УДК 255:29.1

СИСТЕМА МОНІТОРИНГУ МЕРЕЖНОЇ АКТИВНОСТІ КОРИСТУВАЧІВ ОС WINDOWS 7/8

П. П. Костенко
Кременчуцький національний університет імені Михайла Остроградського
вул. Першотравнева, 20, м. Кременчук, 39600, Україна. E-mail: ppkostenko@gmail.com

Розглянуто питання необхідності моніторингу активності користувачів локальної мережі та мережі Інтернет на базі операційної системи Windows 7/8. Проаналізовані існуючі засоби отримання інформації про мережну активність користувачів операційних систем сімейства Windows. Обрані технології реалізації детального моніторингу в режимі реального часу. Розроблені алгоритми та визначені класи і методи, які дозволяють визначати інформацію про пакети, що передаються мережею, їх основні параметри та протоколи, за допомогою яких здійснюється передача даних. Розроблено клієнт-серверний програний додаток NetActivityControl, який проводить збір детальної інформації про діяльність користувачів мережі, передає статистичні дані до серверної частини додатку. Клієнтська частина додатку надає детальну інформацію про діяльність користувача: IP-адреси і порти відправника та отримувача, тип протоколу передачі даних, вміст пакету. Розроблене програмне забезпечення надає можливість адміністраторам комп'ютерних мереж контролювати потоки даних і приймати рішення щодо обмеження доступу до ресурсів мережі.

Ключові слова: моніторинг користувачів, комп'ютерна мережа, ipworks.

СИСТЕМА МОНИТОРИНГА СЕТЕВОЙ АКТИВНОСТИ ПОЛЬЗОВАТЕЛЕЙ ОС WINDOWS 7/8

П. П. Костенко
Кременчугский национальный университет имени Михаила Остроградского
ул. Первомайская, 20, г. Кременчуг, 39600, Украина. E-mail: ppkostenko@gmail.com

Рассмотрены вопросы необходимости мониторинга активности пользователей локальной сети и сети Интернет на базе операционной системы Windows 7/8. Проанализированы существующие средства получения информации о сетевой активности пользователей операционных систем семейства Windows. Выбраны технологии реализации детального мониторинга в режиме реального времени. Разработаны алгоритмы и определены классы и методы, позволяющие определять информацию о пакетах, передаваемых по сети, их основные параметры и протоколы, с помощью которых осуществляется передача данных. Разработано клиент-серверное приложение NetActivityControl, которое проводит сбор детальной информации о деятельности пользователей сети, передает статистические данные в серверной части приложения. Клиентская часть приложения предоставляет подробную информацию о деятельности пользователя: IP-адрес, порт отправителя и получателя, тип протокола передачи данных, содержимое пакета. Разработанное программное обеспечение позволяет администраторам компьютерных сетей контролировать потоки данных и принимать решения, по ограничению доступа к ресурсам сети.

Ключевые слова: мониторинг пользователей, компьютерные сети, ipworks.

АКТУАЛЬНІСТЬ РОБОТИ. Однією з важливих особливостей сучасних корпоративних мереж є їх розмір, який часто обчислюється тисячами, а й іноді й десятками тисяч комп'ютерів. Діяльність користувачів таких комп'ютерних мереж може бути розподілена між різними комп'ютерами, а складні задачі часто вирішується не окремими користувачами, а групами користувачів. При цьому проблеми адміністрування системи та контролю її ресурсів є актуальними.

Основними цілями контролю роботи користувачів є забезпечення інформаційної безпеки, виявлення випадків некоректного, непрофесійного або нецільового використання ресурсів, контроль роботи, як окремих користувачів, так і груп користувачів, оцінка характеристик функціонування корпоративної мережі і параметрів використання ресурсів та ін.

Окрім цього, задача керування мережними ресурсами тісно пов'язана із забезпеченням якості обслуговування користувачів комп'ютерної мережі, оскільки коректне використання наявних ресурсів мережі підвищує якість послуг, які надаються даною мережею. До того ж, нові типи мережних додатків, що застосовуються в сучасних комп'ютерних системах і мережах різної спеціалізації (наприклад, сервіси потокового мультимедіа, Internet-конференції, IP-телефонія та VoIP, HTTP/FTP-трафік та ін.), висувають різні вимоги до продуктивності мережі. Це, в свою чергу, обумовлює необхідність розробки нових і покращення існуючих технологій забезпечення якості обслуговування в комп'ютерних системах і мережах.

