

ДОДАТОК А

Графічний матеріал кваліфікаційної роботи

Програмні засоби моніторингу корпоративної комп'ютерної мережі

Кваліфікаційна робота

Виконав:
студент гр. КІУКІ-21-3
Шевченко Д.О.

Керівник:
ас. каф. ЕОМ
Бондаренко М.Е.

Мета та завдання

2

Метою кваліфікаційної роботи є розробка та реалізація програмних засобів моніторингу корпоративної комп'ютерної мережі з використанням сучасних інструментів аналізу, зокрема Matlab для моделювання й обробки даних та Google Colab для хмарної візуалізації та інтерпретації результатів, що дозволяє підвищити якість контролю та своєчасного реагування на порушення в роботі мережевої інфраструктури..

Завдання:

- проаналізувати існуючі підходи до моніторингу комп'ютерних мереж і виділити їх переваги та обмеження;
- сформулювати вимоги до системи моніторингу з урахуванням особливостей корпоративної мережі;
- розробити алгоритм збору, обробки та аналізу мережевих даних на основі синтетичних або реальних показників;
- реалізувати програмну частину моніторингової системи у середовищі Matlab з подальшою інтеграцією до Google Colab;
- провести тестування розробленої системи та виконати аналіз результатів моніторингу.

Багаторівнева архітектура функціонування системи

3



Аналіз популярних систем

4

Критерій	Zabbix	PRTG	Nagios	SolarWinds
Ліцензування та вартість	Відкритий код Безкоштовно	Комерційна €1600+/рік	Відкритий код Безкоштовно	Комерційна \$2000+/рік
Складність розгортання	3	1	4	2
Масштабованість	Висока	Середня	Висока	Висока
Підтримувані протоколи	SNMP, Agent, JMX, IPMI, SSH, HTTP, LDAP, ODBC	SNMP, WMI, Packet Sniff, NetFlow, HTTP, SSH	SNMP, NRPE, NSCA, SSH, HTTP, SMTP, Plugins	SNMP, WMI, NetFlow, sFlow, IPFIX, API, Syslog
Візуалізація та звіти	★★★★☆	★★★★★	★★★★☆	★★★★★
Спільнота та підтримка	A	B	A	A
Гнучкість налаштувань	90%	70%	80%	75%

