

Міністерство освіти і науки України  
Національний технічний університет  
«Дніпровська політехніка»  
Інститут електроенергетики  
(інститут)  
Факультет інформаційних технологій  
(факультет)  
Кафедра інформаційних технологій та комп'ютерної інженерії  
(повна назва)

**ПОЯСНЮВАЛЬНА ЗАПИСКА  
кваліфікаційної роботи магістра**

Студента Кабака Олега Валерійовича  
(ПІБ)

Академічної групи 123М-19-1  
(шифр)

Спеціальності 123 Комп'ютерна інженерія  
(код і назва спеціальності)

за освітньо-професійною програмою 123 Комп'ютерна інженерія  
(офіційна назва)

освітній рівень магістр  
(назва освітнього рівня)

на тему: «Обґрунтування структури комп'ютерної системи ділянки  
доменного виробництва»

Виконавець: студент 6 курсу, групи 123М-19-1 \_\_\_\_\_  
(підпис)

Кабак О. В.  
(прізвище та ініціали)

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинг.	інституційною	
Кваліфікаційної роботи	доц. Ткаченко С.М.			
Розділів:				
Теоретичний розділ	доц. Ткаченко С.М.			
Синтез системи	доц. Ткаченко С.М.			
Розроблення програмного забезпечення	ас. Бешта Л.В.			
Експериментальний розділ	доц. Ткаченко С.М.			

Рецензент				
-----------	--	--	--	--

Нормоконтролер	Проф.Цвіркун Л.І.			
----------------	-------------------	--	--	--

Дніпро  
2020

«ЗАТВЕРДЖУЮ»  
Завідувач кафедри  
Інформаційних технологій та  
комп'ютерної інженерії  
проф. Гнатушенко В.В.

"1" жовтня 2020 р.

**ЗАВДАННЯ**  
**на кваліфікаційну роботу**

*магістра*  
(назва освітньо-кваліфікаційного рівня)

студенту групи 123м-19-1 Кабаку Олегу Валерійовичу  
(група) (прізвище, ім'я та по батькові)

**Тема кваліфікаційної роботи** "Обґрунтування структури комп'ютерної системи ділянки доменного виробництва"

затверджена наказом ректора НТУ "Дніпровська політехніка"  
від «22» 10 2020 р. № 888-с

<b>Розділ</b>	<b>Зміст</b>	<b>Термін виконання</b>
<b>Стан питання і завдання дослідження</b>	<i>На основі матеріалів виробничих практик, інших науково-технічних джерел обґрунтувати необхідність досліджень роботи комп'ютерної системи управління і мережі рівня підприємства.</i>	<i>01.10.2020 р.</i>
<b>Теоретична частина</b>	<i>На основі науково-технічних джерел проаналізувати теоретичні методи дослідження комп'ютерних мереж</i>	<i>15.10.2020 р.</i>
<b>Синтез комп'ютерної системи підприємства</b>	<i>На основі характеристик комп'ютерної системи розробити програму людино-машинного інтерфейсу системи управління та налаштування комп'ютерної мережі.</i>	<i>01.11.2020 р.</i>
<b>Експериментальний розділ</b>	<i>Розробити математичну модель комп'ютерної мережі, провести обчислювальний експеримент і розробити рекомендації що до покращення параметрів.</i>	<i>15.11.2020 р.</i>
<b>Графічна частина</b>	<i>Графічні результати роботи подати у вигляді рисунків схем таблиць на 10 арк. формату А4.</i>	<i>05.12.2020 р.</i>

Завдання видав, кер. роботи

доц. Ткаченко С.М.  
(підпис)

Завдання прийняв до виконання

Кабак О.В.  
(підпис)

Дата видачі завдання 01.10.2020 р.

Термін подання дипломної роботи до ДЕК 10.12.2020 р.



## РЕФЕРАТ

Пояснювальна записка: 112 с., 44 рис., 5 табл., 2 додатки, 24 джерела.

**Об'єкт дослідження:** Комп'ютерна мережа підприємства, що лежить в основі забезпечення роботи, менеджменту, фінансових служб, транспорту та матеріально-технічного забезпечення. Також комп'ютерна мережа забезпечує онлайн контроль стану виробничого процесу по виплавці металу і стану виробничого обладнання в цілому.

**Мета:** Удосконалення комп'ютерної мережі підприємства.

Визначити вузли мережі, на які найсильніше впливає інформаційне перевантаження мережі.

Визначити умови, та параметри властивостей технічних пристроїв мережі за яких може виникнути втрата працездатності.

Розробити рекомендації що до модернізації мережі для підвищенні її стійкості до перевантажень.

Визначення найбільш «слабких» вузлів комп'ютерної мережі, які у випадку перевантаження приводять до втрати працездатності мережею і розробка рекомендацій для вирішення такої проблеми.

**Практичні результати:** визначення недоліків і шляхів удосконалення комп'ютерної мережі досягаються завдяки використанню наукового підходу до вирішення поставлених завдань. Застосовані наукові розробки з моделювання комп'ютерних систем і мереж показали свою достовірність у багатьох прикладах їх застосувань.

Результати перевірки у вигляді таблиць, графіків описані і наводяться у пояснювальній записці та додатках.

СИСТЕМА, КОМП'ЮТЕРНА МЕРЕЖА, МЕТАЛУРГІЯ,  
НАЛАШТУВАННЯ, МАТЕМАТИЧНА МОДЕЛЬ, ТЕОРІЯ МАСОВОГО  
ОБСЛУГОВУВАННЯ

## ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	6
Вступ	7
1 Стан питання і постановка завдання дослідження	9
1.1 Галузь застосування комп'ютерної системи	9
1.2 Комп'ютерна система управління тиском	13
1.2.1 Сучасні технічні засоби і методики організації управління процесами доменного виробництва	15
1.2.2 Інформаційне забезпечення системи	16
1.3 Функціональні особливості комп'ютерної системи	19
1.4 Організація комп'ютерної системи підприємства	20
1.4.1 Завдання дослідження	22
2 Теоретична частина	24
2.1 Розрахунок основних характеристик трафіку комп'ютерної мережі	24
2.1.1 Особливості сегмента мережі що об'єднує ПЛК систем управління	24
2.1.2 Комп'ютерна мережа підприємства	27
2.1.3 Імітаційна модель мережі в пакеті Cisco Packet Tracer	28
2.2 Аналітичні моделі комп'ютерних мереж	30
2.2.1 Загальна характеристика систем масового обслуговування	30
2.2.2 Оцінка ефективності розімкнутої багатоканальної системи масового обслуговування	34
2.2.3 Ефективність замкнутої багатоканальної системи масового обслуговування	37
2.2.4 Алгоритм згортання	39
3 Синтез комп'ютерної системи	44
3.1 Структура комплексу технічних засобів комп'ютерної системи	44
3.2 Характеристики використаних апаратних засобів системи управління	44
3.2.1 Розробка функціональної схеми автоматизації	47
3.2.2 Розробка схеми електричної принципової	37
3.3 Розробка програмного забезпечення	50
3.3.1 Призначення і область застосування програми	50
3.3.2 Обґрунтування технічних характеристик програми	51
3.3.2.1 Постановка завдання на розробку програми	51
3.3.2.2 Опис алгоритму функціонування програми	52
3.3.3 Опис і обґрунтування вибору методу організації вхідних та вихідних даних	53

3.3.3.1	Опис і обґрунтування вибору та складу технічних та програмних засобів	53
3.3.4	Опис розробленої програми	53
3.3.4.1	Загальні відомості	53
3.3.4.2	Функціональне призначення	54
3.3.4.3	Опис логічної структури програми	54
3.3.4.4	Використані технічні засоби	55
3.3.4.5	Виклик і завантаження програми	55
3.3.4.6	Вхідні і вихідні дані	55
3.3.5	Очікувані техніко-економічні показники	56
3.4	Людино-машинний інтерфейс	56
4	Експериментальний розділ	59
4.1	Розробка математичної моделі мережі як замкнутої системи масового обслуговування	59
4.2	Розрахунок параметрів мережі по її моделі	61
4.2.1	Параметри роботи мережі без впливу шкідливого програмного забезпечення	64
4.2.2	Параметри роботи мережі під впливом вірусних програм	65
4.3	Робота мережі із скоригованими характеристиками проблемних вузлів	67
	Висновки	72
	Перелік посилань	74
	Додаток А. Текст програми комп'ютерної системи керування тиском доменної печі	77
	Додаток Б. Текст програми математичної моделі мережі	94

## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ ТА ТЕРМІНІВ**

ОЗП—оперативнийзапомнюючий пристрій;

КЗ—контрольована зона;

АРМ—автоматизоване робоче місце;

КІСП – комп’ютерна інформаційна система підприємства;

ЛОМ – локальна обчислювальна мережа;

ПЛК – програмований логічний контролер;

LAN – локальна мережа;

КМ – комп’ютерна мережа;

СМО – система масового обслуговування;

МеМО – мережа масового обслуговування.

## ВСТУП

Безперервне зростання потужності доменних печей і впровадження нових методів інтенсифікації технологічного процесу істотно підвищили продуктивність доменного виробництва та збільшили потоки сировини, енергії і продуктів плавки, що використовуються у виробничому процесі.

Особливість доменної печі як об'єкта автоматизації полягає в тому, що кожен з трьох основних процесів, що в ній протікають (газодинаміка, теплообмін та масообмін), має свій оптимальний рівень, при якому досягаються найкращі показники плавки. Мистецтво управління доменною піччю полягає в тому, щоб вести газодинамічні, тепло і масообмінні процеси в ній на оптимальному рівні. Однак на практиці це досягається не завжди.

Функціональними органами управління великих цехів є планово-диспетчерське, технологічне бюро, бюро праці і заробітної плати тощо. Для забезпечення стратегічного, поточного й оперативного керівництва підприємством використовується функціональний апарат управління, безпосередньо підлеглий директору (президенту) і його заступникам. Кожний з них керує певною частиною роботи з управління виробничим процесом і має в розпорядженні відповідні функціональні служби.

У зв'язку з появою сучасних технологій обробки інформації та значною модернізацією обчислювальних засобів виникає попит на централізований збір інформації про технологічні об'єкти підприємства поширення її (надання доступу) в сучасних комп'ютерних мережах підприємства, а також на оновлення автоматизованих систем керування, які в подальшому можуть обслуговуватися сучасними спеціалістами.

***Мета роботи і завдання дослідження.*** Удосконалення комп'ютерної мережі підприємства. Визначити вузли мережі, на які найсильніше впливає інформаційне перевантаження мережі. Визначити умови, та параметри властивостей технічних пристроїв мережі за яких може виникнути втрата продуктивності. Розробити рекомендації що до модернізації мережі для підвищенні її стійкості до перевантажень.



**Об'єкт дослідження.** Комп'ютерна мережа підприємства, що лежить в основі забезпечення роботи, менеджменту, фінансових служб, транспорту та безпеки в організаційній структурі. Також комп'ютерна мережа забезпечує онлайн контроль стану виробничого процесу по виплавці металу і стану виробничого обладнання в цілому.

**Предмет і методи дослідження.** Предметом дослідження є структура комп'ютерної мережі, інформаційні властивості мережі а також технічні можливості апаратних засобів з урахуванням системи моніторингу і управління плавкою металу у доменній печі.

Для виконання завдань дослідження використовуються методи теорії масового обслуговування, розроблена математична модель комп'ютерної мережі як мережі масового обслуговування. Проведено дослідження властивостей мережі з різними параметрами інформаційного середовища та апаратних засобів.

**Ідея роботи.** Визначення найбільш «слабких» вузлів комп'ютерної мережі, які у випадку перевантаження приводять до втрати працездатності мережею і розробка рекомендацій для вирішення такої проблеми.

Практичні результати – визначення недоліків і шляхів удосконалення комп'ютерної мережі досягаються завдяки використанню наукового підходу до вирішення поставлених завдань. Застосовані наукові розробки з моделювання комп'ютерних мереж показали свою достовірність у багатьох прикладах їх застосувань.

# 1            С Т А Н            П И Т А Н Н Я            І            П О С Т А Н О В К А

## ЗАВДАННЯ ДОСЛІДЖЕННЯ

Доменна піч є постійно діючим апаратом шахтного типу. Завантаження шихти здійснюється зверху, через типовий завантажувальний пристрій, який одночасно є газовим затвором доменної печі. У домні відновлюють багату залізну руду.

Для успішної плавки чавуну у доменній печі повинні завжди виконуватися основні правила. По-перше, температура за всім об'ємом печі й тепло повинні забезпечувати протікання необхідних реакцій у потрібному місці та в певний час. Це відбувається за рахунок руху назустріч один одному двох потоків. Газ від згоряння палива піднімається від низу до верху, а шихта, що нагрівається теплом газу, спускається згори вниз. По-друге, шлак повинен утворюватися тільки тоді, як закінчиться відновлення заліза та необхідних домішок з руди. Тут важливо правильно підібрати під сорт чавуну тугоплавкість шлаку. Це необхідно для того, щоб шлак передчасно не сплавив руду, що призведе згодом до зміни складу чавуну і може викликати збій в процесі плавки [3].

### **1.1 Галузь застосування комп'ютерної системи**

Конструкція доменної печі влаштована таким чином, що шихта потрапляє в чашу через засипні пристрій (рис. 1.2), що виконаний у вигляді невеликого конуса, розташованого вгорі. Далі з чаші, потрапляючи на великий конус при його опусканні, шихта надходить у піч. Така система не дозволяє газу з доменної печі проникати в навколишнє середовище. Після завантаження малий конус та воронка для прийому сировини повертаються на кут, кратний 60 градусам. Це необхідно для того, щоб шихта розподілялася рівномірно.

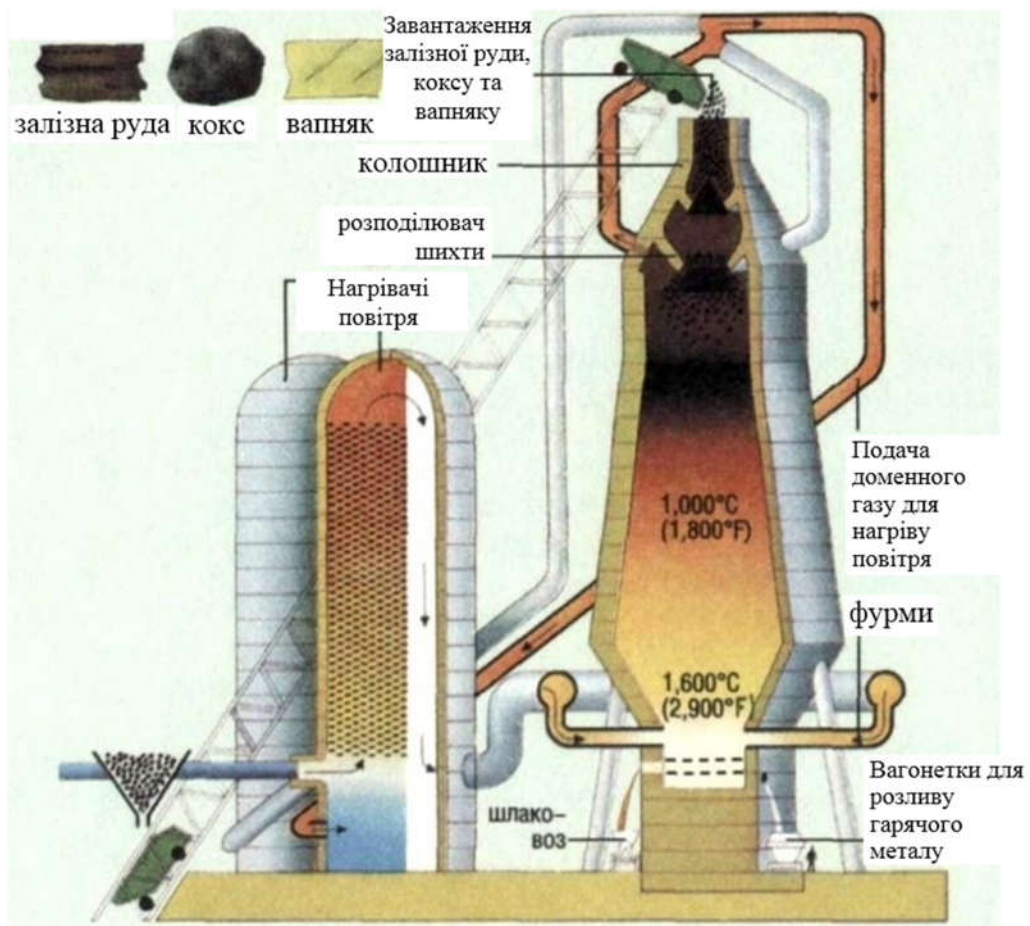


Рисунок 1.1 - Технологічна схема доменної печі

По мірі роботи металургійної печі шихта розплавляється та спускається далі вниз, звільняючи місце для нових порцій сировини. Корисний об'єм домни повинен бути завжди повністю заповнений. Сучасна доменна піч може мати корисний об'єм від 2 000 до 50 000 м<sup>3</sup>. Її висота може сягати 35 м, що майже втричі більше її діаметра. Така конструкція придумана не випадково: принцип роботи доменної печі заснований на русі матеріалів і газів назустріч один одному, що дозволяє збільшувати використання тепла до 85 %.

Горн та лещадь виконуються з цегли, що має у своєму складі велику кількість глинозему, або з вуглецевих блоків. Вони розташовані всередині сталевого кожуха й постійно у процесі роботи охолоджуються водою, що надходить по двом водопровідним системам з холодильників особливої конструкції. Причому коли перша система працює, друга в цей час знаходиться в резерві. Заплечики, шахта й розпар виготовляються з шамотної

цегли. Колошник оброблений плитами зі сталі, порожнини всередині яких повністю заповнені шамотом, а купол печі – плитами з чавуну.

У процесі роботи потрібні допоміжні пристрої і механізми, що забезпечують якісну плавку чавуну. Необхідними є пристрої для підйому і завантаження вихідної сировини в піч (рис. 1.1).

Доменна піч вимагає постійного обслуговування, особливо за умови випуску шлаку й чавуну. Для цього пристосовані ливарні двори, які обладнані мостовими кранами. Нагріте повітря для роботи печі та, як наслідок, високу температуру плавки при меншій кількості повітря, забезпечують повітрянагрівачі. Наприклад, у піч, що має корисний об'єм 2 000 м<sup>3</sup>, таке обладнання повинно подавати в хвилину 3 800 м<sup>3</sup> повітря, температура якого складає 1 200 градусів за Цельсієм. Пара, що утворюється на основі змішування зволоженого холодного повітря з повітрям на виході повітрянагрівача, повинна бути постійно вологою з різним ступенем вологості.

Стиснене повітря, що необхідне для спалювання палива, надходить у піч завдяки повітродувним машинам. Тиск гарячого дуття сучасних печей у зоні фурм сягає 410 кПа. Очищення колошникового газу відбувається за допомогою газоочисника.

