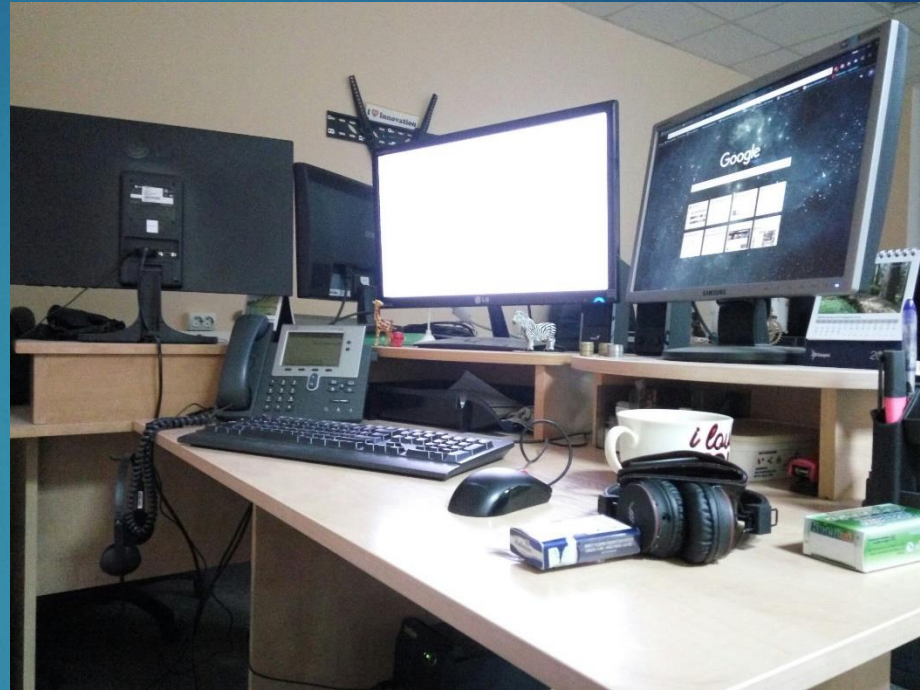


Метод тестування мережі з VoIP



ВИКОНАВ:

ГУМЕНЮК Р. С., СТ. ГР. КІ-15МН


НАУКОВИЙ КЕРІВНИК:

К.Т.Н., ДОЦЕНТ ЗАХАРЧЕНКО С. М.

Актуальність роботи полягає -

У збільшенні попиту ринку на VoIP-технології, а задля надання високої якості голосових сигналів в мережі потрібні надійні методи тестування, за допомогою яких можливо провести точну оцінку якості цих самих сигналів





Об'єктом дослідження є - процес тестування мереж на предмет відповідності вимогам VoIP.

Предметом дослідження є методи тестування мереж для VoIP

Наукова новизна полягає у вдосконаленні методу тестування мережі з VoIP шляхом враховування тимчасових сплесків (спайків), що покращує якість процесу тестування та дозволяє збільшити надійність та продуктивність роботи мережі.

Мета тестування мереж, незалежно від того, хто здійснює тестування або де це зроблено, полягає у тому, щоб забезпечити необхідну якість роботи мережі.

Для досягнення мети магістерської дипломної роботи були визначені наступні завдання:

- 1) Проаналізувати існуючі методи тестування мережі з VoIP;
- 2) Розробити вдосконалений метод тестування мережі з VoIP;
- 3) Провести порівняльну характеристику нового методу з існуючими.