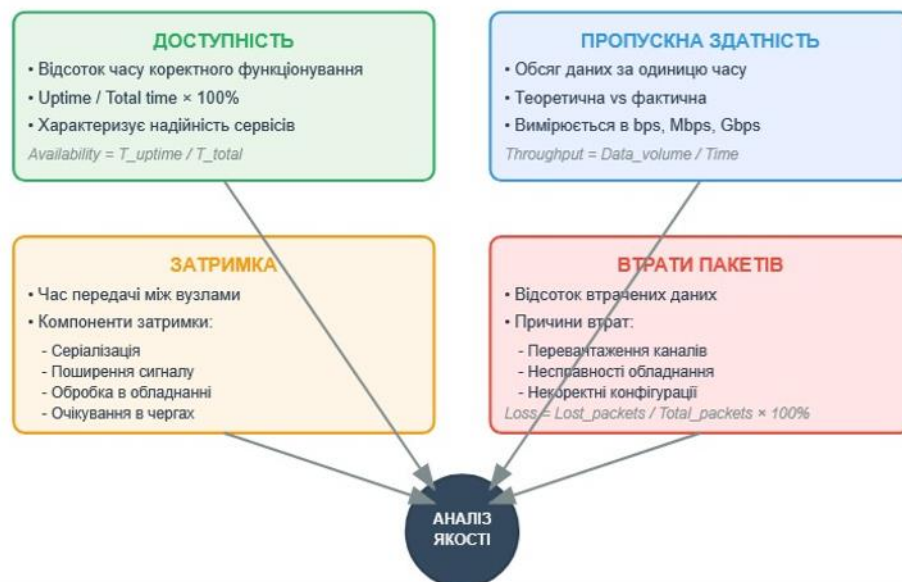
Порівняння протоколів моніторингу

5

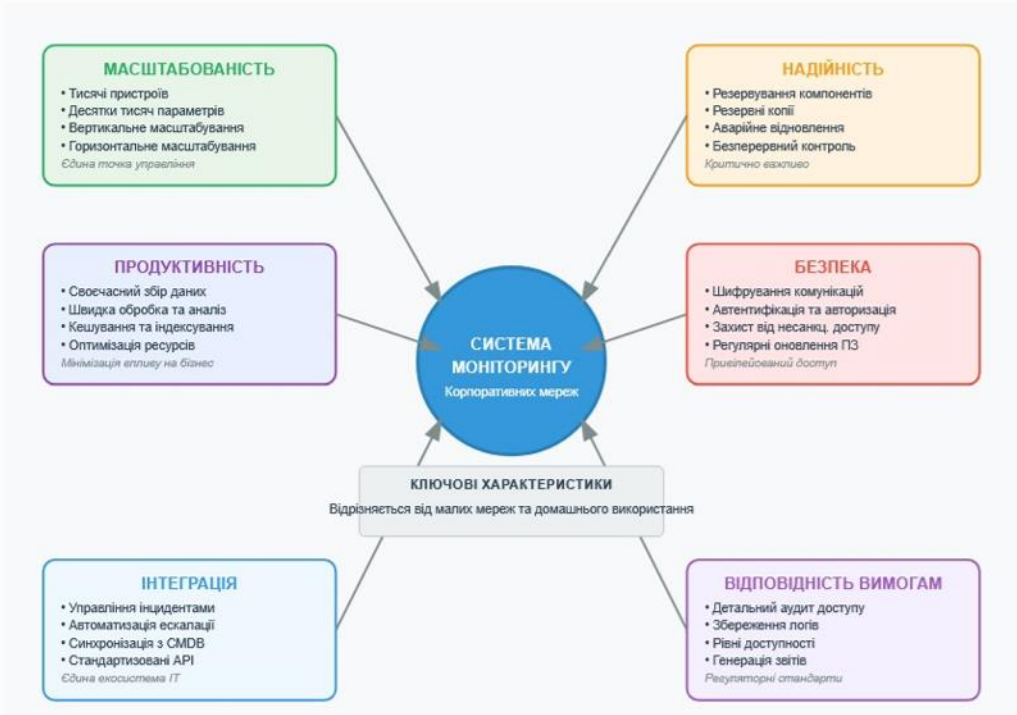
Критерій	SNMP	ICMP	NetFlow	Syslog
Призначення	Збір метрик пристроїв	Тестування зв'язності	Аналіз трафіку	Збір логів подій
Навантаження	● Середнє	● Низьке	● Високе	● Середнє
Детальність	80%	30%	90%	60%