Доменна піч представляє собою безперервно діючий апарат шахтного типу. Завантаження шихти здійснюється зверху, через засипний апарат 7 (рис. 1.2), який одночасно є газовим затвором доменної печі. У домні відновлюють багату залізну руду, агломерат або окатиші [3].

Доменна піч складається з п'яти конструктивних елементів: верхньої циліндричної частини – колошника 6, що необхідний для завантаження та ефективного розподілення шихти у печі; найбільш великої за висотою конічної частини, що розширюється, – шахти 5, в якій відбуваються процеси нагріву матеріалів та відновлення залізу з оксидів; найбільш широкої циліндричної частини – розпару 4, в якому відбуваються процеси розм'якшення та плавлення відновленого заліза; конічної частини, що

звужується, – заплечиків 3, де створюється відновлювальний газ (монооксид вуглецю); циліндричної частини – горна 2 для накопичення рідинних продуктів доменного процесу (чавуну та шлаку). Така конструкція печі забезпечує рух матеріалів та газів назустріч друг другу, що дозволяє збільшити використання тепла до 85 %.

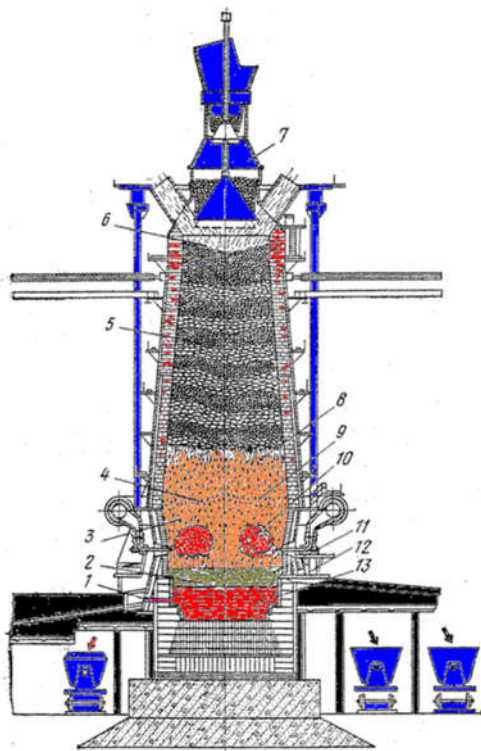


Рисунок 1.2 - Розріз (а) та профіль робочого простору доменної печі (б):

- |                          |                                |                              |
|--------------------------|--------------------------------|------------------------------|
| 1 – чавунна летка;       | 2 – горн;                      | 3 – заплечики;               |
| 4 – розпар;              | 5 – шахта;                     | 6 – колошник;                |
| 7 – засипний апарат;     | 8 – горизонт створення чавуну; | 9 – горизонт створення шлаку |
| 10 – зона горіння коксу; | 11 – шар шлаку;                | 12 – шлакова летка;          |
| 13 – розплавлений чавун  |                                |                              |

У верхній частині горна знаходяться фурми – отвори для подачі під тиском нагрітого до високої температури та збагаченого киснем повітря (гарячого дуття). Це дозволяє економити значну кількість коксу в процесі виготовлення чавуну. Гаряче дуття створюється повітрянагрівачами (рис.1.2) – спеціальними апаратами регенеративного типу (кауперами), що застосовуються разом із доменною піччю.

## 1.2 Компютерна система управління тиском

Більшість доменних печей наразі працюють з підвищеним тиском газів під колошником (76 – 152 кПа) за умови тиску горнових газів 300 – 400 кПа. Це значно зменшує перепад тиску між фурмами та колошником печі, знижує швидкість газу у стовпці шихтових матеріалів, що сприяє більш рівному ходу печі. Зменшення швидкості газу дозволяє більш повно використовувати хімічну енергію газів та вести доменний процес більш форсовано. При цьому зменшуються витрати коксу на 3 – 5 % та підвищується продуктивність доменної печі на 3 – 6 %.

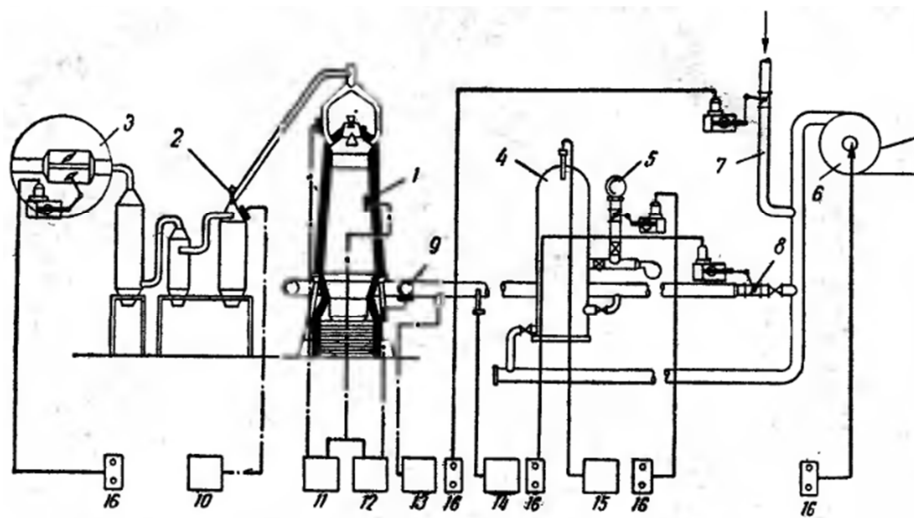


Рисунок 1.3 - Функціональна схема доменної печі

1 – доменна піч; 2 – очищувач газу; 3 – дросельна група; 4 – повітрянагрівач; 5 – газопровід доменної печі; 6 – турбоповітродувка; 7 – паропровід; 8 – змішувальний клапан; 9 – кільцевий повітропровід; 10 – датчик тиску на колошнику; 11 – датчик верхнього перепаду по шахті; 12 – датчик нижнього перепаду; 13 – датчик вологості дуття; 14 – датчик температури дуття; 15 – датчик температури купола; 16 – пульт дистанційного керування

Для забезпечення оптимальної швидкості газу, коли витрати коксу та продуктивність печі мають свої найліпші показники, в окремих зонах шахти печі повинні підтримуватися різні перепади тиску [4]. Так, між кільцевим повітропроводом гарячого дуття 9 (рис. 1.3) та середньою точкою по шахті печі так званий нижній перепад тиску повинен дорівнювати 76 кПа.

Зміна перепадів тиску в різних зонах шахти доменної печі забезпечується зміною тиску газу на колошнику, або зміною витрат гарячого дуття.

У нашому випадку розглядається канал автоматичного керування «витрати гарячого дуття - перепад тиску у нижній частині шахти доменної печі». А зміна тиску газу на колошнику для нашого випадку є збуренням в системі автоматичного управління.

Зміна витрат гарячого дуття доменної печі досягається через регулювання продуктивності турбоповітродувки 6 (рис. 1.3). Турбоповітродувка 6 має асинхронний двигун потужністю 2 500 кВт, отже у якості виконуючого пристрою системи автоматичного управління можна використовувати перетворювач частоти. Неперервна зміна швидкості обертання валу двигуна відбувається лише у діапазоні 75 – 100 % від його номінальної частоти обертання. Це забезпечує зміну витрат гарячого дуття у фурмах, відповідно, від 2100 до 2800 м<sup>3</sup>/хв. Для більш ефективного керування перепадом тиску у нижній частині шахти доменної печі слід забезпечити внутрішній контур автоматичного керування за витратами гарячого дуття, отже необхідно вимірювати витрати гарячого дуття. Для вимірювання гарячого дуття зазвичай застосовують окрему імпульсну лінію довжиною декілька десятків метрів, що під'єднується до повітропроводу гарячого дуття.

Також, звичайно, необхідно вимірювати перепад тиску у нижній частині шахти доменної печі, що може змінюватись у діапазоні від 65 до 90 кПа. Зазвичай перепад тиску по шахті печі замірюють сильфони диференційним манометром компенсаційного типу (поз. 12 на рис.1.3) між кільцевим повітропроводом гарячого дуття 9 (рис.1.3) та середньою точкою по шахті печі. Середня точка заміру статичного тиску в шахті печі розміщена на 8 м нижче рівня заслони при загальній висоті стовпа матеріалів у шахті 21м.

### **1.2.1 Сучасні технічні засоби і методики організації управління процесами доменного виробництва**

Задачі підтримки газодинамічних, тепло і масообмінних процесів на оптимальному рівні вирішують інтелектуальні системи підтримки прийняття рішень операторами печі. Ці системи формують уставки для регуляторів основних робочих параметрів печі, що знаходяться на нижньому рівні (рівні ПЛК) автоматизованої системи управління технологічним процесом та входять до складу локальних систем автоматичного управління [1].

Система автоматичного управління перепадом тиску у нижній частині шахти доменної печі є одною з локальних систем управління, що виконують завдання інтелектуальної системи управління масообмінними процесами у шахті доменної печі. Отже, основною задачею для системи автоматичного управління перепадом тиску у нижній частині шахти доменної печі є якомога точне відтворення зміни реального перепаду тиску за зміною уставки даного робочого параметру печі. Розглянемо більш детально особливості процесу управління масообмінними процесами у шахті печі.

Рух вниз шихтових матеріалів обумовлюється плавленням рудно-флюсового частини шихти і горінням коксу. Якщо шар шихти опускається рівномірно, то хід печі називається рівним і при цьому спостерігається максимальна продуктивність. Якщо спостерігається підвисання й обриви шихти, то все це призводить до нерівномірного ходу і знижує продуктивність – це небажане явище.

На хід печі впливають два суперечливих фактори. Для збільшення продуктивності печі бажано збільшити витрату дуття. Але для кожного реального стану профілю шахти, який визначається наростанням гарнісажу, існує певне критичне значення витрати дуття, при якому динамічний напір викликає припинення сходу шихти, що веде до зниження продуктивності. Завданням управління ходом печі є визначення та підтримка такого стану, при якому використовується максимально можлива витрата дуття без порушення рівномірного сходу шихти.



Непрямими показниками, що характеризують сход шихти, є локальні перепади тиску газу за висотою шахти. Використовується нижній перепад тиску, що формується датчиком між гарячим дуттям і серединою шахти, верхній перепад тиску та загальний перепад тиску. Ці локальні перепади характеризують втрати тиску на різних ділянках шахти за її висотою, а ці втрати, у свою чергу, характеризують силу, що тримає шихту. Також контролюється загальні витрати дуття на піч.

Керуючими впливами на хід шихти є: порядок завантаження матеріалів в піч через засипний апарат (управління зверху), управління витратою та вологістю дуття ( управління знизу).

У нашому випадку відбувається керування ходом шихти через зміну витрат гарячого дуття доменної печі. Хід шихти контролюється непрямим способом через локальний нижній перепад тиску за шахтою доменної печі. При цьому паралельно відбувається управління газодинамічними процесами через зміну тиску на колошнику печі, що може впливати на нижній перепад тиску за шахтою печі. Тому, окрім відтворення уставки з максимальною точністю другою задачею системи автоматичного управління перепадом тиску у нижній частині шахти доменної печі є стабілізація перепаду тиску на заданому рівні за умови наявності збурення у системі (за умови зміни тиску газу на колошнику печі) [2,4].

### **1.2.2 Інформаційне забезпечення системи**

Розглянемо організаційну структуру відділу головного енергетика (рис. 1.4).

Цех - головна виробнича ланка заводу, яку складають основні, допоміжні дільниці і інші підрозділи, що приймають участь в виготовленні продукції або напівфабрикатів. Також, як і завод, цех має лінійно-функціональну організаційну структуру управління.



Рисунок 1.4 – Організаційна структура

Організаційна структура управління - це сукупність певним чином пов'язаних між собою управлінських ланок. Вона характеризується кількістю органів управління, порядком їхньої взаємодії та функціями, які вони виконують. Головне призначення організаційної структури - забезпечити ефективну діяльність управлінського персоналу. Вона безпосередньо пов'язана з виробничою структурою підприємства. Принцип формування структури управління - організація та закріплення тих або інших функцій управління за підрозділами (службами) апарату управління.

Організаційна структура апарату управління характеризується різною кількістю ланок, частіше за все використовується система "трюх": директор (президент, менеджер) - начальник цеху - майстер. Кожний з них несе особисту відповідальність за доручену йому ділянку роботи [22].

Начальник цеху відповідає за всі сторони роботи цеху і виконує всі функції з технологічного і господарського керівництва цеху за допомогою підлеглого йому апарату управління. Функціональними органами управління великих цехів є планово-диспетчерське, технологічне бюро, бюро праці і заробітної плати тощо. Начальник цеху підпорядковується безпосередньо директору.

Великі дільниці цеху (відділення, прольоти) очолюють начальники ділянок (старші майстри), яким підлеглі змінні майстри. Майстер є керівником і організатором виробництва та праці на дільниці. Він підпорядковується безпосередньо начальнику цеху, а там, де є начальники дільниць або змін, - відповідно начальнику дільниці або зміни. Групою робітників, об'єднаних у бригаду, керує бригадир, який є старшим робітником і не звільняється від виробничої роботи, одержуючи доплату до тарифної ставки за виконання своїх обов'язків.

Діяльністю підприємства керує директор (президент, менеджер), який може бути як власником майна, так і найманим працівником (в останньому випадку з ним підписується контракт). Для забезпечення стратегічного, поточного й оперативного керівництва підприємством використовується функціональний апарат управління, безпосередньо підлеглий директору (президенту) і його заступникам. Кожний з них керує певною частиною роботи з управління виробничим процесом і має в розпорядженні відповідні функціональні служби. В апараті заводоуправління (управління фірмою) виділяються функціональні структурні підрозділи (відділи, служби), а в цехах, як правило, - бюро. Апарат управління складається з таких основних служб:

- оперативного керівництва підприємством;
- управління персоналом;
- економічної й фінансової діяльності;
- перероблення інформації;
- адміністративного управління;
- маркетингу;
- зовнішніх економічних зв'язків;
- технічного розвитку тощо.

### **1.3 Функціональні особливості комп'ютерної системи**

Комп'ютерна система повинна виконувати наступні функції:

Збір інформації. Повинен забезпечуватися прийом інформації від датчиків на технологічному об'єкті та передача цих даних для подальшого аналізу та обробки.

Аналіз та обробка інформації. Комп'ютерна система повинна на підставі отриманих даних визначати необхідність внесення змін в хід технологічного процесу.

Зберігання оперативних даних системи, даних для формування аналітичних звітів, документів системи, сформованих у процесі роботи звітів. Ця функція повинна забезпечити періодичне резервне копіювання і збереження даних на додаткових носіях інформації.

Формування звітності. У комп'ютерній системі повинна забезпечуватися можливість формування різних видів звітів. Ця функція повинна забезпечувати механізми гнучкого налаштування, а також інструментарій щодо формування нових звітних форм.

Керування. Система повинна мати можливість здійснювати безпосереднє вмикання/вимикання виконуючих пристроїв на технологічному обладнанні.

Для виконання вищеперерахованих функцій система повинна мати інформацію про стан та зміни технологічних параметрів.

Частина комп'ютерної системи, що відповідає за перебіг технологічного процесу, повинна функціонувати в реальному масштабі часу. Час реакції системи має бути не більше 500 мс.

Пропускна здатність, яка дозволить комфортно і без затримок працювати з системою, за умов використання каналу тільки в межах впроваджуваної комп'ютерної системи повинна становити не менше 3 Мбіт/с.

Крім того мережа Ethernet використовується для обміну документами між адміністративними підрозділами, тому мінімальною пропускнуою здатністю каналу повинна бути не меншою за 10 Мбіт/с.

#### **1.4 Організація комп'ютерної системи підприємства**

Виходячи з вимог система повинна забезпечувати управління об'єктом отже до її складу повинна входити підсистема управління технологічним обладнанням. Дана підсистема складається з пристроїв збору інформації (датчики тиску на середньому та нижньому рівнях доменної печі, датчик витрат гарячого дуття), еталонів стану обладнання (перепад тиску в доменній печі), системи автоматичного контролю стану обладнання, програми управління яка повинна реалізувати управління дуттям (турбоповітродувкою).

Крім того система повинна забезпечувати візуалізацію та контроль, за технологічним процесом, тому до неї повинна входити підсистема інформаційного забезпечення роботи оператора. Дана підсистема складається з реєстрації параметрів процесу, людино-машинного інтерфейсу та сигналізації досягнення параметрами заданих значень [4].

Також системою повинно забезпечуватися архівування технологічних процесів які відбуваються, цьому в неї повинна бути підсистема ведення архівів параметрів та подій, включно з базою даних та резервним сховищем.

Розроблена структурна схема інформаційних потоків наведена на рис. 1.5. Дана структура забезпечує управління перепадом тиску у нижній частині доменної печі, збір даних про технологічний процес, візуалізацію отриманих даних, збереження їх в базі даних та створення їх резервних копій. Крім того дана структура забезпечує контроль, за обладнанням та сигналізацію досягнення параметрами управління заданих значень.

Таким чином згідно з вимогами підсистема автоматизованого управління технологічним обладнанням представляє собою апаратно програмний комплекс до якого входять датчики тиску розміщені на

середньому і нижньому рівнях доменної печі, датчик витрат дуття, об'єкт управління та турбоповітродувку.

Підсистема інформаційного забезпечення роботи оператора представляє собою апаратно програмний комплекс на базі персонального комп'ютера з SCADA системою zenonSupervisor.

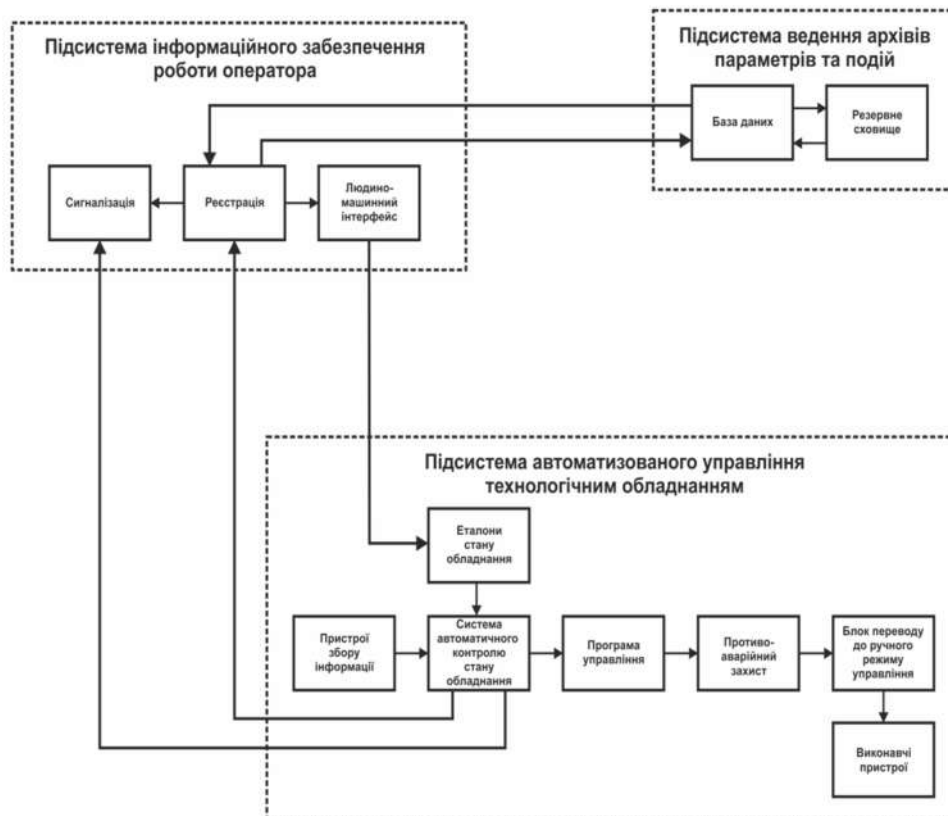


Рисунок 1.5 – Структурна схема інформаційних потоків

Підсистема ведення архівів параметрів та подій представляє собою окремий програмний модуль SCADA системи zenon.