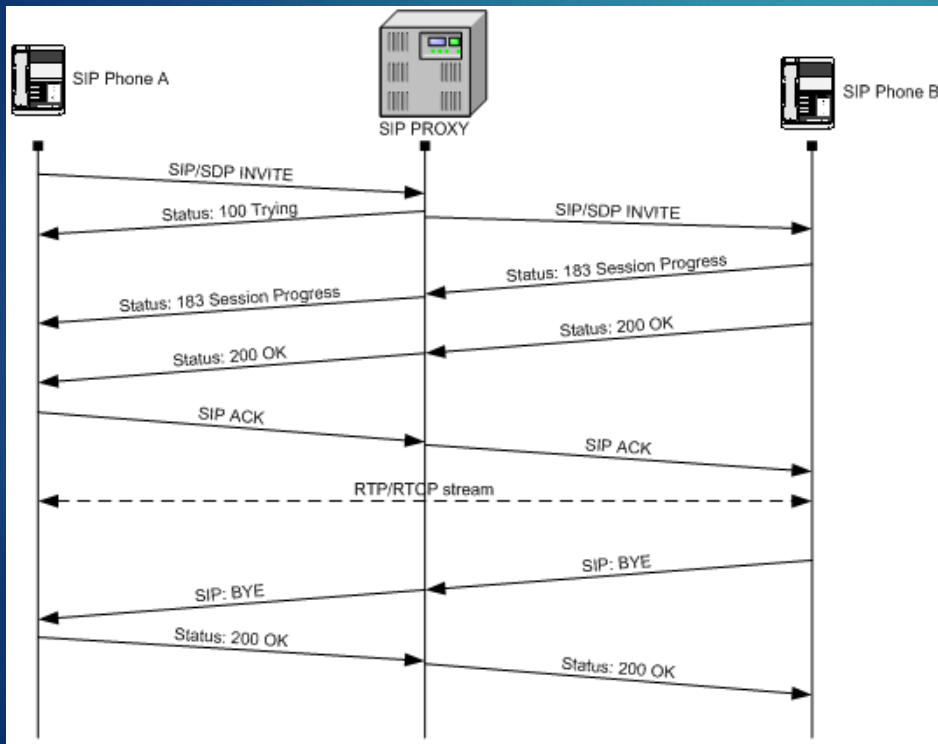
Кажучи простою мовою, VoIP - це технологія, що дозволяє вести медіа дані через Інтернет. В більшості випадків – голосові сигнали, а саме дзвінки

Переваги:

1. Масштабовані рішення - VoIP може бути гарною ідеєю, щоб почати з малого і працювати на масштабування бізнесу. З АТС, наприклад, не може бути можливо точної оцінки свого зростання.
2. Можна отримати істотну економію на міжміських і міжнародних телефонних дзвінках, скориставшись послугами телефонної служби VoIP. Телефонні системи, що з'єднують офіси, відділення компанії дозволяють здійснювати безкоштовні телефонні дзвінки.
3. Не має потреби в виділеній телефонній лінії - використовується комп'ютерна мережа. Телефонна система VoIP дозволяє підключати звичайні телефони до стандартного комп'ютерного мережевого інтерфейсу (який може спільно використовувати сусідній комп'ютер). Програмні телефони можуть підключатися безпосередньо до ПК.

Принципи роботи:

