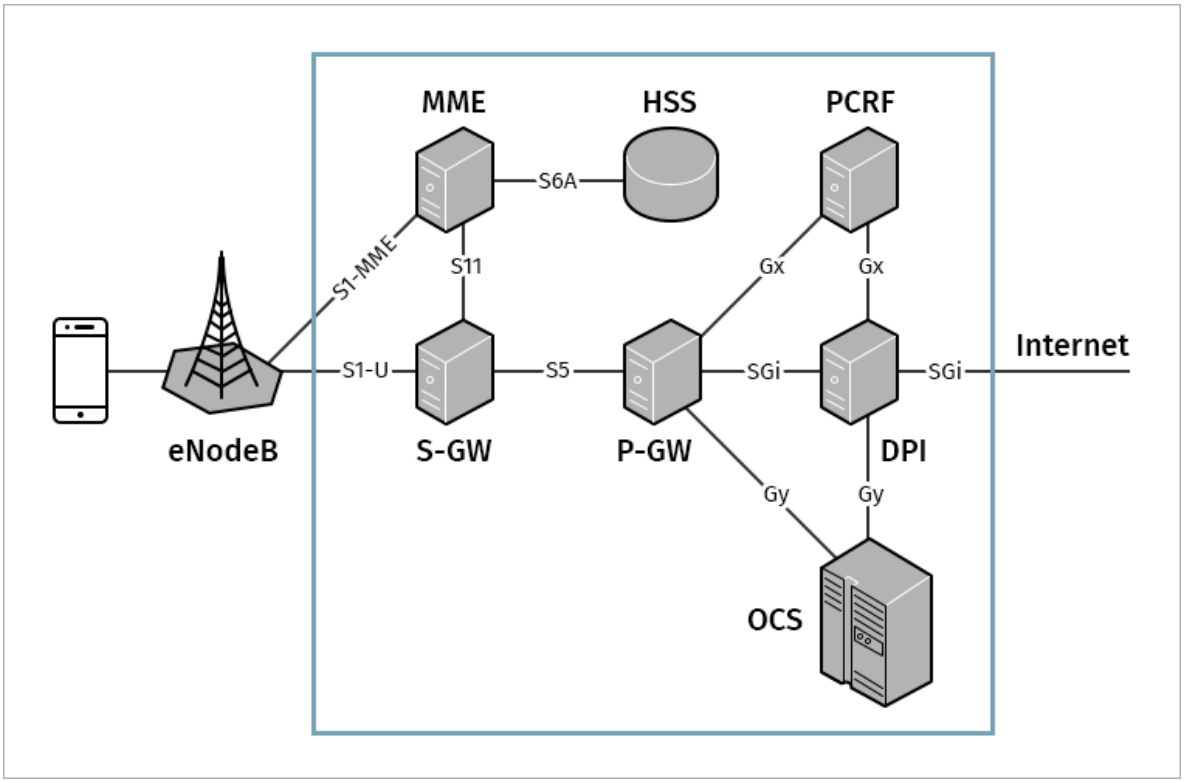


Рис. 1.1. Архітектура побудови мобільної мережі

Кафедра КСУ				НАУ 21 03 02 000 ПЗ			
Виконав	*Гусаков С.С.*			Аналіз небезпек мережевих атак і необхідність контролю мережевого трафіку	***Літера***	***Аркуш***	***Аркушів***
Керівник	*Марченко Н.Б.*				*Д*	*12*	*66*
Консульт.					СП 501Бз 123		
Норм. контр.	*Тупота Є.В.*						
Зав. Каф.	*Литвиненко О.Є.*						

Ядро мережі 4*G* має обробляти мільйони пакетів і сотні тисяч транзакцій в секунду. У цих умовах будь-яке звернення до пам'яті, будь-який додатковий виклик віртуальної функції вносить свій внесок в загальну продуктивність. Так що доводиться проводити глибокий рефакторинг коду. Початково проводиться синхронізація кешей *L*1-*L*2 ядер, яка впливала на швидкість доступу до оперативної пам'яті через вирівнювання структур даних.

Вичавивши розумний максимум з високорівневою оптимізації, ми перейшли до низькорівневим. Аналіз показав, що процесори чекають даних: *L*1 *missed*, показник *Cycles per Instruction* високий (4-20 тактів на інструкцію). Це змусило шукати рішення по превентивної завантаженні даних з пам'яті в кеш процесора.

В процесі роботи над ядром 4*G* доводиться враховувати, що кожне підприємство має власні специфічні вимоги і запити. Зокрема, багатьом потрібна гладка міграція від використовуваних аналогових і цифрових стандартів голосового зв'язку до приватної мережі нового покоління. Тому оператори реалізують взаємодію ядра з мережами попередніх поколінь. В окремих сегментах бізнесу існують власні вимоги регулятора, а також корпоративні правила, що стосуються внутрішнього зв'язку. Це також доводиться мати на увазі при розгортанні мережі (рис. 1.2).

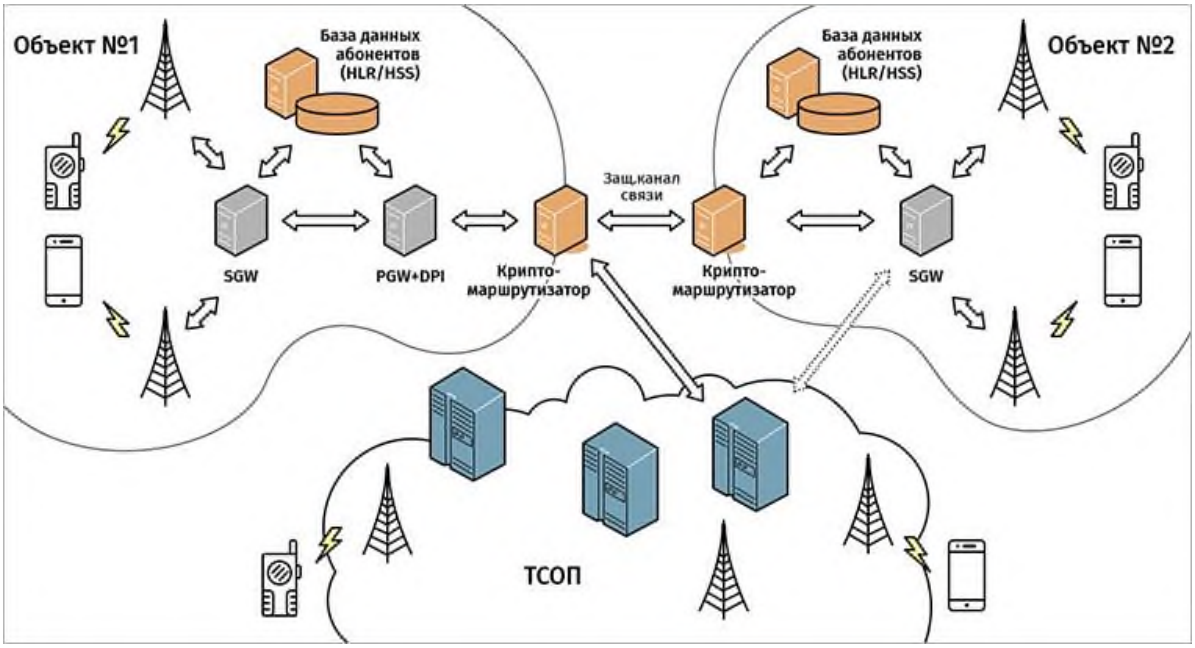


Рис. 1.2. Варіант побудови відомчої мережі *LTE* [9]

В цілому такі приватні вимоги і інтеграції серйозно ускладнюють життя нашим розробникам. Наш улюблений приклад на цю тему – історія з загальнодоступним мобільним оператором з далекої африканської країни. Хоча це трохи інший ринок, історія відмінно ілюструє ідею, не розкриваючи нічиїх корпоративних секретів. У тій мережі ми впровадили наш *PGW* (*Packet data network Gateway*). Він справлявся зі своїми функціями, поки один з абонентів не почав скаржитися на відсутність мобільного інтернету. Аналіз логів показав, що від абонента приходило цілих три стандартних запиту *Create Session Request* з різницею в кілька мілісекунд. Наша система обробляла ці запити відповідно до 3*GPP*, який, природно, не передбачає можливості такої відправки запитів. У підсумку все сесії абонента закривалися – мобільний інтернет відключався.

1.2. Аналіз необхідності розвантаження мобільних мереж

На початковому етапі поширення смартфонів (2005-2006 рр.) Мобільні оператори не були зацікавлені в розвитку *Wi-Fi* через побоювання в канібалізації сервісів на основі 3*G*. Тому великі оператори намагалися лобіювати свої інтереси, схиляючи виробників пристроїв не використовувати *Wi-Fi* в смартфонах або зменшувати можливості цієї технології в прошивці. З тих пір ситуація змінилася, і багато операторів активізують розгортання мереж *Wi-Fi*, в тому числі з метою розвантаження мобільного трафіку. Одним з основних драйверів зростання популярності *Wi-Fi* став високий попит на цю технологію з боку користувачів. У багатьох випадках мережі 3*G* не забезпечували достатнього рівня сигналу всередині приміщень, і в цьому випадку хорошою альтернативою став *Wi-Fi*-доступ. Відновленню використання цієї технології в чималому ступені сприяли поява і швидке зростання популярності *iPhone*. Бізнес-стратегії найбільших операторів і вендорів щодо *Wi-Fi* були переглянуті:

– Компанія *AT & T* придбала в 2008 р оператора хот-спотів *Wayport* за $ 275 млн, що є важливим етапом в стратегії *AT & T* за експансії оператора на ринок публічного *Wi-Fi* доступу;

– у 2011 р *Qualcomm* придбала компанію *Atheros* за $ 3 млрд. На думку лідера ринку чіпсетів, *Wi-Fi* повинен бути в кожному смартфоні;

– з 2016-го року у операторів всього світу набуває поширення *SIM*-аутентифікація абонентів для отримання доступу до *Wi-Fi*. Таку процедуру впровадили*T-Mobile, Orange, SK Telecom, Korea Telecom, PCCW, China Mobile, China Telecom* та ін.

Популярність використання *Wi-Fi* серед користувачів смартфонів зростає з року в рік і до теперішнього часу досягла істотних величин. Наприклад, в Південній Кореї споживання трафіку через *Wi-Fi* (за часом перебування в мережі і обсягом трафіку) становить близько 50%. У Великобританії, за даними *Bango,* через *Wi-Fi* відбувається приблизно половина всіх мобільних веб-сесій через смартфони. У *Wi-Fi* мережі хот-спотів компанії *AT & T* (США) більше 70% з'єднань ініційовано зі смартфонів, до кінця 2 кварталу 2011 р кількість таких сесій наблизилося до позначки 250 млн. За даними *Wireless Broadband Alliance* і *Informa Telecoms & Media,* в листопаді 2018 року обсяг трафіку *Wi-Fi* при підключенні зі смартфонів вперше перевищив трафік *Wi-Fi* з ноутбуків (40% і 39% від сукупного *Wi-Fi* трафіку, відповідно). При цьому частка планшетних комп'ютерів виявилася істотно нижчою і склала 17%. Основною перевагою використання *Wi-Fi* для розвантаження трафіку є висока поширеність даної технології в світі. Так, за оцінкою *Strategy Analytics*, чверть всіх домогосподарств в світі в даний час має розгорнуті мережі *Wi-Fi,* в 2021 р їх кількість досягне 42%. У країнах-лідерах (Південна Корея, Великобританія і Німеччина) цей показник перевищує 70%. В Україні проникнення *Wi-Fi* в приватному секторі нижче середньосвітового рівня – 22,9%. При цьому швидкість передачі даних в мережах «домашнього» *Wi-Fi*, за даними *Carrypad.com*, лише незначно нижче, ніж в мережах *LTE,* а за часом затримки (*latency*) *Wi-Fi* навіть виграє у *LTE*. Основною перевагою використання *Wi-Fi* для розвантаження трафіку є висока поширеність даної технології в світі. Так, за оцінкою *Strategy Analytics*, чверть всіх домогосподарств в світі в даний час має розгорнуті мережі *Wi-Fi,* в 2021 р їх кількість досягне 42%. У країнах-лідерах (Південна Корея, Великобританія і Німеччина) цей показник перевищує 70%. В Україні проникнення *Wi-Fi* в

приватному секторі нижче середньосвітового рівня – 22,9%. При цьому швидкість передачі даних в мережах «домашнього» *Wi-Fi*, за даними *Carrypad.com*, лише незначно нижче, ніж в мережах *LTE*, а за часом затримки (*latency*) *Wi-Fi* навіть виграє у *LTE* (табл. 1.1).

Таблиця 1.1.

Порівняння результатів випробувань технології *LTE* і домашнього *Wi-Fi* з'єднання

Параметр	*Wi-Fi*	4*G LTE*
Смуга пропуску	13,6 Мбіт/с / 4,2 Мбіт/с (*DL* / передача)	20,9 Мбіт/с / 5,4 Мбіт/с (*DL* / передача)
Затримка	32 мс	65 мс
Коливання часу затримки (джиттер)	30 мс	10 мс
Відсоток втрачених даних за період дослідження	0%	0%

Крім того, стандарт *Wi-Fi* широко поширений в готелях, на підприємствах громадського харчування, на громадському наземному транспорті, в метро, в аеропортах і залізничних вокзалах тощо. Згідно з дослідженням, проведеним асоціацією *Association of Corporate Travel Executives* (США), 80% менеджерів корпорацій вважають наявність *Wi-Fi* доступу головним критерієм при виборі готелю.

Іншими перевагами *Wi-Fi* є невисока вартість розгортання і обслуговування (в порівнянні з мережами 3*G* / 4*G*), висока швидкість передачі даних, робота в неліцензованому спектрі в більшості країн.

Разом з тим, *Wi-Fi* притаманні і ряд недоліків (табл. 1.2). Зокрема, дана технологія характеризується обмеженою потужністю, сильними перешкодами і складним управлінням безпекою та ін. Крім того, в даний час в більшості випадків (за винятком декількох впроваджень *SIM*-аутентифікації) доступ до

точки *Wi-Fi* для користувачів смартфонів незручний: потрібно ввести пароль , а в деяких випадках пройти реєстрацію та ін.

Таблиця 1.2.

Переваги і недоліки технології *Wi-Fi* для розвантаження мобільного трафіку

Переваги	Недоліки
Висока поширеність технології *Wi-Fi*, велика частка пристроїв з підтримкою *Wi-Fi*	Обмежена потужність, сильні перешкоди (інтерференція)
Висока швидкість передачі даних	Складні управління безпекою
Низька вартість розгортання і обслуговування в порівнянні з мережами 3*G* / 4*G*	Незручно для користувачів (потрібне введення пароля)
Не потрібне ліцензування спектру	Мало смартфонів з підтримкою *IEEE* 802.11*a*

Але незважаючи на існуючі недоліки *Wi-Fi,* мобільні оператори активно впроваджують цю технологію для розвантаження трафіку. Найбільша кількість точок доступу *Wi-Fi* в світі зафіксовано у оператора *Free* (Франція): 4 млн. При цьому співвідношення базових станцій і точок доступу *Wi-Fi* у цій компанії складає 1:4000. У п'ятірку лідерів входять також *China Mobile* (2,83 млн), *Softbank* (0,27 млн), *KDDI* і *Korea Telecom* (*KT*) (по 0,1 млн).

*Я*понському операторові *KDDI* вдалося через *Wi-Fi* і *WiMAX* розвантажити в годину пік в червні 2012 р 32% трафіку даних зі смартфонів. Цільовий показник на березень 2013 року – 50% розвантажувати трафіку. В цілому оператори використовують різні стратегії розвантаження мобільного трафіку. Наприклад, *Softbank Mobile* активно використовує для цієї мети технології *Wi-Fi* і фемтосетей, комбінуючи їх з розвантаженням в діапазон 1,5 *ГГц*, в якому у оператора працює мережа *DS-HSDPA*. *France Telecom Orange* робить акцент на існуючі *Wi-Fi*-мережі (домашні і публічні) і на спрощення аутентифікації користувачів. *AT&T,* володіючи досить масштабними мережами *Wi-Fi* і Фемтостільників, комбінує платний і безкоштовний доступ по *Wi-Fi,* а

користувачам пакетів послуг надає значні знижки на обладнання фемтосот (точку доступу). *Verizon* на відміну від свого конкурента *AT & T* не використовує широко *Wi-Fi* для розвантаження мобільного трафіку. Оператор вважає, що *Wi-Fi* має обмежене застосування в сценаріях для розвантаження трафіку макромережі і робить акцент на Фемтостільників.

Основні тренди по використанню розвантаження мобільного трафіку за допомогою *Wi-Fi* – це впровадження *SIM*-аутентифікації (не потрібно вводити пароль) і інтеграція *Wi-Fi* в мережі операторів мобільного зв'язку.

Як показано вище, найбільші оператори стільникового зв'язку в світі швидко нарощують кількість *Wi-Fi* хот-спотів з метою розвантаження мереж стільникового зв'язку. Розгортання мереж *Wi-Fi* більш ефективно з економічної точки зору, ніж розширення мереж стільникового зв'язку. При цьому більшість операторів не стягує плату з абонентів за *Wi-Fi*-доступ, тим самим збільшуючи лояльність користувачів і знижуючи їх відтік.

Обмежуючи або тарифікуємо *Wi-Fi*-трафік, оператори ризикують зменшити лояльність користувачів, підштовхуючи їх до ручного вибору безкоштовних публічних точок доступу *Wi-Fi*, де це можливо. В Україні ринки *Wi-Fi* і фемтосот в порівнянні з розвиненими країнами розвинені дуже слабо. Одна з причин пов'язана з проблемами регулювання. Так, з січня 2012 року набрав чинності наказ Роскомнадзора, який встановлює плату за спектр виходячи з нової методики, розробленої Мінкомзв'язку. Ця методика не відносить *Wi-Fi* до «перспективних технологій», тому плата за частоти для *Wi-Fi* встановлена більш висока.

Висока плата за радіочастотний спектр і складність отримання частот для розгортання зовнішніх (вуличних) точок доступу *Wi-Fi* є основними стримуючими факторами розвитку масштабних проектів *Wi-Fi* в Україні.

Відсутність чіткого розуміння у абонентів переваг використання фемтосот – другий за популярністю технології розвантаження трафіку – є одним з основних бар'єрів для більш широкого її застосування. В Україні використання фемтосот також стримується регуляторними проблемами. Втім, поступово українське галузеве законодавство адаптується до реалій ринку. Так,

держкомісія з радіочастот (ГКРЧ) на найближчому засіданні 19 грудня планує дозволити використовувати фемтосоти в мережах *GSM* (900 і 1800 МГц) і *LTE* (791-862 МГц і 2,5-2,69 ГГц) додатково до діючого дозволу на фемтосоти 3*G* (1,9-2,1 ГГц).

За оцінкою *J'son&Partners Consulting* [13], до теперішнього часу оператори "великої трійки" встановили менше 2 тис. Зареєстрованих фемтостанцій. У США, на найбільшому ринку фемтосот, діє приблизно 1,5-2 млн таких пристроїв – більше половини всіх встановлених фемтосот в світі. В оптимістичному сценарії *J'son&Partners* очікує до 1 млн діючих фемтосот в Україні до кінця 2021 р.

1.3. Вплив *DoS*–атак на збільшення мобільного трафіку

Використання SNMP для цілей спостереження

Моніторинг значною мірою покладається на свій основний протокол - Простий протокол управління мережею (SNMP). Коротко розширимо термін.

SNMP - це протокол прикладного рівня, визначений Радою архітектури Інтернету (IAB) у RFC1157 для обміну інформацією управління між мережевими пристроями. Він є частиною набору протоколів Transmission Control Protocol / Internet Protocol (TCP / IP). SNMP є одним із широко прийнятих протоколів, що використовуються для управління та моніторингу елементів мережі. Більшість професійних мережевих елементів постачаються в комплекті з агентом SNMP. Ці агенти повинні бути ввімкненими та налаштованими для взаємодії із Системою управління мережею (NMS).

Основні компоненти SNMP та його функції:

Архітектура SNMP складається з:

1) Менеджер SNMP
2) Керований пристрій
3) Агент SNMP
4) Бази даних управлінської інформації (інакше відомі як бази управлінської інформації або MIB)

5) Менеджер SNMP - (Зазвичай Система управління мережею - NMS) спілкується з кількома агентами SNMP, реалізованими в мережі.
6)
7) Керований пристрій - або елемент мережі - це частина мережі, яка вимагає певної форми моніторингу та управління, наприклад, маршрутизатори, комутатори, сервери, робочі станції, принтери, ДБЖ тощо.
8) Агент SNMP- це програма, яка входить до складу керованого пристрою. Увімкнення цього агента дозволяє йому локально збирати базу даних управління інформацією з пристрою, щоб зробити її доступною менеджеру SNMP за запитом. Ці агенти можуть бути стандартними (наприклад, Net-SNMP) або специфічними для постачальника (наприклад, агент HP Insight)

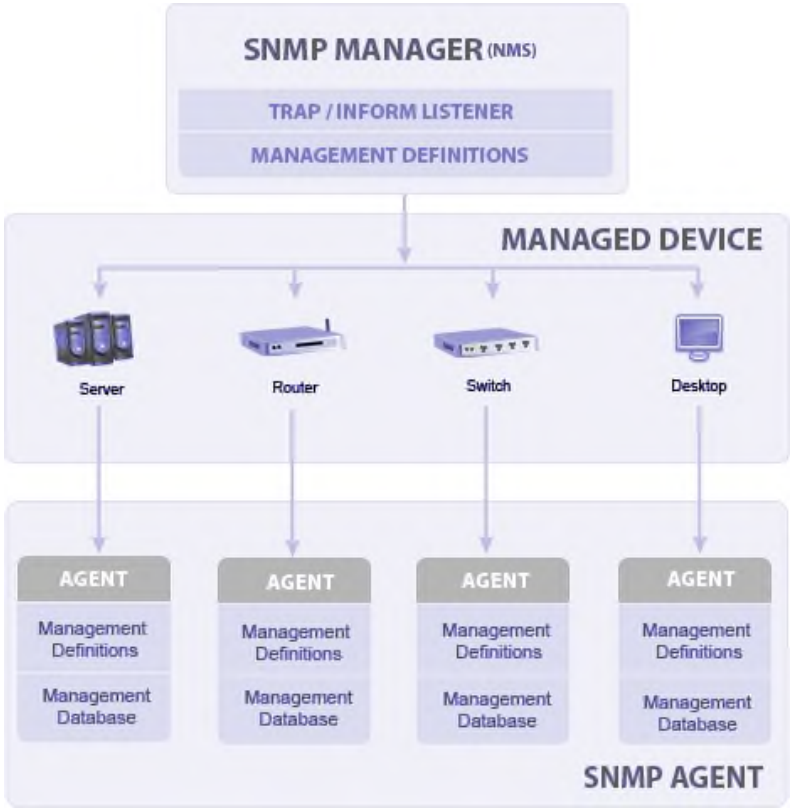


Рис. 1.2. Принцип роботи SNMP

Рис. 1.2. Принцип роботи SNMP

9) Інформаційна база управління / база даних- Загальнодоступна база даних між Агентом і Менеджером називається Управлінська інформаційна база (MIB).