Зв'язок між наведеним апаратним комплексом згідно з вимогами забезпечується за допомогою загальнопромислової мережі підприємства, а між програмним забезпеченням за допомогою стандартних протоколів.

### **1.4.1 Завдання дослідження**

Завдання аналізу продуктивності комп'ютерної мережі (КМ) є однією з найважливіших серед завдань проектування КМ. У загальному вигляді задача аналізу продуктивності КМ полягає в тому, щоб оцінити показники продуктивності конкретної КМ при заданих параметрах її технічного, програмного забезпечення і зовнішнього середовища. До таких параметрів можуть ставитися швидкодії пристроїв, характеристики складності програм, інтенсивності потоків вимог на виконання програм, умови зовнішнього оточення (пропускна здатність орендованих ліній зв'язку, наявність випадкових і навмисних впливів інформаційного та іншого характеру, що призводять до непередбачуваного функціонування КМ), і інші.

Вирішуючи завдання аналізу продуктивності, необхідно враховувати випадкову природу багатьох факторів, від яких залежить продуктивність КМ. Так, випадковими часто є моменти надходження в КМ вимог, обсяги інформації що підлягає обробці, послідовність необхідних для її обробки операцій.

Складність структури КМ і необхідність обліку випадкових факторів роблять завдання аналізу продуктивності КМ дуже складною. Тому все більш широке поширення для аналізу продуктивності КМ отримує метод математичного моделювання.

Теорія масового обслуговування пов'язана з розробкою і аналізом математичних, тобто абстрактних, моделей, які описують процес обслуговування деяких об'єктів, що надходять на вхід обслуговуючого приладу у вигляді деякого потоку, і утворює в загальному випадку чергу на вході обслуговуючого приладу.

Метою використання СМО як моделі є аналіз якості функціонування систем-оригіналів.

У свою чергу, МеМО використовують для визначення найважливіших системних характеристик інформаційних систем:

продуктивності; часу доставки пакетів; ймовірності втрати повідомлень і блокування в вузлах; області допустимих значень навантаження, при яких забезпечується необхідну якість обслуговування та ін.

Головним завданням дослідження у кваліфікаційній роботі є визначення допустимих значень навантаження на існуючу мережу за яких мережа буде забезпечувати обробку запитів з мінімальною чергою.

Для забезпечення запасу по навантаженню необхідно промоделювати мережу з параметрами, що імітують дію шкідливого програмного забезпечення і визначити здатність мережі виконувати свої основні функції в цих умовах.

Розробити варіанти поліпшення роботи мережі в умовах значного інформаційного навантаження.



## **2 ТЕОРЕТИЧНА ЧАСТИНА**

### **2.1 Розрахунок основних характеристик трафіку комп'ютерної мережі**

#### **2.1.1 Особливості сегмента мережі що об'єднує ПЛК систем управління**

У мережах Profibus контролери з'єднуються, використовуючи технологію головний (підлеглий, при якій тільки один пристрій (головний) може ініціювати передачу (зробити запит). Інші пристрої (підлеглі) передають прошені головним пристроєм дані, або виконують прошені дії. Типовий головний пристрій включає провідний (HOST) процесор і панелі програмування. Типовий підпорядкований пристрій - програмований контролер.

Головний контролер може адресуватися до індивідуального підлеглого або може ініціювати широкомовну передачу повідомлення на усі підпорядковані пристрої. Підпорядкований пристрій повертає повідомлення у відповідь на запит, що адресується саме йому. Відповіді не повертаються при широкомовному запиті від головного [1,14].

У нашій системі АСКОК передбачається використання двох видів посилок: посилки контролю і управління. Посилки контролю - запит від головного (пульта управління) і відповідь від підлеглого (контролера). Посилки управління - команди від головного і звіт про нормальний прийом команди підлеглим.

Типові фрейми повідомлень протоколу Profibus представлені на рис.2.1...2.4.

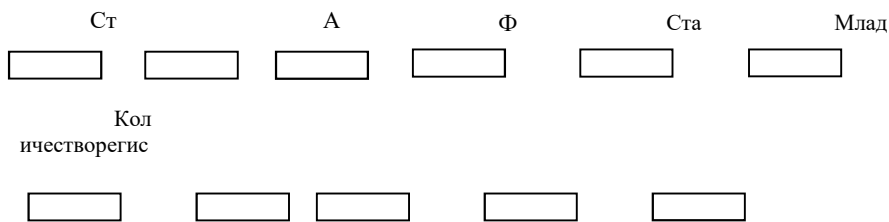


Рисунок 2.1 - Запит від головного

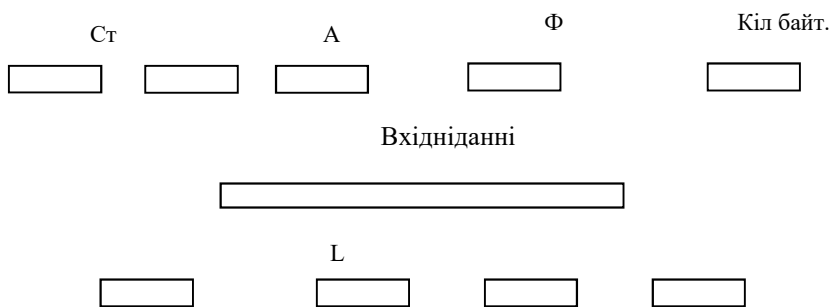


Рисунок 2.2 - Відповідь від підлеглого



Рисунок 2.3 – Відповідь від головного

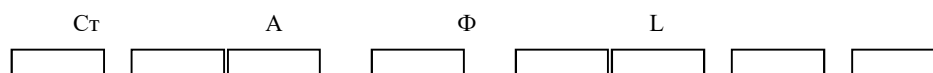


Рисунок 2.4 - Звіт про нормальний прийом підлеглим

Розрахуємо реальну швидкість опитування:

$$V_{опр} = K_{опт} \frac{V_{\kappa}}{n_{уст} \cdot l_{нооп} \cdot l_{байта}} \text{ біт/с} \quad (2.1)$$

Де:  $K_{opt}$  - коефіцієнт оптимізації (для Profibus 0,63);  $V_k$  – швидкість каналу: 9600, 19200, 57600, 115200 біт/с (для довжини 1 000 м, швидкість обмежена 19 200 біт/с);  $n_{уст}$  – кількість контролерів (типове значення 2...6);  $l_{noop}$  – довжина посилок між головним і підпорядкованим під час опитування у байтах (21 байт - максимальна довжина посилки, що відповідає максимально інформативною);  $l_{байта}$  - умовна довжина байту.

$$V_{opr} = 0,63 \frac{19200}{6 \cdot 21 \cdot 11} = 8,72 \text{ кбіт/с.}$$

Розрахуємо реальну швидкість управління :

$$V_{упр} = K_{opt} \frac{V_k}{n_{уст} \cdot l_{ууп} \cdot l_{байта} + l_{лопр} * l_{байта}} \text{ біт/с} \quad (2.2)$$

Де:  $l_{ууп}$  - довжина посилки управління, приймемо умовно рівну  $l_{noop}$

$$V_{упр} = 0,63 \frac{19200}{6 * 21 \cdot 11 + 21 \cdot 11} = 7,48 \text{ біт/с.}$$

Розрахуємо отриманий час опитування і час команд управління.

$$T_{упр} = \frac{1}{V_{упр}} \text{ с} \quad (2.3)$$

$$T_{opr} = \frac{1}{V_{opr}} \text{ с} \quad (2.4)$$

$$T_{упр} = \frac{1}{7,48} = 0,13 \text{ с.}$$

$$T_{opr} = \frac{1}{8,72} = 0,114 \text{ с.}$$

Проаналізувавши отриманий результат можна сміливо стверджувати, що час опитування і час команд управління, що передаються по каналу зв'язку, не вносить істотної затримки, так це як час значно менше часу можливого оновлення інформації [23].

### 2.1.2 Комп'ютерна мережа підприємства

В рамках роботи розроблена комп'ютерна мережа:

мережний блок: 192.168.48.0/20;

кількість ПК у кожній LAN: LAN1 – 85, LAN2 – 40, LAN3 – 7, LAN4 – 100, LAN5 – 120;

VLAN згідно табл. 2.1.

Таблиця 2.1 – VLAN

Номер VLAN	Ім'я VLAN	Примітка
1	default	Не використовується
13	Accounting	Для бухгалтерії
33	Resources Department	Для відділу кадрів
43	Guest	Для гостей
99	Management	Для управління пристроями
100	Native	Власна

Задавати назви пристроям за наступним правилом:

*Kabak\_тип пристрою\_номер пристрою.*

Під час налаштування комп'ютерної мережі встановлені паролі:

– консолі *ivty* – *cisco123\_ss*;

– привілейованого режиму – *class123\_ss*.

При налаштуванні ssh:

користувач: *123\_16\_SK\_Kabak*;

пароль користувача: *Kabak*.

Порти на комутаторах розподілені відповідно до таблиці 2.1.

### 2.1.3 Імітаційна модель мережі в пакеті Cisco Packet Tracer

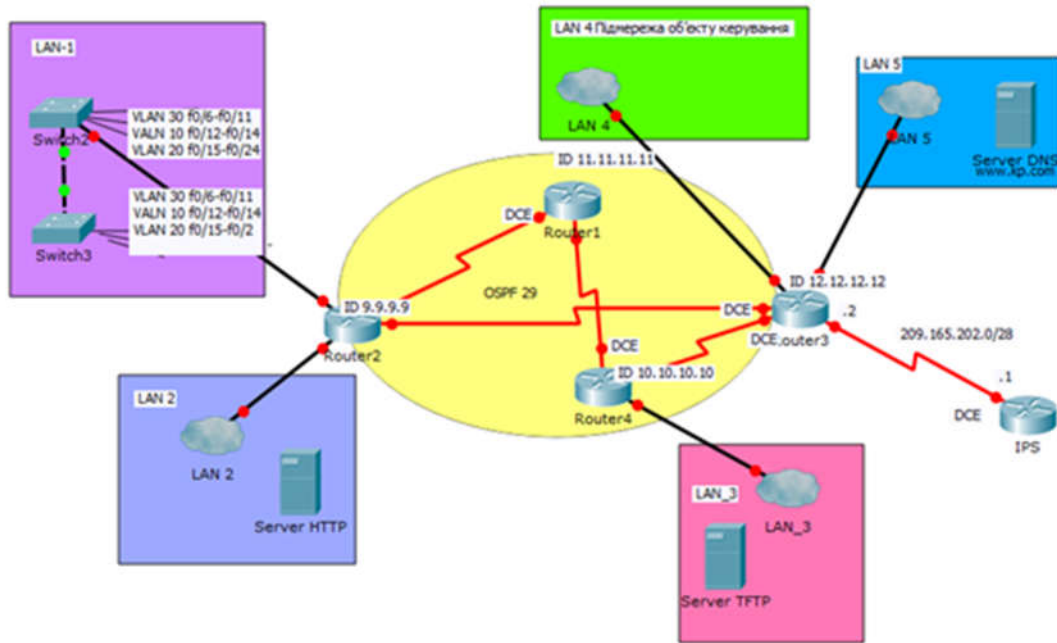


Рисунок 2.5—Модель мережі

Під час налаштування пулів DHCP і NAT встановлені такі дані:

- ім'я пулу DHCP - pool\_vlanN;
- ім'я пулу NAT – Internet123\_ss;
- пул адресів NAT - 209.165.202.5 – 209.165.202.30;
- номер списку доступу – 10.

При традиційному поділі на підмережі до всіх підмереж застосовується одна і та ж маска підмережі. Це означає, що всі підмережі містять однакове число доступних адрес вузлів та використовують одну і ту ж маску підмережі.

При використанні VLSM довжина маски підмережі залежить від числа бітів, запозичених для окремої підмережі, тобто від «змінної» частини маски підмережі змінної довжини. VLSM дозволяє розділити простір мережі на нерівні частини.

При використанні VLSM мережу спочатку розділяється на підмережі, а потім підмережі, в свою чергу, також розбиваються на підмережі. Цей процес

можна повторювати багаторазово для створення підмереж різних розмірів [10].

Згідно завдання, у кожній підмережі повинно бути не менше 8 комп'ютерів. При цьому:

перші можливі для використання IP-адреси призначено інтерфейсам і підінтерфейсам маршрутизаторів у LAN;

другі з можливих IP-адрес призначено комутаторам у кожній LAN;

сервери налаштовано і їм привласнено IP-адреси за правилом: IP-адрес дорівнює першому можливому адресу у мережі+9+1;

останні з використовуваних IP-адрес призначено вузлам;

в мережах VLAN використовується адресація кінцевих пристроїв по протоколу DHCP.

З рис.2.5 видно, що топологія об'єднує 5 мереж з хост-вузлами, 3 мережі маршрутизаторів, 1 мережа зовнішнього шлюзу з заданою адресою мережі 209.165.202.0/28. Мережі маршрутизаторів та зовнішнього шлюзу потребують по 2 IP-адреси кожна.

В табл.2.2 представлена схема IP-адресації мереж організації методом VLSM.

Таблиця 2.2 – Схема адресації мереж

Ім'я	Кіл-	Адрес мережі	Маска мережі	Діапазон адрес
------	------	--------------	--------------	----------------

мережі	сть вузлів			
LAN5	120	192.168.48.0	255.255.255.128	192.168.48.1 – 192.168.48.126
LAN4	100	192.168.48.128	255.255.255.128	192.168.48.129 – 192.168.48.254
LAN1	85	192.168.49.0	255.255.255.128	192.168.49.1 – 192.168.49.126
VLAN13		192.168.49.0	255.255.255.224	192.168.49.1 – 192.168.49.30
VLAN23		192.168.49.32	255.255.255.224	192.168.49.33 – 192.168.49.62
VLAN33		192.168.49.64	255.255.255.224	192.168.49.65 – 192.168.49.94
VLAN99		192.168.49.96	255.255.255.224	192.168.49.97 – 192.168.49.126
LAN2	4 0	192.168.49.128	255.255.255.192	192.168.49.129 – 192.168.49.190
LAN3	7	192.168.49.192	255.255.255.240	192.168.49.193 – 192.168.49.206
WAN1-2	2	192.168.49.208	255.255.255.252	192.168.49.193 – 192.168.49.206
WAN2-3	2	192.168.49.212	255.255.255.252	192.168.49.209 – 192.168.49.210
WAN2-4	2	192.168.49.216	255.255.255.252	192.168.49.213 – 192.168.49.214
WAN2-3	2	192.168.49.220	255.255.255.252	192.168.49.217 – 192.168.49.218

## 2.2 Аналітичні моделі комп'ютерних мереж

### 2.2.1 Загальна характеристика систем масового обслуговування

Дослідженням черг теорія масового обслуговування займається шляхом вивчення своєрідного класу систем, званих *системами масового обслуговування* (СМО) [3]. Кожна така система об'єднує деяку кількість технічних пристроїв (іноді людей – виконавців), що паралельно діють, що називають *каналами обслуговування* (можливий випадок, коли канал один). Вони можуть самостійно (незалежно) виконувати всі операції, що лежать в основі функціонування СМО, і збільшувати тим самим пропускну спроможність системи, яка визначається числом обслуговувань, завершених в одиницю часу.

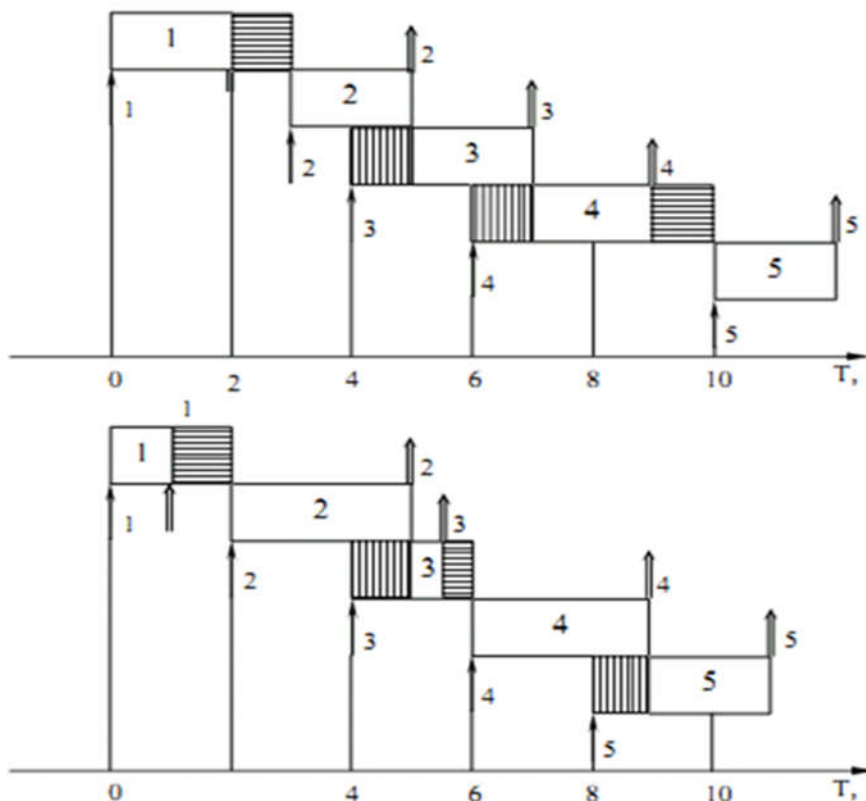


Рисунок 2.6 - Детермінований графік роботи при імовірнісному характері надходження заявок(б), при розсіюванні часу (в):

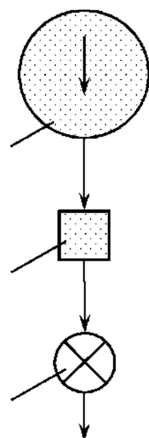


У будь-якій СМО існує небезпека виникнення черги заявок (вимог) в одні періоди часу і простою устаткування (виконавців) через відсутність замовлень в інші періоди, тому головна задача досліджень полягає в пошуку зв'язків між показниками (критеріями) ефективності тієї або іншої системи, її структурою і умовами роботи. В якості критеріїв вибирають середній час обслуговування заявок, що поступають, середню (або максимальну) довжину черги, кількість каналів обслуговування і ін.