Основні метрики мережевих систем

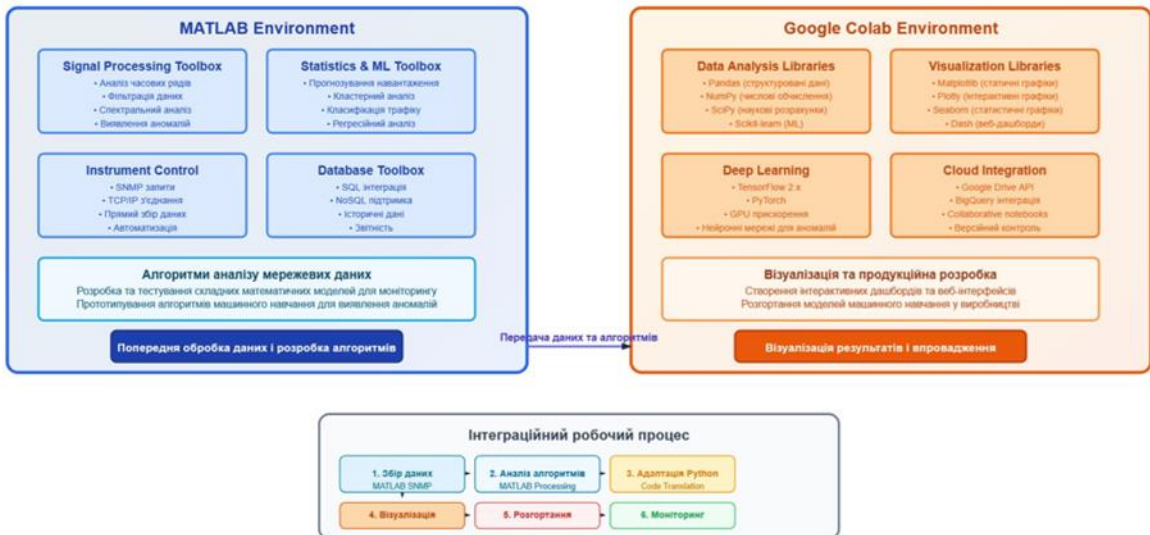
6



Вимоги до моніторингу



Інтеграція Matlab та Google Colab для розробки систем мережевого моніторингу



Інтеграційний робочий процес. Переваги гібридного методу

9



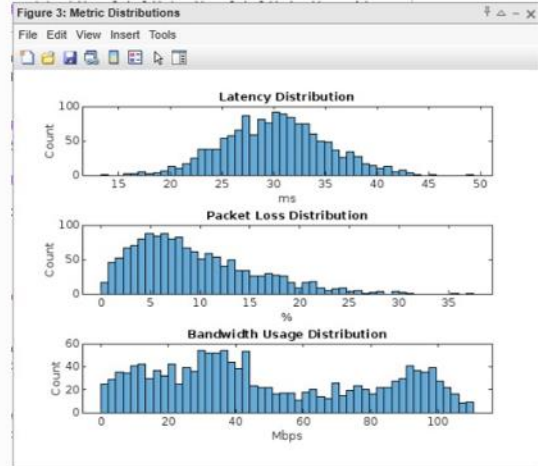
Інтеграційний робочий процес. Переваги гібридного методу

9



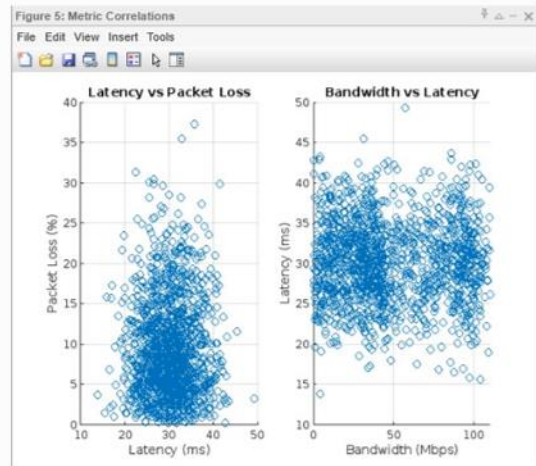
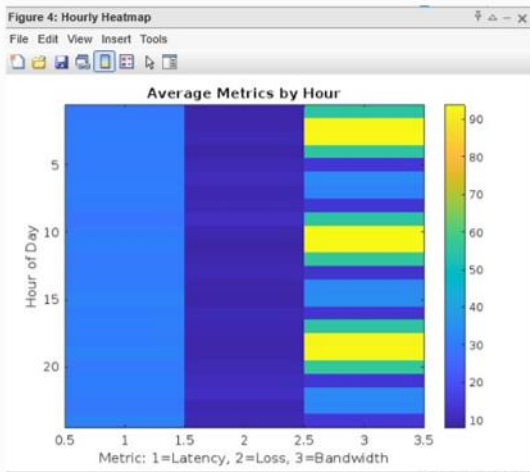
Результати роботи. Реалізація в Matlab