Коротше кажучи, файли MIB - це набір питань, які менеджер SNMP може задати агенту. Агент збирає ці дані локально і зберігає їх, як визначено в MIB.

MIB містять стандартний набір статистичних та контрольних значень, визначених для керованих пристроїв у мережі. Протокол SNMP також дозволяє розширити ці стандартні значення зі значеннями, характерними для конкретного агента, за допомогою приватних MIB. Тож менеджер SNMP повинен знати про ці стандартні та приватні питання для кожного типу Агента.

1) Функції SNMP

- snmp_get_quick_print - Отримує поточне значення параметра швидкого друку бібліотеки UCD
- snmp_get_valueretrieval - Поверніть метод, як будуть повертатися значення SNMP
- snmp_read_mib - Зчитує та аналізує файл MIB в активному дереві MIB
- snmp_set_enum_print - Повертає всі значення, які є переліченнями, із значенням перерахування замість необробленого цілого числа
- snmp_set_oid_numeric_print - Повернути всі об'єкти, включаючи їх відповідний ідентифікатор об'єкта в межах вказаного
- snmp_set_oid_output_format - Встановіть вихідний формат OID
- snmp_set_quick_print - Встановіть значення quick_print в бібліотеці UCD SNMP
- snmp_set_valueretrieval - Вкажіть метод повернення значень SNMP
- snmp2_get - Отримати об’єкт SNMP
- snmp2_getnext - Виберіть об’єкт SNMP, який слідує за вказаним ідентифікатором об’єкта
- snmp2_real_walk - Повернути всі об'єкти, включаючи їх відповідний ідентифікатор об'єкта, у межах вказаного
- snmp2_set - Встановіть значення об'єкта SNMP
- snmp2_walk - Отримати всі об’єкти SNMP від агента
- snmp3_get - Отримати об’єкт SNMP

- snmp3_getnext - Виберіть об’єкт SNMP, який слідує за вказаним ідентифікатором об’єкта
- snmp3_real_walk - Повернути всі об'єкти, включаючи їх відповідний ідентифікатор об'єкта, у межах вказаного
- snmp3_set - Опис
- snmp3_walk - Отримати всі об’єкти SNMP від агента
- snmpget - Отримати об’єкт SNMP
- snmpgetnext - Виберіть об’єкт SNMP, який слідує за вказаним ідентифікатором об’єкта
- snmprealwalk - Повернути всі об'єкти, включаючи їх відповідний ідентифікатор об'єкта, у межах вказаного
- snmpset - Встановіть значення об'єкта SNMP
- snmpwalk - Отримати всі об’єкти SNMP від агента
- snmpwalkoid - Запит дерева інформації про сутність мережі

Використовуючи ці функції, пристрої керування можуть працювати з будь-якими об'єктами до рівня MAC-адреси. Іншими словами, якщо ваш тостер підключений до мережі, ви отримаєте сигнал, коли тости будуть готові.

Рішення системи управління від CA

Unicenter NSM дозволяє організаціям розгортати та оптимізувати складну, безпечну та надійну інфраструктуру, яка підтримує бізнес-цілі. Крім того, це допомагає забезпечити безперервний стан здоров'я та працездатність критичної інфраструктури за допомогою інноваційних та інтелектуальних методів, що дозволяють організаціям контролювати витрати, зберігаючи або підвищуючи реакцію на зміну пріоритетів бізнесу.

Потужність Unicenter NSM походить від надзвичайно гнучкого підходу до управління системами, розширеного завдяки широкій інтеграції з іншими продуктами в асортименті ЦС, а також із сторонніми додатками для створення рішень, які обмінюються інформацією через загальну структуру подій та спільний інтерфейсний портал . Це забезпечує потужну перевірку бачення CA EITM.

Рішення створено для задоволення таких потреб:

Мінімізація ризиків

10) Підвищення операційної ефективності та зменшення ризику завдяки можливостям самокерування та самовідновлення
11) Забезпечення рольового доступу до систем страхування відповідності
12) Підтримка високої готовності для повних можливостей відмов

Спрощення управління та зниження витрат

13) Розгортання, конфігурація та управління серверними ресурсами швидко та з постійними результатами
14) Оцініть та плануйте оновлення серверів, міграції та консолідацію
15) Спрощення управління серверами за допомогою моделювання серверів та сервісно-метричного аналізу

Покращення сервісу

16) Виявляйте деградацію перед масштабними проблемами або перебоями
17) Об’єднайте різні системи моніторингу та управління ефективністю в єдине рішення
18) Діагностуйте та ізолюйте повторювані проблеми за допомогою даних про ефективність в реальному часі та в реальному часі
19) Переглядайте елементи з точки зору додатків, щоб узгодити мережеву та системну інфраструктуру з бізнес-пріоритетами.
20) Скористайтеся багатою аналітикою та звітами для оптимізації та планування потужності
21) Розмістіть пріоритети на подіях та інцидентах на основі впливу на бізнес

Якщо автоматизація прискорює рутинні завдання, то самоуправління забезпечує негайну реакцію на незаплановані події шляхом адаптації до мінливих умов.

Ефективне самоуправління вимагає добре керованої інфраструктури, перш ніж воно зможе розпочати свою магію. Навколишнє середовище повинно забезпечувати моніторинг у режимі реального часу та автоматичну облицювання основи, щоб швидко та точно виявляти ненормальні ситуації. Повідомлення

повинно бути негайним, незалежно від того, чи проблемою можна керувати самостійно, чи вимагатиме втручання. Для самокерування також потрібна добре зрозуміла інфраструктура, де всі елементи ідентифіковані та зіставлені з бізнес-процесами, які вони підтримують.

Деякі приклади самокерованих завдань включають:

22) Процес не вдається на сервері. Технологія агента виявляє несправність і перезапускає цей конкретний процес на основі встановленого правила.

23) Ви щойно отримали нові активи конкретного типу, і тепер ви несете відповідальність за моніторинг їхніх нових додатків та послуг, але незрозуміло, за якими показниками слід слідкувати або за якими пороговими показниками. Цю проблему слід вирішити, запросивши агента перевірити сервер, розробити список ресурсів для моніторингу та розпочати моніторинг протягом декількох хвилин. З часом системи повинні налаштовуватися на точність і автоматично встановлювати стандартні базові показники споживання для ресурсів, що контролюються.

24) Сервер має проблему, і квиток підтримки автоматично генерується, зменшуючи час, необхідний людині, щоб розпізнати проблему та сформувати білет проблеми.

Опис рішення

Unicenter NSM забезпечує роботу, доступність та ефективність роботи інфраструктури. Він постійно оцінює та надає можливості самокерування серверам. Він підтримує важливі бізнес-процеси, керуючи базовою інфраструктурою та визначаючи пріоритети питань на основі їх впливу на бізнес. Завдяки Unicenter NSM вирішення проблем прискорюється, а рівні обслуговування покращуються - і все це при одночасному зниженні витрат на підтримку. Unicenter NSM також забезпечує централізовану та уніфіковану платформу управління для управління різнорідними IT-інфраструктурами, значно зменшуючи витрати та складність управління інфраструктурою, зберігаючи поточні інвестиції.

Unicenter NSM:

25) Централізує управління всією IT-інфраструктурою за допомогою єдиного стандартного інтерфейсу користувача, спрощуючи обробку складних та різнорідних середовищ

26) Використовує спільну базу даних управління, орієнтовану на активи, щоб колективні знання могли бути видобуті та реалізовані різними спеціальностями управління

27) Веде каталог IT-активів з постійним відкриттям

Як описано в розділі SNMP, все рішення базується на архітектурі агента-менеджера. CA має власний розподільник агента під назвою TNG, який виконує основні функціональні можливості зв'язку SNMP між вузлами. Існують різні клієнти, які дозволяють контролювати Windows, UNIX, мережеві та вузли зберігання. Також існує певний рівень підтримки служб кластеризації. Варто пам’ятати, що ціле рішення є дуже гнучким і базується на конфігурації агента як найменшого зерна. Це означає, що ви можете легко налаштувати параметри контролю кожного окремого вузла лише для цього вузла. Беручи до уваги масштаби рішення на всьому підприємстві, це може мати велике значення.

2d-карта CA Unicenter - це один із продуктів, який поставляється в ряд із рішенням для управління системами. Це схоже на плату з вузлами, які представляють різні системи, починаючи від кластерів серверів, вузлів до мережевих принтерів та сховищ.

На наступному малюнку показано, як це виглядає:

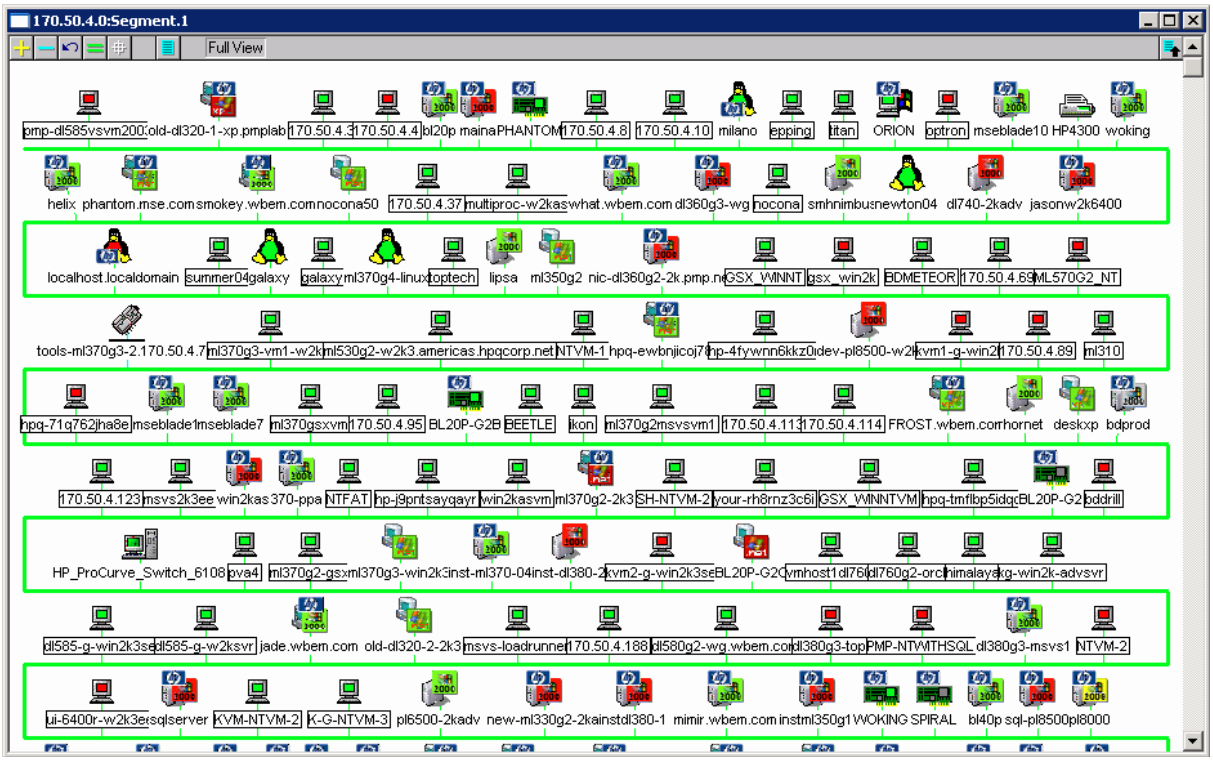


Рис. 1.3. Огляд 2d-карти CA Unicenter

Піктограми показують тип вузла, а зв'язки між ними відображають мережеве підключення, тому легко керувати мережами та офісними локальними мережами. Я працював із цим рішенням приблизно два роки, і мушу визнати, що система дуже гнучка і не така складна в налаштуванні. Що стосується недоліків, то їх можна вважати дещо старими; він не підтримує деяких важливих питань щодо моніторингу мережі та кластерів. Це була основна причина, чому наша компанія вирішила перейти на програмне забезпечення HP ITSM.

1.1.1. ITIL та HP ITSM

Перш ніж зупинитися на програмному забезпеченні HP ITSM, нам доведеться знати, про що йде мова в ITIL, оскільки на цьому побудовано ціле рішення.

ITIL - це загальнодоступна структура, яка описує найкращі практики управління IT-послугами. Він забезпечує основу для управління IT, "обгортку послуг", і зосереджує увагу на постійному вимірюванні та поліпшенні якості наданих IT-послуг, як з точки зору бізнесу, так і з боку клієнтів. Цей фокус є головним фактором світового успіху ITIL і сприяв його плідному використанню

та ключовим перевагам, отриманим тими організаціями, що застосовують методи та процеси у своїх організаціях.

Деякі з цих переваг включають:

28) підвищена задоволеність користувачів та споживачів IT-послугами
29) покращена доступність послуг, що безпосередньо призводить до збільшення прибутку та прибутку бізнесу
30) фінансова економія завдяки зменшенню переробки, втраченому часу, покращеному управлінню ресурсами та їх використанню
31) покращений час випуску на ринок нових товарів та послуг
32) Покращене прийняття рішень та оптимізований ризик.

ITIL був опублікований між 1989 і 1995 рр. Канцелярським бюро Її Величності (HMSO) у Великобританії від імені Центрального агентства зв'язку та телекомунікацій (CCTA), яке зараз входить до складу Управління урядової торгівлі (OGC). Його раннє використання було в основному обмежено Великобританією та Нідерландами. Друга версія ITIL була опублікована як набір перероблених книг між 2000 і 2004 роками.

Початкова версія ITIL складалася з бібліотеки з 31 супутньої книги, що охоплює всі аспекти надання IT-послуг. Потім цю початкову версію було переглянуто та замінено на сім, більш тісно пов'язаних та послідовних книг (ITIL V2), об'єднаних у загальних рамках. Ця друга версія стала загальновизнаною і зараз використовується в багатьох країнах тисячами організацій як основа для ефективного надання IT-послуг. У 2007 році ITIL V2 був замінений вдосконаленою та консолідованою третьою версією ITIL, що складається з п'яти основних книг, що охоплюють життєвий цикл служби, разом з офіційним вступом.

П'ять основних книг охоплюють кожну стадію життєвого циклу служби (рис. 1), починаючи від початкового визначення та аналізу бізнес-вимог у стратегії обслуговування та дизайну послуг, через міграцію в середовище реального часу в рамках переходу до служби, до функціонування та вдосконалення роботи служби та постійне вдосконалення послуг.

Рис. 1.4. Структура ITIL

Інтелектуальна інформація ITIL була розроблена для задоволення відомих критеріїв SMART:

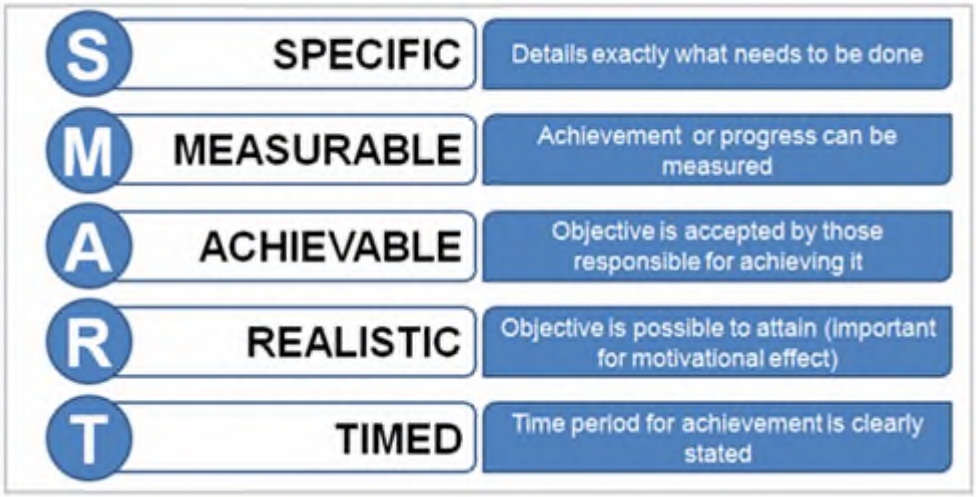


Рис. 1.5. Пояснення критеріїв SMART

Основна служба ITIL служить для наступних операцій:

- Управління рівнем обслуговування (SLM)

SLM узгоджує, узгоджує та документує відповідні цілі IT-послуг з бізнесом, а потім відстежує та видає звіти про доставку відповідно до узгодженого рівня обслуговування.

Мета процесу УУЗР полягає у забезпеченні того, щоб усі оперативні послуги та їх ефективність послідовно, професійно вимірювались у всій IT-організації, а також щоб послуги та вироблені звіти відповідали потребам бізнесу та клієнтів.

Основна інформація, що надається процесом УУЗР, включає угоди про рівень обслуговування (SLA), угоди про операційний рівень (OLA) та інші угоди про підтримку, а також підготовку Плану вдосконалення послуг (SIP) та Плану якості послуг.

- Управління потужністю

Управління потужністю включає управління бізнесом, послугами та компонентами протягом усього життєвого циклу. Ключовим фактором успіху в управлінні потенціалом є забезпечення його врахування на етапі проектування. Метою управління потенціалом є забезпечення фокусу та управління для всіх питань, що стосуються потенціалу та ефективності, що стосуються як послуг, так і ресурсів, а також відповідність можливостей IT до узгоджених вимог бізнесу.

Інформаційна система управління потенціалом (CMIS) є наріжним каменем успішного процесу управління потенціалом. Інформація, що міститься в CMIS зберігається та аналізується усіма підпроцесами управління потенціалом для надання технічних та управлінських звітів, включаючи план потужності.

- Управління доступністю

Мета управління доступністю полягає у забезпеченні фокусу та управління всіма проблемами, пов'язаними з доступністю, що стосуються послуг, компонентів та ресурсів, гарантування того, що цільові показники доступності у всіх сферах вимірюються та досягаються, і що вони відповідають або перевищують поточні та майбутні узгоджені потреби бізнесу рентабельним способом.

Управління доступністю повинно проходити на двох взаємопов’язаних рівнях і мати на меті постійну оптимізацію та попереджувальне покращення доступності IT-послуг та організації, що їх підтримує.

Є два ключові аспекти:

33) реактивна діяльність: моніторинг, вимірювання, аналіз та управління подіями, інцидентами та проблемами, пов'язаними з недоступністю послуг

34) ініціативна діяльність: попереджувальне планування, дизайн, рекомендації та покращення доступності. Діяльність з управління доступністю повинна враховувати наявність, надійність, ремонтопридатність та зручність обслуговування як на рівні обслуговування, так і на рівні компонентів, особливо тих, що підтримують життєво важливі бізнес-функції (VBF).

Процес управління доступністю повинен базуватися на Інформаційній системі (AMIS), яка містить усі виміри та інформацію, необхідну для надання відповідної інформації бізнесу на рівнях обслуговування. AMIS також допомагає у розробці Плану доступності.

Тепер ми підійшли до опису сімейства програм HP OpenView.

HP працює у сфері IT-обслуговування вже майже десять років і пропонує широкий спектр програмного забезпечення, орієнтованого на підприємства:

1) Менеджер вузлів HP OpenView (OV NNM)
2) HP Operations Manager (OM) - моніторинг систем та додатків за допомогою агентів
3) HP OMW - Operations Manager (Windows) (раніше OVOW, раніше VantagePoint Operations для Windows)
4) HP OMU - Operations Manager (Unix) (раніше OVOU, раніше VantagePoint Operations для Unix, іноді згадується як ITO)
5) HP OpenView ServiceCenter (раніше Peregrine ServiceCenter) - тепер HP Software Service Manager
6) HP OpenView AssetCenter (раніше Peregrine AssetCenter)
7) HP OpenView Connect-It - інструмент інтеграції даних та процесів
8) HP OpenView SOA Manager

9) Програмне забезпечення HP Universal CMDB (uCMDB)

10) Управління корпоративними системами (ESM)

Я маю досвід роботи з деякими з цих систем, і найкраще в них - це основна цілісність їх та глобальний масштаб. Вони мають великий потенціал ERP-систем, оскільки це програмне забезпечення дозволяє збирати всю інформацію про IT-інфраструктуру, наприклад, до ліцензування.

HP ITSM поєднує світовий досвід консультантів HP Services із перевіреними продуктами та партнерами HP OpenView. HP ITSM допомагає отримати максимальну віддачу від IT завдяки впровадженню процесів та інструментів для надання та підтримки важливих для бізнесу IT-послуг. До того ж, зосередившись на ефективності IT-процесів, управлінні послугами та безперервності, ви можете бути впевнені, що ваша компанія отримає потрібну інформацію потрібним людям у потрібний час.

Служби HP ITSM мають неперевершені можливості, які допоможуть вам досягти цілей управління послугами.

Одна з найбільших та найбільш обізнаних працівників, сертифікованих ITIL / ITSM, з широкими глобальними можливостями:

35) Понад 80 освітніх центрів у всьому світі, які є уповноваженими центрами іспитів ITIL

36) 35 центрів підтримки

37) Більше 1000 спеціалістів із сертифікованої підтримки фонду ITIL

38) Понад 80 критично важливих для бізнесу консультантів, що спеціалізуються на оперативній підтримці ITCM, 2500 співробітників, які займаються бізнес-сервісом та

рішення для управління інфраструктурою

39) Досвід навчання понад 80 000 IT-фахівців рішенням ITIL / ITSM та HP OpenView

40) Компанія HP GlobalSoft Ltd. була однією з перших у світі компаній, яка отримала сертифікацію BS 15000, що є стандартом якості в ITSM.