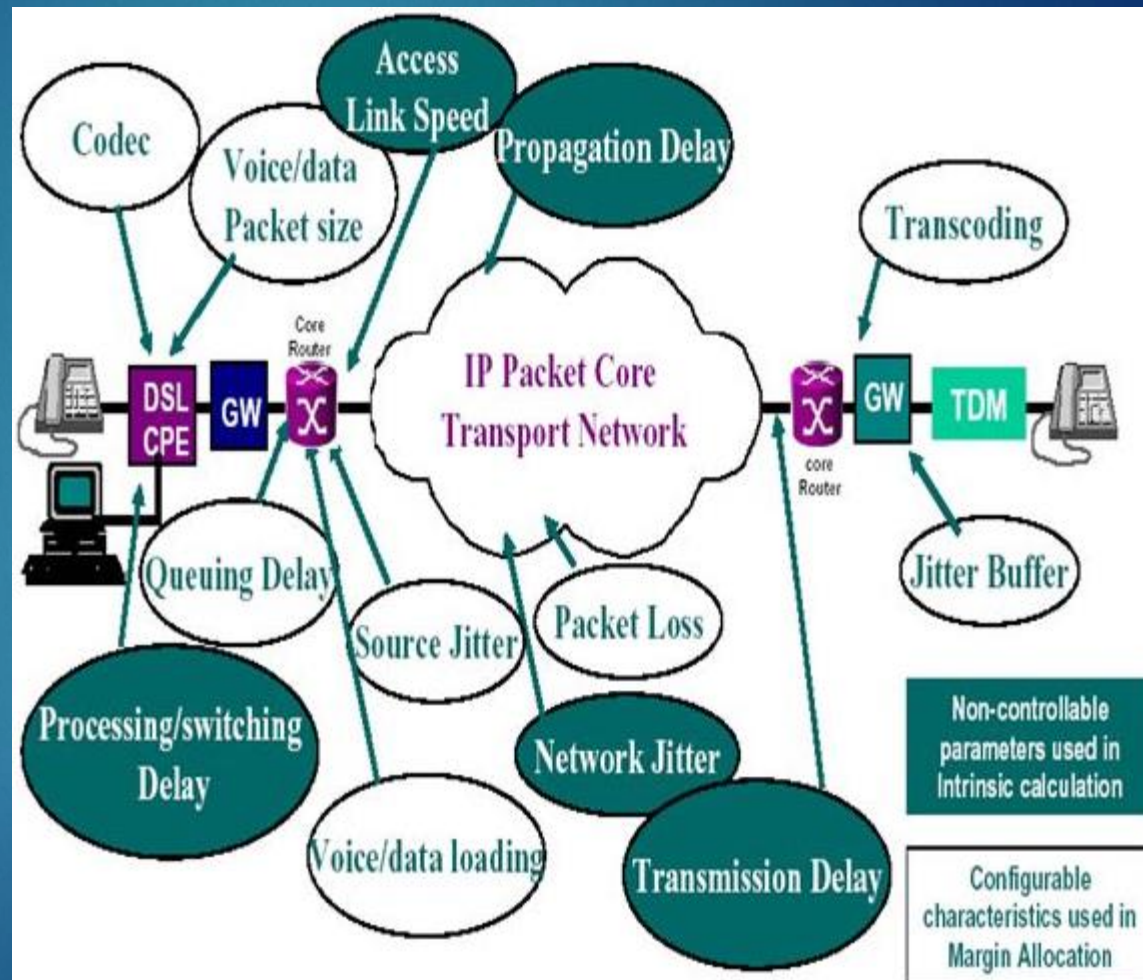
Розуміння особливостей пакета IP дає краще розуміння VoIP протоколів, зокрема UDP і протоколу ініціювання сеансу зв'язку SIP



Дані	Контроль		Аудіо/ Відео	Реєстрація
T.120	H.225.0 Call Signaling	H.245 Conference Control	RTP/RTCP	H.225.0 RAS
TCP			UDP	
Мережевий рівень				
Канальний рівень				
Фізичний рівень				

Фактори, що впливають на якість передачі голосу в системі VoIP :

1. Затримка
2. Втрата пакетів
3. Луна



Тестування VoIP:

- **Технологія тестування MOS** (визначається при проходженні сигналу від мовця (джерело) через систему зв'язку до слухача (приймач), оцінюється як арифметичне середнє від усіх оцінок, виставлених експертами після прослуховування тестового тракту передачі)
- **Суб'єктивні методики** (опитування людей на предмет якості мовлення в мережі - по 5-бальній шкалі) та **об'єктивні методики тестування**



- Рішення E-модель (багатокритеріальна оцінка якості переданої мови в мережах IP через R-фактор, що визначається діапазоном значень від 0 до 100 і обраховується за рахунок суми всіх викривлень, його можна співставити з суб'єктивною оцінкою MOS)

Суміжне використання рішень у методиці:



У реальному мережевому трафіку існує зв'язок між затримкою проходження пакета d і коефіцієнтом втрати пакетів l_d , це - називається поширення Паретто

Значення R	MOS	Оцінка користувача
90	4,3	Відмінно
80	4,0	Добре
70	3,6	Задовільно
60	3,1	Нижче задовільного
50	2,6	Погано

Як зазначено в рекомендації ITU-T

$$R = R_0 - I_d - I_e$$

$$MOS = 1 + 0.035 \cdot R + R \cdot (R - 60) \cdot (100 - R) \cdot 7 \cdot 10^{-6}, \text{ при } 0 < R < 100$$

Адаптивні буферні методи

Буфер компенсує відхилення значень затримки від середнього значення. Пакети, що прибувають на вузол-приймач, відтворюються не відразу, а з певною затримкою. Чим більша флуктуація затримки, тим більший розмір джитер-буферу потрібний для її компенсації, інакше частина пакетів буде відкинута, якщо пакети прийдуть пізніше часу відтворення.