11



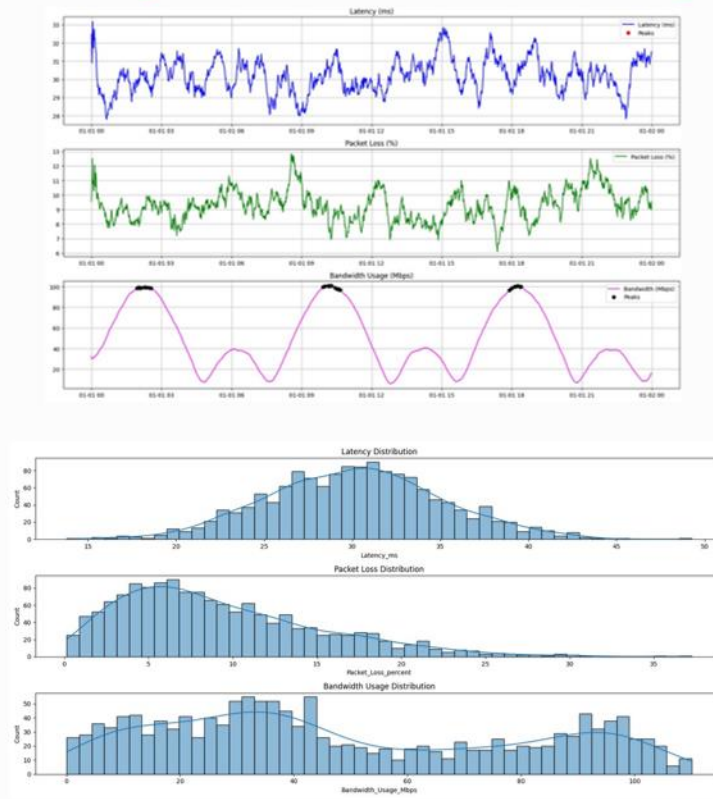
Результати роботи. Реалізація в Matlab

12



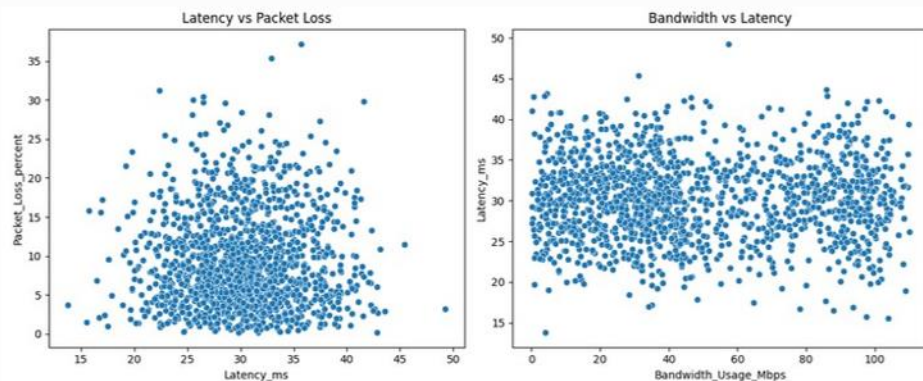
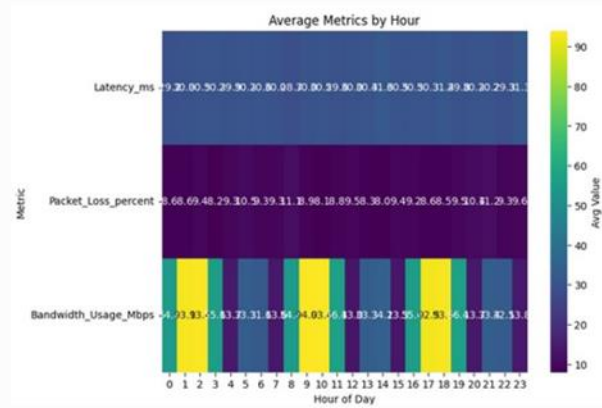
Результати роботи. Реалізація в Google Colab

13



Результати роботи. Реалізація в Google Colab

14



Висновки

15

У ході виконання кваліфікаційної роботи було реалізовано програмні засоби моніторингу корпоративної комп'ютерної мережі, які передбачають інтеграцію локального інструментарію Matlab із хмарними засобами обробки даних у середовищі Google Colab. Застосування багаторівневої архітектури моніторингу забезпечило послідовний перехід від збору первинних даних до їх аналітичної інтерпретації, що дало змогу не лише виявляти поточні аномалії, а й формувати основи для прогнозного аналізу стану мережі.

Результати синтетичного моделювання показали доцільність використання обраних метрик: затримки, пропускну здатності, втрат пакетів та доступності як основних індикаторів продуктивності та надійності мережевої інфраструктури. Побудовані візуалізації дозволили ідентифікувати типові сценарії навантаження та часові інтервали з підвищеним ризиком порушення якості сервісів. Реалізовані механізми згладжування, виявлення пікових значень та вивчення кореляцій між параметрами продемонстрували високу ефективність для формування оперативної аналітичної інформації.

Інтеграція Matlab і Google Colab у межах єдиної аналітичної платформи дозволила забезпечити як точну математичну обробку даних, так і мобільність, візуальну інтерпретацію та спільний доступ до результатів. Такий підхід є особливо актуальним у контексті розподілених ІТ-систем, де важливим є як контроль у реальному часі, так і гнучкість аналітичних засобів. Запропонована система є масштабованою та адаптивною, що дозволяє застосовувати її як у навчальних, так і в промислових умовах.