Постійне дотримання HP стандартами такі як ITIL та підтримка групи користувачів, Форум ITSM (itSMF):

41) Глобальний член itSMF
42) Член правління SMF USA
43) Член-засновник розділів SMF в США, Канаді, Японії, Індії, Сінгапурі, Угорщині та Польщі
44) Бере участь у написанні та рецензуванні книг ITIL
45) Бере участь у розробці схеми сертифікації itSMF BS 15000
46) HP / Compaq були як провідними авторами, так і розробниками ITIЛ Microsoft Operations Framework (MOF)
47) ITIL є основою для еталонної моделі ITSM від HP.

IT-організації, які прагнуть підвищити свою операційну ефективність, надаючи та підтримуючи свої IT-послуги та узгоджуючи ці послуги зі своїми бізнес-цілями, звернулися до Бібліотеки IT-інфраструктури (ITIЛ) за рекомендацією. ITIL забезпечує основу визначень високого рівня для процесів управління IT-послугами (ITSM).

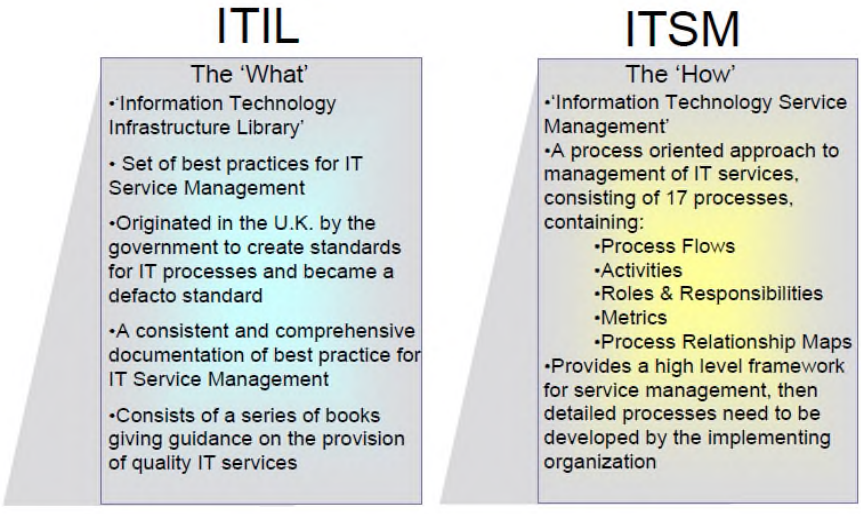


Рис. 1.6 Поняття ITIL та ITSM - "Що" та "Як"

Впроваджуючи рішення ITSM, засноване на ITIL, IT-організації часто стикаються з довгими та дорогими циклами визначення процесів, а також з внутрішнім опором "новому винаходу колеса". Крім того, численні ітерації, необхідні для узгодження процесу організації, що дозволяє конфігурацію

технології, з її процесами, процедурами та робочими інструкціями, як правило, збільшують витрати та зусилля щодо впровадження та подовжують терміни.

Найкращі практики HP ITSM для HP OpenView Service Desk підтримують спрямований дизайнерський підхід до найкращих практик, надаючи заздалегідь визначену, попередньо налаштовану основу для вашої реалізації HP OpenView Service Desk.

Це передове рішення найкращих практик включає:

• Довідкова модель HP ITSM

• П'ять заздалегідь визначених процесів управління послугами, укомплектовані детальними процедурами та робочими інструкціями, представлені в інтуїтивно зрозумілому веб-форматі

• Управління інцидентами та запитами на обслуговування

• Управління проблемами

•Управління конфігурацією (включаючи персонал та організацію)

•Управління змінами

• Управління на рівні обслуговування

•Попередньо налаштована база даних HP OpenView Service Desk із системними налаштуваннями, що підтримують модель

•Посібник із встановлення, який містить інструкції щодо модифікації HTML-сторінок, що спрощує моделювання процесів, процедур та робочих інструкцій

Компоненти моделі ITIL (інциденти, зміни завдань, проблеми, запити) є основою сервісного центру HP OpenView. Це велике рішення для управління квитками, яке використовується для IT-інфраструктури.

Це вікно входу в Центр обслуговування:

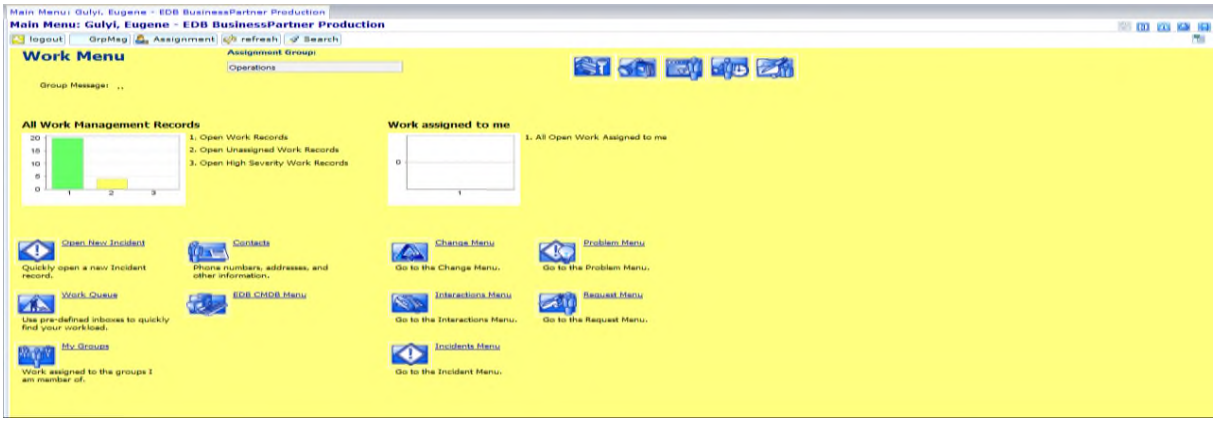
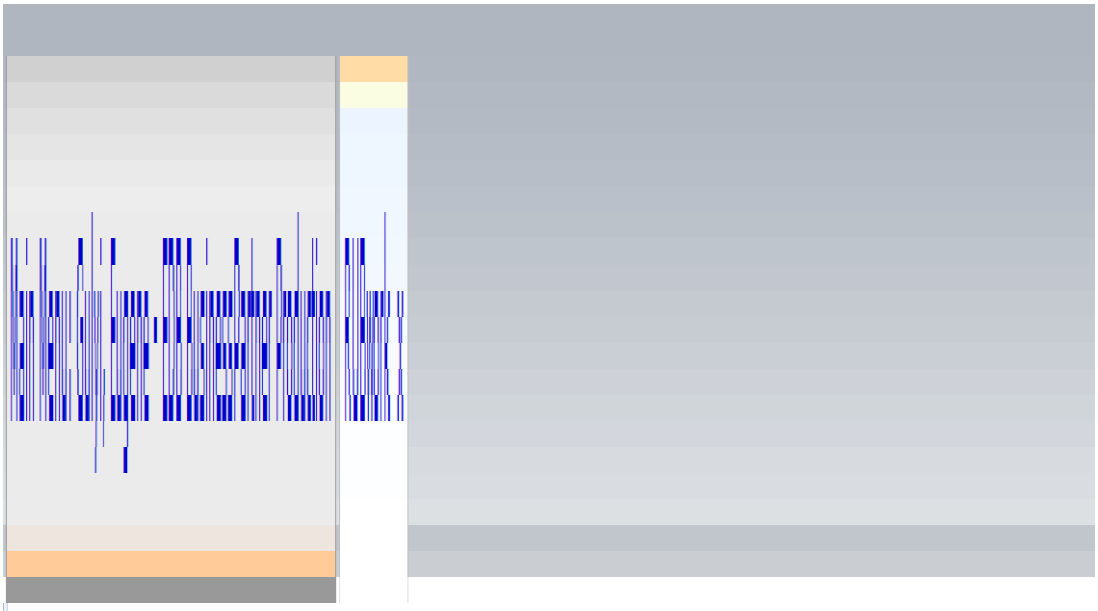
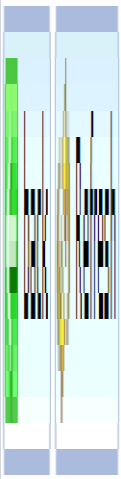


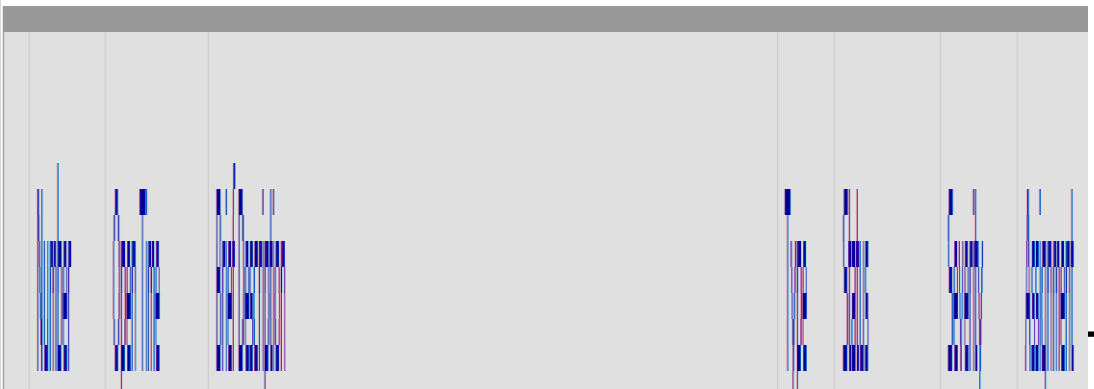
Рисунок 1.7. Система обробки квитків сервісного центру HP OpenView

Припустимо, що я людина, яка виконує оперативні роботи з клієнтами. У мене є моя група присвоєння (яка є основою розподілу прав та оперативної відповідальності), яка називається “Операції”. Записи управління роботою (ліва схема) представляють існуючу чергу роботи моєї групи завдань. За задумом існують 3 двокрапки: записи про відкриті роботи, записи про непризначені роботи та записи про високу ступінь важкості. Це призначено для великих груп доручень для управління своїми завданнями відповідно до пріоритету. У контрактах можуть бути записи SLA, які визначають час відповіді на кожен вид квитка. Наприклад, інциденти з найвищим пріоритетом 1 повинні бути вирішені за одну годину, або до компанії буде застосовано штраф згідно з SLA.

Наступне зображення показує приклад черги:







Мал. 1.8. Черга квитків у сервісному центрі

Ви можете бачити цитати, випадки та проблеми (починаючи з Q, I та P відповідно). Також зверніть увагу на пріоритети інцидентів та проблем - вони обчислюються на основі двох показників: обсягу проблеми та тяжкості. Наприклад, якщо проблема існує лише для одного працівника, але має критичну ступінь серйозності, пріоритет буде встановлений на 3 з 5, а якщо це стосується кількох користувачів або всього підприємства, це може бути 1-м або 2-м залежно від ситуації.

Тож загалом це дуже зручне рішення щодо управління квитками, яке задовольняє організаційні потреби та міжнародні стандарти.

1.5. Постановка завдання дослідження

Програмний засіб складатиметься з двох основних модулів – основного системного модуля для перехвату мережевого трафіку (сервер, згідно з «клієнт-серверною» архітектурою), та модуля, що надає графічний інтерфейс користувача (*GUI*), завдяки якому можна виконувати аналіз трафіку.

У відповідності до сформульованої мети у даному проєкті було поставлено наступні завдання:

– провести аналіз сучасного стану наукових досліджень у напрямку впровадження засобів обліку та аналізу трафіку мобільних мереж;

– розробити програмний комлекс з повним спекторм звітності для мережевих адміністраторів;

– розробити системний модуль перехвату, оцінки та аналізу трафіку мобільних мереж, який буде проводити аналіз протоколів передачі даних та розшифровувати пакети;

– розробити модулі графічного інтерфейсу користувачів.

1.7. Висновки до розділу

В даному розділі була досліджена актуальність моніторингу мережевого трафіку, пошуку несправностей та критеріїв ефективності. У міру розвитку і ускладнення засобів, і методів мережевого моніторингу було помічено підвищення рівня загроз для використовуваних інформаційних технологій, адже з кожним роком з'являються все нові і нові атаки на мережі передачі даних, а також прості збої. У відповідь на них з'являються нові або вдосконалюються старі методи моніторингу та аналізу інформаційно-технічної інфраструктури.

Підбираючи приватні інструменти для використання їх в моніторингу мобільної мережі, спочатку слід вирішити, чи слід використовувати системи, що добре зарекомендували себе – які вже використовувалися багато років, або ж нові. Було запроновано нову систему, рішення комбінованого моніторингу – кращий напрямок для подальшої роботи.

РОЗДІЛ 2
АНАЛІЗ СУЧАСНИХ ЗАСОБІВ МОНІТОРИНГУ МЕЖЕВОГО ТРАФІКУ

Що стосується того, що Nagios Core повинен бути встановлений в системі Linux для нормальної роботи, ми вибрали Debian Linux як операційну систему, а не керуючу. Це стосується багатьох питань, як правило, високих стандартів та вражаючої надійності. Debian (як і більшість систем Linux) використовує сховища для зберігання пакетів. Це веб-сховища, в яких зберігається програмне забезпечення, яке найчастіше використовується, тому ми могли б автоматично встановити Nagios. Але з міркувань належної компіляції та конфігурації в нашій роботі ми виконали ручну установку. Пізніше ми використали сховище для встановлення надбудови NRPE на Ubuntu.

По-перше, нам доведеться створити користувача / групу Nagios з належних міркувань безпеки. Усі файли та папки, пов'язані з Nagios, повинні належати спеціальному користувачеві:

```
#adduser nagios
```

Це має створити обліковий запис користувача та групу за замовчуванням з однаковим іменем (nagios). Це перевіряє:

```
# groupadd nagios
```

Цю групу можна використовувати як групу, яку Nagios використовує як групу команд.

Наступним кроком є створення інсталяційного каталогу. Ми створити базовий каталог, де ми хотіли б встановити Nagios наступним чином:

#mkdir / usr / local / nagios

Змініть власника базового каталогу встановлення на користувача та групу Nagios, які ви додали раніше, таким чином:

#chown -R nagios: nagios / usr / local / nagios

Після виконання всіх процедур користувача ми завантажуємо останню версію nagios за таким посиланням:

http://www.nagios.org/download/

Розпакування дистрибутива Щоб розпакувати дистрибутив Nagios, ми використовуємо таку команду

#tar xzf nagios-2.6.tar.gz # cd nagios-версія (nagios-2.6)
Потім ми запускаємо скрипт налаштування

./configure --prefix = / usr / local / nagios --with-cgiurl = / nagios / cgi-bin --with-htmurl = / nagios / --with-nagios-user = nagios --with-nagios- group = nagios --with-command-group = nagios

Де

--prefix = / usr / local / nagios - це коренева папка Nagios - with-cgiurl = / nagios / cgi-bin - це папка CGI Nagios - with-htmurl = / nagios / - це папка HTML / веб-сайт Nagios - with-nagios-user = nagios є користувачем Nagios - with-nagios-group = nagios є групою Nagios - with-command-group = nagios - це група команд

Nagios, яка має користувача веб-сервера (Apache) та користувача nagios як members.Usage: ./configure [OPTION] ... [VAR = VALUE] ...

зробити все

це складе Nagios та CGI.

зробити встановлення

За замовчуванням, `make install 'встановить всі файли в` / usr / local / nagios / bin', `/ usr / local / nagios / lib 'тощо.

Ця команда встановить бінарні файли та файли HTML (документація та головна веб-сторінка).

#make install-init

Це встановить сценарії запуску

make install-config

Це встановлює конфігураційні файли * SAMPLE * у / usr / local / nagios / тощо. Нам доведеться змінити ці зразки файлів, перш ніж ми зможемо використовувати Nagios.

Це завершить установку. Тепер нам потрібно переглянути структуру каталогу та розташування файлів.

#cd / usr / local / nagios

Ми бачимо п’ять різних підкаталогів. Оскільки ми будемо робити багато змін конфігураційних файлів вручну, нам потрібно буде знати всю внутрішню структуру підкаталогів Nagios. Короткий опис того, що містить кожен каталог, подано нижче.

Зміст підкаталогу

основна програма bin / Nagios

etc / Main, ресурси, об'єкт та конфігураційні файли CGI слід помістити сюди

sbin / CGI

файли спільного доступу / HTML (для веб-інтерфейсу та онлайн-документації)

var / Порожній каталог для журналу, файлу стану, файлу збереження тощо.

var / archives Порожній каталог архівованих журналів

var / rw Порожній каталог зовнішнього файлу команд

Тепер нам потрібно сконцентруватись на одному / usr / local / nagios / etc каталозі, де зберігаються ці конфігураційні файли: cgi.cfg-sample

nagios.cfg-зразок

більший.cfg-зразок

misccommands.cfg-зразок

checkcommands.cfg-sample

minimal.cfg-зразок

resource.cfg-зразок

Вище наведено зразки файлів конфігурації, які нам потрібні для перейменування цих файлів у файли .cfg.

mv більший.cfg-зразок більший.cfg

Ми виконаємо більше дій із цим файлом у подальшому процесі конфігурації.

1. Компіляція Nagios із вбудованою Perl

Якщо ми хочемо використовувати вбудований інтерпретатор Perl, спочатку нам потрібно скомпілювати Nagios з підтримкою для нього. Для цього ми запускаємо скрипт налаштування з додаванням опції --enable-embedded-perl. Якщо ми хочемо, щоб вбудований інтерпретатор кешував внутрішньо скомпільовані сценарії, додайте також параметр --with-perlcache.

Приклад:

./configure --enable-embedded-perl --with-perlcache інші варіанти ...

Перезапустивши скрипт налаштування з новими параметрами, перекопіюйте Nagios.

Використання перекладача Perl, специфічне для плагіна

Починаючи з Nagios 3, ми можемо вказати, які плагіни та скрипти Perl слід чи не запускати під вбудованим інтерпретатором Perl. Це особливо корисно, якщо у вас є клопітні сценарії Perl, які погано працюють з інтерпретатором Perl.

Щоб явно повідомити Nagios, чи слід використовувати вбудований інтерпретатор Perl для певного сценарію perl, додайте один із наведених нижче записів до вашого сценарію / плагіна Perl.

Щоб сказати Nagios використовувати інтерпретатор Perl для певного сценарію, ми додаємо цей рядок до сценарію Perl:

nagios: + epn

Щоб сказати Nagios НЕ використовувати вбудований інтерпретатор Perl для певного сценарію, додайте цей рядок до сценарію Perl:

nagios: -epn

Кожен рядок повинен знаходитись у перших 10 рядках сценарію, щоб Nagios його виявив.

Тепер у нас є повністю встановлений Nagios, над яким можна працювати. Наступними кроками було б встановити плагіни та розпочати налаштування Nagios.

2. Встановіть плагіни Nagios

Щоб Nagios мав якусь користь, нам потрібно буде встановити плагіни nagios. Зазвичай плагіни встановлюються в каталозі libexec / нашої установки Nagios (тобто / usr / local / nagios / libexec). Плагіни - це сценарії або двійкові файли, які виконують усі перевірки служби та хосту, що становлять моніторинг.

Ми завантажуємо останню версію плагінів nagios за таким посиланням:
http://sourceforge.net/projects/nagiosplug/

Зніміть завантажений файл

tar -zxvf nagios-plugins- <версія> .tar.gz [тут nagios-plugins-1.4.5.tar.gz]

Це створить новий каталог nagios-plugins- <версія> [тут nagios-plugins-1.4.5]

Перейдіть до каталогу та запустіть Налаштувати сценарій

cd nagios-plugins-1.4.5
./configure

Виконайте наступне, щоб створити та встановити плагіни

make && make install

Це має встановити модулі в каталозі / usr / local / nagios / libexec.

Існує кілька правил, які повинні реалізовувати всі плагіни Nagios, що робить їх придатними для використання Nagios. Усі плагіни надають опцію --help, яка відображає інформацію про плагін та його роботу. Ця функція дуже допомагає, коли ми намагаємося відстежувати нову послугу за допомогою плагіна, який ми раніше не використовували.

check_ssh (nagios-plugins 1.3.0-alpha1)

Це показує нам, що плагін check_ssh приймає один необхідний хост параметрів та два необов'язкові параметри, час очікування та порт.

Зараз Базова інсталяція nagios завершена. Після цього нам потрібно налаштувати веб-інтерфейс для nagios.

3. Конфігурація веб-інтерфейсу

Тепер слід налаштувати веб-інтерфейс та налаштувати аутентифікацію користувача для nagios. Це також описує, як нам вдалося змусити CGI використовувати автентифікацію.

Після встановлення Nagios та плагінів настав час створити інтерфейс веб-інтерфейсу для nagios. Для цього нам потрібно налаштувати псевдонім веб-інтерфейсу та псевдонім сценарію CGI на нашому веб-сервері.