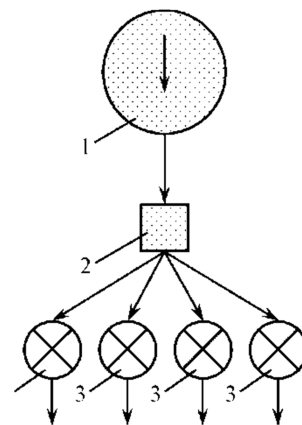
Теорія масового обслуговування розглядає різні системи масового обслуговування. Всі вони складаються з трьох основних елементів: джерела вимог, накопичувача і вузла обслуговування. Якщо вузол обслуговування складається всього з одного обслуговуючого пристрою (виконавця), то така система називається *одноканальною* (рис. 2.7 а, рис. 2.8 а), при більшому ж числі пристроїв (виконавців) – *багатоканальною* (рис. 2.7 б, рис. 2.8 б).

Розрізняють розімкнені (рис. 2.7) та замкнуті (рис. 2.8) системи масового обслуговування. Нерідко число вимог, які можуть поступати в систему, буває необмеженим або таке припущення є зручним [8].

Ознакою замкнутої системи є те положення, що обслужені вимоги не покидають систему, а повертаються в джерело вимог (рис. 2.8). При цьому інтенсивність вхідного потоку залежить від кількості вимог, що знаходяться в системі.



а)



б)

Рисунок 2.7 - Розімкнута одноканальна (а) і багатоканальна (б) система масового обслуговування

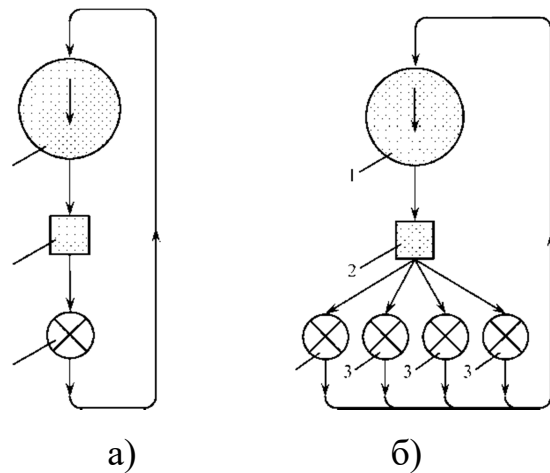


Рисунок 2.8 - Замкнута одноканальна (а) і багатоканальна (б) система масового обслуговування

Розрізняють ще системи масового обслуговування з втратами (з відмовами) і без втрат (з чергою). Перша група систем характеризується тим, що вимога не може чекати почала обслуговування або система обслуговування відмовляє вимозі, якщо всі обслуговуючі пристрої (виконавці) зайняті.

Системи без втрат (з чергою) називаються *системами з необмеженим очікуванням*. Є також системи з *обмеженим очікуванням*, в яких вимоги можуть чекати обслуговування тільки обмежений час, після закінчення якого вони покидають систему.

Розрізняють також системи з пріоритетом і без пріоритету. У системах без пріоритету вимоги обслуговуються в порядку їх надходження. Якщо деякі з вимог поступають на обслуговування насамперед (незалежно від того, коли вони поступили в накопичувач), такі системи називаються *системами масового обслуговування з пріоритетом*.

З усіх перерахованих СМО детально розглянемо лише дві, що якнайповніше відповідають умовам нашого завдання: розімкнену і замкнуту багатоканальні системи [6].

### 2.2.2 Оцінка ефективності розімкненої багатоканальної системи масового обслуговування.

Розімкнута багатоканальна система масового обслуговування (система з декількома виконавцями) також, як і інша СМО, може знаходитися в множині різних станів. Позначимо через  $E_n$  стан, при якому в системі (накопичувачі та вузлі обслуговування) знаходиться  $n$  вимог. За час  $dt$  в системі можливі наступні переходи з певною вірогідністю з одного стану в інше (табл. 1.). Тут  $S$  – число пристроїв (каналів), що обслуговуються;  $\lambda$  – інтенсивність потоку вимог.

Таблиця 2.3 – Вірогідності переходу системи в можливі стани

Співвідношення $n$ і $S$	Зміна стану	Вірогідність переходу
$n \leq S$	$E_0 \rightarrow E_0$	$1 - \lambda dt$
	$E_0 \rightarrow E_1$	$\lambda dt$
	$E_1 \rightarrow E_0$	$\frac{\mu}{S} dt$
	$E_n \rightarrow E_n$	$1 - (\lambda + \frac{\mu n}{S}) dt$
	$E_n \rightarrow E_{n+1}$	$\lambda dt$
	$E_n \rightarrow E_{n-1}$	$\frac{n\mu}{S} dt$
$n > S$	$E_n \rightarrow E_n$	$1 + (\lambda + \mu) dt$

	$E_n \rightarrow E_{n+1}$	$\lambda dt$
	$E_n \rightarrow E_{n-1}$	$\mu dt$

Для розрахунку СМО необхідно визначити вірогідність нульового стану системи  $P_0$  і поточного стану системи  $P_n$ :

$$(2.1) \quad P_0 = 1 / \left( \frac{S^n \psi^n}{S!(1-\psi)} + \sum_{n=0}^{S-1} \frac{S^n \psi^n}{n!} \right)$$

$$P_n = P_0 \frac{S^n \psi^n}{n!}$$

, якщо  $n \leq S$  та

$$P_n = P_0 \frac{S^n \psi^n}{S!}$$

, якщо  $n > S$ ,

$$\psi = \frac{\lambda}{\mu}$$

де.

Величина  $\psi$  грає важливу роль. Вона виражає ступінь насиченості системи і називається коефіцієнтом використання системи, або завантаження системи. Якщо  $\psi > 1$  сталого режиму роботи не вийде, і черга ростиме необмежено, оскільки інтенсивність обслуговування  $\mu$  менше інтенсивності надходження вимог  $\lambda$ . Тому при розв'язанні задач масового обслуговування приймає, що  $\psi \leq 1$ .

Крім того, по величині  $\mu$  можна визначити середнє число вільних (таких, що простоюють) каналів обслуговування:

$$\bar{\rho} = (1 - \psi)S. \quad (2.2)$$

Таким чином, ми маємо рівняння, необхідні для вирішення задачі масового обслуговування з урахуванням вимог розімкненої системи з пристроями. Знаючи вірогідність стану системи масового обслуговування  $P_0$  і  $P_n$ , легко визначити всі основні параметри, що характеризують цю систему:

- вірогідність того, що всі обслуговуючі пристрої вільні;
- вірогідність того, що всі обслуговуючі пристрої зайняті;
- середній час очікування початку обслуговування вимоги при  $S$  обслуговуючих пристроях;
- середня довжина черги (середнє число вимог, що чекають обслуговування);
- середнє число вимог, що знаходяться в системі обслуговування;
- середнє число вільних (зайнятих) обслуговуючих пристроїв, коефіцієнт їх використання в часі і ін.

Проте обмежимося обчисленням всього двох показників, достатніх для визначення ефективності системи масового обслуговування і вибору оптимального числа обслуговуючих пристроїв: середньої довжини черги  $\bar{\gamma}$  і коефіцієнта використання обслуговуючих пристроїв  $K_a$ .

Середня довжина черги  $\bar{\gamma}$  визначається наступним чином. Вимога може знаходитися в черзі, якщо кількість вимог  $n$ , що знаходяться в системі, буде перевищувати кількість пристроїв  $S$ , що обслуговуються. Якщо  $n = S + 1$  в черзі буде одна вимога, при  $n = S + 2$  – дві і т.д. В загальному випадку кількість вимог в черзі складає  $\gamma = n - S$ . Середня довжина черги визначається рівнянням:

$$\bar{\gamma} = \frac{S^S \psi^{S+1}}{S!(1-\psi)^2} P_0 \quad (2.3)$$

Для визначення коефіцієнта використання за часом пристроїв, що обслуговували, розглянемо достатньо великий інтервал часу  $T$ . На протязі цього часу через систему пройде в середньому  $\lambda T$  вимог. За цей же час пристрої, що обслуговують (при повному завантаженні) змогли обслужити  $\mu T$  вимог. Коефіцієнт використання за часом обслуговуючих пристроїв буде дорівнювати:

$$K_a = \frac{\lambda T}{\mu T} = \frac{\lambda}{\mu} = \psi \quad (2.4)$$

Інакше коефіцієнт використання пристроїв дорівнює коефіцієнту використання системи.

### 2.2.3 Оцінка ефективності замкнутої багатоканальної системи масового обслуговування

Розглянемо випадок, коли вузол обслуговування призначений для обслуговування кінцевого і притому постійного числа вимог. Як тільки вимоги обслужені, вони повертаються на початок. Більше того, припустимо, що початок має інтенсивність, пропорційну числу вимог, що містяться в ньому [5,7].

Інтенсивність потоку вимог на  $\lambda'$ . Якщо в деякий момент часу число запитів, що очікують обслуговування і обслужених,  $n$  (система знаходиться в  $E_n$  стані), то кількість обслужених запитів дорівнює  $(m - n)$  і інтенсивність потоку, що входить, визначиться як  $(m - n)\lambda'$ , яка буде змінюватися дискретним чином (стрибкоподібно).

В результаті рішення диференціальних рівнянь станів замкнутої багатоканальної СМО отримані співвідношення, в яких  $\psi' = \lambda' / \mu$ :

$$\frac{P_n}{P_0} = C_m^n S^n \psi'^n, \quad (0 < n \leq S);$$

$$\frac{P_n}{P_0} = C_m^n \frac{n! S^S}{S!} \psi'^n, \quad (S < n \leq m);$$

$$P_0 = \frac{1}{\sum_{n=0}^S C_m^n S^n \psi'^n + \sum_{n=S+1}^m C_m^n \frac{n! S^S}{S!} \psi'^n},$$

де  $\lambda'$  – питома інтенсивність потоку вимог, що доводиться на одно вимогу, потенційно наявну в джерелі;  $\mu$  – загальна інтенсивність обслуговування;  $C_n^m$  – число поєднань із  $m$  по  $n$ .

Приймаючи до уваги, що  $C_n^m = \frac{m!}{n!(m-n)!}$  та замінивши  $\mu$  через  $s\mu'$ ,

де  $\mu'$  – інтенсивність обслуговування одним каналом, тоді отримані співвідношення можна записати:

$$P_n = \frac{m! \psi'^n}{n!(m-n)!} P_0, \quad \text{якщо } 0 \leq n \leq S; \quad (2.6)$$

$$P_n = \frac{m! \psi'^n}{S^{n-S} S!(m-n)!} P_0, \quad \text{якщо } S < n \leq m; \quad (2.7)$$

$$P_0 = \sum_{n=0}^S \frac{m! \phi^n}{n!(m-n)!} + \sum_{n=S+1}^m \frac{m! \phi^n}{S^{n-S} S!(m-n)!}, \quad (2.8)$$

де  $\phi = \chi / \mu'$ .

Слід сказати, що розрахунки за визначенням оптимального числа каналів для джерел на 300-400 вимог громіздкі і важко піддаються ручному рахунку (визначення величин  $m!$  і  $(m-n)!$  та ін.). Крім того, система на 400 вимог може знаходитися в 401 стані, і якщо складно визначити вірогідність

одного стану, то ця трудність посилюється необхідністю підрахунків вірогідності стану. З цього положення може бути два виходи [2, 3]:

- вирішувати задачу по формулах розімкнених СМО, заздалегідь припускаючись похибки;
- розрахунок вести на ЕОМ, склавши програму по отриманих формулах для замкнутої СМО з декількома обслуговуючими пристроями.

#### 2.2.4 Алгоритм згортання

Відповідно до цього методу алгоритм розрахунку нормалізуючої константи

$$(2. G_M(N) = \sum_{n \in (N, M)} \prod_{i=1}^M Z_i(n_i),$$

де множник  $Z_i(n_i)$  має вигляд

$$Z_i(n_i) = \begin{cases} e_i^{n_i} / \prod_{j=1}^{n_i} \mu_i(j), & \text{якщо центр } i \text{ залежить від навантаження,} \\ (e_i / \mu_i)^{n_i}, & \text{якщо центр } i \text{ не залежить від навантаження,} \end{cases}$$

а простір станів мережі

$$S_{(2.10)}(N, M) = \left\{ \mathbf{n} = (n_1, n_2, \dots, n_M) \mid \sum_{i=1}^M n_i = N, n_i \geq 0, i = \overline{1, M} \right\}$$

зводиться до простої ітеративної процедури, суть якої полягає в наступному.

Розглянемо функцію

$$(2.11) g(n, m) = \sum_{n \in S(n, m)} \prod_{i=1}^m Z_i(n_i).$$

Очевидно, що при



$$g(N, M) = G_M(N) \text{ та } g(n, M) = G_M(n) \text{ для } n = \overline{0, N}$$

при  $m > 1$  маємо

$$\begin{aligned} g(n, m) &= \sum_{k=1}^n \left[ \sum_{n \in S(n, m), n_m = k} \prod_{i=1}^M Z_i(n_i) \right] = \sum_{k=0}^n Z_m(k) \sum_{n \in S(n-k, m-1)} \prod_{i=1}^{m-1} Z_i(n_i) = \\ &= \sum_{k=0}^n Z_m(k) g(n-k, m-1). \end{aligned}$$

Таким чином, для мережі, що складається з центрів, інтенсивність обслуговування яких залежить від навантаження

$$(2.1) \quad G_M(N) = \sum_{k=1}^N Z_M(k) G_{M-1}(N-k).$$

Якщо центр  $t$  не залежить від навантаження, той вираз (2.12) можна спростити. В цьому випадку

$$(2.1) \quad Z_m(k) = (e_m / \mu_m)^k = x_m^k = x_m Z_m(k-1).$$

Підставивши (2.14) у вираз (2.12), після спрощень одержуємо

$$(2.15) \quad g(n, m) = g(n, m-1) + x_m g(n-1, m).$$

Формули (2.15) і (2.12) дозволяють здійснювати рекурентне обчислення  $g(n, m)$  при початкових умовах  $g(n, 1) = Z_1(n), n = \overline{1, N}$ , та  $g(0, m) = 1, m = \overline{1, M}$ . У табл. 2.4 і 2.5 показаний схематичний опис алгоритму Бузена. Стівці таблиці заповнюються послідовно зверху вниз. Якщо центр  $t$  не залежить від навантаження, то елементи стівця  $t$  обчислюються по формулі (2.15), інакше – використовується вираз (2.13) і для відшукування  $Z_j(k)$  застосовується рекурентний вираз

$$Z_i(k) = \begin{cases} \frac{e_i}{\mu_i(k)} Z_i(k-1), & \text{якщо вузол } i \text{ залежить від навантаження,} \\ x_i Z_i(k-1), & \text{якщо вузол } i \text{ не залежить від навантаження.} \end{cases}$$

Шукане значення нормалізуючої константи  $G_M(N)$  знаходиться у нижньому правому куті таблиці.

Таблиця 2.4 Метод Бузена. Центри не залежать від навантаження.

				1	m-	m		M
				1		1		1
								$g(1, M)$ $G_M(1)$
						$g(n-1, m)$ ↓		...
-1				$g(n, m)$		$g(n, m)$		$g(n-1, M)$ $G_M(n-1)$
								$g(n, M)$ $G_M(n)$
								...
								$g(N, M)$ $G_M(N)$

Таблиця 2.5 Метод Бузена. Центри залежать від навантаження.

	1		m-1	m	..	M
	1		$g(0, m-1)Z_m(n)$	→	..	1
	$Z_1(1)$		$g(1, m-1)Z_m(n-1)$	→		$g(1, M)$ $G_M(1)$
	...		...	→		...
-1	$Z_1(n-1)$		$g(n-1, m-1)Z_m(n)$	→		$g(n-1, M)$ $G_M(n-1)$
	$Z_1(n)$		$g(n, m-1)Z_m(0)$	→		$g(n, M)$ $G_M(n)$
	...					...
	$Z_1(N)$					$g(N, M)$ $G_M(N)$

Подальше спрощення обчислювальної процедури можливе в окремому випадку мереж, що залежать від навантаження, коли  $m$ -й центр мережі містить  $A_m$  однакових обслуговуючих приладів. При цьому інтенсивність обслуговування  $m$ -го центру:

$$\mu_m(k) = \begin{cases} k\mu_m, & \text{якщо } 0 \leq k \leq A_m, \\ A_m\mu_m, & \text{якщо } k > A_m; \end{cases}$$

$$Z_m(k) = \begin{cases} \frac{x_m^k}{k!}, & \text{якщо } 0 \leq k \leq A_m, \\ \frac{x_m^k}{A_m! A_m^{k-A_m}}, & \text{якщо } k > A_m \end{cases}$$

Останній вираз можна записати у рекуррентному вигляді

$$(2.16) \quad Z_m(k) = x_m d_m(k) Z'_m(k-1), k \geq 1,$$

де  $Z'_m(k) = x_m^k$ ;

$$d_m(k) = \begin{cases} [k!]^{-1}, & \text{якщо } 0 \leq k \leq A_m \\ [A_m! A_m^{k-A_m}]^{-1}, & \text{якщо } k > A_m + 1. \end{cases}$$

Підставляючи (2.16) в (2.13), одержуємо

$$(2.17) \quad g(n, m) = g(n, m-1) + x_m \left[ \sum_{k=0}^{A_m-1} \frac{1}{k+1} Z_m(k) g(n-k-1, m-1) + \sum_{k=A_m}^{n-1} \frac{1}{A_m} Z_m(k) g(n-k-1, m-1) \right]$$

Після перетворень маємо остаточно

$$g(n, m) = g(n, m-1) +$$

$$\left( + \frac{x_m}{A_m} \left[ g(n-1, m) + \sum_{k=0}^{A_m-2} \left( \frac{A_m-k-1}{k+1} \right) Z_m(k) g(n-k-1, m-1) \right] \right)$$

Для мережі, що не залежить від навантаження ( $A_m = 1, m = \overline{1, M}$ ), вираз (2.18) співпадає з (2.15). Якщо  $A_m = 2$ , то

$$Q(n, m) = g(n, m-1) + x_m [g(n-1, m) + g(n-1, m-1)]/2.$$

вираз (2.18) дозволяє значно скоротити обчислення порівнянню з прямим використанням формули (2.12) для випадку, коли центр  $t$  складається з кількох однакових обслуговуючих центрів [6].

Відмітимо, що нормалізуючі константи, які обчислені в результаті роботи описаного вище алгоритму не є однозначними, оскільки залежать від конкретного вибору величин  $e_j$ , які визначаються з системи рівнянь

$$(2.20) \sum_{i=1}^M e_j P_{ij}, j = \overline{1, M}.$$

Розв'язок цієї системи рівнянь, єдиний з точністю до мультиплікативної константи, тому конкретне значення нормалізуючи констант  $G_M(n)$  можна одержати шляхом довільного завдання  $e_{i^*}$ , наприклад  $e_{i^*} = 1$ .