ДОДАТОК Б

Програмний код

```

%% === ІМПОРТ ДАНИХ ===
filename = 'network_data.csv'; % Збережи CSV-файл з Python-
генерації під цією назвою
data = readtable(filename);
data.Timestamp = datetime(data.Timestamp, 'InputFormat', 'yyyy-
MM-dd HH:mm:ss');

%% === ПОПЕРЕДНЯ ОБРОБКА ===
% Згладжування метрик (30-хвилинне ковзне середнє)
latency_smoothed = movmean(data.Latency_ms, 30);
packet_loss_smoothed = movmean(data.Packet_Loss_percent, 30);
bandwidth_smoothed = movmean(data.Bandwidth_Usage_Mbps, 30);

%% === ВИЯВЛЕННЯ ПІКОВИХ НАВАНТАЖЕНЬ ===
[latency_peaks, latency_locs] = findpeaks(latency_smoothed,
'MinPeakHeight', 45);
[bandwidth_peaks, bw_locs] = findpeaks(bandwidth_smoothed,
'MinPeakHeight', 90);

%% === ВІЗУАЛІЗАЦІЯ ===
figure('Name','Network Monitoring
Dashboard','NumberTitle','off');

subplot(3,1,1);
plot(data.Timestamp, latency_smoothed, 'b'); hold on;
plot(data.Timestamp(latency_locs), latency_peaks, 'ro');
title('Latency (ms)'); ylabel('ms'); grid on;

subplot(3,1,2);
plot(data.Timestamp, packet_loss_smoothed, 'g');
title('Packet Loss (%)'); ylabel('%'); grid on;

subplot(3,1,3);
plot(data.Timestamp, bandwidth_smoothed, 'm'); hold on;
plot(data.Timestamp(bw_locs), bandwidth_peaks, 'ko');
title('Bandwidth Usage (Mbps)'); ylabel('Mbps'); xlabel('Time');
grid on;

%% === ПРОСТА ОЦІНКА СТАНУ ===
critical_latency = sum(latency_smoothed > 50);
critical_loss = sum(packet_loss_smoothed > 5);
critical_bandwidth = sum(bandwidth_smoothed > 95);

fprintf('\n=== NETWORK STATUS REPORT ===\n');
fprintf('High Latency Events (>50ms): %d\n', critical_latency);
fprintf('High Packet Loss Events (>5%%): %d\n', critical_loss);

```

```

fprintf('Bandwidth Spikes (>95Mbps): %d\n', critical_bandwidth);

if critical_latency > 10 || critical_loss > 10
    fprintf('WARNING: Network is experiencing critical
performance issues.\n');
else
    fprintf('Network performance is within acceptable
thresholds.\n');
end
figure('Name','Metric Distributions');

subplot(3,1,1);
histogram(data.Latency_ms, 50);
title('Latency Distribution'); xlabel('ms'); ylabel('Count');

subplot(3,1,2);
histogram(data.Packet_Loss_percent, 50);
title('Packet Loss Distribution'); xlabel('%'); ylabel('Count');

subplot(3,1,3);
histogram(data.Bandwidth_Usage_Mbps, 50);
title('Bandwidth Usage Distribution'); xlabel('Mbps');
ylabel('Count');
% Агрегування по годинам
data.Hour = hour(data.Timestamp);
hourly_latency = groupsummary(data.Latency_ms, data.Hour,
'mean');
hourly_loss = groupsummary(data.Packet_Loss_percent, data.Hour,
'mean');
hourly_bw = groupsummary(data.Bandwidth_Usage_Mbps, data.Hour,
'mean');

heat_matrix = [hourly_latency, hourly_loss, hourly_bw];

figure('Name','Hourly Heatmap');
imagesc(heat_matrix);
colorbar;
title('Average Metrics by Hour');
xlabel('Metric: 1=Latency, 2=Loss, 3=Bandwidth');
ylabel('Hour of Day');
figure('Name','Metric Correlations');

subplot(1,2,1);
scatter(data.Latency_ms, data.Packet_Loss_percent);
xlabel('Latency (ms)'); ylabel('Packet Loss (%)');
title('Latency vs Packet Loss'); grid on;

subplot(1,2,2);
scatter(data.Bandwidth_Usage_Mbps, data.Latency_ms);
xlabel('Bandwidth (Mbps)'); ylabel('Latency (ms)');
title('Bandwidth vs Latency'); grid on;

```