У Debian з Apache2 ми можемо зробити це наступним чином:

Створення файлу конфігурації nagios (або з будь-яким іменем, яке ми хочемо називати псевдонімом) в / etc / apache2 / sites-available / із таким вмістом (скопіюйте та вставте за допомогою VI):

ScriptAlias / nagios / cgi-bin / usr / local / nagios / sbin

<Каталог "/ usr / local / nagios / sbin">
Параметри ExecCGI
AllowOverride None
Наказ дозволити, відмовити
Дозволити від усіх
AuthName "Nagios Access"
AuthType Basic
AuthUserFile /usr/local/nagios/etc/htpasswd.users
Потрібен дійсний користувач
</Directory>

Псевдонім / nagios / usr / local / nagios / share

<Каталог "/ usr / local / nagios / share">
Варіанти Немає
AllowOverride None
Наказ дозволити, відмовити
Дозволити від усіх
AuthName "Nagios Access"

AuthType Basic
AuthUserFile /usr/local/nagios/etc/htpasswd.users
Потрібен дійсний користувач
</Directory>

5. Сценарій заповнення бази даних

Перш ніж використовувати Perl, ми повинні переконатися, що наше встановлене рішення Nagios відповідає двом наступним умовам:

- Nagios компілюється з підтримкою Perl, що робиться за допомогою наступної команди:

./configure --enable-embedded-perl --with-perlcache

- Перевірте, чи має файл $ NAGIOS / etc / nagios.cfg такі параметри, які дозволяють вбудований інтерпретатор Perl:

enable_embedded_perl = 1
use_embedded_perl_implicitly = 1

Основною особливістю розробленої надбудови є тимчасова основа спостереження. Відповідно до теорії ризиків відмови апаратного забезпечення, після того, як пройшов певний проміжок часу в періоді експлуатації апаратного забезпечення, ймовірність відмови зростає нелінійно. Це означає, що кожен пристрій має свій "попереджувальний" та "критичний" час для заміни після встановлення. Основна функціональність розробленого доповнення полягає у спрацьовуванні належного часу сигналів тривалості експлуатації, що перевищують. Для цього завдання потрібна база даних із фактичною інформацією про поточне встановлене обладнання та мережеві пристрої систем. Залежно від обсягу, база даних може лежати у вигляді текстового файлу на моніторинговому "королівському" хості або бути доступною як SQL бази ORACLE через мережевий сервер баз даних. У нашому випадку ми прийняли рішення керувати нею локально, наскільки ми говорили про Nagios Core як добре збалансовану систему для малого обсягу. Nagios Core можна налаштувати для моніторингу величезних інфраструктур, але для цього потрібен чіткий рівень підтримки та управління ресурсами. Отже, ми наблизились до локальної бази даних. Для того, щоб система була зручною, нам доведеться розробити процедуру управління вмістом бази даних за допомогою введення на основі підказки.

Для створення даних, з якими працюватиме плагін, ми розробили сценарій, який створює та модифікує дані у файлі, який буде використовуватися як база даних. Сценарій бере кілька входів, які:

- Ідентифікатор обладнання (рядок)
- Дата встановлення
- Дата передбачуваного попередження
- Дата критичної тривоги

Ці дані слід брати з інформації виробника обладнання; наприклад, це може бути автоматизовано шляхом експорту Perl із веб-сайту. Отримана нижче таблиця є тією, яка використовується в роботі:

Intel_CPU_i7_quad 2011.03.15 2013.03.15 2014.03.15

GGB_MothB_GA-E350N 2011.04.09 2011.04.12 2015.04.15

SGT_Barracuda_320 2011.05.09 2011.02.15 2012.09.15

Що означає Intel CPU i7 quad, материнську плату Gigabyte GA-E350N та жорсткий диск Seagate Barracuda 320 ГБ відповідно.

Сценарій, який записує в файл введення даних на основі підказки:

#! urs / bin / perl

Сценарій введення файлу даних для тестування плагіна

Сценарій Євгена Гулого, Київ, НАУ 2011

print "будь-ласка, введіть етикетку апаратного компонента:";

$ h_label = <STDIN>;

<STDIN> зараховується як швидке введення, ми зчитуємо його у змінну, яка буде використана пізніше.

print "будь ласка, введіть дату встановлення:";

$ h_inst_d = <STDIN>;

надрукувати "будь ласка, введіть приблизну дату попередження:";

$ h_warn_d = <STDIN>;

друк "будь ласка, введіть дату зміни оцінки:";

$ h_crit_d = <STDIN>;

Масив # @ stringz формується для об'єднання даних, які були зібрані з підказки

```
мій @stringz = ("$ h_label", "$ h_inst_d", "$ h_warn_d", "$ h_crit_d \ n");
відкрити (Файл, ">> h_dbase.txt");
foreach (@stringz) {
    чомп;
    #chomp з'їдає останній символ, щоб вихідний друк виглядав нормально
    print (Файл "$ _");
    # ”$ _” - це вбудована змінна Perl, яка дорівнює значенню члена масиву поточної ітерації циклу
}
закрити (Файл);
```

Отже, у нас є корисна база даних, яку можна перевірити за допомогою плагіна. На цьому етапі ми переходимо до розробки плагіна. Тепер ми маємо створити сценарій, який буде зчитувати ці дані та порівнювати кожну дату з поточною. Виходячи з цього, він зможе з’ясувати, чи може будь-яке з раніше встановлених програм бути застарілим. Згодом скрипт повинен вивести статус, який може бути «ОК», «ПОПЕРЕДЖЕННЯ» та «КРИТИЧНИЙ».

Повний код плагіна буде присутній у додатку А.

Для його розробки ми використовували редактор із відкритим кодом gedit.

Після того, як код був готовий, нам довелося зробити кілька налаштувань, щоб плагін справді працював. Сценарій Perl - це фактично метод перевірки, який слід належним чином впровадити в систему моніторингу.

Отже, спочатку ми відкриваємо $ NAGIOS / etc / objects / commands.cfg файл та додайте наступні рядки для визначення команди перевірки:

```
# РОЗРОБЛЕНО ПЕРЛІН ПЛУГІН
визначити команду {
    ім'я_команди check_hardware_availability
    command_line $ USER1 $ / check_hardware_availability
}
```

Тут $ ARGS1 є необов'язковим аргументом, прийнятим плагіном. Він визначає аргументи, які команда може прийняти.

Commands.cfg - це файл, який містить усі команди, які можуть бути використані Nagios Core. Скрипти Perl або bash можуть бути використані як методи визначення команд. Отже, якщо ми хочемо зробити визначення у файлі конфігурації (пізніше), нам потрібно ввести команду check.

Тепер ми модифікуємо файл $ NAGIOS / etc / objects / localhost.cfg, щоб додати перевірку служби.

визначити послугу {
    скористайтеся локальним сервісом
    host_name localhost
    service_description check_hardware_availability
    check_command check_hardware_availability
}

6. Розробка плагіна

Інтерфейс між демоном Nagios і плагіном перевірки послуг дуже простий. За замовчуванням демон Nagios розгалужує процес для запуску плагіна, і в найпростішому випадку плагін видає один рядок описового тексту, а потім виходить з 0 (OK), 1 (ПОПЕРЕДЖЕННЯ), 2 (КРИТИЧНИЙ) або 3 (НЕВІДОМИЙ). Усі коди виходу можна переглянути у вихідному коді плагіна у додатку А.

Нижче ви можете побачити статус localhost:

Рис. 2.4. Статус Localhost (з розробленим плагіном у стані попередження)

У ньому розміщено дві служби - check_ping (для тестування) та check_hardware_availability - наш розроблений на замовлення плагін. Як ви можете помітити, згідно з таблицею бази даних, ми отримали правильний результат ПОПЕРЕДЖЕННЯ для перевірки. Працює чудово!

Ми також внесли деякі зміни конфігурації, щоб увімкнути різні перевірки для цієї послуги, і ось як виглядає остаточна конфігурація:

Рис.2.5. Конфігурація для елемента в моніторингу

Нарешті, у нас є робоче рішення для перевірки часу експлуатації обладнання. Варто також зазначити, що історія спостерігачів записана в журналах, і якщо щось трапиться з моніторинговим хостом Nagios, він може отримати статус у момент, коли служба буде вимкнена.

7. Висновки

Як практичну частину роботи ми розробили робочий набір опцій для моніторингу різнорідної системи комп'ютерів, пов'язаних мережею. Також було виготовлено плагін Perl; це дозволить людям контролювати обладнання за його гарантованим терміном безвідмовної роботи. Ми вважаємо це важливим аспектом, який покращує рішення з відкритим кодом Nagios Core і робить його більш схильним до більш широкого використання.

РОЗДІЛ 3
ОПИСАННЯ ПРОГРАМНОЇ СИСТЕМИ ОБЛІКУ ТА АНАЛІЗУ МЕРЕЖЕВОГО ТРАФІКУ

3.1. Опис системи моніторингу

Nagios® Core ™ - це програма з відкритим кодом та програма для моніторингу мережі. Він спостерігає за хостами та послугами, які ви вказали, попереджаючи вас, коли справи йдуть погано і коли вони покращуються.

Nagios Core спочатку був розроблений для роботи під Linux, хоча це повинно працювати і в більшості інших уніксів.

Деякі з багатьох функцій Nagios Core включають:

Моніторинг мережевих служб (SMTP, POP3, HTTP, NNTP, PING тощо)

Моніторинг ресурсів хоста (навантаження процесора, використання диска тощо)

Проста конструкція плагіна, яка дозволяє користувачам легко розробляти власні перевірки послуг

Паралелізовані перевірки послуг

Можливість визначення ієрархії мережевих хостів за допомогою "батьківських" хостів, що дозволяє виявляти та розрізняти хости, які не працюють, та ті, які недосяжні.

Звертатися до сповіщень про виникнення проблем із сервісом або хостом та їх вирішення (електронною поштою, пейджером або визначеним користувачем способом)

Можливість визначення обробників подій, які запускатимуться під час службових або хост-подій для попереджувального вирішення проблем

Автоматичне обертання файлу журналу

Підтримка впровадження надлишкових хостів моніторингу

Необов'язковий веб-інтерфейс для перегляду поточного стану мережі, сповіщень та історії проблем, файлу журналу тощо.

Структура Nagios:

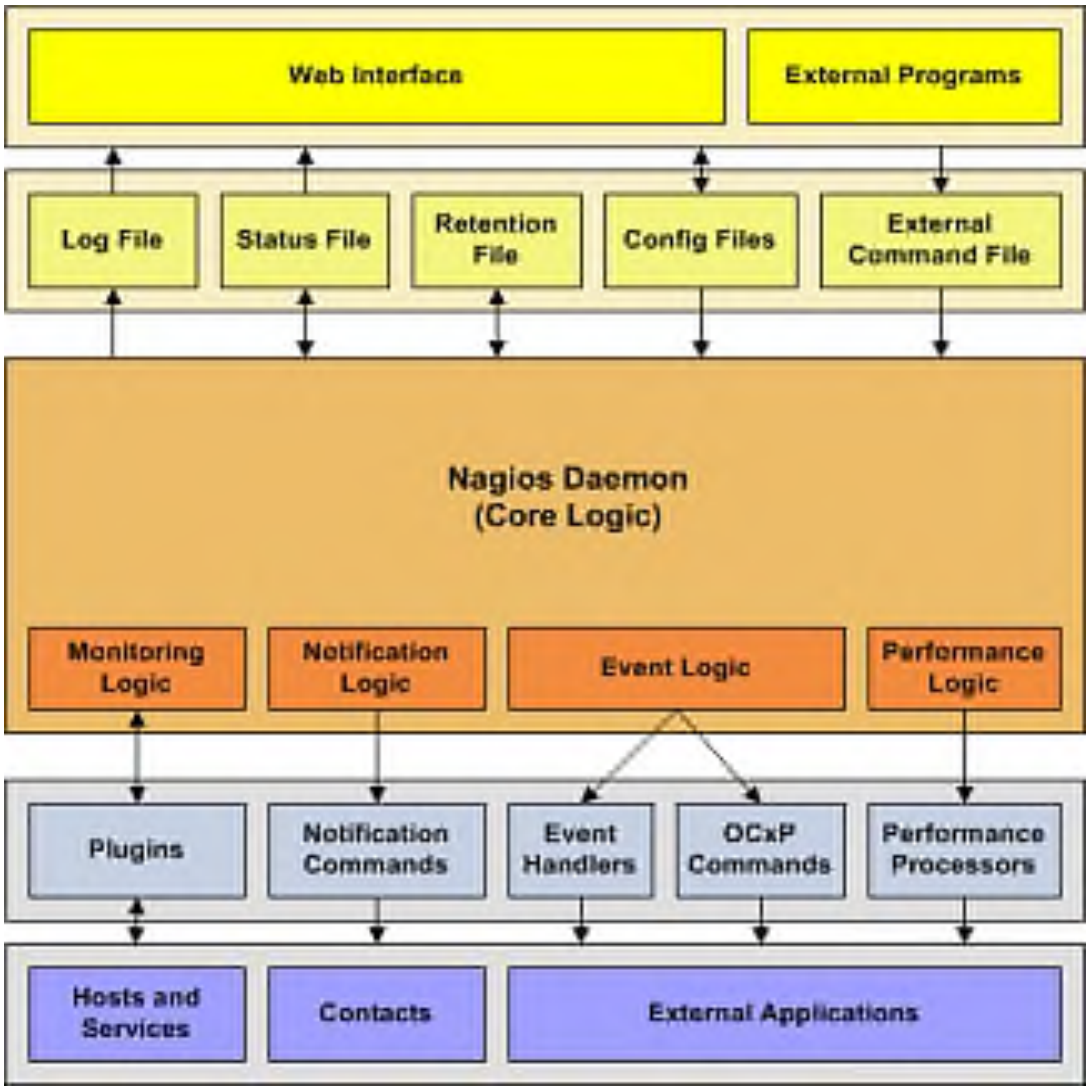


Рис. 1.10 Структура Nagios

Специфікації моніторингу Nagios

Nagios можна використовувати для моніторингу різних середовищ, включаючи:

Windows машини

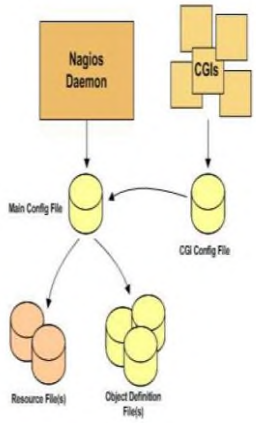
Машини Linux / Unix

Мережеві сервери

Маршрутизатори / комутатори

Мережеві принтери

Загальнодоступні послуги (HTTP, FTP, SSH тощо)



1.11 Архітектура рішення Nagios

Основний файл конфігурації

Основний файл конфігурації містить ряд директив, які впливають на роботу демона Nagios. Цей файл конфігурації читають як демон Nagios, так і CGI. Тут ви захочете розпочати свої конфігураційні пригоди.

Файл (и) ресурсу

Файли ресурсів можна використовувати для зберігання визначених користувачем макросів. Основний сенс наявності файлів ресурсів - використовувати їх для зберігання конфіденційної інформації про конфігурацію (наприклад, паролів), не роблячи їх доступними для ІГІ.

Файли визначення об’єктів

Файли визначення об’єктів використовуються для визначення хостів, служб, груп хостів, контактів, контактних груп, команд тощо. Тут ви визначаєте всі речі, які ви хочете контролювати, і те, як ви хочете їх контролювати. Ви можете вказати один або кілька файлів визначення об'єкта за допомогоюcfg_file та / або cfg_dir db основному файлі конфігурації.

Об'єкти - це всі елементи, які беруть участь у логіці моніторингу та сповіщення. Типи об'єктів включають:

Послуги

Групи обслуговування

Ведучі

Групи господарів

Контакти

Групи контактів

Команди

Періоди часу

Повідомлення ескалації

Залежності сповіщення та виконання

Більше інформації про те, що таке об’єкти та як вони пов’язані між собою, можна знайти нижче.

Де визначаються об’єкти?

Об'єкти можуть бути визначені в одному або декількох файлах конфігурації та / або каталогах, які ви вказуєте, використовуючи директиви cfg_file та / або cfg_dir у головному файлі конфігурації.

Як визначаються об'єкти?

Об'єкти визначаються у гнучкому форматі шаблону, що може значно полегшити управління конфігурацією Nagios в довгостроковій перспективі. Основну інформацію про те, як визначити об’єкти у файлах конфігурації, можна знайти тут.

Ознайомившись з основами того, як визначати об'єкти, вам слід прочитати про успадкування об'єктів, оскільки це зробить вашу конфігурацію надійнішою в майбутньому. Досвідчені користувачі можуть скористатися деякими розширеними функціями визначень об'єктів, як описано в документації до фокусів на об'єкти.

Деякі з основних типів об'єктів пояснюються більш докладно нижче ...

Ведучі є одним із центральних об’єктів у логіці моніторингу. Важливими атрибутами хостів є наступні: Хости, як правило, це фізичні пристрої у вашій мережі (сервери, робочі станції, маршрутизатори, комутатори, принтери тощо).

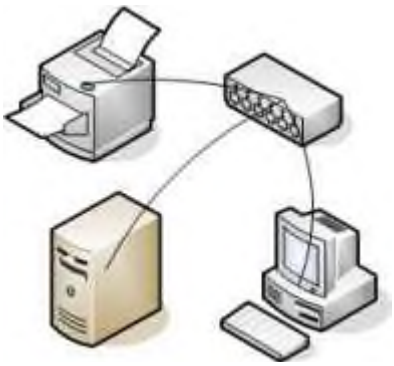


Рис. 1.12 Посилання через мережеві вузли

Хости мають певну адресу (наприклад, IP або MAC-адресу).

Хости мають одну або кілька служб, пов’язаних із ними. Хости можуть мати батьківсько-дочірні стосунки з іншими хостами, часто представляючи реальні мережеві з'єднання, що використовується в логіці досяжності мережі. Групи хостів - це групи одного або декількох хостів. Групи хостів можуть полегшити перегляд стану пов’язаних хостів у веб-інтерфейсі Nagios та спростити вашу конфігурацію за допомогою об’єктних фокусів. Послуги є одним з центральних об'єктів у логіці моніторингу. Послуги пов'язані з хостами і можуть бути:

Атрибути хоста (завантаження процесора, використання диска, час роботи тощо)

Послуги, що надаються хостом (HTTP, POP3, FTP, SSH тощо)

Інші речі, пов’язані з хостом (записи DNS тощо)

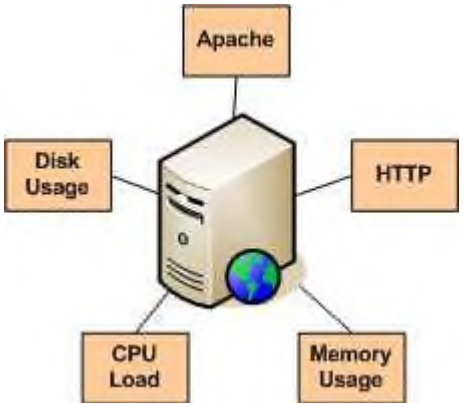


Рис. 1.13 Варіанти моніторингу об’єктів

Групи обслуговування - це групи однієї або декількох служб. Групи служб можуть спростити перегляд стану пов’язаних служб у веб-інтерфейсі Nagios та спростити вашу конфігурацію за допомогою використання об’єктних хитрощів.

Контакти є люди, які беруть участь у процесі сповіщення:

Мати один або кілька методів сповіщення (мобільний телефон, пейджер, електронна пошта, обмін миттєвими повідомленнями тощо)

отримувати повідомлення для хостів та сервісу, за який вони відповідають

Групи контактів - це групи одного або декількох контактів. Групи контактів можуть спростити визначення всіх людей, які отримують повідомлення про виникнення певних проблем із хостом або сервісом.

Часові періоди використовуються для контролю:

Коли хости та послуги можуть контролюватися

Коли контакти можуть отримувати сповіщення

Команди використовуються, щоб повідомити Nagios, які програми, сценарії тощо повинні виконуватися для виконання:

Перевірка хосту та обслуговування

Повідомлення

Обробники подій

Файл конфігурації CGI

Файл конфігурації CGI містить ряд директив, що впливають на роботу CGI. Він також містить посилання на основний файл конфігурації, тому CGI знають, як ви налаштували Nagios і де зберігаються визначення ваших об'єктів.

Технічні характеристики:

Windows:

Рис. 1.14 Технічні характеристики моніторингу систем Windows

Моніторинг приватних служб або атрибутів машини Windows вимагає встановлення на ньому агента. Цей агент діє як проксі-сервер між плагіном Nagios, який здійснює моніторинг, та фактичною службою чи атрибутом машини Windows. Без встановлення агента на вікні Windows, Nagios не зможе контролювати приватні служби або атрибути вікна Windows.

У цьому прикладі ми будемо встановлювати NSClient ++ аддон на машині Windows та використовуючи плагін check_nt для зв'язку з аддоном NSClient ++. Якщо ви дотримувались посібника з швидкого запуску, плагін check_nt вже повинен бути встановлений на сервері Nagios. Інші агенти Windows (напрікладNC_Net), за бажанням, можна використовувати замість NSClient ++ - за умови, що ви трохи зміните визначення команд, служб тощо. Для простоти я розгляну лише використання аддону NSClient ++ у цих інструкціях.

Linux / Unix:

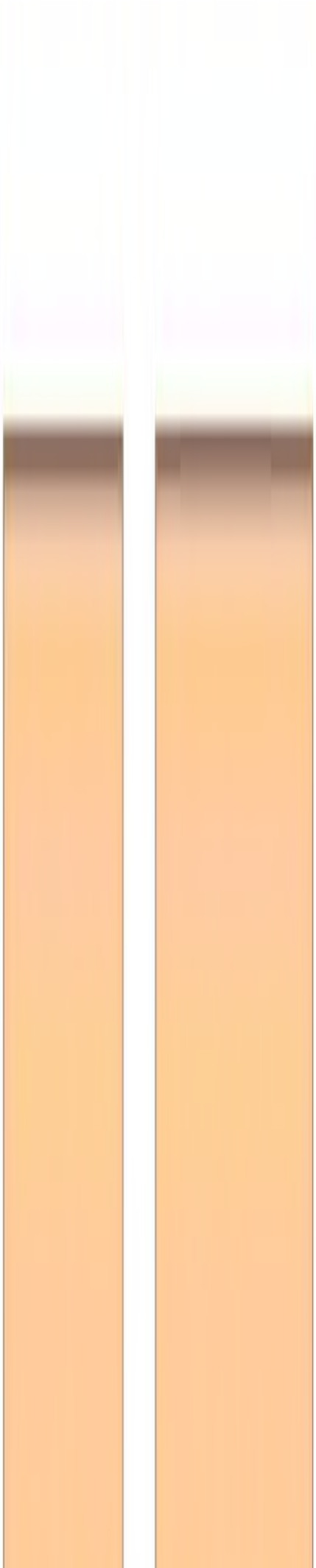


Рис. 1.15 Технічні характеристики моніторингу систем Linux / Unix

Існує кілька різних способів контролю атрибутів або віддалених серверів Linux / Unix. Перший - використання спільних ключів SSH та плагіна check_by_ssh для запуску плагінів на віддалених серверах. Цей метод не буде розглядатися тут, але може призвести до великого навантаження на ваш сервер моніторингу, якщо ви стежите за сотнями або тисячами служб. Причиною цього є накладні витрати на налаштування / знищення з'єднань SSH.