## 3 СИНТЕЗ КОМП'ЮТЕРНОЇ СИСТЕМИ

### 3.1 Структура комплексу технічних засобів комп'ютерної системи

В якості об'єкта управління виступає повітря доменної печі, для якого виконується розробка системи автоматичного управління перепадом тиску у нижній частині доменної печі. Вхідним параметром об'єкта управління є кількість повітря, що задається турбоповітродувкою. Вихідними параметром об'єкта управління є тиск на середньому і нижньому рівнях доменної печі та витрати гарячого дуття.

Виходячи з цього, система управління повинна включати датчики тиску на середньому та нижньому рівнях доменної печі, датчик витрат гарячого дуття, турбоповітродувку, пристрій управління та пульт оператора (рис. 3.1).

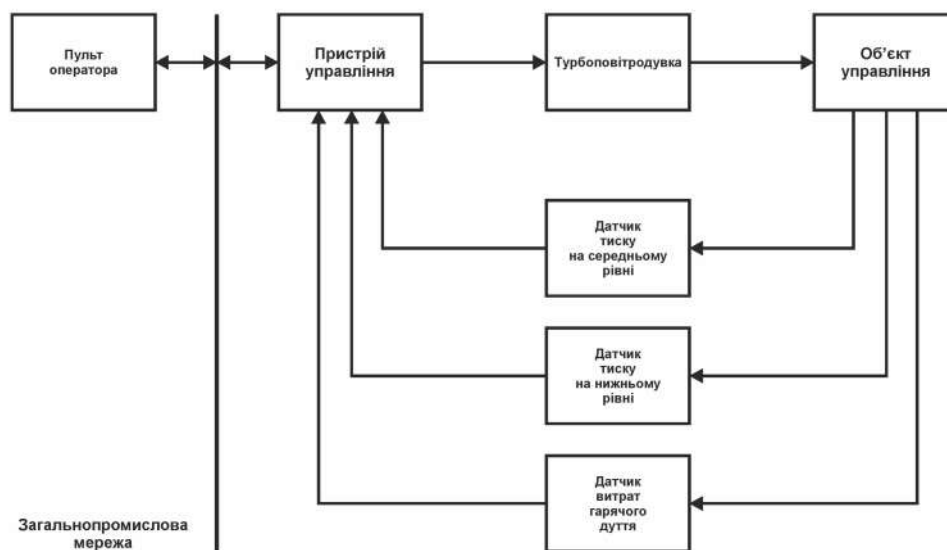


Рисунок 3.1 – Структурна схема системи управління

### 3.2 Характеристики використаних апаратних засобів системи управління

Система повинна управляти повітродувкою на підставі перепаду тиску в нижній частині доменної печі. Тиск на середньому рівні доменної

печі не перевищує 350 кПа. Таким чином для вимірювання тиску може бути використані однакові датчики. У якості датчиків обрано датчик тиску KOBOLD SEN-3297 C535 який має діапазон вимірювання від -1 до 5 бар (100 до 500 кПа), та токовий вихід з діапазоном  $4\div 20$  мА (рис. 3.2).



Рисунок 3.2 – Датчик KOBOLD SEN-3297 C535



Рисунок 3.3 – Датчик KATflow 100

Крім того необхідно вимірювати витрати дуття які можуть змінюватися в діапазоні від 500 до 2500 м<sup>3</sup>/хв. Для цього обрано спеціалізований ультразвуковий накладної витратомір KATflow 100 датчик, який має діапазон вимірювання від 10 до 3000 м<sup>3</sup>/хв, та RS-485 інтерфейс з протоколом Modbus RTU (рис. 3.3).

Управління процесом дуття виконується за рахунок змінення швидкості турбоповітродувки яка базується на трьохфазному асинхронному електродвигуні потужністю 70 кВт. Швидкість турбоповітродувки повинна змінюватися в діапазоні від 35 до 50 Гц. На підставі цього для управління турбоповітродувкою обрано частотний перетворювач Lenze E82EV753K4B201 потужністю 75 кВт та інтерфейс RS-485 з протоколом Modbus RTU (рис. 3.4).



Рисунок 3.4 – Частотний перетворювач Lenze E82EV753K4B201

Відповідно вимогам до системи управління в якості пристрою управління повинен використовуватися програмований логічний контролер компанії VIPA. Цикл роботи контролера повинен бути не більше 100 мс, для забезпечення реакції на змінення температури. Крім того контролер повинен мати не менш 1 Кбайт вільної робочої пам'яті для реалізації програми управління.

Виходячи з того, що до контролеру повинно бути підключено два пристрої з аналоговими виходами та два пристрої з цифровим інтерфейсом RS-485 контролер повинен мати модульну структуру, що забезпечить підключення тільки обраного обладнання та легке розширення подальшого функціоналу.

Так як в структуру системи управління входить підсистема інформаційного забезпечення роботи оператора, яка відповідно до вимог повинна бути реалізована на базі персонального комп'ютера виконуючого

функцію пульта оператора, контролер повинен мати інтерфейс Ethernet [12,14].

Наведеним вимогам відповідає програмований логічний контролер VIPA 214-2BE03. Контролер має час арифметичної операції над речовим числом 40 мкс, об'єм пам'яті програм 144 Кбайта, об'єм робочої пам'яті 96 Кбайт та інтерфейс Ethernet (рис. 3.5).



Рисунок 3.5 – Програмований логічний контролер VIPA 214-2BE03

### 3.2.1 Розробка функціональної схеми автоматизації

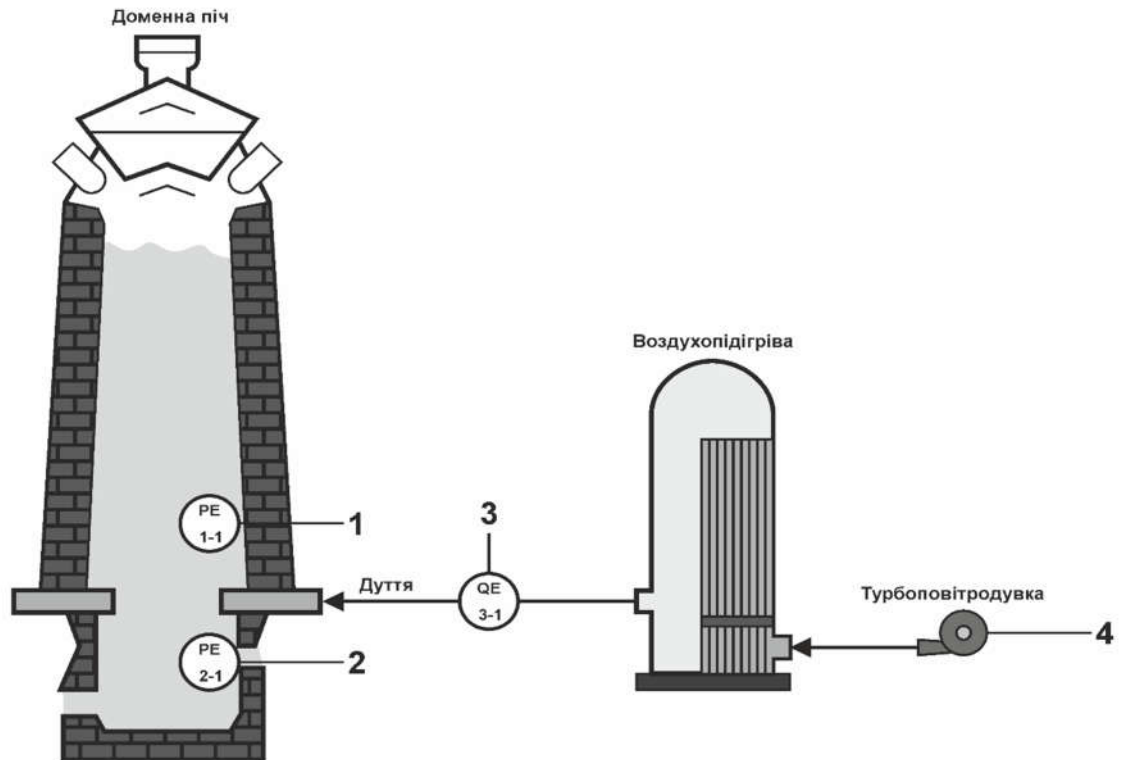
На основі вимог до системи управління та обраного апаратного забезпечення розроблена функціональна схема автоматизації, яка наведена на рис.3.6.

У якості пристрою управління використовується програмований логічний контролер (UY 5 – VIPA 214-2BE03). Програмований логічний контролер підключено до пульта оператора (UYR 6), зв'язок між ними реалізовано за допомогою інтерфейсу Ethernet.

Тиск на середньому та нижньому рівнях вимірюється за допомогою датчиків тиску (PE 1-1,1-2 - KOBOLD SEN-3297 C535) сигнал з яких допомогою перетворювачів (PT 1-2,1-3) надходить до програмованого логічного контролера (UY 5).

Витрати дуття вимірюються відповідним датчиком (QE 3-1 – KATflow 100) який за допомогою перетворювачів (QT 3-2) передається по інтерфейсу RS-485 згідно з протоколом Modbus RTU до програмованого логічного контролера (UY 5).





	1	2	3	4
Прибори за місцем	PT 1-2	PT 2-2	QT 3-2	SC 4-2
Шафа управління	↓ ↓ ↓			UY 5
Підсистема управління формування дуги				UYR 6

Рисунок 3.6 – Функціональна схема автоматизації системи управління

Контролер (UY5) на підставі інформації від датчиків формує необхідне значення швидкості турбоповітряної машини яке надсилається до частотного перетворювача (SC4-2–Lenze E82EV753K4B201), який змінює обертання електродвигуна турбоповітряної машини в діапазоні від 35 до 50Гц.

### 3.2.2 Розробка схеми електричної принципової

На основі функціональної схеми автоматизації та обраного апаратного забезпечення розроблена схема електрична принципова системи управління (рис.3.7).

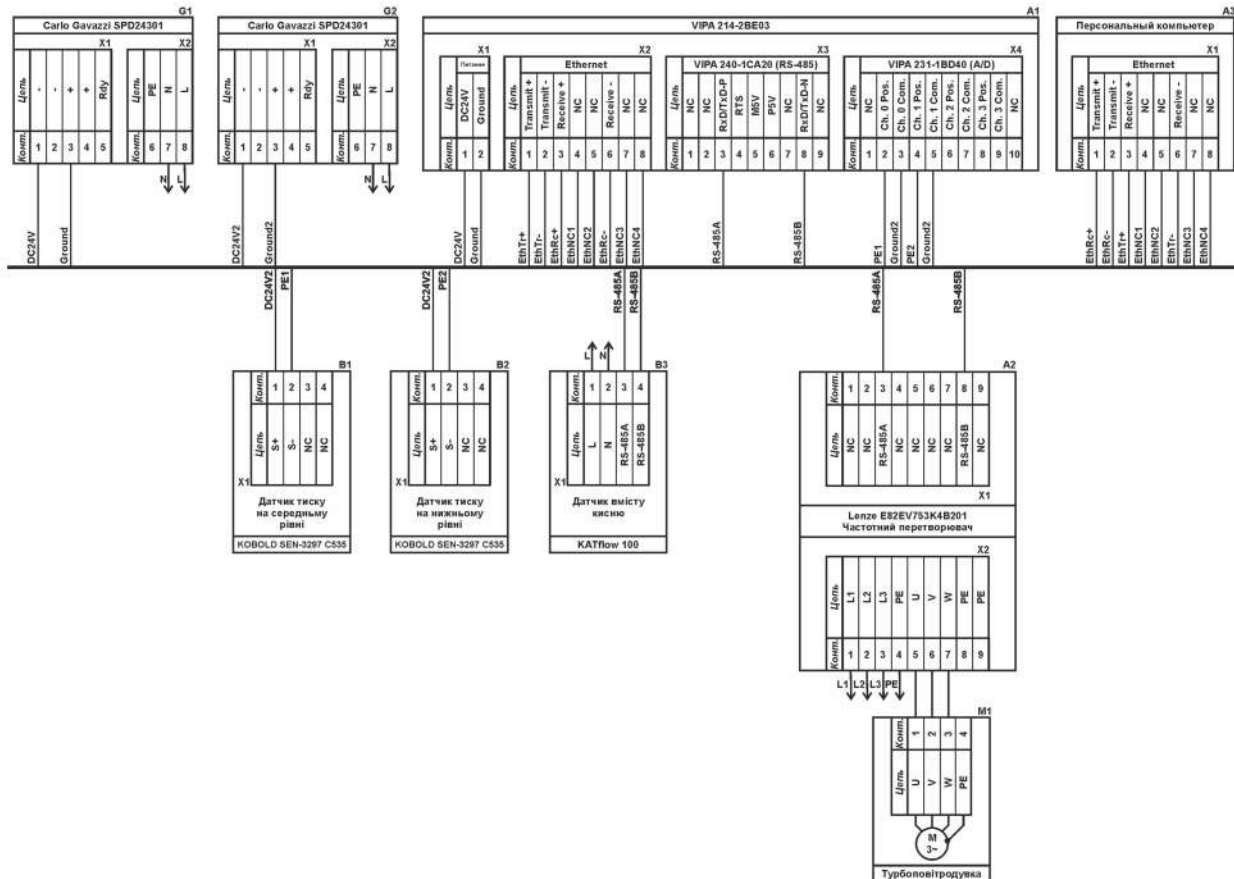


Рисунок 3.7 – Схема електрична принципова системи управління

В підсистемі використовуються два блоки живлення. Блок живлення CarloGavazzi SPD24301 (G1) підключено до програмованого логічного контролеру VIPA 214-2BE03 (A1). Блок живлення CarloGavazzi SPD24301 (G2) підключено до модулю аналогового вводу VIPA 231-1BD40 (A1 – X4) та датчиків тиску KOBOLD SEN-3297 C535 (B1, B2).

Зв'язок між програмованим логічним контролером VIPA 214-2BE03 (A1) та пультом оператора у якості котрого виступає персональний комп'ютер (A3) реалізовано за допомогою інтерфейсу Ethernet (A1 – X2).

Датчик витрат KATflow 100 (B3) підключається до програмованого логічного контролера за допомогою інтерфейсу RS-485. Обмін даними між ними виконується згідно з протоколом Modbus RTU. Датчик живиться від загальнопромислової мережі ~220В.

Управління електродвигуном турбоповітродувки (M1) реалізовано за допомогою частотного перетворювача LenzeE82EV753K4B201 (A2). Програмований логічний контролер (A1) за допомогою інтерфейсу RS-485 згідно з протоколом Modbus RTU надсилає уставку швидкості до частотного перетворювача, який формує трьохфазну напругу з частотою від 35 до 50 Гц, яка надходить на електродвигун.

### **3.3 Розробка програмного забезпечення**

#### **3.3.1 Призначення і область застосування програми**

Програмне забезпечення розроблено під програмований логічний контролер zenon Logic, що входить до складу SCADA системи zenon. Програмне забезпечення об'єкту керування розроблено як ПЗ перевірки функціонування системи керування для драйверу пристрою керування.

Програмне забезпечення є нижнім рівнем програмного забезпечення. Виходячи з вимог програма повинна забезпечувати управління об'єктом отже до її складу повинна входити підсистема управління технологічним обладнанням.

Дана підсистема складається з пристроїв збору інформації (датчики тиску на середньому та нижньому рівнях доменної печі, датчик витрат гарячого дуття), еталонів стану обладнання (перепад тиску в доменній печі), системи автоматичного контролю стану обладнання, програми управління яка повинна реалізувати управління дуттям (турбоповітродувкою).

Невід'ємною частиною розробки системи керування є створення програмного забезпечення людино-машинного інтерфейсу.

Програмне забезпечення системи керування поділено на два окремих програмних блоки: регулятор та об'єкт керування.

Підсистема інформаційного забезпечення роботи оператора представляє собою апаратно програмний комплекс на базі персонального комп'ютера з SCADA системою zenonSupervisor.

### **3.3.2 Обґрунтування технічних характеристик програми**

#### **3.3.2.1 Постановка завдання на розробку програми**

Програмне забезпечення системи автоматизованого управління реалізує функції управління турбоповітродувкою.

Програмне забезпечення людино-машинного інтерфейсу має 5 вікон: інформації про об'єкт, технологічний процес, зміну тиску в колошнику, хронологічний список подій та інформаційний список тривоги. Виходячи з того, що людино-машинний інтерфейс повинен відображати декілька зображень, він повинен складатися з шаблонів на одному з котрих повинні відображатися кнопки переключення зображення, а на іншому самі зображення.

На основному зображенні будуть відображатися інформації про об'єкт управління, технологічний процес, зміну тиску в колошнику, хронологічний список подій та інформаційний список тривоги.

. На наступному етапі опрацювання, на основі наявних даних, проводиться розбиття сформульованої завдання розробки програмного забезпечення на окремі підзадачі, блоки, процедури і функції. Після цього складається алгоритм роботи програми і відбувається побудова детальної схеми алгоритму програми, за якою згодом буде написана керуюча програма.

#### **3.3.2.2 Опис алгоритму функціонування програми**

Виходячи з технічних вимог і прийнятого протоколу обміну, програма нижнього рівня АСУ повинна виконувати наступні функції:

- взаємодія через гальванічно розв'язаної лінії зв'язку за узгодженим протоколом між контролером і пультом оператора;
- отримання сигналів від датчиків технологічного обладнання;
- видачу керуючих сигналів на виконавчі механізми технологічного обладнання;
- стеження за аварійними ситуаціями в роботі технологічних машин, виконання аварійної зупинки приводів;
- зберігання на верхньому рівні і видачу на нижній еталонів спрацьовування обладнання для регулювання подачі повітря.

Реалізація програми можлива тільки з урахуванням таких особливостей застосовуваного промислового контролера. ПЛК VIPA System 200V підтримує мову програмування FBD (функціональні блоки), таким чином, програма управління АСУ розділена на функціональні складові. Головним блоком програми є основною алгоритм опитування, який циклічно виконується контролером. Його головним завданням є виклик необхідних в даний момент функціональних блоків програми. Так в функціональні блоки виділені алгоритми:

- алгоритм управління обладнанням;
- алгоритм відстеження та відпрацювання аварійних зупинок;
- алгоритм регулювання тиску;
- алгоритм регулятора.

Поряд з основними алгоритмами необхідно провести початкову установку значень змінних при включенні контролера. Це можна виконати в блоці, за замовчуванням активному при запуску VIPA 200V - у другому, після чого активізувати інший блок виконання.

### **3.3.3 Опис і обґрунтування вибору методу організації вхідних та вихідних даних**

- Програма повинна використовувати такі вхідні дані:
- сигнали спрацьовування датчиків;

- в ручному режимі запуск процесу кнопкою пуск з панелі оператора.

Програма повинна видати наступні вихідні дані:

- інформацію про стан технологічного обладнання у вигляді одинично-позиційного коду станів датчиків, графічного відображення.