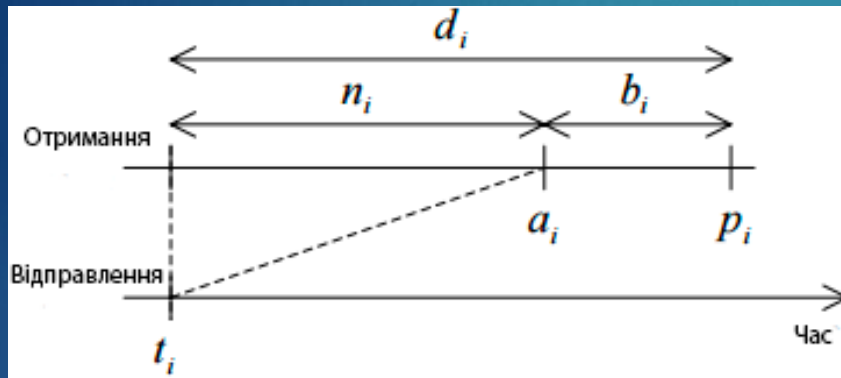


Схема шляху пакету

Бувають:

1. **Метод середнього значення** (Цей метод використовує мережеву затримку d_i та її варіацію v_i для оцінки затримки проходження p_i пакету i)
2. **Покращений метод середнього значення** (відмінність полягає у тому що затримка d_i обраховується іншим чином у випадку, коли затримка пакету у мережі n_i більше d_{i-1})
3. **Метод мінімальної затримки** (використовує мінімальну затримку пакету d_i отриману в потоці, аби передбачити наступну затримку в наступному потоці)

4. **Метод реагування на сплеск** (Одним з найбільш поширених явищ, які спостерігаються в системі VoIP є те, що деякі з пакетів раптово страждають від високої затримки проходження. В результаті голосові пакети не досягають приймача протягом деякого часу з подальшим прибуттям великої кількості їх одночасно. Термін цього явища – спайк.

) - Коли сплеск виявлений - метод змінюється в режим Spike, що проводить моніторинг всіх затримок. Якщо б не режим Spike, то цей метод по функціоналу був би ідентичним першому, але за винятком того, значення α встановлюється 0,875 як константа

5. **Метод вікна** - Під час сплеску він використовує затримку першого пакету поточного мовного потоку, як затримку проходження для цього мовного потоку. відображає два параметра - затримка і втрата пакетів для отримання оптимального значення

$$MOS = 4.10 - 0.195l + 2.64 \times 10^{-3} d - 1.86 \times 10^{-5} d^2 + 1.22 \times 10^{-8} d^3$$

$l = l_n + l_d$ де l_n - втрата пакетів в мережі, і l_d - втрата пакетів, викликана затримкою пакетів, відкинутих буфером джиттера.

6. **Метод M-MOS** (Він захоплює як затримку так і відсоток втрати останнього з w прийнятих пакетів, щоб максимізувати попереднє MOS значення, а потім щоб отримати відповідну мережеву затримку в якості майбутньої затримки проходження, на відмінну від попереднього методу, де передбачається відсоток втрат з використанням розподілу Парето.)

Висновки з аналізу

Метод має:

- ▶ Має підлаштовуватись під зміни затримки мережі.
- ▶ Має містити можливість зробити змінні затримки невеликими.
- ▶ Повинен мати хороший механізм виявлення спайку.
- ▶ Може передбачити затримку проходження (час), коли відбувається спайк.
- ▶ Може уникнути зіткнень між двома послідовними пакетами або мовними потоками.
- ▶ Може зменшити затримку проходження після виникнення спайку
- ▶ Повинен бути придатним для будь-яких мережевих обставин.