Іншим поширеним методом моніторингу віддалених хостів Linux / Unix є використання Аддон NRPE. NRPE дозволяє виконувати плагіни на віддалених хостах Linux / Unix. Це корисно, якщо вам потрібно відстежувати локальні ресурси / атрибути, такі як використання диска, завантаження процесора, використання пам'яті тощо на віддаленому хості.

Мережеві принтери:

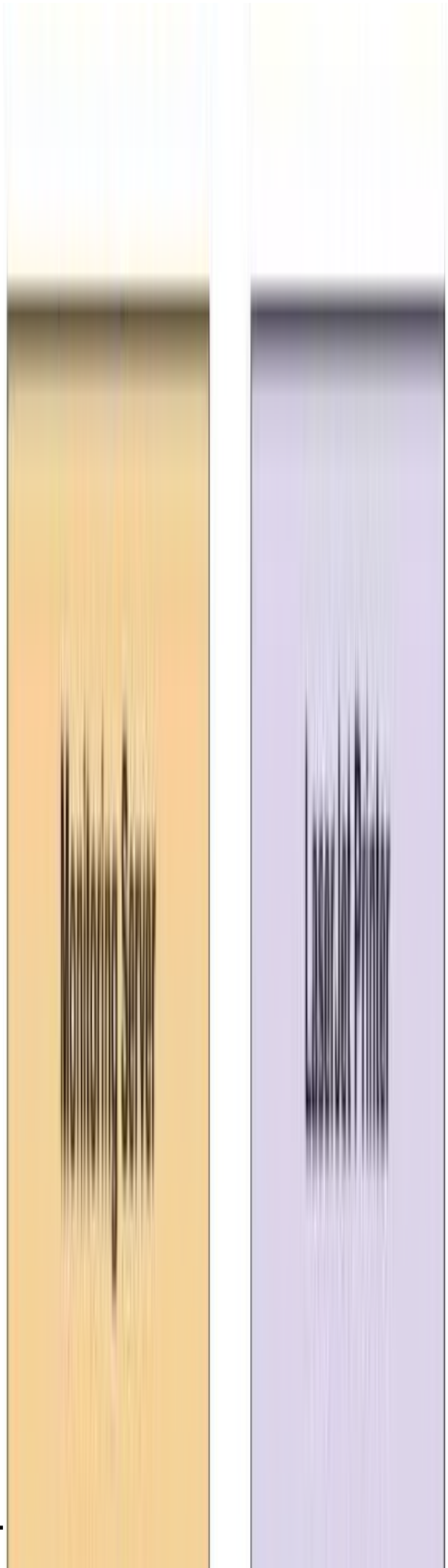


Рис. 1.16 Технічні характеристики моніторингу мережевих принтерів

Моніторинг стану мережевого принтера досить простий. У принтерах із підтримкою JetDirect зазвичай увімкнено SNMP, що дозволяє Nagios контролювати свій стан за допомогою плагіна check_hpjd. Плагін check_hpjd буде скомпільований та встановлений лише в тому випадку, якщо ми встановимо у вашій системі пакети net-snmp та net-snmp-utils.

Маршрутизатори / комутатори:

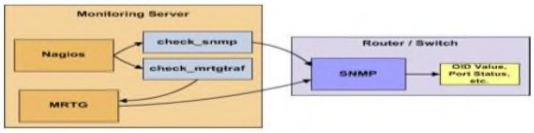


Рис. 1.17 Технічні характеристики моніторингу маршрутизаторів / комутаторів

Моніторинг комутаторів і маршрутизаторів може бути простим або більш задіяним - залежно від того, яке обладнання у вас є і що ви хочете контролювати. Оскільки вони є критично важливими компонентами інфраструктури, ви, без сумніву, захочете відстежувати їх хоча б якимсь основним способом. Комутатори та маршрутизатори можна легко контролювати, "пропингуючи" їх, щоб визначити втрату пакетів, RTA тощо.

Якщо ваш комутатор підтримує SNMP, ви можете контролювати стан порту тощо за допомогою плагіна check_snmp та пропускної здатності (якщо ви використовуєте MRTG) за допомогою плагіна check_mrtgtraf. Плагін check_snmp буде скомпільовано та встановлено лише у тому випадку, якщо у вашій системі встановлені пакети net-snmp та net-snmp-utils. Перш ніж продовжувати, переконайтесь, що плагін існує в / usr / local / nagios / libexec. Якщо цього не сталося, встановіть net-snmp та net-snmp-utils та перекомпілюйте / переінсталюйте плагіни Nagios.

Плагіни Nagios.

На відміну від багатьох інших інструментів моніторингу, Nagios не включає ніяких внутрішніх механізмів для перевірки стану хостів та служб у

вашій мережі. Натомість Nagios робить всю брудну роботу на зовнішні програми (які називаються плагінами).

Що таке плагіни? Плагіни - це скомпільовані виконувані файли або сценарії (сценарії Perl, сценарії оболонки тощо), які можна запускати з командного рядка для перевірки стану, хосту чи служби. Nagios використовує результати плагінів для визначення

поточний стан хостів та служб у вашій мережі. Nagios буде запускати плагін щоразу, коли виникає потреба перевірити статус служби чи хоста. Плагін робить щось (зверніть увагу на дуже загальний термін), щоб виконати перевірку, а потім просто повертає результати Nagios. Nagios обробляє результати, які він отримує від плагіна, та вживає будь-яких необхідних дій (запуск обробників подій, розсилання сповіщень тощо).

Плагіни як шар абстракції Плагіни виступають як абстракційний рівень між логікою моніторингу, наявною в демоні Nagios, та реальними службами та хостами, які контролюються. Перевага цього типу архітектури плагінів полягає в тому, що ви можете відстежувати майже все, що вам прийде в голову. Якщо ми можемо автоматизувати процес перевірки чогось, ми можемо відстежувати це за допомогою Nagios.

Вже існує багато плагінів, створених для моніторингу основних ресурсів, таких як завантаження процесора, використання диска, швидкість пінгу тощо. Якщо ви хочете відстежувати щось інше, перегляньте документацію щодо написання плагінів і прокрутіть власний.

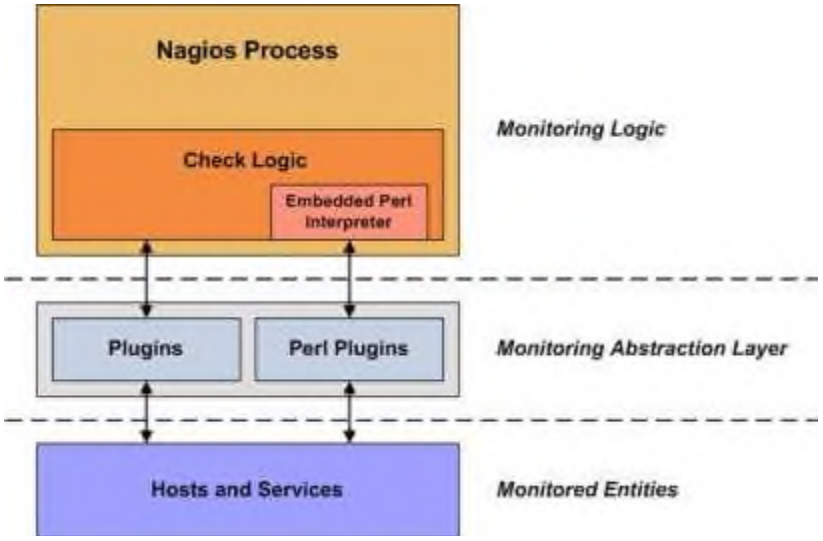


Рис. 1.18 Архітектура плагінів Nagios

Недоліком цього типу архітектури плагінів є той факт, що Nagios абсолютно не уявляє, що саме ви контролюєте. Ви можете стежити за статистикою мережевого трафіку, частотою помилок даних, помірною кімнатою, напругою процесора, швидкістю обертання вентилятора, навантаженням процесора, дисковим простором або здатністю вашого суперфантастичного тостера правильно підрум'янювати хліб вранці ... Нагіос не Не розумію специфіки того, що відслідковується - він просто відстежує зміни в стані цих ресурсів. Тільки самі плагіни точно знають, що вони відстежують і як виконувати фактичні перевірки.

Які плагіни доступні?

На даний момент доступні плагіни для моніторингу багатьох різних типів пристроїв та послуг, зокрема:

HTTP, POP3, IMAP, FTP, SSH, DHCP

Навантаження процесора, використання диска, використання пам'яті, поточні користувачі

Сервери Unix / Linux, Windows та Netware

Маршрутизатори та комутатори

Мова програмування Perl для розробки плагінів

Особливості та можливості Perl

Система моніторингу Nagios Core надає вбудовану підтримку Perl. Для обробки коду він використовує вбудований інтерпретатор Perl. Це робить Nagios ще більш гнучким та ремонтопридатним інструментом, оскільки розширює об'єкти потенційного моніторингу. Оскільки Perl має власну багату історію, давайте трохи розширимо термін.

Особливості Perl:

Потужний, стабільний, зрілий, портативний

Perl - це високопродуктивна, багатофункціональна мова програмування, що розробляє понад 20 років. Perl 5 працює на понад 100 платформах від

портативних до мейнфреймів. Perl підходить як для швидкого створення прототипів, так і для масштабних проектів розробки.

Місія критична

Використовується для критично важливих проектів у державному та приватному секторах.

Кодекс високої якості

Внутрішній код Perl сертифіковано на наявність низька щільність дефектів і бути вільним від дефекти безпеки згідно з Покриття аналіз.

Об’єктно-орієнтована, процедурна та функціональна

Підтримує об'єктно-орієнтоване, процедурне та функціональне програмування.

Легко розширюється

У Комплексі доступно понад 21 000 модулів з відкритим кодом

Маніпулювання текстом

Perl включає потужні інструменти для обробки тексту, що робить його ідеальним для роботи з HTML, XML та всіма іншими мовами розмітки та природними мовами.

Підтримка Unicode

Підтримує Версія Unicode 5.

Інтеграція баз даних

Інтерфейс інтеграції баз даних Perl (DBI) підтримує сторонні бази даних, включаючи Oracle, Sybase, Postgres, MySQL і багато інші.

Інтерфейс бібліотеки C / C ++

Інтерфейси Perl із зовнішніми бібліотеками C / C ++ через XS або ЛАГІТ.

Вбудовуваний

Інтерпретатор Perl може бути вбудований в інші системи, такі як веб-сервери і сервери баз даних.

Відкрите джерело

Perl є Відкрите джерело програмне забезпечення, ліцензований під його Художня ліцензія, або Загальна публічна ліцензія GNU (GPL).

Використання вбудованого перекладача Perl

Nagios можна скомпілювати з підтримкою вбудованого інтерпретатора Perl. Це дозволяє Nagios виконувати плагіни Perl набагато ефективніше, ніж в іншому випадку, тому це може вас зацікавити, якщо ви сильно покладаєтесь на плагіни, написані на Perl. Без вбудованого інтерпретатора Perl Nagios виконує плагіни Perl (і не Perl), розгалужуючи та виконуючи плагіни як зовнішню команду. Коли використовується вбудований інтерпретатор Perl, Nagios може виконати плагіни Perl, просто зробивши виклик бібліотеки.

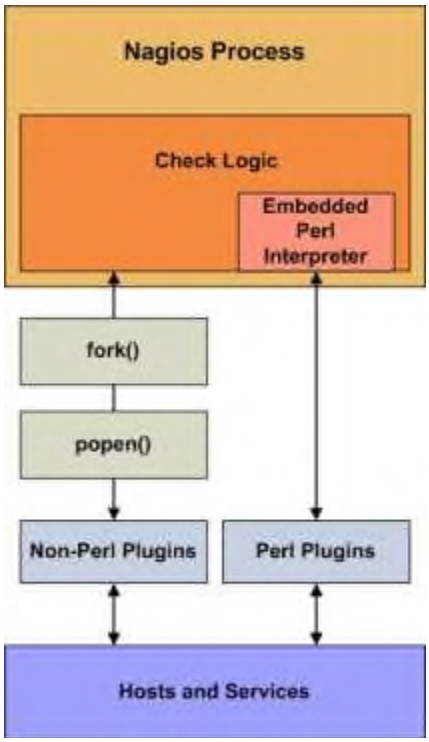


Рис. 1.19 Логіка перевірки Nagios

Стівен Девіс надав оригінальний вбудований код інтерпретатора Perl кілька років тому. Стенлі Хопкрофт був основною особою, яка допомагає вдосконалити вбудований код інтерпретатора Perl і прокоментував переваги / недоліки його використання. Він також дав кілька корисних підказок щодо створення плагінів Perl, які належним чином працюють із вбудованим інтерпретатором.

Слід зазначити, що "ePN", як використовується в цій документації, відноситься до вбудованих Perl Nagios, або, якщо вам більше подобається, Nagios, складених із вбудованим інтерпретатором Perl.

Деякі переваги ePN (вбудований Perl Nagios) включають:

Nagios буде витрачати набагато менше часу на роботу ваших плагінів Perl, оскільки він більше не розгалужується для виконання плагіна (кожен раз, коли завантажується інтерпретатор Perl). Натомість він виконує ваш плагін, здійснюючи виклик бібліотеки.

Це значно зменшує системний вплив плагінів Perl та / або дозволяє виконувати більше перевірок за допомогою плагіна Perl, ніж ви могли б в іншому випадку. Іншими словами, у вас менше стимулів писати плагіни іншими мовами, такими як C / C ++ або Expect / TCL, які, як правило, мають час розробки принаймні на порядок повільніший за Perl (хоча вони працюють приблизно в десять разів також швидше - TCL виняток).

Якщо ви не програміст на C, то ви все одно можете отримати величезну кількість пробігу з Nagios, дозволивши Perl робити всі важкі дії, не вимагаючи від Nagios сповільнення. Однак зауважте, що ePN не прискорить ваш плагін (крім усунення часу завантаження інтерпретатора). Якщо ви хочете швидкі плагіни, то розгляньте Perl XSUB (XS) або C після того, як ви переконаєтесь, що ваш Perl налаштований і що у вас є відповідний алгоритм (Benchmark.pm є безцінним для порівняння продуктивності елементів мови Perl).

Використання ePN - чудова можливість дізнатись більше про Perl.

Недоліки

Недоліки ePN (вбудований Perl Nagios) майже однакові з Apache mod_perl (тобто Apache із вбудованим інтерпретатором) порівняно зі звичайним Apache:

Програма Perl, яка чудово працює з простими Nagios, може не працювати з ePN. Можливо, вам доведеться змінити наші плагіни, щоб вони працювали.

Плагіни Perl важче налагодити за допомогою ePN, ніж під звичайним Nagios.

Наш ePN матиме більший РОЗМІР (розмір пам'яті), ніж звичайний Nagios.

Деякі конструкції Perl не можна використовувати або можуть поводитися інакше, ніж ви очікували.

Можливо, нам доведеться знати про "більше ніж один спосіб зробити це" і вибрати спосіб, який здається менш привабливим або очевидним.

Нам знадобляться більші знання Perl (але нічого дуже езотеричного чи подібних матеріалів про внутрішні компоненти Perl - якщо ваш плагін не використовує XSUBS).

Використання вбудованого перекладача Perl Якщо ви хочете використовувати вбудований інтерпретатор Perl для запуску плагінів та скриптів Perl, ось що вам потрібно буде зробити:

Скомпілюйте Nagios з підтримкою вбудованого інтерпретатора Perl (див. Інструкції нижче).

Увімкніть опцію enable_embedded_perl у головному файлі конфігурації.

Встановіть параметр use_embedded_perl_implicitly відповідно до ваших потреб. Цей параметр визначає, чи слід використовувати інтерпретатор Perl за замовчуванням для окремих плагінів та сценаріїв Perl.

За бажанням увімкніть або вимкніть запуск певних плагінів та сценаріїв Perl за допомогою вбудованого інтерпретатора Perl. Це може бути корисним, якщо деякі сценарії Perl мають проблеми із запуском інтерпретатора Perl. Див. Інструкції нижче для отримання додаткової інформації щодо цього.

Базова теорія надійності (використовується для розробки ідей плагіна)

Рівень відмов є частота з яким an інженерно система або компонентвідмов, що виражається, наприклад, у відмовах за годину. Це часто позначається символомГрецький лист λ (лямбда) і є важливим у інженерія надійності.

Частота відмов системи зазвичай залежить від часу, при цьому частота змінюється протягом життєвого циклу системи. Наприклад, рівень відмов автомобіля за п'ятий рік служби може бути в рази більше, ніж рівень відмов за перший рік служби. Не слід очікувати заміни вихлопної труби, капітального ремонту гальм або наявності основнихспосіб передавання проблеми в новому транспортному засобі.

Рівень відмов у дискретному розумінні

Частоту відмов можна визначити наступним чином:

Загальна кількість відмов у елементі населення, розділене на загальний час, витрачений цією сукупністю, протягом певного інтервалу вимірювання за вказаних умов. (MacDiarmid та ін.)

Хоча частоту відмов, λ (t), часто розглядають як ймовірністьщо збій трапляється у визначений інтервал, не давши відмови до часу t, це насправді не є ймовірністю, оскільки він може перевищувати 1. Помилкове вираження рівня відмов у% може призвести до неправильного сприйняття міри, особливо якщо вона буде вимірюватися від ремонтованих систем та декількох систем з непостійною частотою відмов або різним часом роботи. Це можна визначити за допомогоюфункція надійності або функція виживання R (t), ймовірність відмови до часу t.

Failure_rate (t) = f (t) / R (t), де f (t) - час до (першого) розподілу відмов, а R (t) - 1 - F (t):

$$\lambda = \frac{R(t_1) - R(t_2)}{(t_2 - t_1) \cdot R(t_1)} = \frac{R(t) - R(t + \triangle t)}{\triangle t \cdot R(t)}$$

за інтервал часу (t2 - t1) від t1 (або t) до t2 і Δt визначається як (t2 - t1). Зверніть увагу, що це аумовна ймовірність, отже, R (t) у знаменнику.

3.6. Налаштування апаратного і програмного середовища комплексу обліку і аналізу мобільного трафіку

Базова архітектура апаратного оснащення системи обліку і аналізу мобільного трафіку представлена на рис. 3.8.

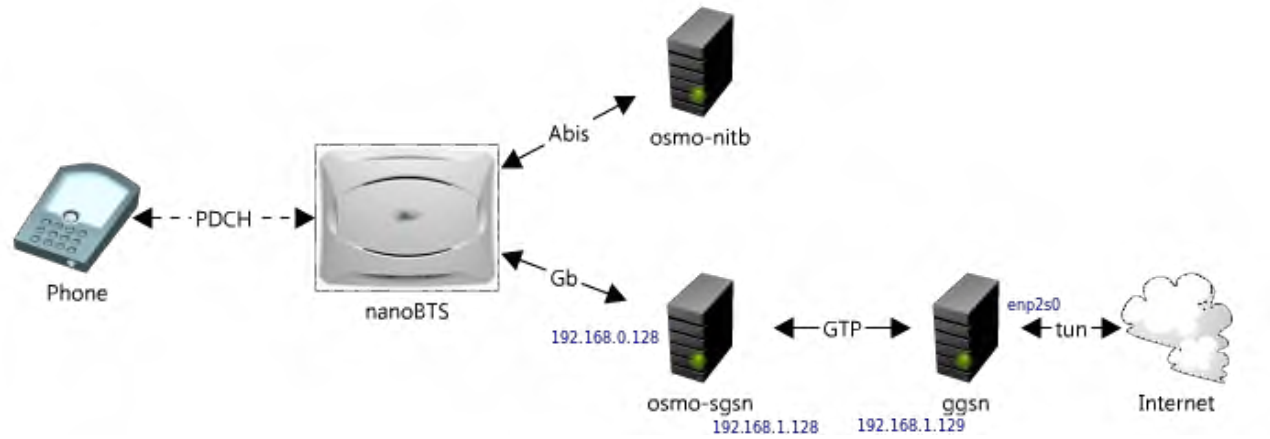


Рис. 3.8. Архітектура апаратного оснащення системи обліку і аналізу мобільного трафіку

Розглянемо основні рівні передачі мобільного трафіку:

– *PDCH* – це *Packet Data Channel*. Для передачі пакетних даних повинен використовуватися особливий тип логічного каналу. До сих пір оператори були змушені використовували *TCH/H* для обслуговування голосових викликів. Тепер є можливість замінити *TCH/H* на *PDCH*;

– *nanoBTS* – передбачається використовувати *OsmoBTS* в зв'язці з двома *osmocombb*-сумісними телефонами для створення базової станції, як і робили раніше;

– *osmo-nitb* – необхідна мінімальна конфігурація для активації *GPRS* сервісу та виконати перекомпіляцію *osmo-nitb* з підтримкою *osmo-sgsn*;

– *osmo-sgsn* – *Serving GPRS Support Node*. Ядро *GPRS* мережі, що є аналогом *MSC* для голосових викликів.

– *ggsn* – *GPRS Gateway Support Node*. Даний вузол стоїть на кордоні між *GPRS Core network* (*GTP*) та Інтернетом. Легко збирається і підключається до решти модулів *osmocom*.

Необхідні для реалізації функції:

– контроль доставки пакетів даних користувачам;

– взаємодія з реєстром власних абонентів мережі *HLR* або аутентифікація (перевірка дозволу на запит користувачами послуги); механізм збігається з механізмом аутентифікації в *GSM*;

– моніторинг знаходження в режимі *online* користувачів;

– перетворення кадрів *GSM* у формати, які використовуються протоколами *TCP* / *IP* глобальної комп'ютерної мережі *Internet*;

– реєстрація або «прикріплення» (*attachment*) абонентів, що знову «з'явилися» в зоні дії мережі;

– шифрування даних; алгоритм шифрування в технології *GPRS* (*GEA*1, *GEA*2, *GEA*3) відрізняються від алгоритмів шифрування в *GSM* (*A*5 / 1, *A*5 / 2, *A*5 / 3), але розроблені на їх основі;

– збір надходить білінгової інформації, пересилання її в головний офіс і т. п.

На рис. 3.8. пропущений ще один компонент *PCU – Packet Control Unit*.

PCU виконує деякі функції *BSC*, але тільки для пакетних даних. Для його реалізації буде використаний *osmo-pcu*. На модифікованій схемі (рис. 3.9) *PCU* присутній.