Так як програмований логічний контролер компанії VIPA перетворює аналогові сигнали у цифровий код з діапазоном від 0 до 27648, то отримане значення від SCADA системи zenon було нормалізовано шляхом переводу з діапазону від 0 до 27648 до діапазону значення продуктивності насосу від 0 до 100%. Крім того було перетворено вихід об'єкту керування з діапазону фізичного значення від -50 до 180 °C, до діапазону цифрового значення від 0 до 27648.

### **3.3.3.1 Опис і обґрунтування вибору та складу технічних та програмних засобів**

Для розробки системи управління обраний логічний контролер VIPA System 200V. Контролери VIPA System 200V - програмовані логічні контролери, призначені для використання в якості віддалених інтелектуальних вузлів в розподілених мережах технологічного рівня, а також для автономної роботи в якості пристроїв управління невеликими установками. Середовище розробки програмного забезпечення VIPA WinPLC7. VIPA WinPLC7 - програмний пакет, призначений для:

- конфігурації і параметризації апаратних засобів;
- проектування комунікацій (ProfibusDP slave / master, Ethernet);
- створення і налагодження програм;
- архівування;
- діагностики контролерів VIPA.

Оболонка має дружній інтерфейс для всіх етапів роботи. Це єдина оболонка для всіх типів контролерів VIPA, а також для контролерів Siemens S7\_300 / 400.

WinPLC7 містить різні інструменти для створення проекту:

- конфігурація апаратного забезпечення;

- символний редактор, для визначення змінних через символне позначення;

- редактор програм;

- документування та здійснення симуляції роботи контролера.

Технічні і програмні засоби для розробки програми обумовлені довідковим керівництвом по застосуванню контролерів VIPA200. Мова програмування - FBD (функціональні блоки), середовище розробки - ZenOn (інструментальне середовище - безкоштовна, підтримує всі мови стандарту MEK 61131-3).

### **3.3.4 Опис розробленої програми**

#### **3.3.4.1 Загальні відомості**

Програма розробляється для прошивки в контролер нижнього рівня. Програмне забезпечення для АСУ розроблено на Techno FBD і являє собою схему, що складається з набору функціональних блоків, пов'язаних між собою через входи і виходи.

Програмне забезпечення розміщується у внутрішньому ПЗП контролера.

#### **3.3.4.2 Функціональне призначення**

Виходячи з технічних вимог і прийнятого протоколу обміну, програма нижнього рівня повинна виконувати наступні функції:

- запуск обладнання подачі повітря згідно технологічного процесу;
- реалізацію опитування датчиків, контролюючих технологічні характеристики об'єкта автоматизації;
- пуск і зупинка двигунів гарячого дуття;
- регулювання дуття;
- реалізацію відпрацювання процедури аварійної зупинки;
- контроль тиску;

### **3.3.4.3 Опис логічної структури програми**

Головний організаційний блок програми: Main, функції DState, функції DeadZone і функції IntegralGain.

Блок Main викликає блок DeadZone і відстежує ознака настання чергового події в системі (сигнали стоп / пуск, необхідність регулювання, зміна значень технологічних величин).

Блок DState відпрацьовує технологічний процес роботи обладнання.

Функція DeadZone викликається блоком DState для перекладу даних вимірювання різниці тиску в дискретні ознаки відповідності технологічним еталонним значень параметрів. Функція IntegralGain викликається блоком DState для перекладу даних вимірювання тиску в повітропроводі в дискретні.

Натискання на пульті оператора кнопки «Пуск» або під час вступу команди запуску відбувається запуск роботи устаткування.

Заданий рівень тиску забезпечується регулюванням продуктивності повітродувки.

### **3.3.4.4 Використані технічні засоби**

Модуль контролера VIPA200V - 1 шт.

Кабель програмування - 1 шт.

Модуль перетворювача інтерфейсу SM702 - 1 шт .

Модуль дискретного вводу 221-1BF00 - 2 шт .

Модуль дискретного виводу 222-1HF00 - 1 шт .

Модуль аналогового введення 231-1BD60 - 2 шт .

Система програмування STEP7 v.3.0.6. і старша.

### **3.3.4.5 Виклик і завантаження програми**

Виклик програми на виконання відбувається по включенню живлення контролера

Програма спочатку завантажується в енергонезалежну пам'ять VIPA200V з персонального комп'ютера за допомогою системи STEP7 за допомогою кабелю програмування, де і знаходиться весь термін експлуатації



системи. Виклик підпрограми відбувається при включенні живлення контролера після проходження підпрограмою початкової ініціалізації.

#### **3.3.4.6 Вхідні і вихідні дані**

*Вхідні змінні.*

Тиск на середньому рівні – PE 1-1

Тиск на нижньому рівні PE 1-.

Витрати дуття QE 3-1.

*Вихідна змінна*

Витрати дуття QE 3-1

M1 Пуск / зупинка двигуна

#### **3.3.5 Очікувані техніко-економічні показники**

Технічне рішення дозволяє виконувати більш точно регулювання тиску в колошнику. В результаті підвищується якість продукції, зменшуються енерговитрати що веде до виникнення економічного ефекту.

### **3.4 Людино-машинний інтерфейс**

На технологічному зображенні наведена доменна піч та вказані основні складові елементи.

За для контролю зміни тиску в колошнику розроблено відповідне зображення яке наведено на рис. 3.8. На зображенні за допомогою тренду відображається зміна тиску в колошнику доменної печі

За для відображення списку подій використано зображення типу “Chronologic event list”. Відповідно до вимог в хронологічному списку подій записується інформація про зміни тиску в колошнику доменної печі, закриття та відкриття регульованого клапана. Розроблене зображення наведено на рис. 3.9.

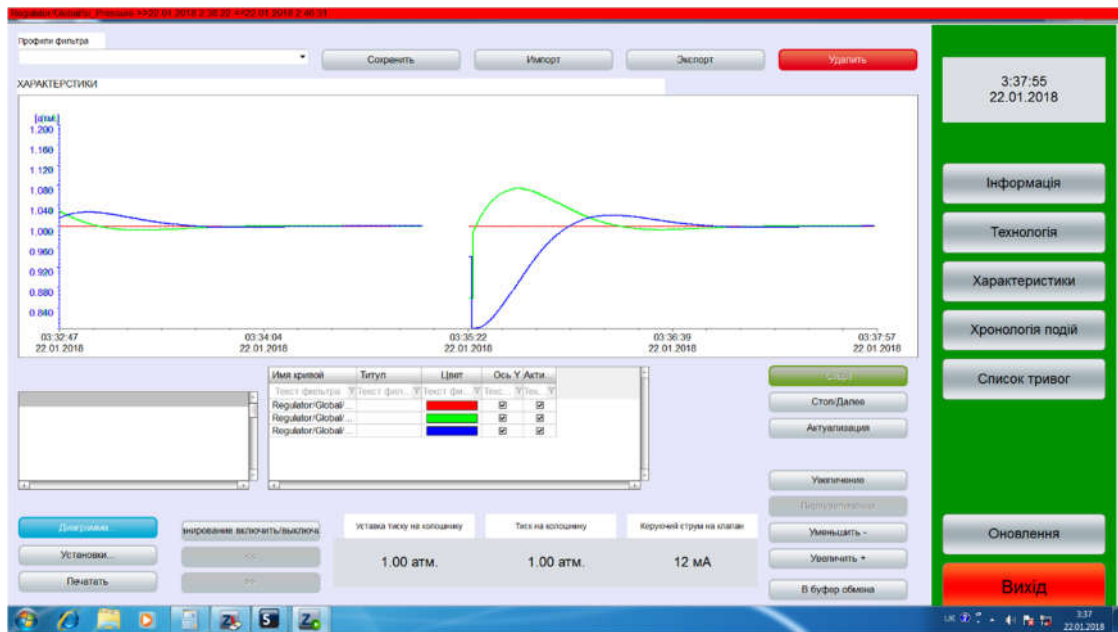


Рисунок 3.8 – Зображення тренду зміни тиску в колошнику доменої печі

За для відображення списку тривог використано зображення типа “Alarm message list”. Відповідно до вимог в інформаційному списку тривог записується інформація про досягнення кінцевих точок процесу. Розроблене зображення інформаційного списку тривог наведено на рис. 3.10.

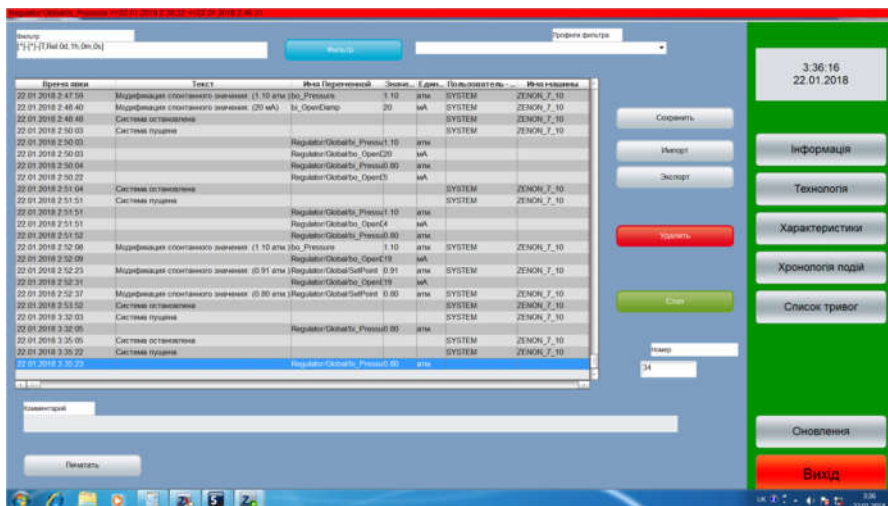


Рисунок 3.9 – Зображення хронологічного списку подій

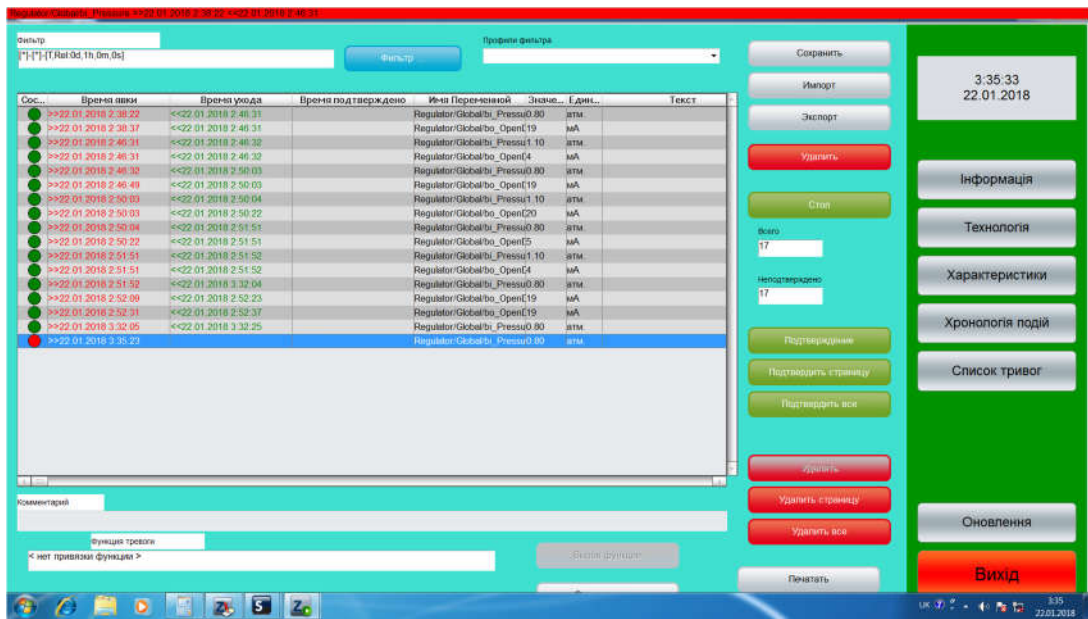


Рисунок 3.10 – Зображення інформаційного списку тривоги

Для зв'язку зображень з даними програмованого логічного контролера який реалізує об'єкт управління та програмного програмованого логічного контролера який реалізує регулятор використані змінні, список котрих наведено на рис. 3.11.

## 4 ЕКСПЕРИМЕНТАЛЬНИЙ РОЗДІЛ

### 4.1 Розробка математичної моделі мережі як замкнутої системи масового обслуговування

Досить важливий випадок являє собою мережу, в якій циркулює постійне число заявок, які не покидають мережу і не можуть увійти в неї з зовні. Це відповідає схемі Джексона.

Знайдемо тепер характеристики вузлів мережі, якими є завантаження, інтенсивність потоку заявок, середнє число заявок у вузлах і середній час перебування заявок в вузлі [5,6] (Рис. 4.1).

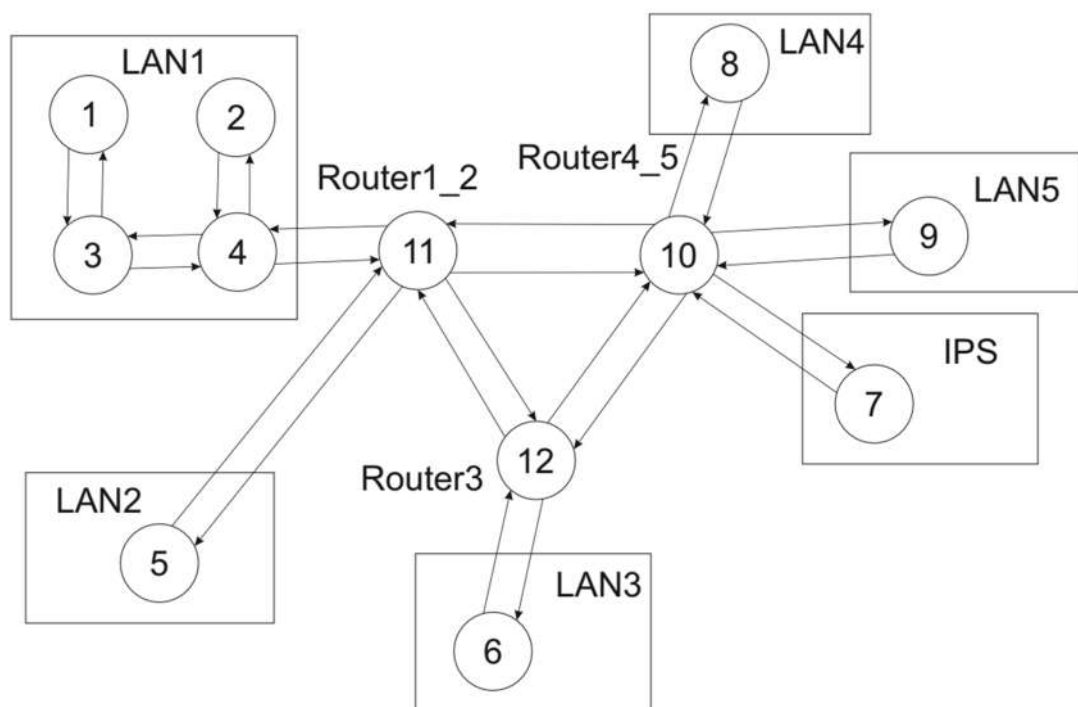


Рисунок 4.1 – Структура математичної моделі комп'ютерної мережі

Структура моделі комп'ютерної мережі відповідає структурі мережі рівня підприємства, яка була розроблена у бакалаврській роботі.

Зв'язки між елементами структури – це ймовірність передачі пакета від одного до другого вузла. Кожен вузол – це система масового обслуговування.

Ймовірність того що вузол зв'язується сам з собою дорівнює нулю.

Ймовірність того що вузли, які відповідають за маршрутизацію зв'язуються між собою приблизно рівна, ймовірність того, що найбільш інтенсивно буде йти зв'язок із серверами та інтернетом найбільш висока.

Результатом є маршрутна матриця розмірністю 12 x 12.

$$Pr := \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.4 & 0 & 0 & 0.6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.25 & 0.25 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.15 & 0.15 & 0.2 & 0 & 0.25 & 0.25 \\ 0 & 0 & 0 & 0.2 & 0.2 & 0 & 0 & 0 & 0 & 0.3 & 0 & 0.3 \\ 0 & 0 & 0 & 0 & 0 & 0.4 & 0 & 0 & 0 & 0.3 & 0.3 & 0 \end{pmatrix}$$

Матриця стовпчик

$$\tau = \begin{array}{|c|c|} \hline & 0 \\ \hline 0 & 3 \\ \hline 1 & 3 \\ \hline 2 & 3 \\ \hline 3 & 3 \\ \hline 4 & 3 \\ \hline 5 & 3 \\ \hline 6 & 3 \\ \hline 7 & 3 \\ \hline 8 & 3 \\ \hline 9 & 3 \\ \hline 10 & 3 \\ \hline 11 & 3 \\ \hline 12 & \\ \hline \end{array}$$

Показує час обробки одного повідомлення в відповідному вузлі. Для початкових умов моделювання ми задаємо однакові умови для усіх вузлів моделі.

## 4.2 Розрахунок параметрів мережі по її моделі

Далі методом Гауса розраховується матриця стовпчик з передаточними коефіцієнтами

$$e_m := \begin{pmatrix} 1 \\ 1.5 \\ 2.5 \\ 6 \\ 3 \\ 6 \\ 2.7 \\ 2.7 \\ 3.6 \\ 18 \\ 15 \\ 15 \end{pmatrix}$$

Задаємо матрицю  $m$ , коефіцієнти якої означають кількість конвеєрів обробки пакетів в кожному із вузлів системи масового обслуговування.

	0
0	1
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1
11	1

Для розрахунків приймаємо, що в кожному пристрої знаходиться лише один конвеєр обробки пакетів.

Матриця В це матриця яка визначає з якою ймовірністю у відповідному вузлі (строчки) буде знаходитися у чеканні обробки пакетів (номер стовпчика).

	0	1	2	3	4	5
0	0.965	0.034	$1.018 \cdot 10^{-3}$	$2.35 \cdot 10^{-5}$	0	
1	0.947	0.05	$2.263 \cdot 10^{-3}$	$7.778 \cdot 10^{-5}$	$1.545 \cdot 10^{-6}$	
2	0.912	0.081	$6.139 \cdot 10^{-3}$	$3.553 \cdot 10^{-4}$	$1.192 \cdot 10^{-5}$	
3	0.79	0.173	0.032	$4.681 \cdot 10^{-3}$	$3.956 \cdot 10^{-4}$	
4	0.895	0.096	$8.735 \cdot 10^{-3}$	$6.099 \cdot 10^{-4}$	$2.472 \cdot 10^{-5}$	
5	0.79	0.173	0.032	$4.681 \cdot 10^{-3}$	$3.956 \cdot 10^{-4}$	
6	0.905	0.087	$7.126 \cdot 10^{-3}$	$4.464 \cdot 10^{-4}$	$1.622 \cdot 10^{-5}$	
7	0.905	0.087	$7.126 \cdot 10^{-3}$	$4.464 \cdot 10^{-4}$	$1.622 \cdot 10^{-5}$	
8	0.874	0.113	0.012	$1.045 \cdot 10^{-3}$	$5.127 \cdot 10^{-5}$	
9	0.369	0.294	0.2	0.105	0.032	
10	0.474	0.291	0.155	0.064	0.015	
11	0.474	0.291	0.155	0.064	0.015	
12						
13						
14						
15						
16						

Відповідно до алгоритму Бузена розраховуються середні значення по кожному із вузлів мережі.