Такі проблеми, як перевірка фактичної завантаженості робітників, ефективності використання ресурсів комп'ютеризованого робочого місця та комп'ютерної мережі взагалі, аналіз часу, що витрачається працівниками на сторонні речі (спілкування по Інтернету, перегляд фільмів, запуск ігор, відвідування web-сторінок, не пов'язаних із робочою діяльністю та ін.), гарантування неможливість використання робочого часу на подібні дії або обмеження такого часу, забезпечення інформаційної безпеки функціонування та інше постають на кожному підприємстві.

Коректне та адекватне вирішення перелічених проблем дозволить підвищити ефективність роботи підприємства, зменшити витрати на адміністрування за допомогою людського ресурсу, підвищити трудову дисципліну на робочих місцях, що, в свою чергу, подовжить термін експлуатації наявного програмного та апаратного забезпечення, підвищити безпеку та цілісність даних та ін.

Таким чином, керування мережними ресурсами на основі моніторингу активності користувачів

комп'ютерної мережі є актуальною практичною задачею.

Мета роботи – отримання інформації про діяльність користувачів ОС WINDOWS 7/8 для забезпечення ефективного керування ресурсами комп'ютерної мережі.

МАТЕРІАЛ І РЕЗУЛЬТАТИ ДОСЛІДЖЕНЬ. Постійний контроль за роботою локальної комп'ютерної мережі (ЛКМ), що становить основу будь-якої корпоративної мережі, необхідний для підтримки її в працездатному стані. Використання автономних засобів контролю допомагає адміністратору мережі виявити проблемні ділянки і обладнання мережі, а їх відключення або реконфігурацію він може виконувати в цьому випадку вручну.

Процес контролю роботи мережі зазвичай ділять на два етапи: моніторинг та аналіз [1].

На етапі *моніторингу* виконується процедура збору первинних даних про роботу мережі, а саме статистику про кількість циркулюючих в мережі кадрів і пакетів різних протоколів, стан портів концентраторів, комутаторів і маршрутизаторів і т.д.

На етапі *аналізу* виконується більш складний і інтелектуальний процес осмислення зібраної на етапі моніторингу інформації, співставлення її з даними, отриманими раніше, і формулювання припущень про можливі причини сповільненої або ненадійної роботи мережі, що аналізується.

Всі засоби моніторингу й аналізу мереж можна умовно розділити на декілька великих класів (рис. 1).

Рисунок 1 – Класифікація засобів моніторингу й аналізу комп'ютерних мереж

1. Системи управління мережею (Network Management Systems) – централізовані програмні системи, які збирають дані про стан вузлів і комунікаційних пристроїв мережі, а також дані про трафік в мережі [2]. Такі системи покликані проводити моніторинг, аналіз і здійснювати в автоматичному чи напівавтоматичному режимі управління мережею – включення та відключення портів пристроїв, зміну параметрів адресних таблиць мостів, комутаторів і маршрутизаторів тощо. Представниками даних систем є HPOpenView, SunNetManager, IBMNetView.

2. Засоби управління системою (System Management), виконують функцію, що аналогічні до систем керування мережею, але по відношенню до комунікаційного обладнання. Також вона здатна виконувати найпростіший аналіз мережевого трафіка [3]. До найбільш відомих систем управління системами належать LANDesk, IBM Tivoli, Microsoft Systems Management Server, HP OpenView, Novell ZENworks і CA Unicenter.

3. Вбудовані системи діагностики і управління (Embedded Systems), які виконуються у вигляді програмно-апаратних модулів, що встановлюються в комунікаційне обладнання, а також у вигляді програмних модулів, вбудованих в операційні системи [4]. Такі системи здійснюють управління тільки одним пристроєм. Прикладом засобів цього класу є модуль управління концентратором Distributed 5000, що реалізує функції автосегментації портів при виявленні несправностей, приписування портів внутрішнім сегментам концентратора та ін. Як правило, вбудовані модулі управління також виконують роль SNMP-агентів, які надають дані про стан пристрою системам управління.

4. Аналізатори протоколів (Protocol analyzers) представляють собою програмні або апаратно-програмні системи, які обмежуються, на відміну від систем управління, лише функціями моніторингу й аналізу трафіку в мережах [4]. Аналізатор протоколів проводить захоплення і декодування пакетів мережевих протоколів. Аналізатори протоколів функціонують на основі правил регулярних виразів вмісту пакетів даних. Одним із найефективніших і найбільш вживаних є програмний продукт Wireshark.

Існуючі програмні комплекси проводять локальний аналіз трафіку чи збір загальної статистики на головному сервері мережі (сегменту мережі). У даній роботі пропонується клієнт серверний програмний додаток, який покликаний поєднати два окремих програмних модулі (клієнтський аналіз трафіку та серверний збір статистики роботи з ресурсами) для забезпечення ефективного керування ЛКМ.