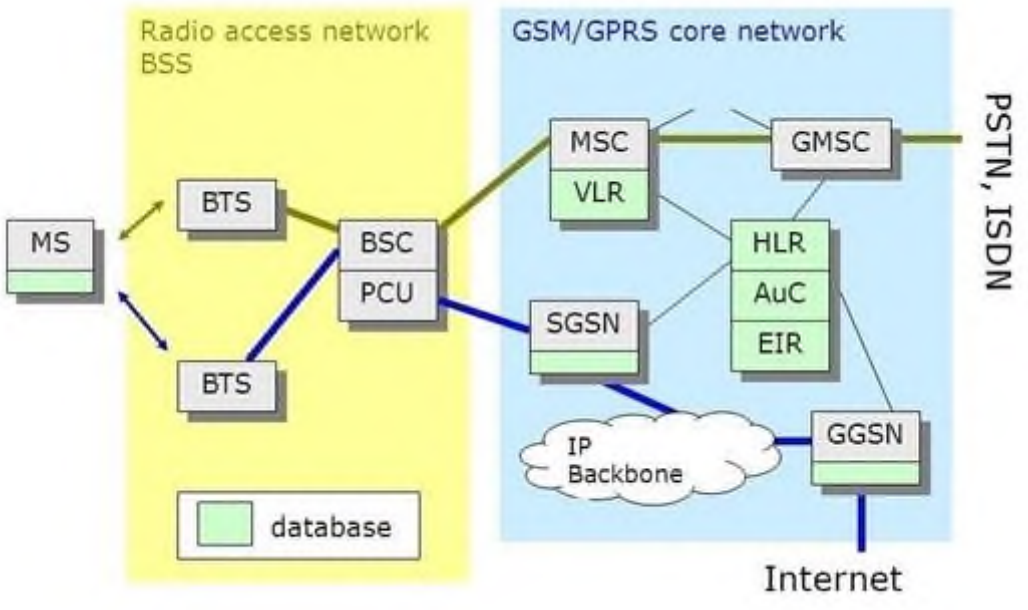


Рис. 3.9. Модифікована архітектура апаратного оснащення системи обліку і аналізу мобільного трафіку

Конфігураційні файли зберігаються в /*root*/.*osmocom*. Перш ніж використовувати конфігураційні файли, потрібно ввести коректні *IP* адреси замість ВІРТУАЛЬНИЙ_*IP* і ОСНОВНОЙ_*IP*, а так само *GSM*900 або *DCS*1800 замість діапазону і номер *ARFCN* замість каналу.

Передбачається, що всі компоненти будуть працювати на одному пристрої, так що нам потрібно створити віртуальний інтерфейс для мережевого адаптера. *IP* адреси для *GGSN* і *SGSN* повинні відрізнятися.

Також потрібно дозволити транзитні пакети і налаштувати *NAT*, так як ми будемо «роздавати» Інтернет всім абонентам мережі. (Мережу 192.168.0.0/24 міняти не потрібно, вона буде закріплена за інтерфейсом *tun*0, який з'явиться при запуску всіх компонентів *GPRS* інфраструктури).

Можуть бути проблеми з *DNS* трафіком, тоді рекомендується додати ще одне правило в *iptables*:

*iptables -t nat -I PREROUTING -i tun*0 *-p udp --dport* 53 *-j DNAT --to-dest* 8.8.

Запуск всіх сервісів виконується наступними командами:

cd /root/.osmocom

*osmo-nitb -s -c /root/.osmocom/open-bsc.cfg -l /root/.osmocom/hlr.sqlite*3 *-P -C --debug = DSQL*: *DLSMS*: *DRLL*: *DCC*: *DMM*: *DRR*: *DMSC* : *DHO*: *DGPRS*: *DNS*: *DLLC*: *DCTRL*

cd /root/.osmocom

ggsn -c /root/.osmocom/ggsn.conf -f -d

cd /root/.osmocom

osmo-sgsn -c /root/.osmocom/osmo_sgsn.cfg -d DRLL: *DCC*: *DMM*: *DRR*: *DNM*: *DMSC*: *DHO*: *DGPRS*: *DNS*: *DLLC*: *DCTRL*

cd / *Root* / *osmocom* / *trx* / *src*

host / *osmocon* / *osmocon -m c*123*xor -p* / *dev* / *ttyUSB*0 *-s* / *tmp* / *osmocom_l*2 *-c target* / *firmware* / *board* / *compal_e*88 / *trx.highram.bin -r* 99

cd / *Root* / *osmocom* / *trx* / *src*

host / *osmocon* / *osmocon -m c*123*xor -p* / *dev* / *ttyUSB*1 *-s /tmp/osmocom_l*2.2 *-c target* / *firmware* / *board* / *compal_e*88 / *trx.highram.bin -r* 99

cd / *Root* / *osmocom* / *trx* / *src* / *host* / *layer*23 / *src* / *transceiver* /

./transceiver -a SCH -2 *-r* 99

cd /root/.osmocom

osmo-bts-trx --debug DRSL: *DOML*: *DLAPDM -r* 99

В результаті буде отримано в консолі *osmo-pcu* наступне зображення (рис. 3.10).

```
root@osmobox:~/.osmocom# osmo-pcu -c /root/.osmocom/osmo-pcu.conf
<000b> telnet_interface.c:101 telnet at 127.0.0.1 4240
<0001> osmobts_sock.cpp:227 Opening OsmoPCU L1 interface to OsmoBTS
<0001> osmobts_sock.cpp:286 osmo-bts PCU socket /tmp/pcu_bts has been connected
<0001> osmobts_sock.cpp:290 Sending version 0.2.915-241f5 to BTS.
<0001> pcu_l1_if.cpp:107 Sending 0.2.915-241f5 TXT as PCU_VERSION to BTS
<0001> pcu_l1_if.cpp:404 BTS available
<0008> gprs_ns.c:244 NSVCI=65534 Creating NS-VC
<0008> gprs_ns.c:244 NSVCI=101 Creating NS-VC
<0008> gprs_ns.c:1602 NSEI=101 RESET procedure based on API request
<0008> gprs_ns.c:427 NSEI=101 Tx NS RESET (NSVCI=101, cause=O&M intervention)
<0001> pcu_l1_if.cpp:119 Sending activate request: trx=0 ts=1
<0001> pcu_l1_if.cpp:531 PDCH: trx=0 ts=1
<0008> gprs_ns.c:976 NSVCI=101 Rx NS RESET ACK (NSEI=101, NSVCI=101)
<0008> gprs_ns.c:536 NSEI=101 Tx NS UNBLOCK (NSVCI=101)
<0008> gprs_ns.c:1386 NSEI=101 Rx NS UNBLOCK ACK
<000a> gprs_bssgp_pcu.cpp:490 NS-VC 101 is unblocked.
<0009> gprs_bssgp_pcu.cpp:769 Sending reset on BVCI 0
<0009> gprs_bssgp_bss.c:289 BSSGP (BVCI=0) Tx BVC-RESET CAUSE=O&M intervention
<0009> gprs_bssgp_pcu.cpp:777 Sending reset on BVCI 2
<0009> gprs_bssgp_bss.c:289 BSSGP (BVCI=2) Tx BVC-RESET CAUSE=O&M intervention
<0009> gprs_bssgp_pcu.cpp:785 Sending unblock on BVCI 2
<0009> gprs_bssgp_bss.c:269 BSSGP (BVCI=2) Tx BVC-BLOCK
```

Рис. 3.10. Консоль налаштування *osmo-pcu*

А консоль *osmo-nitb* має наступний вигляд (рис. 3.11).

```
<0005> abis_nm.c:699 OC=CHANNEL(03) INST=(00,00,00) Opstart ACK
<0005> abis_nm.c:703 OC=CHANNEL(03) INST=(00,00,01) Set Channel Attributes ACK
<0005> abis_nm.c:381 OC=CHANNEL(03) INST=(00,00,01) STATE CHG: OP_STATE=Enabled AVAIL=OK(ff)
<0005> abis_nm.c:699 OC=CHANNEL(03) INST=(00,00,01) Opstart ACK
<0005> abis_nm.c:703 OC=CHANNEL(03) INST=(00,00,02) Set Channel Attributes ACK
<0005> abis_nm.c:381 OC=CHANNEL(03) INST=(00,00,02) STATE CHG: OP_STATE=Enabled AVAIL=OK(ff)
<0005> abis_nm.c:699 OC=CHANNEL(03) INST=(00,00,02) Opstart ACK
<0005> abis_nm.c:703 OC=CHANNEL(03) INST=(00,00,03) Set Channel Attributes ACK
<0005> abis_nm.c:381 OC=CHANNEL(03) INST=(00,00,03) STATE CHG: OP_STATE=Enabled AVAIL=OK(ff)
<0005> abis_nm.c:699 OC=CHANNEL(03) INST=(00,00,03) Opstart ACK
<0005> abis_nm.c:703 OC=CHANNEL(03) INST=(00,00,04) Set Channel Attributes ACK
<0005> abis_nm.c:381 OC=CHANNEL(03) INST=(00,00,04) STATE CHG: OP_STATE=Enabled AVAIL=OK(ff)
<0005> abis_nm.c:699 OC=CHANNEL(03) INST=(00,00,04) Opstart ACK
<0005> abis_nm.c:703 OC=CHANNEL(03) INST=(00,00,05) Set Channel Attributes ACK
<0005> abis_nm.c:381 OC=CHANNEL(03) INST=(00,00,05) STATE CHG: OP_STATE=Enabled AVAIL=OK(ff)
<0005> abis_nm.c:699 OC=CHANNEL(03) INST=(00,00,05) Opstart ACK
<0005> abis_nm.c:703 OC=CHANNEL(03) INST=(00,00,06) Set Channel Attributes ACK
<0005> abis_nm.c:381 OC=CHANNEL(03) INST=(00,00,06) STATE CHG: OP_STATE=Enabled AVAIL=OK(ff)
<0005> abis_nm.c:699 OC=CHANNEL(03) INST=(00,00,06) Opstart ACK
<0005> abis_nm.c:703 OC=CHANNEL(03) INST=(00,00,07) Set Channel Attributes ACK
<0005> abis_nm.c:381 OC=CHANNEL(03) INST=(00,00,07) STATE CHG: OP_STATE=Enabled AVAIL=OK(ff)
<0005> abis_nm.c:699 OC=CHANNEL(03) INST=(00,00,07) Opstart ACK
<0005> abis_nm.c:381 OC=BTS(01) INST=(00,ff,ff) BTS 0 reported connected PCU version 0.2.915-241f5
```

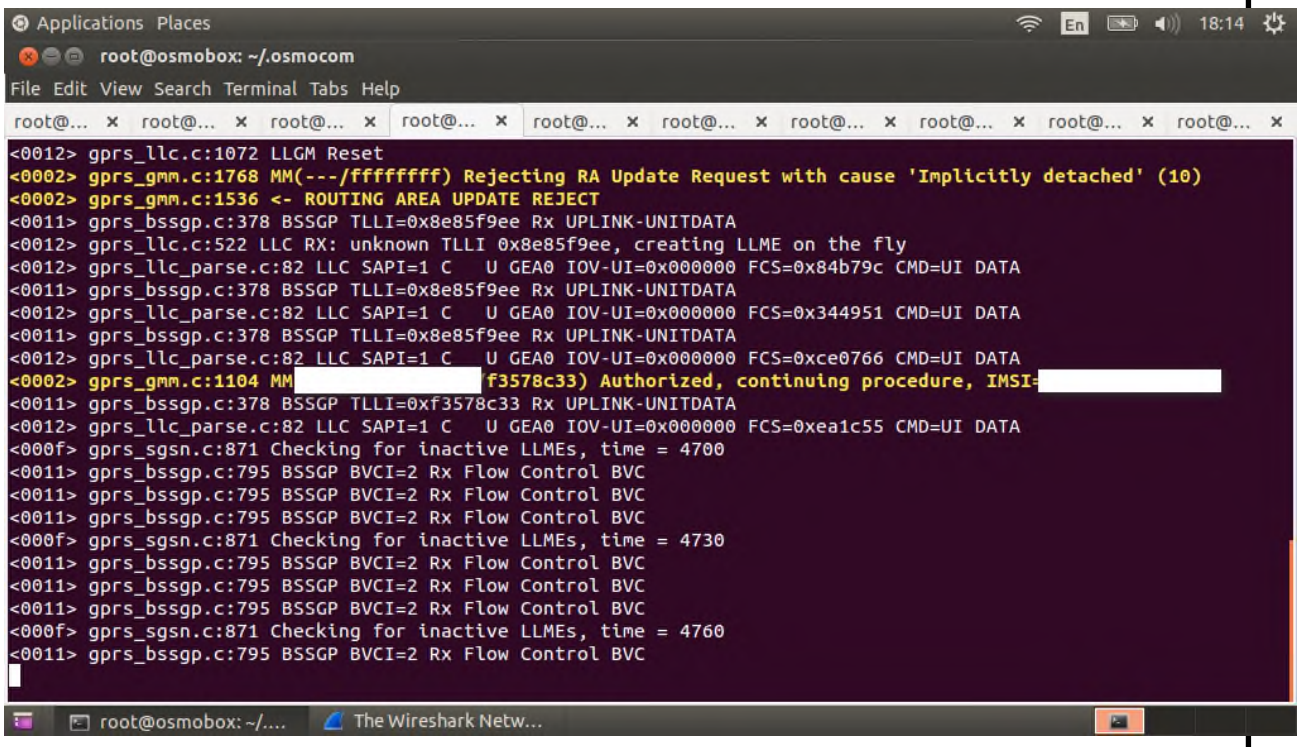
Рис. 3.11. Консоль налаштування *osmo-nitb*

При роботі в такій мережі телефон вважає, що він в роумінгу і пакетні дані в роумінгу часто відключені. Тому нічого працювати не буде, якщо не дозволите *GPRS* в роумінгу в настройках телефону. Тепер, при підключенні до мережі будуть наступні записи в консолі з *osmo-pcu* (рис. 3.12)

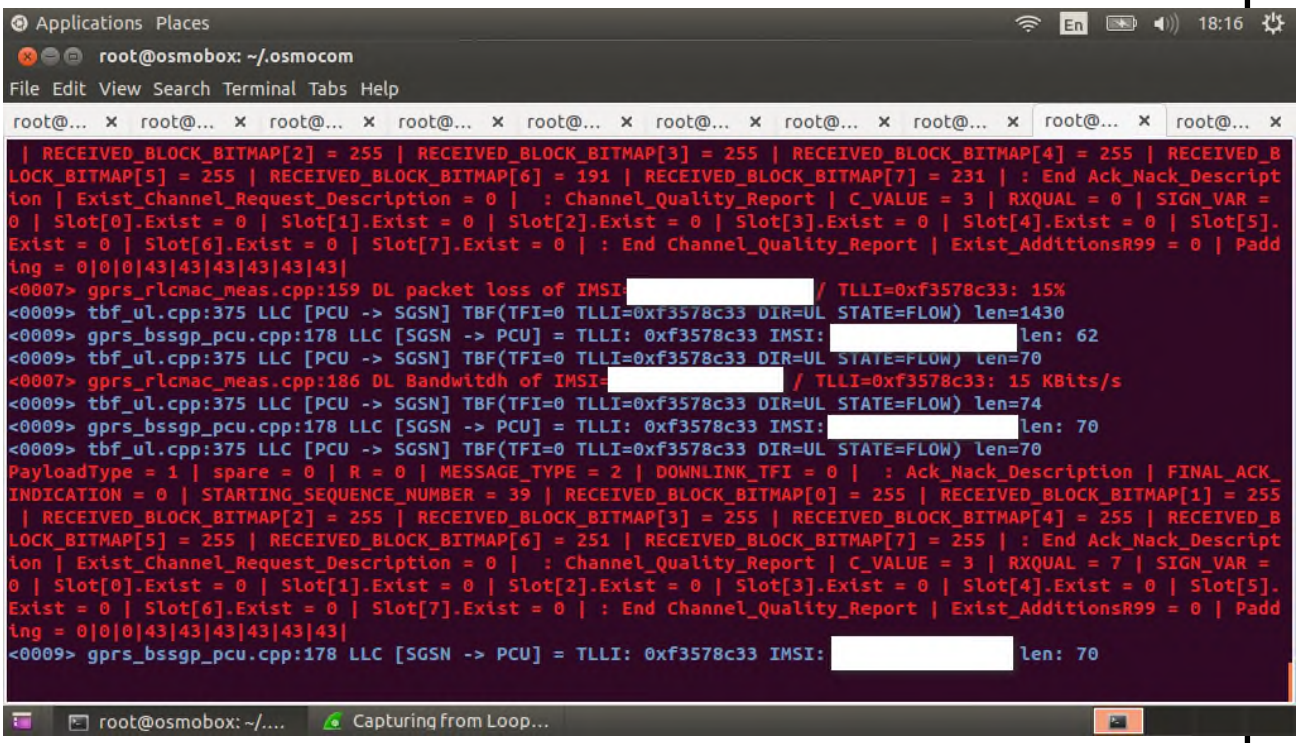
```
P[5] = 0 | RECEIVED_BLOCK_BITMAP[6] = 31 | RECEIVED_BLOCK_BITMAP[7] = 255 | : End Ack_Nack_Description | Exist_C
hannel_Request_Description = 0 |  : Channel_Quality_Report | C_VALUE = 32 | RXQUAL = 0 | SIGN_VAR = 0 | Slot[0].
Exist = 0 | Slot[1].Exist = 0 | Slot[2].Exist = 0 | Slot[3].Exist = 0 | Slot[4].Exist = 0 | Slot[5].Exist = 0 |
Slot[6].Exist = 0 | Slot[7].Exist = 0 | : End Channel_Quality_Report | Exist_AdditionsR99 = 0 | Padding = 0|0|0|
43|43|43|43|43|43|
PayloadType = 1 | spare = 0 | R = 0 | MESSAGE_TYPE = 2 | DOWNLINK_TFI = 0 |  : Ack_Nack_Description | FINAL_ACK_
INDICATION = 0 | STARTING_SEQUENCE_NUMBER = 14 | RECEIVED_BLOCK_BITMAP[0] = 0 | RECEIVED_BLOCK_BITMAP[1] = 0 | R
ECEIVED_BLOCK_BITMAP[2] = 0 | RECEIVED_BLOCK_BITMAP[3] = 0 | RECEIVED_BLOCK_BITMAP[4] = 0 | RECEIVED_BLOCK_BITMA
P[5] = 0 | RECEIVED_BLOCK_BITMAP[6] = 63 | RECEIVED_BLOCK_BITMAP[7] = 255 | : End Ack_Nack_Description | Exist_C
hannel_Request_Description = 0 |  : Channel_Quality_Report | C_VALUE = 31 | RXQUAL = 0 | SIGN_VAR = 0 | Slot[0].
Exist = 0 | Slot[1].Exist = 0 | Slot[2].Exist = 0 | Slot[3].Exist = 0 | Slot[4].Exist = 0 | Slot[5].Exist = 0 |
Slot[6].Exist = 0 | Slot[7].Exist = 0 | : End Channel_Quality_Report | Exist_AdditionsR99 = 0 | Padding = 0|0|0|
43|43|43|43|43|43|
<0007> gprs_rlcmac_meas.cpp:186 DL Bandwitdh of IMSI=[redacted] / TLLI=0xf3578c33: 0 KBits/s
PayloadType = 1 | spare = 0 | R = 0 | MESSAGE_TYPE = 2 | DOWNLINK_TFI = 0 |  : Ack_Nack_Description | FINAL_ACK_
INDICATION = 1 | STARTING_SEQUENCE_NUMBER = 15 | RECEIVED_BLOCK_BITMAP[0] = 0 | RECEIVED_BLOCK_BITMAP[1] = 0 | R
ECEIVED_BLOCK_BITMAP[2] = 0 | RECEIVED_BLOCK_BITMAP[3] = 0 | RECEIVED_BLOCK_BITMAP[4] = 0 | RECEIVED_BLOCK_BITMA
P[5] = 0 | RECEIVED_BLOCK_BITMAP[6] = 127 | RECEIVED_BLOCK_BITMAP[7] = 255 | : End Ack_Nack_Description | Exist_
Channel_Request_Description = 0 |  : Channel_Quality_Report | C_VALUE = 28 | RXQUAL = 3 | SIGN_VAR = 0 | Slot[0]
.Exist = 0 | Slot[1].Exist = 0 | Slot[2].Exist = 0 | Slot[3].Exist = 0 | Slot[4].Exist = 0 | Slot[5].Exist = 0 |
 Slot[6].Exist = 0 | Slot[7].Exist = 0 | : End Channel_Quality_Report | Exist_AdditionsR99 = 0 | Padding = 0|0|0
|43|43|43|43|43|43|
<0007> gprs_rlcmac_meas.cpp:159 DL packet loss of IMSI=[redacted] / TLLI=0xf3578c33: 0%
```

Рис. 3.11. Записи в консолі з *osmo-pcu*

І запис про авторизацію в консолі *osmo-sgsn*



А при активації *GRPS* сервісу в телефоні ви побачите, що передача даних почалася

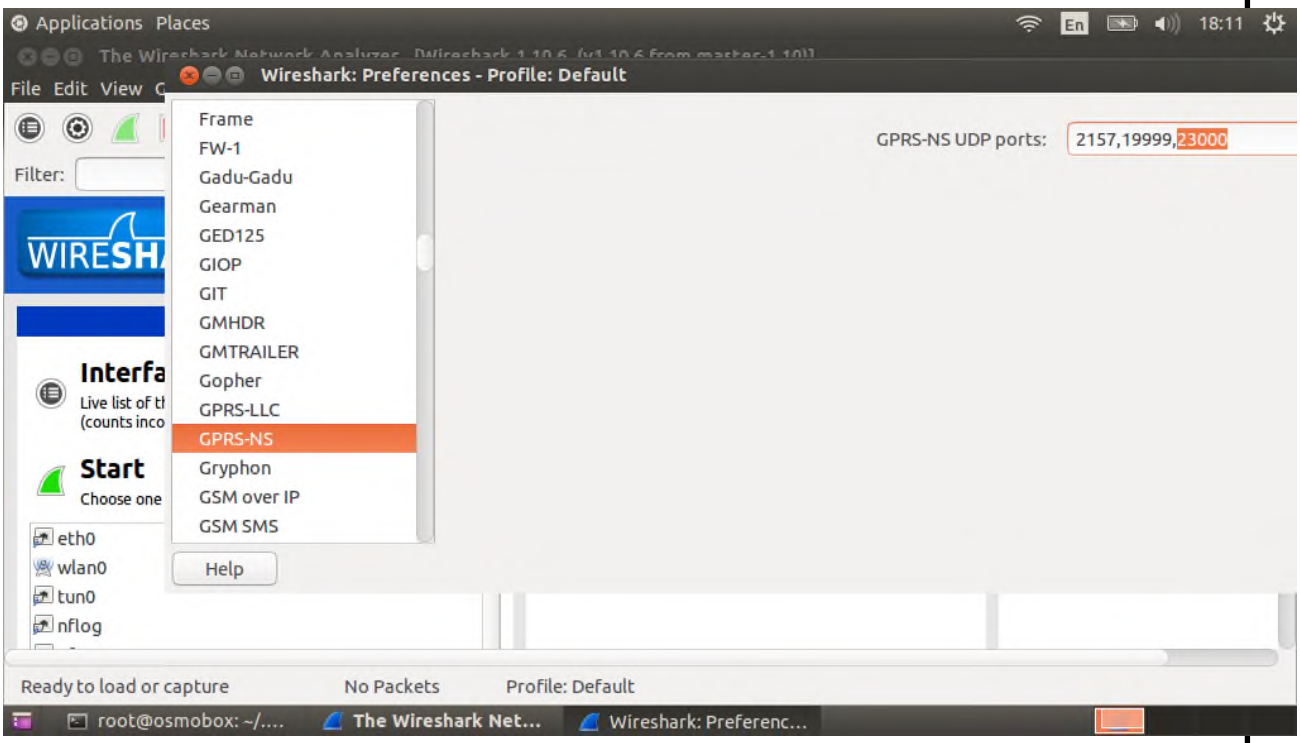


Зверніть увагу на той факт, що швидкість передачі даних в *GPRS* дуже низька, в той же час сучасні телефони при отриманні доступу до мережі тут же починають процес перевірки оновлень, пошти, новин. Всі програми починають оновлювати свої дані. Це може привести до того, що буде важко відкрити щось в браузері, так як, крім низької пропускної здатності, можуть відбуватися втрати пакетів.