Огляд вдосконаленого методу

Коефіцієнт R може виражатися рівнянням 3.1, яке містить два параметри: I_e і I_d . Якщо зіставити R в рівняння 3.4 то ми можемо отримати нове рівняння у вигляді

$$MOS = 4.409 - 0.0194 \times (I_d + I_e) - 0.837 \times 10^{-3} \times (I_d + I_e)^2 + 7 \times 10^{-6} \times (I_d + I_e)^3$$

Для попередніх методів як Spike-DET, Window та M-MOS, механізми виявлення сплеску не можуть добре працювати в різних умовах мережі. При виявленні сплеску, ці три методи більш фіксовані. Для подолання цієї проблеми, пропонується вдосконалений метод по коригуванню сплеску. Мається на увазі концепція сполучення буферного методу як регулювання затримки сплеску і MOS-основаного методу буфера.

Так як затримка проходження зазвичай оцінюється вище в змінній затримці і нижче коли затримка змінюється безперервно, поріг значень повинен бути придатним для цих змін, а для шляху по мережі, ми можемо встановити поріг, щоб був нижче стійкої затримки в мережі.

Параметр α може бути належним чином використаний при розробці адаптивного порога.. Для алгоритму, поріг, встановлений як значення ENTER:

$$ENTER = k - 0.006 \cdot \alpha^2 + 118 + T(k),$$

$$\text{де } \begin{cases} T(k) = 0 \text{ при } k \leq 150 \\ T(k) = 150 \cdot \ln\left(\frac{k}{150}\right) \text{ при } k \geq 150. \\ \alpha = 100 \text{ при } \alpha > 100 \end{cases}$$

Конструкція ENTER в основному базується на можливості взаємозв'язку між фіксованою затримкою і варіаційною серед прийнятих пакетів. При збільшеній фіксованої затримки, дається більший діапазон змінної затримки. При поданні невеликої фіксованої затримки, передбачається маленький діапазон змінної затримки.

Метод виявлення кінця спайку схожий на метод Spike-Det. Ми також використовуємо фактор var в якості X. Проте значення EXIT пропонуємо встановити на 20. Коли $X < EXIT$, ми додаємо ще одну умову, щоб розрізнити чи є це миттєвий спайк, чи він може бути протягом тривалого періоду (рис. 4.1). Коли метод виявив кінець сплеска, і якщо це – миттєвий сплеск алгоритм в той самий час перезавантажує збір довідкових даних, а також перераховує параметр α , але і залишає значення параметра K , щоб уникнути завищеної або заниженої оцінки затримки проходження в наступних мовних потоку. Якщо це не є моментальний спайк, тоді лише зібрані дані, які починаються на початку спайку, будуть зарезервовані, аби слідувати мережевим затримкам, щоб уповільнити зниження затримки оцінки проходження.

```
IF (X < EXIT)  
{ IF (current packet network delay > door ×  
previous NORMAL mode talkspurt playout delay)  
It might be a long time scale spike;  
ELSE  
It is a momentary spike;}
```