Середовищем розробки програмного забезпечення обрано Embarcadero® RAD Studio XE3 Version 17. Embarcadero RAD Studio XE3 – середовище швидкої розробки додатків (RAD) для Microsoft Windows фірми Embarcadero Technologies.

Поточна версія Embarcadero RAD Studio XE3 об'єднує Delphi XE3 і C ++ Builder XE3 в єдине інтегроване середовище розробки. Надає можливість компілювати під різні платформи, зокрема крім win32 є можливість компілювати під win64, і під Mac OS X (візуальний інтерфейс пропонується створювати на базі нової кроссплатформенной бібліотеки FireMonkey).

Використання даного середовища дозволить розробити кросплатформений програмний додаток.

Програмний додаток покликаний підвищити ефективність використання комп'ютерної мережі фірми за рахунок збору в режимі реального часу статистики про використання каналу передачі даних та постійне передання поточних результатів роботи користувачів до головного серверу.

Програмний додаток має надавати наступні функції в клієнтській частині:

– функціонування в ОС MS Windows останніх версій та підтримка x64 розрядних ОС;

– слідкування за мережевою активністю користувача (ів) робочої станції;

– збір даних про протоколи, що використовуються користувачем;

– збір даних про шляхи передачі пакетів в мережі від/до користувача;

– збір даних про наявні відкриті порти та порти, що використовуються в межах даної сесії передачі даних;

– збір даних про об'єм трафіку переданого/прийнятого користувачем.

Серверна частина має сприймає звіти отримані від користувачів мережі та відслідковувати сплески активності чи перевантаження окремого сегменту або мережі в цілому.

Клієнтська частина проекту складається з трьох модулів:

– frm_Main.pas – містить основний код програмного додатку та класи, що описують головне вікно програми.

– frm_About.pas – містить код програмного додатку та класи, що описують вікно програми, що містить інформацію про програму та її розробника.

– ipworks9.dll – бібліотека, що містить набір функцій для вирішення основних завдань програмного додатку.

– У проекті реалізовані наступні методи:

– procedure bStartClick(Sender: TObject) – метод, що викликається при настання події OnClick та реалізує моніторинг мережної активності користувачів;

– procedure lvwPacketsClick(Sender: TObject) – метод, що використовується для отримання інформації про основні властивості кожного з пакетів, що передаються мережею;

– procedure IPMonitor1IPPacket(Sender: TObject; const SourceAddress: string; SourcePort: Integer; const DestinationAddress: string; DestinationPort, IPVersion, TOS, Id, Flags, Offset, TTL, Checksum, IPProtocol: Integer; Payload: string) – метод для отримання інформації про IP-адреси та порти відправника та отримувача, а також версію IP-протоколу;

– Проект є програмою, розраховану для використання на ПК, що входять до складу мережі та реалізований у вигляді програмного додатку NetActivityControl. Основна частина реалізована у вигляді dll-бібліотеки у ipworks9.dll.

– IPWorks – спеціалізований фреймворк, що включає підтримку понад 40 мережевих бібліотек. IPWorks дозволяє швидко інтегрувати підтримку будь якого протоколу до програмного додатку.

Діаграма об'єктів програмного додатка наведена на рис. 2.