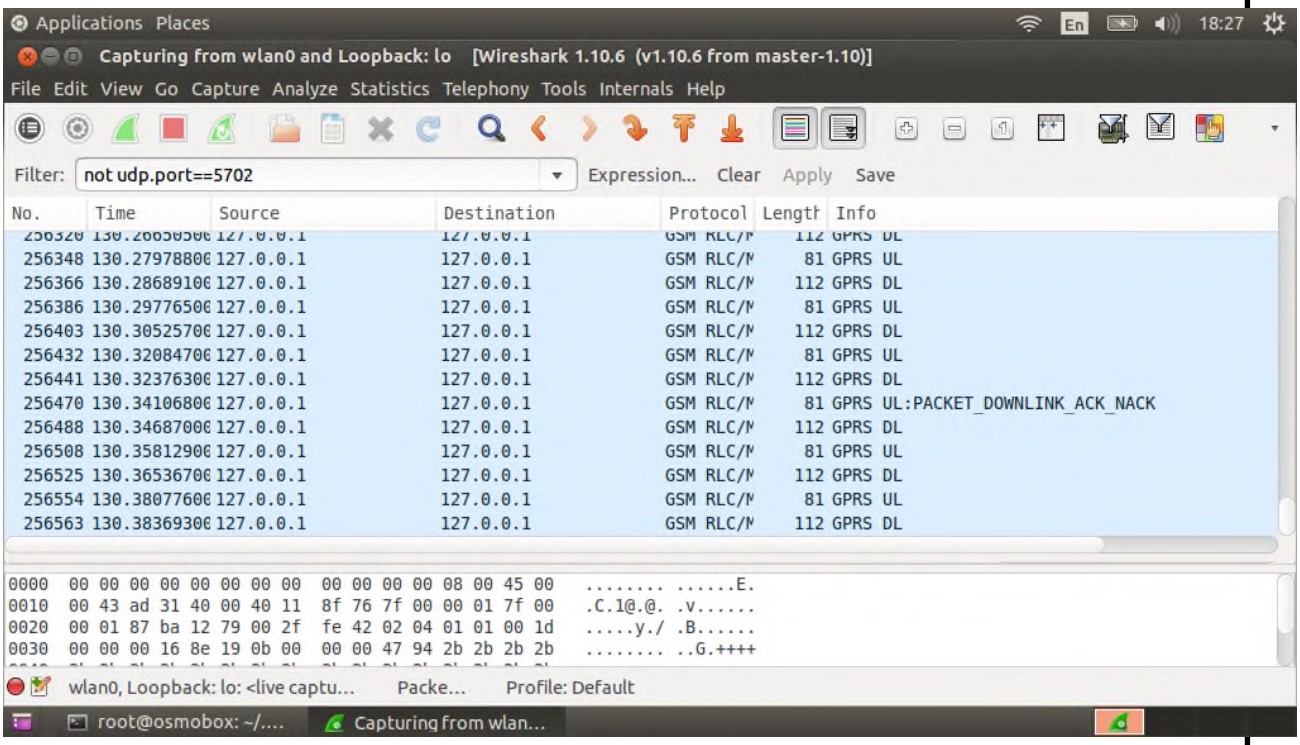
Можна виправити становище обмеживши доступ на машині, яка роздає Інтеренет для підмережі 192.168.0.0/24 (*tun*0), залишивши доступними тільки деякі ресурси.

Робочі вікна програмного комплексу аналізу трафіку

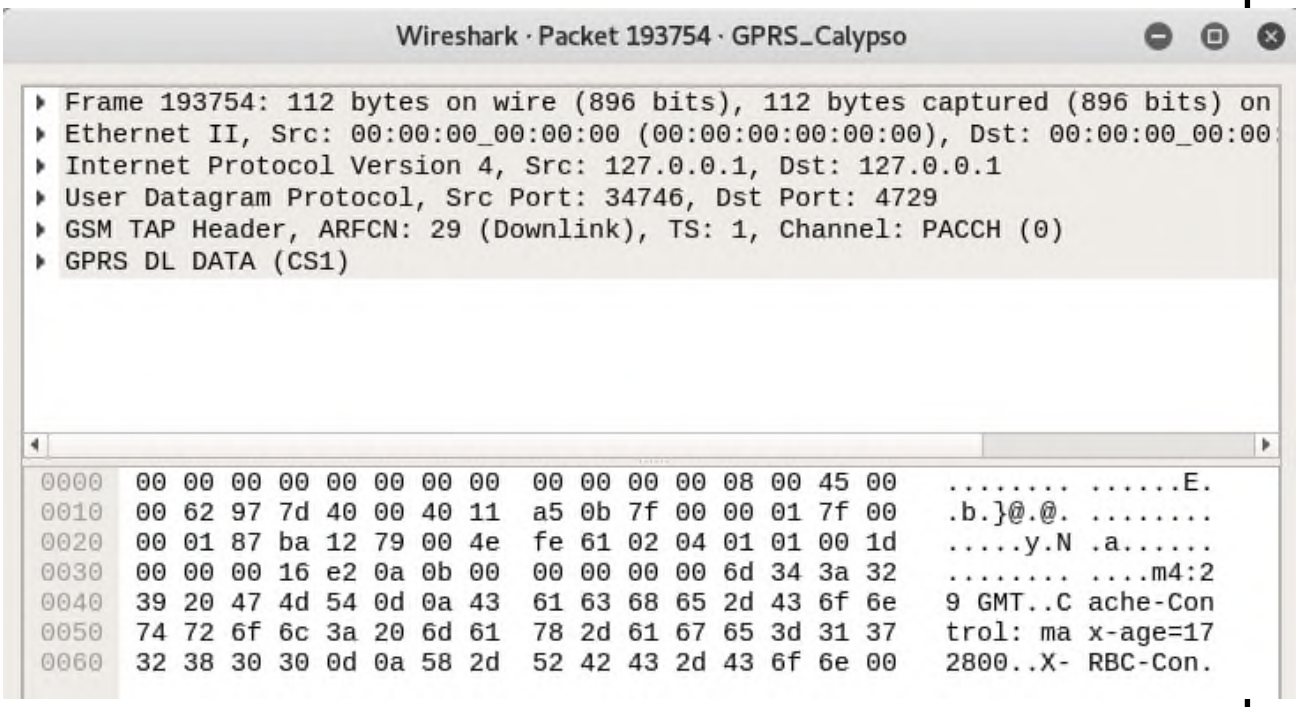
Щоб *wireshark* автоматично розбирав *GPRS* трафік, потрібно в настройках протоколу *GPRS-NS* додати порт 23000.



Підключившись до мережі, я буду прослуховувати інтерфейс *wlan*0 і вивчати трафік.

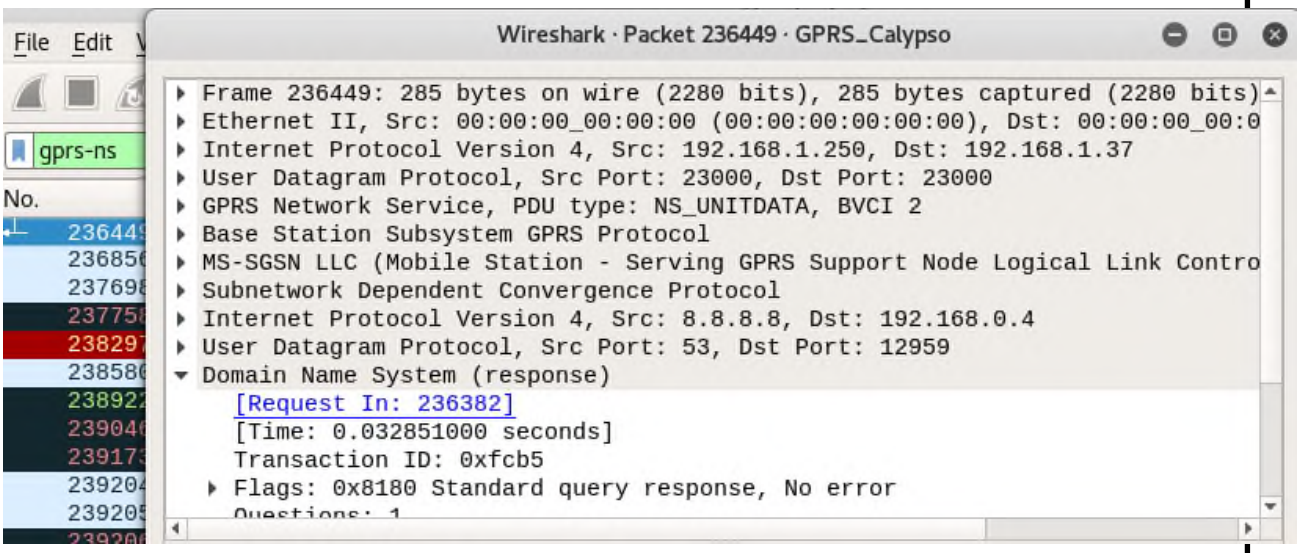


Ми бачимо *GSM* пакети (зверніть увагу на *ASCII* уявлення даних. Видно, що це *HTTP* запит)

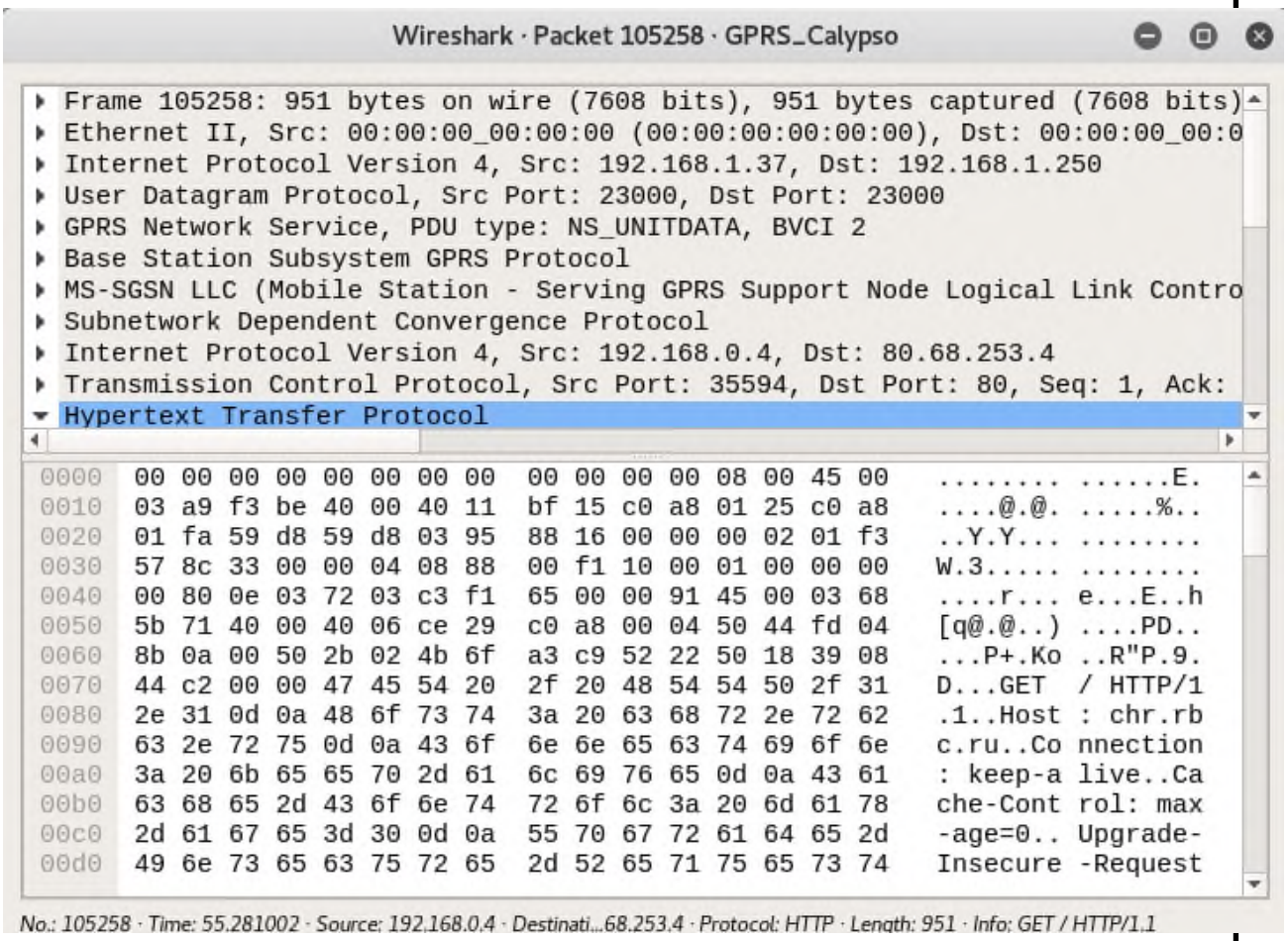


Тут можна знайти так само класичний *TCP* / *IP* трафік, наприклад *HTTP* або *DNS* запити. Можна використовувати фільтр *gprs-ns*. Зверніть увагу на вкладеність *TCP* / *IP* протоколів в *GSM* протоколи.

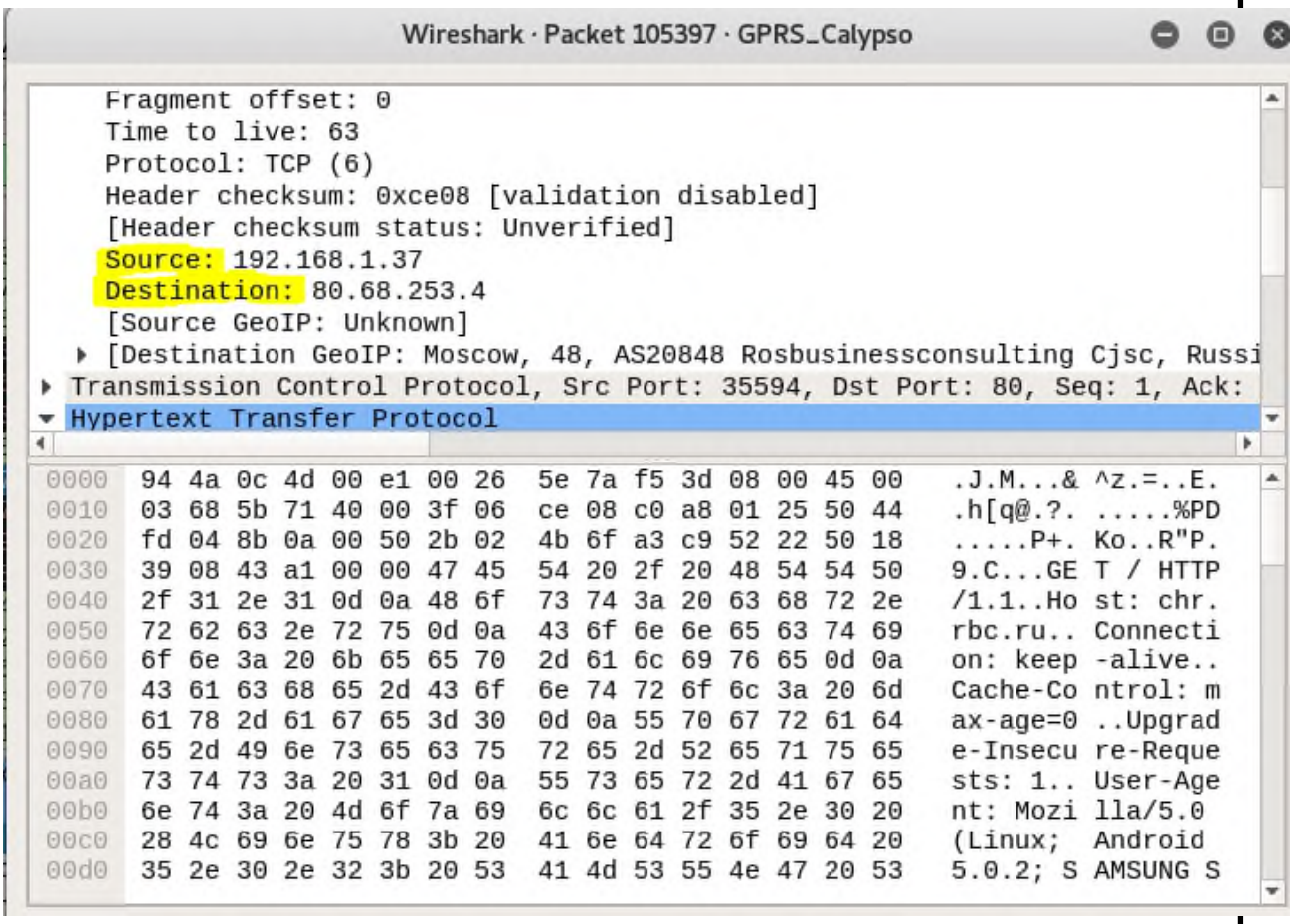
DNS



HTTP



Природно, нам доступний і класичний *TCP* / *IP* трафік, який вже йде безпосередньо від *wlan*0 в Інтернет

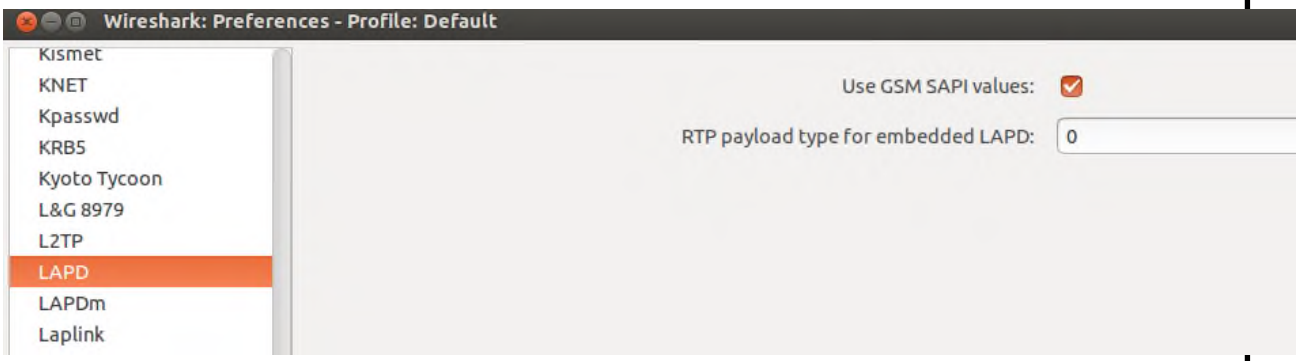


На цьому етапі отримуємо повний контроль над трафіком і можемо провести повний спектр *MitM* атак проти абонентів *GSM* мережі.

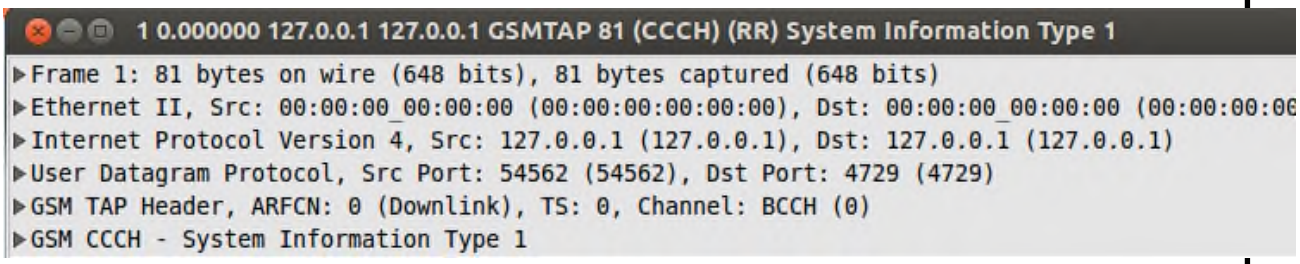
У разі створення фейковий базової станції, абонент стає недоступний для дзвінків ззовні і в стандартній конфігурації сам не може більше зателефонувати комусь із своєї телефонної книги, але зате у нього буде доступний Інтернет і висока ймовірність, що він спробує ним скористатися. Тут він і може бути атакований зловмисником.

GSMTAP

Для коректного відображення протоколів *GSM* було налаштовано релиретк (рис.)