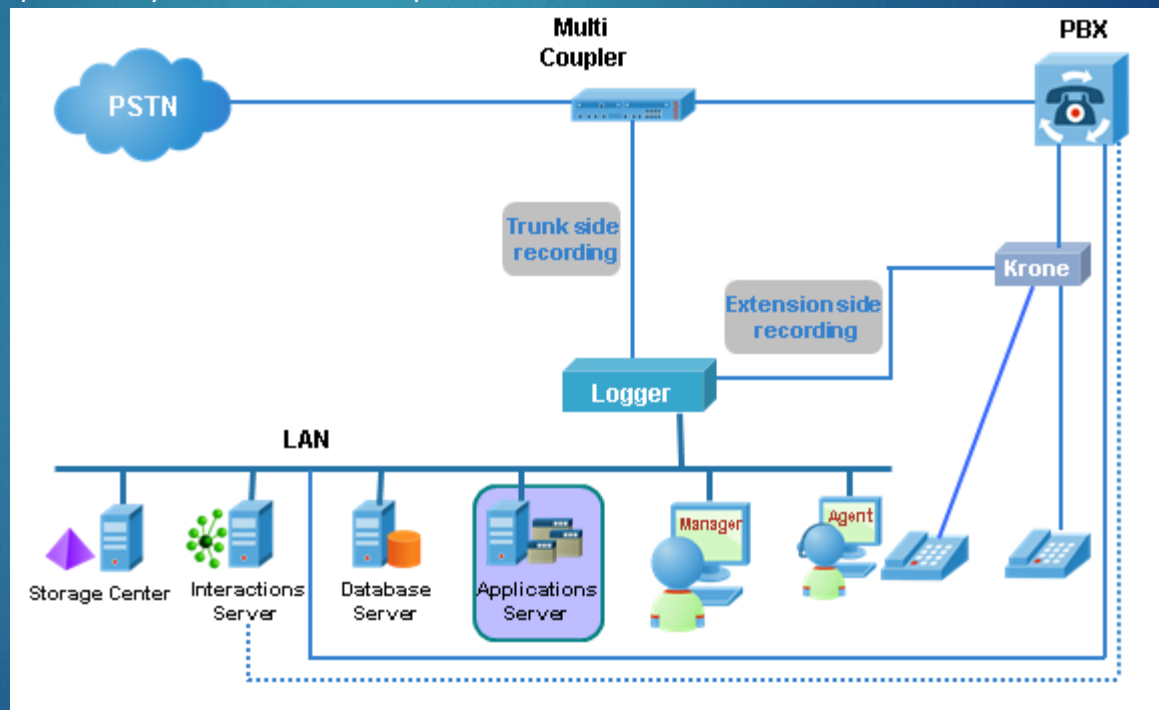
Коли метод виявив кінець сплеска, і якщо це – миттєвий сплеск алгоритм в той самий час перезавантажує збір довідкових даних, а також перераховує параметр α , але і залишає значення параметра K , щоб уникнути завищеної або заниженої оцінки затримки проходження в наступних мовних потоку. Якщо це не є моментальний спайк, тоді лише зібрані дані, які починаються на початку спайку, будуть зарезервовані, аби слідувати мережевим затримкам, щоб уповільнити зниження затримки оцінки проходження.

Реалізація методу тестування

Щоб проводити дослідження необхідно мати платформу. В якості такої була використана віртуалізована система запису дзвінків «NICE SYSTEMS Ltd».

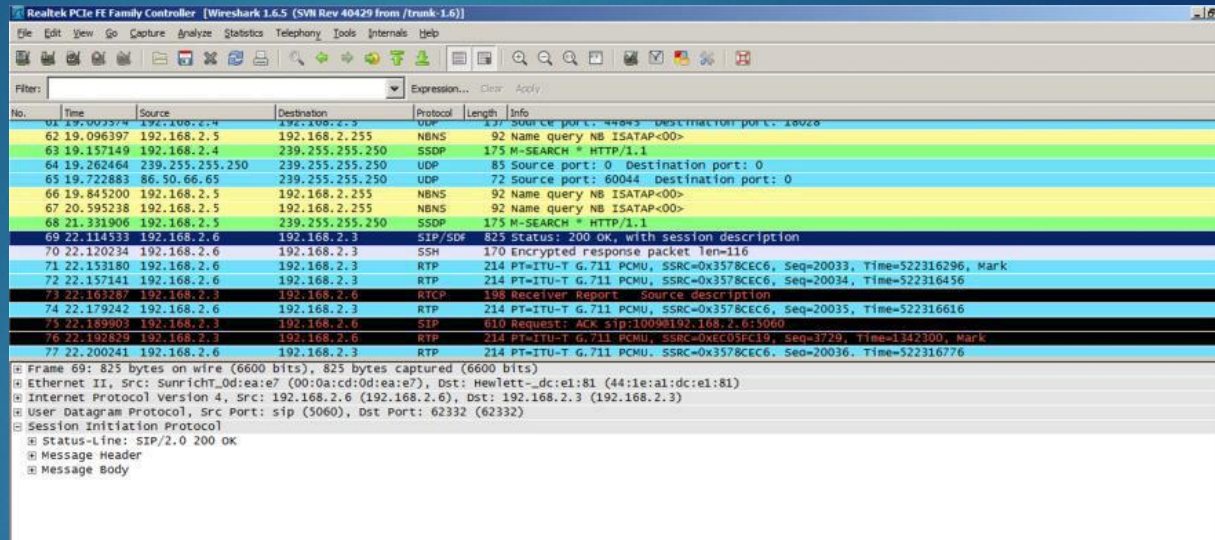
Результати дослідження характеристик і законів розподілу деяких подій різного роду трафіку, існуючого в мультисервісних мережах, будуть застосовуватися при визначенні вимог до якості послуг.