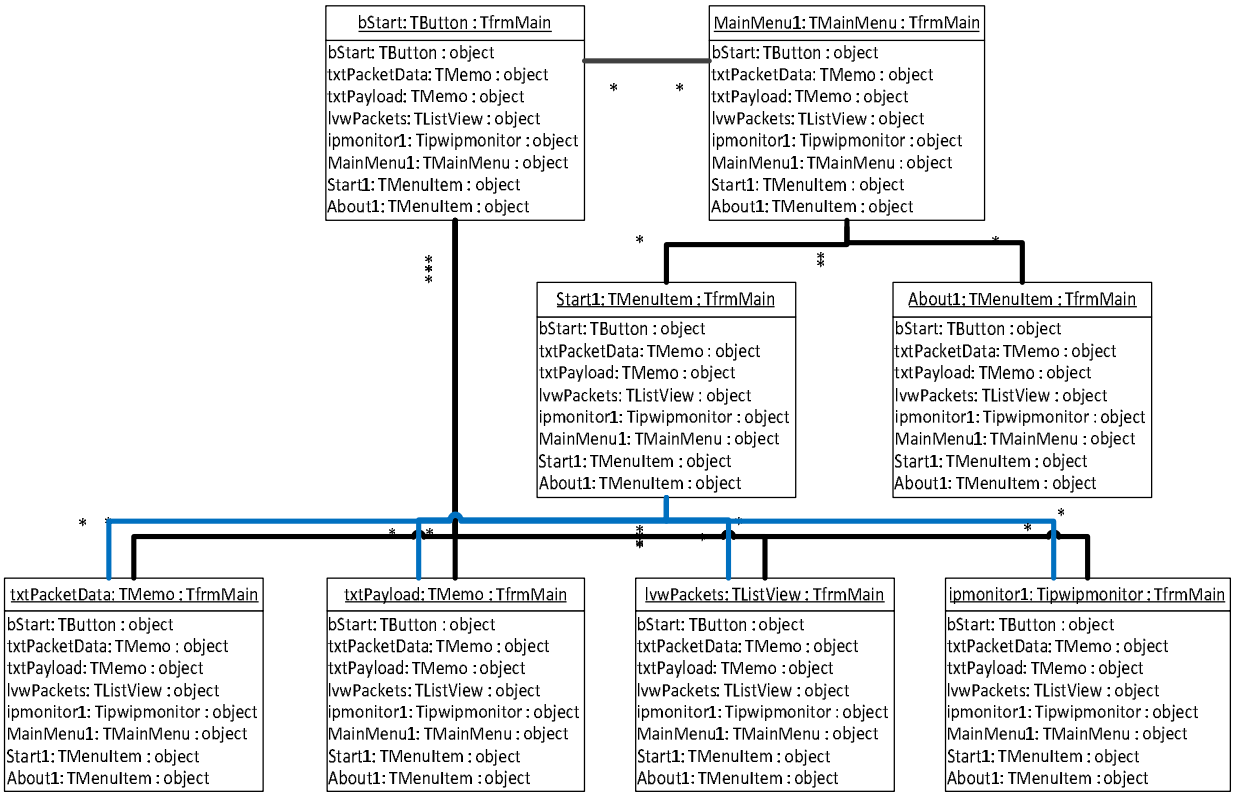

Рисунок 2 – Діаграма об'єктів NetActivityControl

Головне вікно клієнтської частини наведено на рис. 3 і складається із чотирьох блоків:

– головне меню програми;

– блок представлення пакетів;

– блок даних про пакети;

– блок вмісту пакетів.

Особливу увагу слід звернути на програмний код отримання пакету (лістинг 1). Фреймворк IPWorks у рамках модулю ipmonitor надає детальну інформацію адресу і порт відправника, адресу і порт отримувача та параметри обслуговування Type Of Service (TOS), що дозволяє впровадити в мережевий монітор механізми Quality Of Service (QoS). Блок TOS реалізує механізми Multi-Protocol Label Switching (MPLS), які дозволяють ефективно розподіляти трафік на основі пріоритетних бітів [5–9].

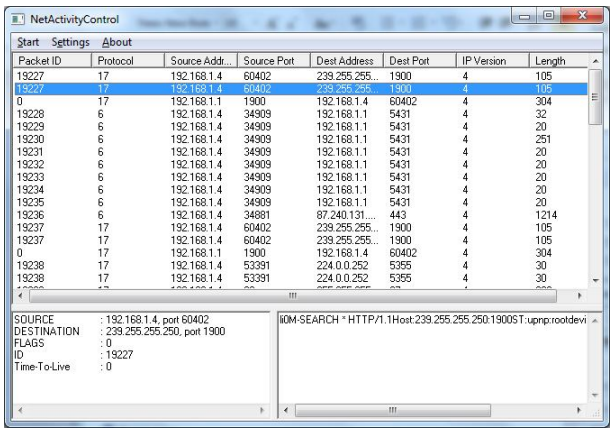

Рисунок 3 – Головне вікно клієнтської програми

```
procedure TfrmMain.ipmonitor1IPPacket(Sender: TObject;
  const SourceAddress: String; SourcePort: Integer;
  const DestinationAddress: String; DestinationPort, IPVersion, TOS,
Id,
  Flags, Offset, TTL, Checksum, IPProtocol: Integer; Payload: String);
var
    pi:Packet_INFO;
begin
    pi.Checksum := Checksum;
    pi.DestinationAddress := DestinationAddress;
    pi.DestinationPort := DestinationPort;
    pi.Flags := Flags;
    pi.Id := Id;
    pi.IPProtocol := IPProtocol;
    pi.IPVersion := IPVersion;
    pi.Offset := Offset;
    pi.Payload := Payload;
    pi.SourceAddress := SourceAddress;
    pi.SourcePort := SourcePort;
    SetLength(Packets, Length(Packets)+1);
    Packets[packetcount] := pi;
    packetcount := packetcount + 1;
    lvwPackets.Items.Add();
    lvwPackets.Items.Item[lvwPackets.Items.Count-1].Caption :=
IntToStr(pi.Id);
    lvwPackets.Items.Item[lvwPackets.Items.Count-
1].SubItems.Add(IntToStr(pi.IPProtocol));
    lvwPackets.Items.Item[lvwPackets.Items.Count-
1].SubItems.Add(pi.SourceAddress);
    lvwPackets.Items.Item[lvwPackets.Items.Count-
1].SubItems.Add(IntToStr(pi.SourcePort));
    lvwPackets.Items.Item[lvwPackets.Items.Count-
1].SubItems.Add(pi.DestinationAddress);
    lvwPackets.Items.Item[lvwPackets.Items.Count-
1].SubItems.Add(IntToStr(pi.DestinationPort));
    lvwPackets.Items.Item[lvwPackets.Items.Count-
1].SubItems.Add(IntToStr(pi.IPVersion));
    lvwPackets.Items.Item[lvwPackets.Items.Count-
1].SubItems.Add(IntToStr(Length(pi.Payload)));
    LenPack := LenPack + Length(pi.Payload);
end;
```