Інтенсивність вхідного потоку пакетів у кожному вузлі

	0
0	$6.477 \cdot 10^{-3}$
1	$9.715 \cdot 10^{-3}$
2	0.016
3	0.039
4	0.019
5	0.039
6	0.017
7	0.017
8	0.023
9	0.117
10	0.097
11	0.097
12	
13	
14	
15	
16	
17	

Середнє число пакетів що чекають на обробку в кожному вузлі

	0
0	0.036
1	0.055
2	0.094
3	0.253
4	0.115
5	0.253
6	0.103
7	0.103
8	0.141
9	1.137
10	0.855
11	0.855
12	
13	
14	
15	
16	

Середній час обробки пакета в вузлі

	0
0	5.575
1	5.66
2	5.836
3	6.516
4	5.927
5	6.516
6	5.872
7	5.872
8	6.039
9	9.755
10	8.797
11	8.797
12	
13	
14	
15	
16	



#### 4.2.1 Параметри роботи мережі без впливу шкідливого програмного забезпечення

Робота комп'ютерної мережі в «нормальному» режимі характеризується наступними параметрами.

Кількість пакетів, які циркулюють у мережі дорівнює 5. Час обробки пакетів у всіх вузлах мережі однаковий і складає 3 часових одиниці. Кількість конвеєрів обробки пакетів у кожному вузлі мережі дорівнює 1.

За таких вихідних даних отримано графіки, що показують усереднені характеристики кожного з вузлів.

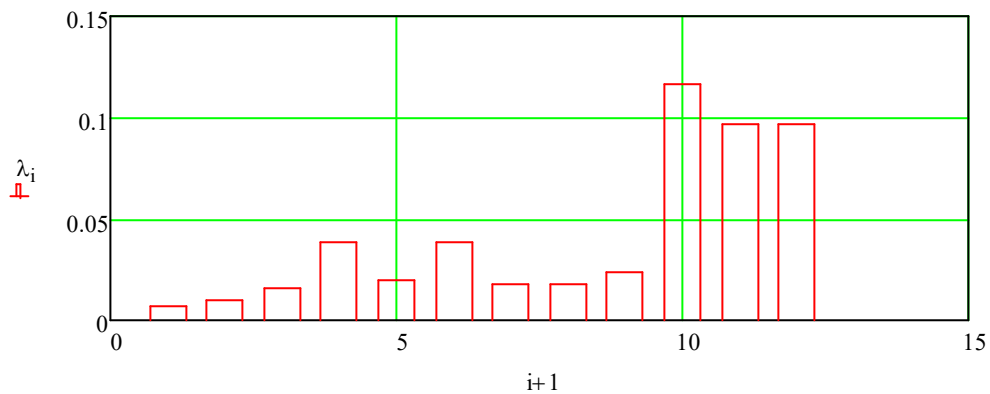


Рисунок 4.2 – Інтенсивність потоку, що входить у вузол

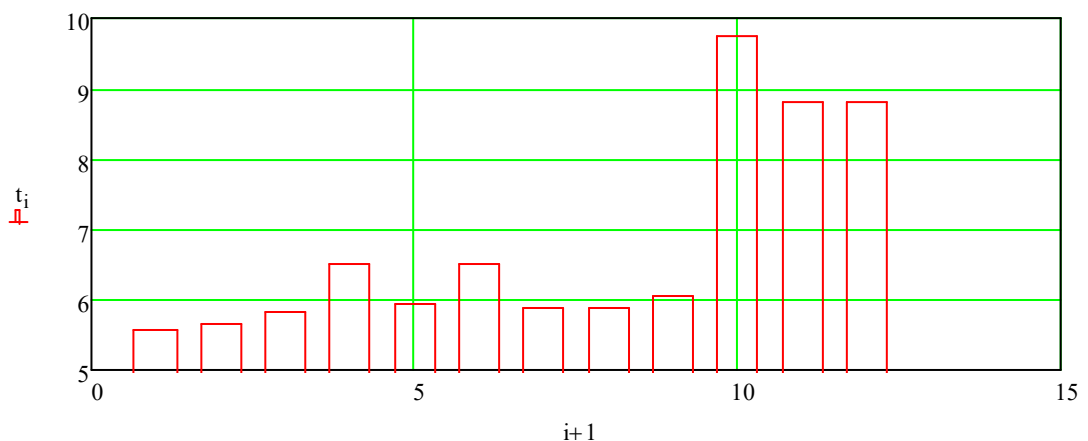
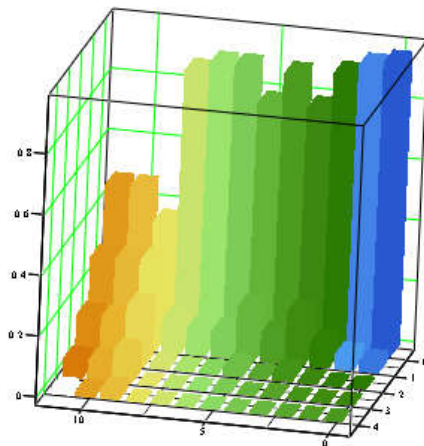


Рисунок 4.3 – Середній час перебування пакета у вузлі

У всіх вузлах мережі усереднені параметри показують, що повідомлення обробляються швидко і без черги. Виключенням є вузли 10, 11, 12.

Рисунок 4.5 показує з якою ймовірністю у вузлах мережі буде черга.



В

Рисунок 4.5 – Ймовірність черги у вузлах мережі

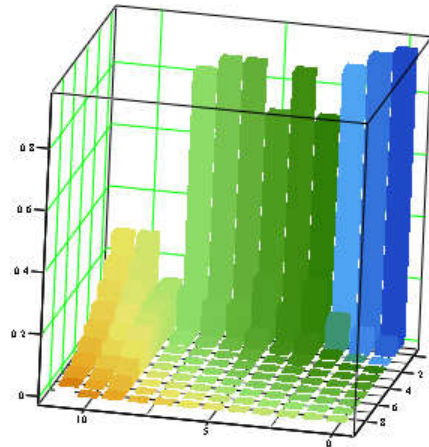
При заданих параметрах вузли 10, 11, 12 мережі є найбільш проблемним.

#### 4.2.2 Параметри роботи мережі під впливом вірусних програм

Вірусне програмне забезпечення вимагають певного обсягу ресурсів, на якому вони виконуються, а також можуть генерувати додатковий трафік в мережі. В залежності від виду вірусів, створювана ними навантаження на ЕОМ і мережу може сильно відрізнятись. Так, наприклад, класичні віруси в загальному випадку не створюють навантаження на мережу зовсім. Деякі ж

мережеві віруси можуть здійснювати таку кількість мережевих запитів, що обчислювальна мережа може і зовсім перестати функціонувати [5].

Для моделювання подібної ситуації задаємо кількість запитів у мережі збільшуємо вдвічі,  $N=10$ .



В

Рисунок 4.6 – Ймовірність черги у вузлах якщо в мережі циркулює 10 пакетів

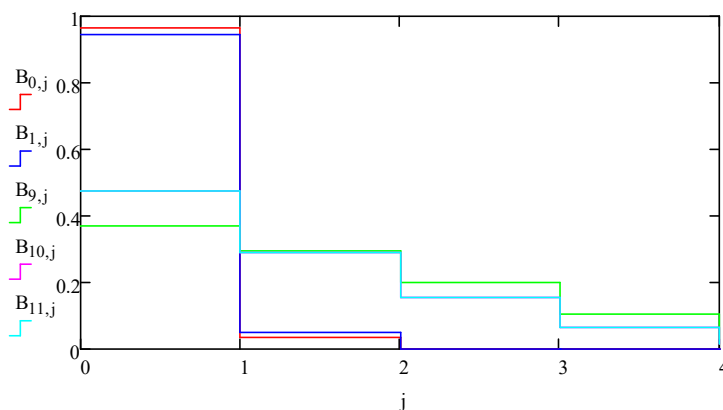


Рисунок 4.7 – Вірогідність черги у вузлах якщо в мережі циркулює 5 пакетів

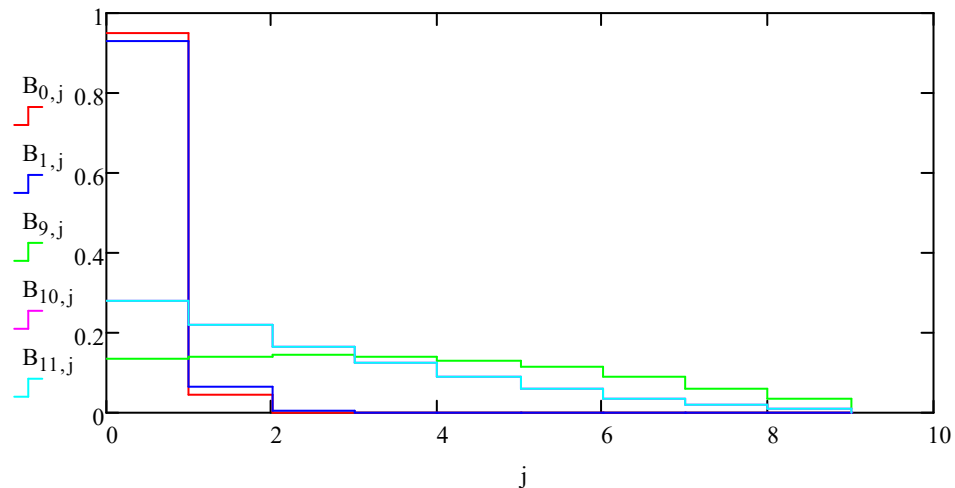


Рисунок 4.8 – Вірогідність черги у вузлах якщо в мережі циркулює 10 пакетів

Результати моделювання показують, що збільшення вдвічі кількості пакетів, що циркулюють в мережі приводить до того, що у вузлах 10, 11, 12 зростає вірогідність того що в черзі уже будуть знаходитися від 2 до 4 пакетів.

### 4.3 Робота мережі із скоригованими характеристиками проблемних вузлів

Корекція характеристик вузлів №5 і №6 проводиться за рахунок підвищення швидкості обробки пакетів.

$$\tau_i :=$$

3
3
3
3
3
3
3
3
3
1
1
1

Відповідно до змін розраховані як усереднені характеристики.

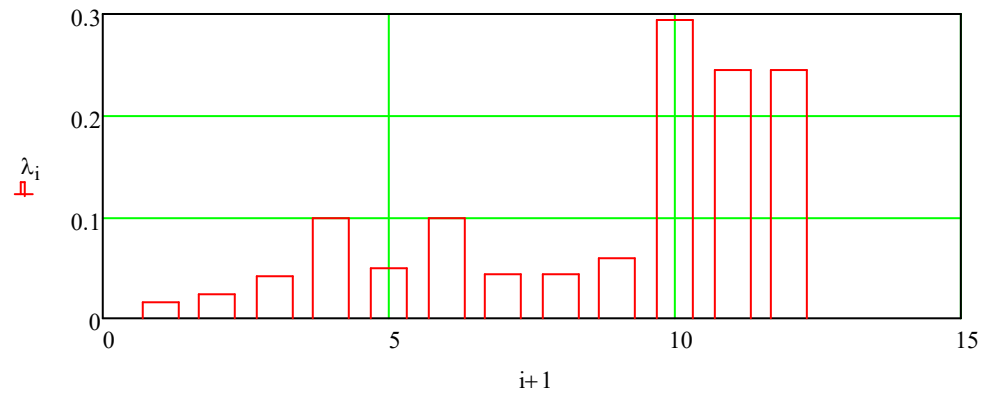


Рисунок 4.9 – Інтенсивність потоку, що входить у вузол

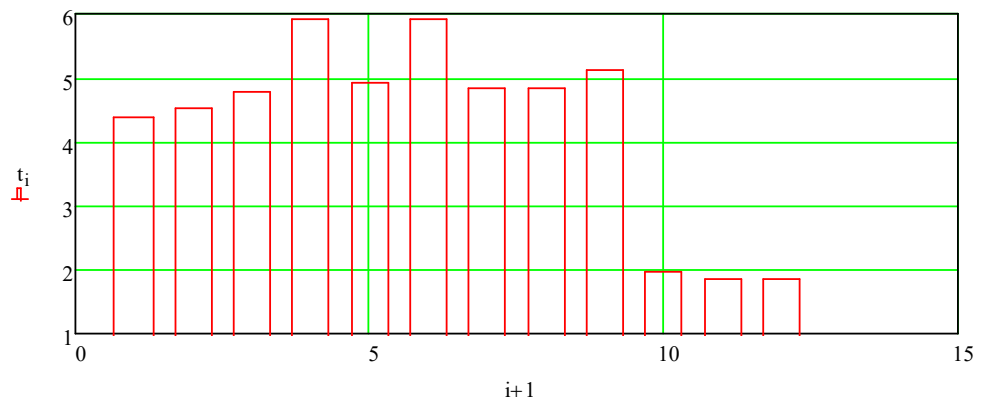


Рисунок 4.10 – Середній час перебування пакета у вузлі

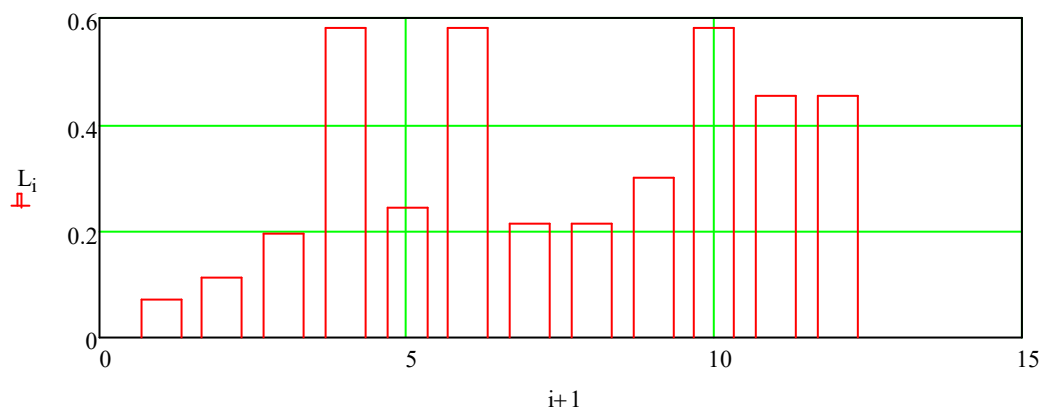
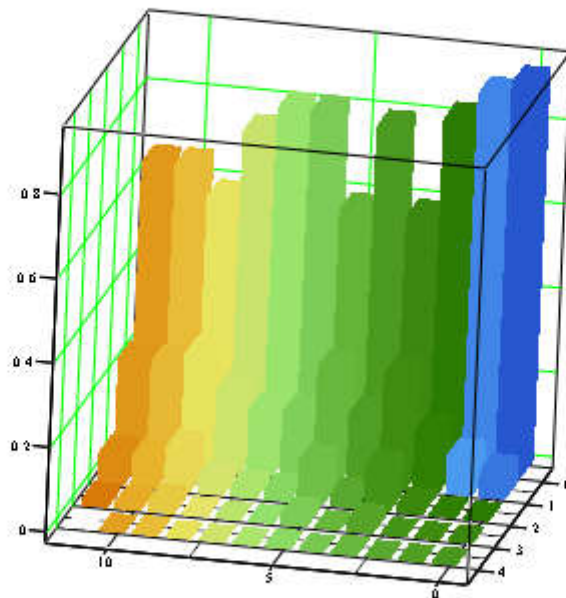


Рисунок 4.11 – Середня кількість пакетів які знаходяться у вузлі

Ймовірність того, що в вузлах мережі може виникнути черга значно знизилася (Рис. 4.12)

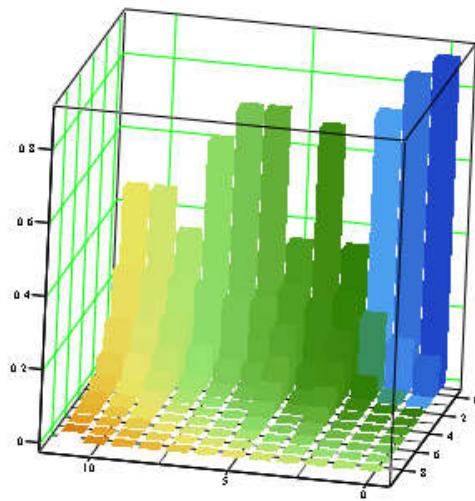


В

Рисунок 4.12 – Ймовірність черги у вузлах якщо в мережі циркулює 5 пакетів

Якщо в мережі із покращеними характеристиками буде циркулювати 10 пакетів тоді вірогідність того, що в вузлах не буде черги зменшується для вузлів 4, 6, 10, 11, 12 (Рис.13).

Підвищення швидкості обробки пакетів у вузлах, які показали найменшу стійкість до перевантаження дозволили певним чином покращити характеристики мережі.



В

Рисунок 4.13 – Вірогідність черги у вузлах якщо в мережі циркулює 10 пакетів

Для замкнутих МеМО циркулює постійне число заявок, то в вузлах мережі не можуть утворюватися черги нескінченної довжини, отже, в ЗМеМО завжди існує сталий режим. Навіть якщо в мережі є дуже «Повільний» вузол, в якому в порівнянні з іншими вузлами занадто довго обробляються заявки, то це може привести тільки до того, що всі заявки будуть постійно накопичуватися в черзі перед цим вузлом, проте їх кількість буде завжди звичайно і в межі дорівнює числу циркулюючих в мережі заявок. Завантаження такого «повільного» вузла буде близьке до одиниці, оскільки постійна наявність черги перед цим вузлом обумовлює безперервну роботу приладів вузла, такий вузол зазвичай являє собою так зване «вузьке місце» мережі. Стационарний ординарний потік без післядії називається

найпростішим (пуассонівським). Інтервали часу між заявками в найпростішому потоці розподілені за експоненціальним законом. Аналітичні дослідження моделей масового обслуговування зазвичай проводяться в припущенні про найпростішому потоці заявок, що обумовлено рядом властивих йому особливостей (підсумовування потоків), що дозволяють у багатьох випадках отримати порівняно прості аналітичні залежності характеристик від параметрів.

Тривалість обслуговування заявок в приладі в найпростішому випадку може бути задана середнім значенням або величиною зворотної - інтенсивністю обслуговування, що характеризує середнє число заявок, яке може бути обслужено приладом за одиницю часу.