Топологія мереж використовується в даній роботі аналогічна тій, що вказана на рис. 4.2:



За допомогою **Wireshark** була отримана практична статистика, на підставі якої були отримані результати.

Кілька телефонних дзвінків були зроблені між софтбонами, встановленими на комп'ютерах, вони були відслідковані з використанням Wireshark. Нижче наведено рисунок захопленого трафіку :

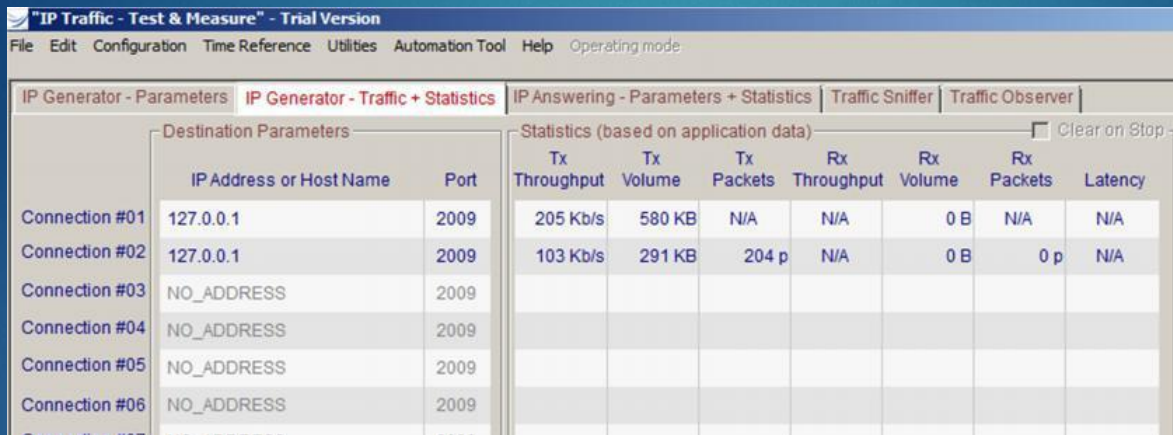


Мережа	Відстань	Спостереження Затримки	Спостереження Втрат
1	Довга	середні-високі джиттери	Мала кількість
2	Довга	середні джиттери	Середня кількість
3	Середня	Немає очевидного джитера	Дуже мала кількість
4	Коротка	Декілька середніх джитерів	Середня-велика кількість

Тестування було зроблено з використанням програмного забезпечення NetDisturb і IP Traffic Generator

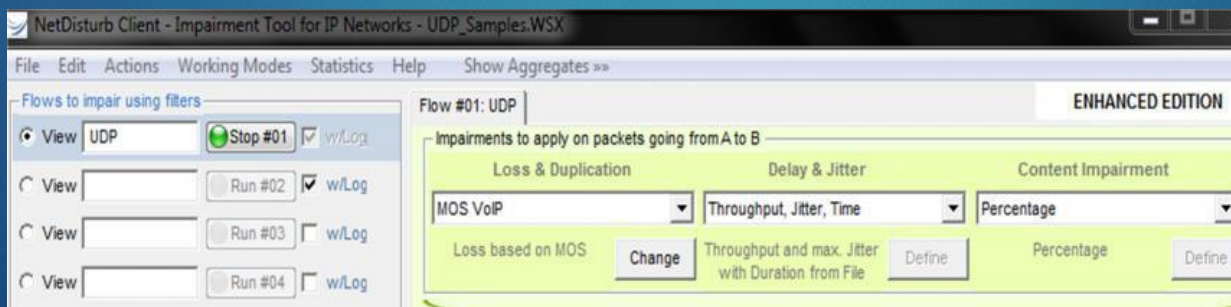
NetDisturb був використаний, з одного боку, аби ввести в основному затримки, джиттер і втрати пакетів в системі.