Лістинг 1 – Вихідний код методу ipmonitor1IPPacket

Відповідно до рис. 3 дані про пакети після парсингу методом (лістинг 1) формуються у структуровану таблицю. Вибір окремого запису з таблиці викликає додатковий метод lvwPacketsClick (лістинг 2), який проводить розбір внутрішньої структури пакету, конвертує дані до кодування UTF-8 та виводить детальну інформацію на головному екрані додатку.

Серверна частина програмного додатку покликана проводити збір статистики на надавати можливість обмежувати роботу клієнтів в мережі.

Система в процесі функціонування отримує наступні вхідні дані

– дані про протоколи, що використовуються користувачем;

– дані про шляхи передачі пакетів в мережі від/до користувача;

– дані про наявні відкриті порти та порти, що використовуються в межах даної сесії передачі даних;

– дані про об’єм трафіку переданого/прийнятого користувачем;

– зведені статистичні дані використання мережевих каналів передачі даних окремими користувачами.

```
procedure TfrmMain.lvwPacketsClick(Sender: TObject);
var i,j,ascvalue: integer;
      ch:char;
begin
   if lvwPackets.Selected <> nil then begin
        txtPacketData.Clear();
        txtPacketData.Text := txtPacketData.Text + 'SOURCE' +
Chr(9) + Chr(9) + ': ' + Packets[lvwPackets.
Selected.Index].SourceAddress + ', port ' + IntToStr(Packets
[lvwPackets.Selected.Index]. SourcePort) + #13#10;
        txtPacketData.Text := txtPacketData.Text + 'DESTINATION'
+ #9+ ': ' + Packets[lvwPackets. Selected.Index].DestinationAddress+
', port ' + IntToStr(Packets[lvwPackets.Selected.Index].
DestinationPort) + #13#10;
        txtPacketData.Text := txtPacketData.Text + 'FLAGS' + #9#9 +
': ' + IntToStr(Packets[lvwPackets. Selected.Index].Flags) + #13#10;
        txtPacketData.Text := txtPacketData.Text + 'ID' + #9#9 + ': ' +
IntToStr(Packets[lvwPackets. Selected.Index].Id) + #13#10;
        txtPacketData.Text := txtPacketData.Text + 'Time-To-Live' +
#9 + ': ' + IntToStr(Packets [lvwPackets.Selected.Index].TTL) +
#13#10;
        txtPayLoad.Clear();
        //txtPayLoad.Text :=
Packets[lvwPackets.Selected.Index].PayLoad;
        //txtPayLoad.Text : '';
        For j := 1 To Length(Packets[lvwPackets.
Selected.Index].Payload) do begin
          ch := Packets[lvwPackets.Selected.Index]. Payload[j];
          ascvalue := ORD(ch);
          If (ascvalue >= 32) And (ascvalue <= 126) Then begin
            txtPayLoad.Text := txtPayLoad.Text + ch;
          end;
        end;
    end;
end;
```

Лістинг 2 – Вихідний код методу lvwPacketsClick

Серверна частина додатку реалізує чотири методи та використовує набір власних і експортованих класів (рис. 4, 5):

– procedure FormCreate(Sender: TObject) – метод, що викликається при настанні події OnCreate та реалізовує початкове становлення значень змінних та формує зовнішній вигляд і наповнення звідних таблиць і полів;

– procedure BitBtn1Click(Sender: TObject) – метод, що викликається при настанні події OnClick, для об’єкту BitBtn1 та реалізовує запуск серверної частини, ініціює початок прослуховування портів передачі даних та очікування надходження звітної інформації про клієнтів;

– procedure BitBtn2Click(Sender: TObject) – метод, що викликається при настанні події OnClick, для об’єкту BitBtn2 та реалізовує зупинку серверної частини, ініціює збереження результатів роботи;

– procedure TcpServer1Accept(Sender: TObject; ClientSocket: TCustomIpClient) – метод, що викликається при настанні події On Accept, для об'єкту TcpServer1 та реалізовує обробку вхідного повідомлення від клієнта мережі, розбір повідомлення на складові на виведення у відповідні графічні блоки (поля) на головній формі.