GSM протоколи будуть інкапсулюватися в *UDP* пакети з заголовком *GSMTAP* при передачі через *Um* інтерфейс або в *TCP* пакети з заголовками *OML*, *RSL* при трасуванні *A-bis* інтерфейсу.



Додатки, здатні генерувати *GSMTAP* трафік діють у такий спосіб:

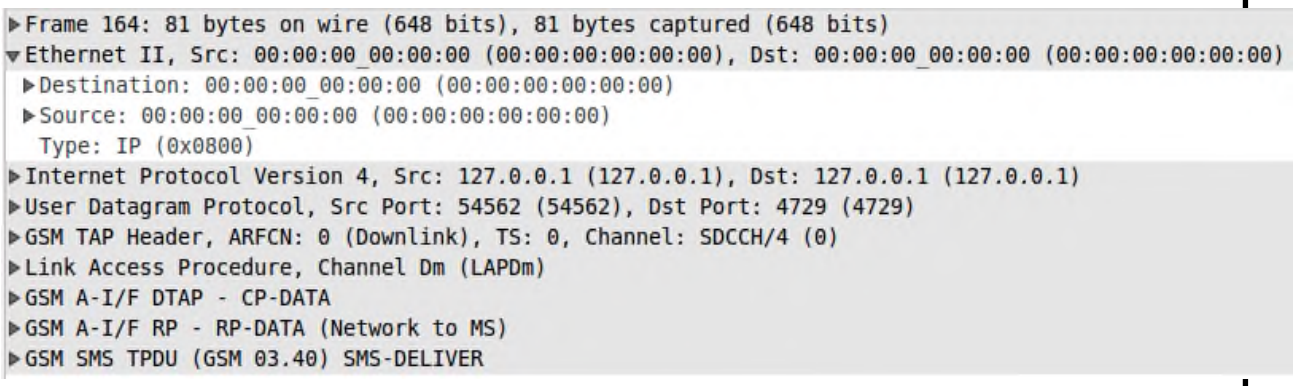
– отримують *Um* фрейм по радіо інтерфейсу;

– додають *GSMTAP* заголовок;

– відправляють все це на вказаний *IP* адреса в *UDP* пакетах (в нашому випадку на *loopback*).

Крім фільтра *GSMTAP*, можна використовувати інші фільтри, що починаються з *gsm*, наприклад *gsm_sms*, для пошуку *SMS*-повідомлень в трафіку.

Але використання фільтра *GSMTAP* дозволить переглядати весь *GSM Um* трафік, оскільки всі інші заголовки вкладені в *GSMTAP*.

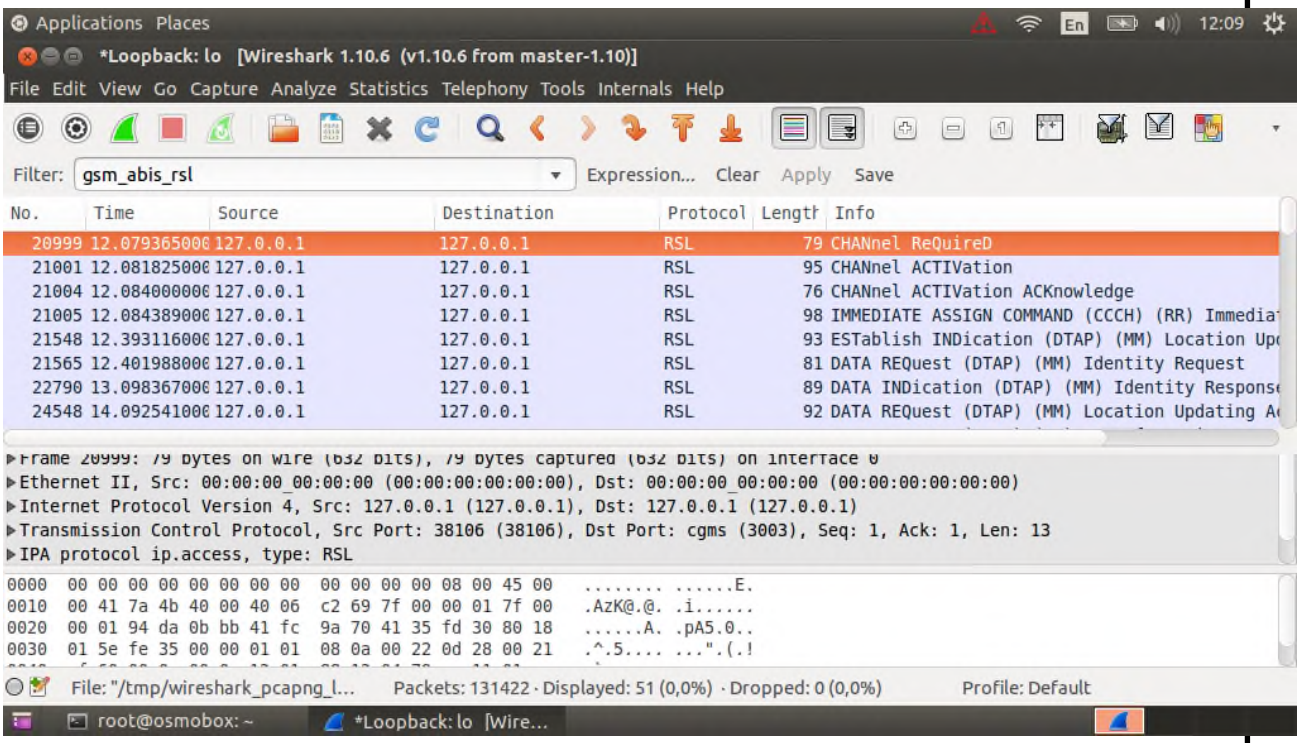


Для вивчення *GSM* трафіку його потрібно спочатку записати. Використовуючи *SDR* пристрій в якості приймача, можна вивчати дані, що передаються на

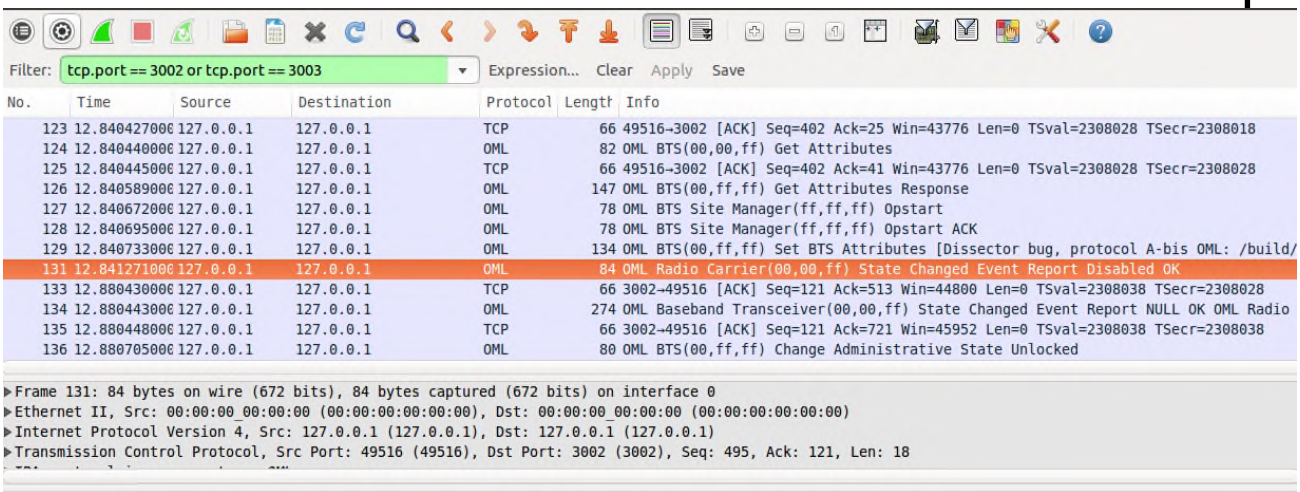
Якщо запустили *GSM* мережу на основі двох *OsmocomBB*-сумісних телефонів можна передавати *GSMTAP* пакети з *OsmoBTS* в *Wireshark* і таким чином вивчати роботу мережі.

A-bis – інтерфейс обміну повідомленнями між *BTS* і *BSC*. Щоб переглянути *RSL* повідомлення, потрібно почати прослуховувати *loopback* інтерфейс (зверніть увагу, що тут використовується *TCP*, а не *UDP*) і ви побачите повідомлення на кшталт цих:

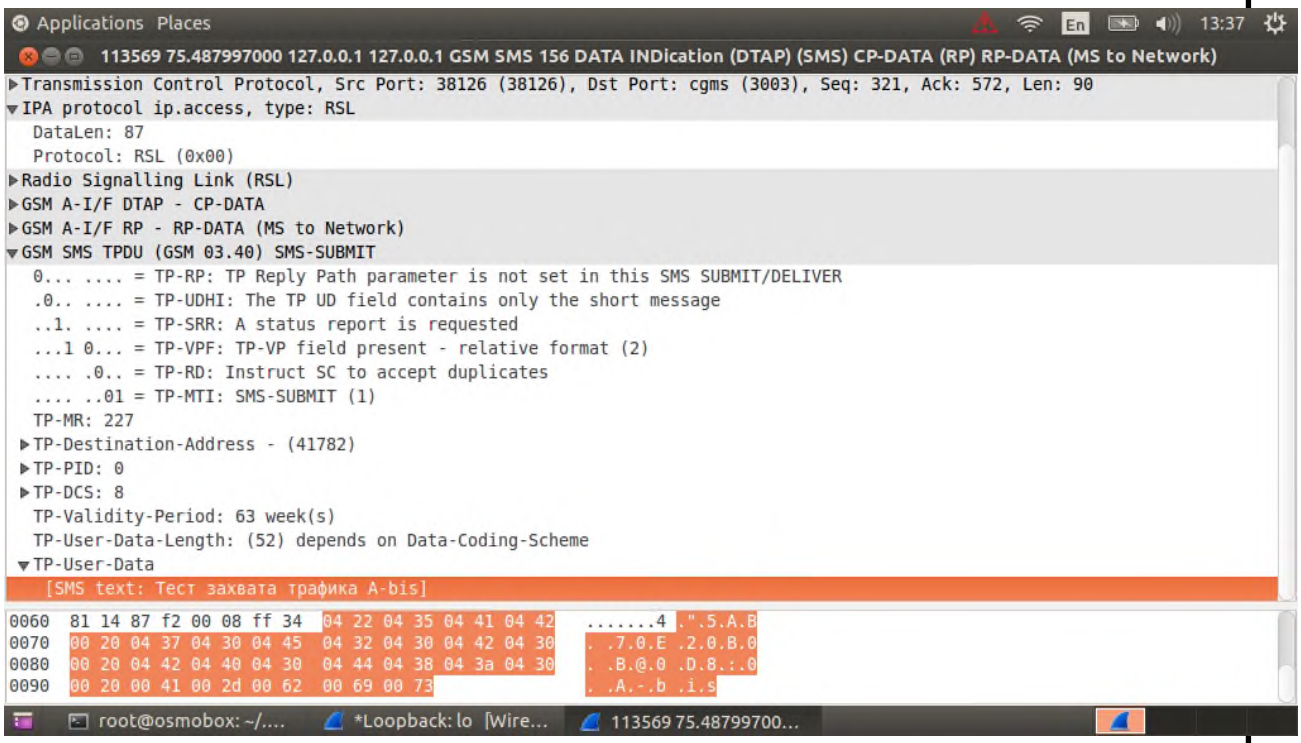
Для *RSL* можна використовувати фільтр *gsm_abis_rsl* (рис.).



Для *OML* можна скористатися фільтром *gsm_abis_oml* або фільтрувати по портам 3002 і 3003.



У той же час *SMS* повідомлення будуть вкладені в *RSL* пакети, а не в *GSMTAP*, як у випадку з передачею через *Um* інтерфейс.



3.6. Висновки до розділу

Ми переглянули різне програмне забезпечення для моніторингу та причини основних потреб моніторингу як явища. Що стосується рішень для всього підприємства, порівняно CA Unicenter та HP OpenView. Підхід CA дуже гнучкий, але частково застарілий. Його системи були розроблені майже десять років тому і потребують серйозного перегляду у питаннях систем кластеризації та підтримки мережевих вузлів. Однак, коли мова йде про звичайні сервери, він добре розуміє системи, які потребують точного моніторингу, такі як нагляд за базами даних та банківськими веб-серверами. Що стосується HP OpenView, він має набагато більш структуроване рішення, яке головним чином інтегровано з міжнародними стандартами бібліотеки IT-інфраструктури. Це можна вважати найбільш висококласним рішенням, яке слід використовувати для великих підприємств та тисяч серверів. З іншого боку, Програмне забезпечення Nagios Core з відкритим кодом забезпечує зручний та ефективний спосіб контролю невеликих приватних інфраструктур за відносно низькою вартістю. Оригінальна основа Linux для цих систем забезпечує операційну стабільність та безпеку, що

може бути дуже важливим для певних органів влади. Беручи до уваги ці причини, ми можемо сміливо заявити, що це рішення слід застосовувати набагато ширше, включаючи державні органи, такі як лікарні, банки та податкові служби, де вартість помилок занадто висока. Ми також розглянули всі позитивні риси Nagios в налаштовуваності. Наскільки ця система відкрита (і навіть вимагає певного стороннього програмного забезпечення для роботи з різними операційними системами), вона має широкий спектр можливостей, які можна вдосконалити. Ми використали цю можливість у роботі і створили надбудову для Nagios Core, який працює на вбудованому інтерпретаторі Perl. Це показує, наскільки легко зануритися та налаштувати відкриту систему, таку як Nagios.

ВИСНОВКИ

У цій роботі ми переглянули інструменти для моніторингу різних сфер комп'ютерного середовища. Ми проаналізували різні види підходів до управління системами та визначили їх плюси та мінуси. Дуже важливою частиною роботи є розробка відкритих технологій систем спостереження. Це показує, що досить просто і зручно створити безпечну та надійну інфраструктуру за низькою вартістю, а отже, збільшити продуктивність IT в цілому. Ми також розглянули розробку нових додаткових функцій до рішення Nagios Core для покращення його можливостей та зручності використання. Сильним моментом є те, що це відкрита система, і тому кожному дозволено та заохочено зробити свій внесок. Ми довели, що цей внесок є порівняно простим.

Я працюю в галузі управління системами вже майже три роки. Ця сфера була серйозною проблемою і раніше, коли мережеві інфраструктури великих підприємств почали широко зростати за попередні кілька десятиліть. Комп’ютери принесли таку автоматизацію, яку люди майже не відчували раніше; з точки зору промислового виробництва це фактично був глобальний прорив. Зі збільшенням кількості завдань, що вирішуються комп’ютерними системами, зростав попит на стабільне середовище. І в цьому пункті люди прийшли до програмного моніторингу, який фактично ввів терміни безвідмовної роботи з часом роботи майже близько 100%.

Коли ви використовуєте свою пластикову картку для оплати подарунка на день народження дочки, розсилки грошей по всьому світу або навіть придбання свіжих фруктів на вечерю - ви не хочете, щоб ваша банківська операція провалилася, і ви точно оберете банк, який надає відповідна послуга. Якщо ви відправляєте своєму начальнику SMS із поясненнями, чому ви запізнилися, ви хочете, щоб його доставили вчасно! Це те, що забезпечують вам невидимі спостерігачі, працюючи день і ніч, щоб люди були впевнені у всьому іншому, ніж інші люди.

Державні органи та великі підприємства розробляють теоретичну базу для всіх IT-послуг як ключову мету для стабільного зростання та підтримки

виробництва. Хорошим прикладом такої участі є Бібліотека інфраструктури інформаційних технологій (ITIL), розроблена Управлінням урядової торгівлі Великобританії (OGC). Ця бібліотека визначає всі процедури управління IT-інфраструктурою, з якими теоретично може зіткнутися будь-яке підприємство. Основними цілями, як очікувалось, є питання управління ризиками та надійності. Це є причиною того, що багато інструментів та програмного забезпечення для управління системою побудовані навколо цього інтелекту.

Щодо потреб, у жовтні 2005 року IDC (Міжнародна корпорація даних) провела велике опитування, яке показало наступний інтерес до моніторингу створення з боку великих гравців IT-галузі:

Звичайно, вони заробляють гроші, але, як ви можете спостерігати, людські помилки посідають друге місце, і в цілому людські помилки плюс скорочення простоїв становлять 35%. Ось в чому справа. Ви більше не можете покладатися на людей при роботі зі складною інфраструктурою.

Оскільки прогрес змушує нас пришвидшити зусилля та покращити сервіс та продукцію, а постійна боротьба за сучасність зростає, люди починають більше дивитись у область надійності комп'ютерних систем та мереж. Банки, лікарні, державні органи, приватні підприємства, великі промислові виробництва - всі вони мають великий попит на безвідмовність у питаннях роботи комп'ютерних систем. Технічний рівень та доступність програмного забезпечення для моніторингу, що існує сьогодні, дозволяють здійснювати цілодобове спостереження. Серед них існують відкриті джерела рішень для цього, тому не так вже й релевантно застосовуватись для приватних комп'ютерних систем, побудованих простими людьми.

Дослідження, яке буде представлено, - це аналіз різних рішень для моніторингу, розроблених для різних цілей. Серед них будуть:

- CA (Computer Associates) - корпоративне рішення Unicenter NSM
- Комплексне рішення управління послугами IT-служби HP (Hewlett-Packard) із усіма інструментами та утилітами управління активами

- Nagios Core (відкритий вихідний код) + практична реалізація на вбудованій системі.

Практична реалізація представлятиме всі можливості програмного забезпечення для системного спостереження, доступне на той момент без жодної ціни. Системи CA та HP будуть переглянуті на практичному досвіді використання в корпоративній сфері. Ми також розробимо спеціальний плагін Perl для програмного забезпечення Nagios Core для контролю надійності обладнання за віком.

Предметом дослідження в цій роботі буде шлях вдосконалення IT-послуг шляхом встановлення автоматичного нагляду. Ми переглянемо такі об’єкти, як різні програмні послуги, орієнтовані на різні сфери застосування. Одним із завдань цього дослідження будуть шляхи вдосконалення існуючих доступних рішень. У нашому випадку ми розробимо спеціальний плагін Perl для системного програмного забезпечення з відкритим кодом Nagios Core для моніторингу технічного стану обладнання, встановленого в середовищі.

СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ДСТУ 3008-95. Документація. Звіти у сфері науки і техніки. Структура і правила оформлення. – 39 с.
2. Бойченко С.В., Іванченко О.В. Положення про дипломні роботи (проекти) випускників Національного авіаційного університету. – К.: НАУ, 2017. – 63 с.
3. *Mockapetris*. *P Domain Names – Concepts and Facilities*. *– Network Working Group*. – 2017.
4. *Schulman A. Computer And Internet Surveillance in the Workplace*: *Rough Notes, US*, 2010-2012 *http*://*www.sonic.net*
5. *Schulman A. The Extent of Systematic Monitoring of Employee E–mail and Internet Use, US*, 2010-2012 *http*://*www.sonic.net.*
6. *Vodafone* інвестував 21,6 млрд гривень у швидкісний інтернет в Україні. – Українці встановили рекорд споживання мобільного інтернету в зимові свята: втричі більше трафіку. – Новина на сайті *Vodafone.ua*. – Постійне посилання: *https*://*www.vodafone.ua*/*news*/*vodafone-investments*. – Дата оновлення: 05.01.2021 р.
7. Голуб А. Правила программирования на *C* и *C*++. – М.:"Бином", 2016. – 241 с.
8. Зобнин Е. Устоять любой ценой. Методы борьбы с *DoS*/*DDoS*–атаками. Журнал "Хакер", №5. – 2020. – С. 35-45.
9. Крис Касперски Компьютерные вирусы изнутри и снаружи. — СПб.: Питер, 2016. – 526 с. – *ISBN* 5–469–00982–3
10. Левоневский Д.К., Фаткиева Р.Р. Разработка системы обнаружения аномалий сетевого трафика. –Научный вестник НГТУ, том 56, № 3. – 2014. – с. 108–114. *ISSN* 1814-1196.
11. Нечаев В.И. Элементы криптографии. Основы теории защиты информации, М. – 2015. – 112 с.

12. Семкин С.Н., Беляков Э.В., Гребенев С.В. Основы организационного обеспечения информационной безопасности объектов информатизации, П.: *BHV* – 2016 – 192с.
13. Українці встановили рекорд споживання мобільного інтернету в зимові свята: втричі більше трафіку. – Новина на сайті *Vodafone.ua*. – Постійне посилання: *https://www.vodafone.ua/news/ukranci-vstanovili-rekord-spozh-mobilnogo-internetu-v-zimovi-svyata*-2021. Дата оновлення: 21.01.2021 р.
14. Цирлов В. Л. Основы информационной безопасности. Краткий курс, Издательство: Феникс. – 2008 – 256 с.
15. Шаньгин В. Ф. Защита компьютерной информации. – М: ДМК. 2008 – 544с.
16. Шелупанов А. А., Мещеряков Р. В., Лось В. П., Белов Е. Б. Основы информационной безопасности, М: Горячая линия –Телеком, 2016. – 544с.
17. *BinaryTides – Raw socket programming on windows with winsock*. – [Електронний ресурс] *http://www.binarytides.com/raw-sockets-using-winsock/*
18. *IBM Knowledge Center – How sockets work*. – [Електронний ресурс] *https://www.ibm.com/support/knowledgecenter/en/ssw_i*5_54/*rzab*6/*howdosockets.htm*
19. *Microsoft Developer Network*. [Електронний ресурс] – Режим доступу до ресурсу: *https://msdn.microsoft.com/ru-ru/library/hh*279654.*aspx*.
20. *Qt – Home*. – [Електронний ресурс] *https://www.qt.io/ru/*
21. Блокнот *IT*-шника – обзор методов анализа и мониторинга сетевого трафика. [Електронний ресурс] *http://it-bloknot.ru/?q=content*/обзор-методов-анализа-и-мониторинга-сетевого-трафика
22. Компьютерные сети и технологии. – [Електронний ресурс] *http://www.xnets.ru/plugins/content/content.php?content*.156.5
23. ООО Стек. – [Електронний ресурс] *http://www.stekspb.ru/autsorsing-it-infrastruktury/articles/it-glossary/local-network-monitoring*.

Додаток А

Лістинг коду основного програмного модулю