IP Traffic Generator, з іншого боку, був використаний для генерації трафіку таким чином, щоб обмежити смугу пропускання мережі.



The screenshot shows the 'IP Traffic - Test & Measure' application window. The 'IP Generator - Traffic + Statistics' tab is active, displaying a table of traffic statistics for various connections. The table has columns for Connection ID, Destination Parameters (IP Address or Host Name and Port), and Statistics (Tx Throughput, Tx Volume, Tx Packets, Rx Throughput, Rx Volume, Rx Packets, and Latency). A 'Clear on Stop' checkbox is also visible.

Connection	Destination Parameters		Statistics (based on application data)						
	IP Address or Host Name	Port	Tx Throughput	Tx Volume	Tx Packets	Rx Throughput	Rx Volume	Rx Packets	Latency
Connection #01	127.0.0.1	2009	205 Kb/s	580 KB	N/A	N/A	0 B	N/A	N/A
Connection #02	127.0.0.1	2009	103 Kb/s	291 KB	204 p	N/A	0 B	0 p	N/A
Connection #03	NO_ADDRESS	2009							
Connection #04	NO_ADDRESS	2009							
Connection #05	NO_ADDRESS	2009							
Connection #06	NO_ADDRESS	2009							
Connection #07	NO_ADDRESS	2009							



The screenshot shows the 'NetDisturb Client' application window. The 'Flow #01: UDP' tab is active, displaying configuration options for impairments to apply on packets going from A to B. The 'Enhanced Edition' label is visible in the top right corner. The configuration is divided into three sections: Loss & Duplication, Delay & Jitter, and Content Impairment. The 'Loss & Duplication' section has a dropdown menu set to 'MOS VoIP' and a 'Change' button. The 'Delay & Jitter' section has a dropdown menu set to 'Throughput, Jitter, Time' and a 'Define' button. The 'Content Impairment' section has a dropdown menu set to 'Percentage' and a 'Define' button. There are also checkboxes for 'w/Log' and buttons for 'Run #01' through 'Run #04'.

Результати тестування

Мережа	Метод	Затримка(мс)	Відкинуті пакети (шт.)	Втрата пакетів (%)	MOS
1	Exp-Avg	300.16	1003	5.40	1.92
	F-Exp-Avg	680.84	68	1.24	1.00
	Min-D	258.44	1970	7.33	2.04
	Spike-Det	239.62	2421	10.02	2.03
	Window	356.04	699	3.33	1.65
	E-MOS	300.33	1345	8.14	2.33
	M-MOS	260.13	1393	5.64	2.22
	Запропонований	221.29	2125	8.13	2.30

ВИСНОВКИ

Проаналізовано сучасний стан технологій передачі даних в комп'ютерних мережах, в результаті чого доведено постійне збільшення відсотку трафіку реального часу, що є критичним до затримок.

Розглянуті основні принципи побудови комп'ютерних мереж з використанням VoIP, базові технології і протоколи. Проведені дослідження характеристик основних методів тестування даних мереж, в результаті чого показано, що більшість із них не дозволяє ефективно врахувати тимчасові зменшення смуги пропускання – спайки.

Запропоновано метод тестування, який полягає в застосуванні адаптивного буферного методу для розрахунку оптимальних характеристик спайку під час передавання головного потоку з наступною адаптацією параметрів тестування.

Виконано практичну реалізацію методу та проведено його дослідження в мережі системи віртуалізації «NICE Systems Ltd». За допомогою спеціалізованих програмних пакетів Wireshark, NetDirurb та Ip Traffic Generator була отримана практична статистика, на основі якої доведено переваги запропонованого методу тестування над аналогами.

Аналіз результатів досліджень показав, що запропонований метод може покращити результати тестування якості мовлення в більшості мережевих умовах, в порівнянні з іншими, і є більш ефективним, а також підходить для будь-якого реального буферного механізму



ДЯКУЮ ЗА УВАГУ