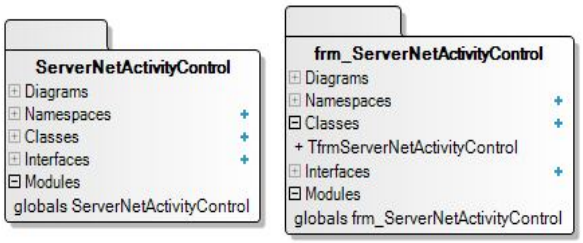

Рисунок 4 – Власні класи сервеної частини додатку

Експортовані класи надають доступ до мережевих методів та функцій, створення TCP з'єднання з клієнтом для керування, передачі широкомовних запитів в мережу для опитування поточного стану клієнтів, тощо.

Головне вікно серверної частини відображено на рис. 6. До інтерфейсу входить два основні блоки: блок керування та блок виводу на екран.

Програма повинна виконуватись у середовищі ліцензійної локалізованої версії операційної системи Microsoft Windows 7/8 з використанням попередньо встановленого пакету сумісності IPWorksV9.

Функціонування програмного додатку описується діаграмою SharePoint рис. 7.

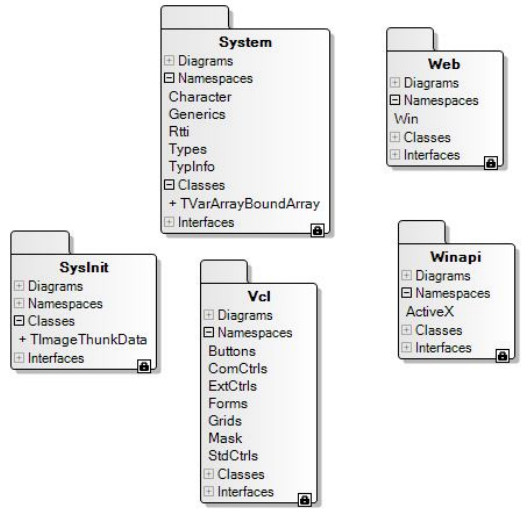

Рисунок 5 – Експортовані класи серверної частини додатку

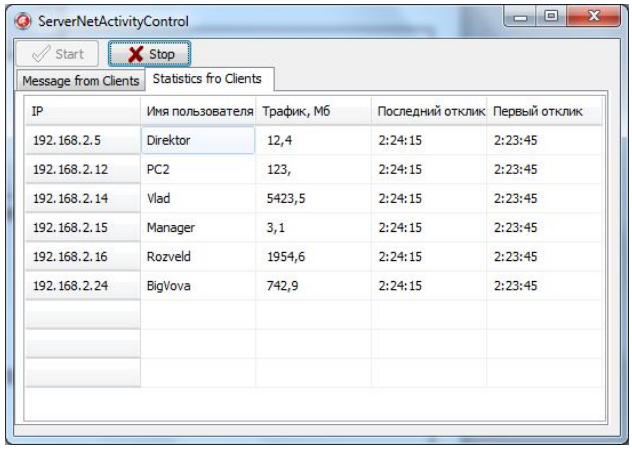

Рисунок 6 – Головне вікно серверної частини додатку

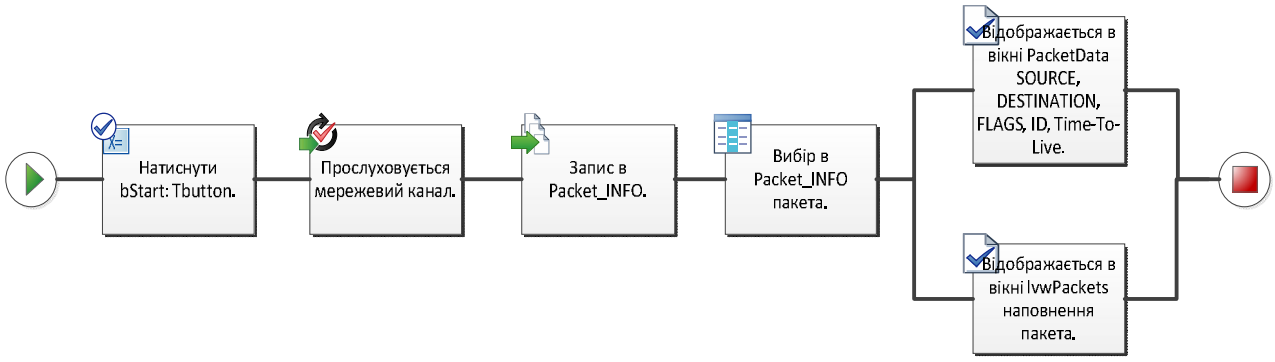

Рисунок 7 – Діаграма функціонування клієнтської частини в нотації SharePoint

Для роботи з програмою необхідно:

1. Встановити на клієнтські робочі станції додаткове програмне забезпечення IPWorksV9 із каталогу: NetActivityControlXE3\IPWorksV9.

2. Встановити клієнтську частину програмного забезпечення моніторингу NetActivityControl.exe.

3. Для перевірки програми необхідно на комп'ютері, підключеному до мережі, запустити програму NetActivityControl.exe та натиснути на кнопку «Start» у вікні, що з'явиться на екрані.

4. Встановити серверне програмне забезпечення ServerNetActivityControl.exe та натиснути на кнопку «Start» у вікні, що з'явиться на екрані.

5. У разі відсутності даних від клієнта перевірити адреси підключення в розділі налаштувань клієнтського ПЗ.

Дослідження проведені на робочих станціях з вказаними характеристиками показали мінімальні часові затримки на збір даних із мережевої карти на аналіз пакетів. Результати тестового навантаження наведені на рис. 8.

У середньому час обробки пакетів збільшився на 4 %, що в абсолютних показниках становить 0,6 с на кожні 587 прийнятих/відправлених пакетів. Для обробки одного пакету програмний додаток вимагає в середньому додатково 0,0011 с.

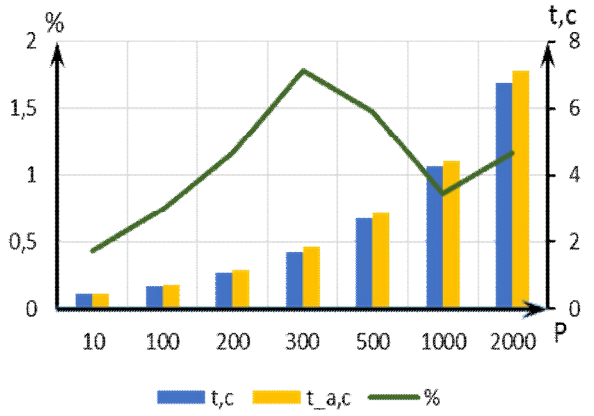

Рисунок 8 – Результати тестового навантаження програмним забезпеченням (t,c – час обробки пакетів без застосування програми, t_a,c – час обробки пакетів з застосуванням програми, % – відсоток затримки)

Програма забезпечує задані характеристики в умовах експлуатації на IBM-сумісних персональних комп'ютерах (ПЕОМ) із наступними основними технічними характеристиками:

− мікропроцесор класу Intel Core2Duo/AMD Athlon з тактовою частотою не менше ніж 2,0 МГц;

− оперативна пам'ять не менше ніж 2048 Мб;

− відеокарта nVidia GeForce 210/ATI Radeon HD 5450;

− відеомонітор із роздільною здатністю не менше ніж 1280 x 1024;

− вільний простір на жорсткому диску не менше 200 Гб.

ВИСНОВКИ. Показано можливість побудови мережевого монітору активності користувачів із можливістю зведення статистики та детального аналізу потоків даних на основі програмного фрейм-мворку IPWorksV9. Розроблений програний додаток має клієнт серверну архітектуру, що надає можливості гнучкого масштабування.

Наведено особливості програмних методів перехоплення мережевих пакетів і розбору їх вмісту. Описано класову структуру клієнт серверного додатка.

Проведений аналіз впливу програмного забезпечення на функціоування клієнтських робочих станцій показав мінімальний приріст часового навантаження в 0,0011с.

ЛІТЕРАТУРА

1. Tanenbaum, Andrew S., David J. Wetherall, Computer networks. – New York: Pearson, 2011. – 5th ed. – 962 p.

2. Lowekamp B., Zangrilli M. Using passive traces of application traffic in a network monitoring systems // IEEE Computer Society, 2004. – PP. 77–86.

3. Чунарьова А.В. Сучасні методи аудиту та моніторингу в задачах захисту інформації. // Проблеми інформатизації та управління. – 2013. – № 3 (43). – С. 87–91.

4. Angrisani, L., Capriglione, D., Ferrigno, L.,,Miele, G. A Methodological Approach for Estimating Protocol Analyzer Instrumental Measurement Uncertainty in Packet Jitter Evaluation // IEEE Transactions on nstrumentation and Measurement. – 2012. –Vol. 61, iss. 5. – PP. 1405–1416.

5. Kher, Vansha Arman, Anuja ; Saini, Davinder S Hybrid evolutionary MPLS Tunneling Algorithm based on high priority bits // International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE). – 2015. – Greater Noida, India. – PP. 495–499.

6. Кортунов В.И., Воробьев А.В. Решение задачи распределения нагрузки на основе динамической модели маршрутизатора // Проблемы телекоммуникаций: электронное научное специализированное издание. – 2011. – № 2 (4). – С. 128–138.

7. Кудзиновская И. П. Метод обнаружения перегрузки сетевых узлов по скорости роста очередей // Наукові записки УНДІЗ. – 2009. – № 2 (10). – С. 74–78.

8. Hubert B., Linux Advanced Routing & Traffic Control HOWTO. – 2003. – 158 p. [Electronic Resource]. – Mode of access: http://www.lartc.org/lartc.pdf

9. Gillen M., Loyall J., Sterling J. Dynamic Quality of Service Management for Multicast Tactical Communications. // Proc. 14th IEEE Computer Society Symposium on Object/Component/Service-oriented Real-time Distributed Computing (ISORC), 2011. – Newport Beach, USA. – PP. 11–18.

MONITORING SYSTEM OF USER NETWORK ACTIVITY OS WINDOWS 7/8

P. Kostenko

Kremenchuk Mykhailo Ostrohradskyi National University

vul. Pershotravneva, 20, Kremenchuk, 39600, Ukraine. E-mail: ppkostenko@gmail.com

The problem of the necessary monitoring of user activity in LAN and Internet network on the basis of Windows 7/8 operating systems is described. The existing means of obtaining information about network activity of users of Windows operating systems are analyzed. The implementation technologies of detailed monitoring in real time are selected. It were developed the algorithms and specified the classes and methods, which allow to determine the information about the package that are transferred through network, their basic parameters and protocols, through which the data is transferring. A client-server application NetActivityControl was developed; it collects detailed information on the activities of network users, reports statistics to the server side of the application. The client-side application provides the detailed information about user activities: IP-addresses and ports of the sender and recipient, the type of data transfer protocol,

the contents of the package. The developed software allows administrators of the computer networks to monitor the data flows and to make decisions on restricting an access to network resources.

Key words: network monitoring, computer network, IPworks.

REFERENCES

1. Tanenbaum, A. S., Wetherall, D. J. (2011), Computer networks, Pirson, New York, USA.

2. Lowekamp, B. Zangrilli, M. (2004), “Using passive traces of application traffic in a network monitoring system”, *13th IEEE International Symposium on High performance Distributed Computin. Proceedings*, Honolulu, Hawaii, USA, June 4-6, 2004, pp. 77–86.

3. Chunaryova, A. V. (2013), “Modern methods of auditing and monitoring of information security problems”, *Problems of information and management*, no. 3(43), pp. 87–91.

4. Angrisani, L., Capriglione, D., Ferrigno, L., Miele, G. (2012), “A Methodological Approach for Estimating Protocol Analyzer Instrumental Measurement Uncertainty in Packet Jitter Evaluation”, *IEEE Transactions on Instrumentation and Measurement*, vol. 61, iss. 5, pp. 1405–1416.

5. Kher, V. A., Saini, D. S. (2015), “Hybrid evolutionary MPLS Tunneling Algorithm based on high priority bits”, *International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE)*, Greater Noida, India, February 25–27, 2015, pp. 495–499.

6. Kortunov, V. I., Vorobev, A. V. (2011), “Solution of the problem of load distribution based on dynamic model of router”, *Problems of telecommunications*, no.2 (4), pp. 128–138.

7. Kudzinovskaya, I. P. (2009), “The method of detection of network nodes overload on the growth rate of the queues”, *Naukovi zapyski UNDIZ*, no. № 2 (10), pp. 74–78.

8. Hubert, B., et al. (2003), Linux Advanced Routing & Traffic Control HOWTO, available at: http://www.lartc.org/lartc.pdf (accessed July 15, 2015).

9. Gillen, M., Loyall, J., Sterling, J. (2011), "Dynamic Quality of Service Management for Multicast Tactical Communications", *Proceedings of 14th IEEE Computer Society Symposium on Object/Component/Service-oriented Real-time Distributed Computing (ISORC*), Newport Beach, USA, March 28–31, pp. 11–18.

Стаття надійшла 30.06.2015.