## ВИСНОВКИ

Поява сучасних технологій обробки інформації та значна модернізація обчислювальних засобів викликає попит на централізований збір інформації про технологічні об'єкти підприємства поширенняї (надання доступу) в сучасних комп'ютерних мережах підприємства, а також на оновлення автоматизованих систем керування, які в подальшому можуть обслуговуватися сучасними спеціалістами.

Основні висновки і результати роботи полягають у наступному:

1. Відповідно до функціональних особливостей комп'ютерної системи керування технологічним процесом виплавки металу у доменній печі розроблений людино-машинний інтерфейс і його програмне забезпечення. Реалізована можливість передавання інформації про стан процесів через комп'ютерну мережу рівня підприємства.
2. Показано, що навантаження на комп'ютерну мережу може викликатися дією шкідливого програмного забезпечення. Що може негативно вплинути на роботу систем управління виробництвом металу і призвести до значних втрат.
3. Вирішення завдань досліджень можливе при застосуванні методів теорії масового обслуговування. Класичні підходи до рішення завдань доволі трудомісткі, тому використаний алгоритм Бузена.
4. Розроблена математична модель компютерної мережі як замкнутої системи масового обслуговування. Обґрунтовано обрані параметри перехідної матриці мережі.
5. Розроблена модель дозволяє визначити усереднені значення інтенсивності вхідного потоку, часу перебування пакета у вузлах, середню кількість пакетів у вузлах. Що відноситься до класу безрозмірних характеристик.
6. Аналіз ймовірностей черги у вузлах мережі показав що найбільша ймовірність черги виникає у вузлах 10, 11, 12.

7. При збільшенні кількості пакетів, що циркулюють у мережі вдвічі у вузлах 10, 11, 12 висока ймовірність виникнення черги з 3 – 4 пакетів. Ця ситуація можлива коли в мережі функціонує шкідливе програмне забезпечення.
8. Для підвищення стійкості мережі до перевантаження необхідно замінити роутери, які є вузлами 10, 11, 12 моделі на більш потужні. Час обробки пакетів у цих пристроїв повинен бути в 3 рази менший ніж у вузлів 1-9.
9. Зважаючи на те, що при лінійному рості навантаження на вузли мережі, ймовірності станів вузлів мережі розподіляються нелінійно виникає додаткове навантаження на вузли 4,6. На основі проведених досліджень можемо рекомендувати удосконалення структури комп'ютерної мережі рівня підприємства.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Автоматика та комп'ютерно-інтегровані технології у промисловості, телекомунікаціях, енергетиці та транспорті: Матеріали Всеукраїнської науково-практичної інтернет-конференції. – Кропивницький: ЦНТУ, 2017. – 228 с.
2. Создание интеллектуальных систем поддержки принятия решений по управлению доменной плавкой: новые подходы [Текст] : монография / [И. Г. Муравьева, Д. Н. Тогобицкая, Ю. С. Семенов и др.] ; НАН Украины, Институт черной металлургии им. З. И. Некрасова. - Киев :Наукова думка, 2019. - 272 с. : граф., рис., табл. - (Проект "Наукова книга"). - Библиогр.: с. 254-266
3. 1. Металургійна теплотехніка: Збірник наукових праць / Національної металургійної академії України. – Дніпропетровськ: ПП Грек О.С. – 2007. – 324 с.
4. Смоляк В.А. и др. Автоматизация производственных процессов металлургической промышленности. – К.: Вища школа, 1976. -416 с.
5. Груздева Л.М., Монахов Ю.М., Монахов М.Ю.Оценка сетевых характеристик компьютерных сетей в условиях информационного вредоносного воздействия: учебное пособие. / Владим. гос. ун-т. – Владимир: Изд-во Владим. гос. ун-та, 2010. – с.
6. Литвинов А. Л. Теорія систем масового обслуговування : навч. посібник / А. Л. Литвинов ; Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова. – Харків : ХНУМГ ім. О. М. Бекетова, 2018. – 141 с..
7. Globa L.S. Approaches and technologies of creating data-processing resources in the telecommunication environment – Електроникаиcвязь. Проблемы электроники ч.2, Киев, 2005
8. Глоба Л.С. Математичні основи побудови інформаційно телекомунікаційних систем. - К.: Політехніка, 2003. -276с

9. ГОСТ 2.702-75. ЕСКД. Правила выполнения электрических схем. – М.: Госстандарт, 1995. – 115 с.
10. VLSM Calculator – калькулятор подсетей с маской переменной длины [Электронный ресурс]. – Режим доступа: URL: <http://www.vlsm-calc.net/>. – Загол. з экрана.
11. Воробьёва Н.И., Корнейчук В.И., Савчук Е.В. Надёжность компьютерных систем. – К.: «Корнійчук», 2002. – 144 с.
12. Мережеве обладнання [Электронный ресурс] – Режим доступа : URL : [https://elmir.ua/routers/router\\_zyxel\\_sbg5500-a.html](https://elmir.ua/routers/router_zyxel_sbg5500-a.html). – Загол. з екрану.
13. Классификация угроз информационной безопасности (Електрон. ресурс)/Способ доступа: URL: [http://www.cnews.ru/reviews/free/oldcom/security/elvis\\_class.shtml](http://www.cnews.ru/reviews/free/oldcom/security/elvis_class.shtml) – Загол. з екрана..
14. Новиков Ю.В., Карпенко Д.Г. Аппаратура локальных сетей: функции, выбор, разработка / Под общей редакцией Ю.В. Новикова. – М.: Эком, 1998. – 288 с.
15. Кулаков Ю.А., Луцкий Г.М. Локальные сети. – К.: Юниор, 1998. – 336 с.
16. Кулаков Ю.А., Омелянский С.В. Компьютерные сети. Выбор, установка, использование и администрирование. – К: Юниор, 1999. – 544 с.
17. Спортак М, Паппас Ф., Рензинг Э. Компьютерные сети. Книга 1. Энциклопедия пользователя: Пер. с англ. – М.: Диасофт, 1998. – 432 с.
18. Баня Е.Н. Компьютерные сети. – К.: Світ, 1999. – 112 с.
19. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы. – СПб.: Питер, 2001. – 172 с.
20. Джеймс Челлис Основы построения сетей: Учебное пособие для специалистов MCSE 1.0. – СПб.: Питер, 1997. – 326 с.
21. Технологии разработки программного обеспечения: / Учебник С. Орлов. – СПб.: Питер, 2002. – 464 с.

22. Microsoft Corporation. Принципы проектирования и разработки программного обеспечения. Учебный курс MSCD/ Пер. с англ. – М.: Издательско-торговый дом «Русская редакция», 2002. – 736 с.

23. Розробка програмного забезпечення комп'ютерних систем. Програмування [Текст]: навч. посібник / Л.І. Цвіркун, А.А. Євстігнєєва, Я.В. Панферова. – 2-ге вид., випр. – Д.: Національний гірничий університет, 2011. – 222 с.

24. Цвіркун Л.І. Глобальні комп'ютерні мережі. Програмування мовою PHP: навч. посібник / Л.І. Цвіркун, Р.В. Липовий, під заг. ред. Л.І. Цвіркуна. – Д.: Національний гірничий університет, 2013. – 239 с.

## **Додаток А**

**Тексти програми комп'ютерної системи керування тиском  
доменної печі**

**Міністерство освіти і науки України**  
**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ**  
**“ДНІПРОВСЬКА ПОЛІТЕХНІКА”**

**ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ**  
**ПРОГРАМА КОМП'ЮТЕРНОЇ СИСТЕМИ КЕРУВАННЯ**  
**ТИСКОМ ДОМЕННОЇ ПЕЧІ**

Текст програми

804.02070743.20005-01 12 01

Листів 17

2020

## АНОТАЦІЯ

Даний документ містить ПЗ реалізації програми керування тиском в доменній печі.

Програма реалізує людино-машинний інтерфейс.



## ЗМІСТ

Стор.		
	Програмні модулі системи керування	
.		
	Програмне забезпечення системи керування	0
.		
	Програмне забезпечення регулятора	1
.		
	Програмне забезпечення людино-машинного інтерфейсу	4
.		

# 1 Програмні модулі системи керування

## Модуль об'єкта керування

```
(*
*
* File: Object.exp
*
* IEC 61131-3 Structured Text (ST) code generated for
subsystem "Model_SYS4/Object"
*
* Model name                : Model_SYS4
* Model version              : 1.44
* Model creator              : WORK
* Model last modified by    :
* Model last modified on    : Sat Dec 14 14:46:39
2019
* Model sample time         : 0s
* Subsystem name            : Model_SYS4/Object
* Subsystem sample time    : 0.1s
* Simulink PLC Coder version : 1.6 (R2013b) 08-Aug-
2013
* ST code generated on     : Sat Dec 14 14:47:14
2019
*
* Target IDE selection      : 3S CoDeSys 2.3
* Test Bench included      : No
*
*)
FUNCTION_BLOCK Object
VAR_INPUT
    ssMethodType: SINT;
    bi_Valve: LREAL;
END_VAR
VAR_OUTPUT
    bo_Pressure: LREAL;
END_VAR
VAR
    DiscreteTransferFcn_states: LREAL;
    DiscreteTransferFcn1_states: LREAL;
    DiscreteTransferFcn2_states: LREAL;
    Delay_DSTATE: ARRAY [0..9] OF LREAL;
END_VAR
VAR_TEMP
    k: DINT;
    i: DINT;
    DiscreteTransferFcn: LREAL;
    DiscreteTransferFcn2: LREAL;
END_VAR
CASE ssMethodType OF
    SS_INITIALIZE:
```

```

        (* InitializeConditions for Delay: '<S1>/Delay' *)
        FOR i := 0 TO 9 DO
            Delay_DSTATE[i] := 0.0;
        END_FOR;
        (* End of InitializeConditions for Delay:
'<S1>/Delay' *)

        (* InitializeConditions for DiscreteTransferFcn:
'<S1>/Discrete Transfer Fcn' *)
        DiscreteTransferFcn_states := 0.0;

        (* InitializeConditions for DiscreteTransferFcn:
'<S1>/Discrete Transfer Fcn1' *)
        DiscreteTransferFcn1_states := 0.0;

        (* InitializeConditions for DiscreteTransferFcn:
'<S1>/Discrete Transfer Fcn2' *)
        DiscreteTransferFcn2_states := 0.0;
        SS_STEP:

        (* Output: '<Root>/bo_Pressure' incorporates:
* Delay: '<S1>/Delay' *)
        bo_Pressure := Delay_DSTATE[0];

        (* DiscreteTransferFcn: '<S1>/Discrete Transfer Fcn'
*)
        DiscreteTransferFcn := 1.084E-5 *
DiscreteTransferFcn_states;

        (* DiscreteTransferFcn: '<S1>/Discrete Transfer
Fcn2' *)
        DiscreteTransferFcn2 := 0.01332 *
DiscreteTransferFcn2_states;

        (* Update for Delay: '<S1>/Delay' incorporates:
* DiscreteTransferFcn: '<S1>/Discrete Transfer
Fcn1' *)
        FOR k := 0 TO 8 DO
            Delay_DSTATE[k] := Delay_DSTATE[k + 1];
        END_FOR;
        Delay_DSTATE[9] := 0.004535 *
DiscreteTransferFcn1_states;
        (* End of Update for Delay: '<S1>/Delay' *)

        (* Update for DiscreteTransferFcn: '<S1>/Discrete
Transfer Fcn' incorporates:
* Update for Inport: '<Root>/bi_Valve' *)
        DiscreteTransferFcn_states := bi_Valve - (-0.9957 *
DiscreteTransferFcn_states);

        (* Update for DiscreteTransferFcn: '<S1>/Discrete
Transfer Fcn1' *)

```

```

        DiscreteTransferFcn1_states := DiscreteTransferFcn2
- (-0.9955 * DiscreteTransferFcn1_states);

        (* Update for DiscreteTransferFcn: '<S1>/Discrete
Transfer Fcn2' *)
        DiscreteTransferFcn2_states := DiscreteTransferFcn -
(-0.9867 * DiscreteTransferFcn2_states);
    END_CASE;
END_FUNCTION_BLOCK
VAR_GLOBAL CONSTANT
    SS_INITIALIZE: SINT := 0;
    SS_STEP: SINT := 1;
END_VAR
VAR_GLOBAL
END_VAR

```

## Модуль регулятора

```

(*)
*
* File: Controller.exp
*
* IEC 61131-3 Structured Text (ST) code generated for
subsystem "Model_SYS4/Controller
"
*
* Model name           : Model_SYS4
* Model version        : 1.44
* Model creator        : WORK
* Model last modified by :
* Model last modified on       : Sat Dec 14 14:46:39
2019
* Model sample time    : 0s
* Subsystem name       : Model_SYS4/Controller
* Subsystem sample time : 0.1s
* Simulink PLC Coder version : 1.6 (R2013b) 08-Aug-
2013
* ST code generated on       : Sat Dec 14 14:48:34
2019
*
* Target IDE selection   : 3S CoDeSys 2.3
* Test Bench included    : No
*
*)
FUNCTION_BLOCK Controller
VAR_INPUT
    ssMethodType: SINT;
    bi_Error: LREAL;
END_VAR
VAR_OUTPUT
    bo_Valve: LREAL;
END_VAR

```

```

VAR
    Integrator_DSTATE: LREAL;
    Filter_DSTATE: LREAL;
END_VAR
VAR_TEMP
    rtb_IntegralGain: LREAL;
    rtb_DeadZone: LREAL;
    rtb_FilterCoefficient: LREAL;
    y: LREAL;
    y_0: LREAL;
END_VAR
CASE ssMethodType OF
    SS_INITIALIZE:

        (* InitializeConditions for DiscreteIntegrator:
        '<S2>/Integrator' *)
            Integrator_DSTATE := 0.0;

        (* InitializeConditions for DiscreteIntegrator:
        '<S2>/Filter' *)
            Filter_DSTATE := 0.0;
    SS_STEP:

        (* Gain: '<S1>/Нормування діапазону1' incorporates:
        * Inport: '<Root>/bi_Error' *)
            rtb_IntegralGain := 400.0 * bi_Error;

        (* Gain: '<S2>/Filter Coefficient' incorporates:
        * DiscreteIntegrator: '<S2>/Filter'
        * Gain: '<S2>/Derivative Gain'
        * Sum: '<S2>/SumD' *)
            rtb_FilterCoefficient := ((9.52707581104004 *
rtb_IntegralGain) - Filter_DSTATE) * 0.05;

        (* Sum: '<S2>/Sum' incorporates:
        * DiscreteIntegrator: '<S2>/Integrator'
        * Gain: '<S2>/Proportional Gain' *)
            rtb_DeadZone := ((1.35707510609239 *
rtb_IntegralGain) + Integrator_DSTATE) + rtb_FilterCoefficient;

        (* Gain: '<S2>/Integral Gain' *)
            rtb_IntegralGain := 0.0316047084829628 *
rtb_IntegralGain;

        (* Saturate: '<S2>/Saturation' *)
        IF rtb_DeadZone >= 100.0 THEN
            (* Output: '<Root>/bo_Valve' *)
            bo_Valve := 100.0;
        ELSIF rtb_DeadZone > 0.0 THEN
            (* Output: '<Root>/bo_Valve' *)
            bo_Valve := rtb_DeadZone;
        ELSE
            (* Output: '<Root>/bo_Valve' *)

```

```

        bo_Valve := 0.0;
    END_IF;
    (* End of Saturate: '<S2>/Saturation' *)

    (* DeadZone: '<S3>/DeadZone' *)
    IF rtb_DeadZone > 100.0 THEN
        rtb_DeadZone := rtb_DeadZone - 100.0;
    ELSIF rtb_DeadZone >= 0.0 THEN
        rtb_DeadZone := 0.0;
    END_IF;
    (* End of DeadZone: '<S3>/DeadZone' *)

    (* Signum: '<S3>/SignPreSat' *)
    IF rtb_DeadZone < 0.0 THEN
        y := -1.0;
    ELSIF rtb_DeadZone > 0.0 THEN
        y := 1.0;
    ELSE
        y := rtb_DeadZone;
    END_IF;
    (* Signum: '<S3>/SignPreIntegrator' *)
    IF rtb_IntegralGain < 0.0 THEN
        y_0 := -1.0;
    ELSIF rtb_IntegralGain > 0.0 THEN
        y_0 := 1.0;
    ELSE
        y_0 := rtb_IntegralGain;
    END_IF;
    (* Switch: '<S2>/Switch' incorporates:
    * Constant: '<S2>/Constant'
    * Logic: '<S3>/AND'
    * RelationalOperator: '<S3>/Equal'
    * RelationalOperator: '<S3>/NotEqual'
    * Signum: '<S3>/SignPreIntegrator'
    * Signum: '<S3>/SignPreSat' *)
    IF (0.0 <> rtb_DeadZone) AND (y = y_0) THEN
        rtb_IntegralGain := 0.0;
    END_IF;
    (* End of Switch: '<S2>/Switch' *)

    (* Update for DiscreteIntegrator: '<S2>/Integrator'
*)
        Integrator_DSTATE := (0.1 * rtb_IntegralGain) +
    Integrator_DSTATE;

    (* Update for DiscreteIntegrator: '<S2>/Filter' *)
    Filter_DSTATE := (0.1 * rtb_FilterCoefficient) +
    Filter_DSTATE;
    END_CASE;
    END_FUNCTION_BLOCK
    VAR_GLOBAL CONSTANT
        SS_INITIALIZE: SINT := 0;
        SS_STEP: SINT := 1;

```

```

END_VAR
VAR_GLOBAL
END_VAR

```

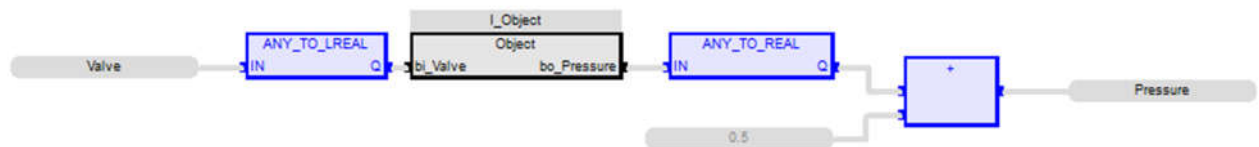
## 2 Програмне забезпечення об'єкта керування

### Список програм

[FBD] Object (\* Основна програма \*)

[ST] Main\_object (\* Об'єкт керування \*)

### Основна програма



### Користувачський функціональний блок

```

(* Output: '<Root>/bo_Pressure' incorporates:
   * Delay: '<S1>/Delay' *)
bo_Pressure := Delay_DSTATE[0];
(* DiscreteTransferFcn: '<S1>/Discrete Transfer Fcn'
*)
DiscreteTransferFcn := LREAL#1.084E-5 *
DiscreteTransferFcn_states;
(* DiscreteTransferFcn: '<S1>/Discrete Transfer
Fcn2' *)
DiscreteTransferFcn2 := LREAL#0.01332 *
DiscreteTransferFcn2_states;
(* Update for Delay: '<S1>/Delay' incorporates:
   * DiscreteTransferFcn: '<S1>/Discrete Transfer
Fcn1' *)
FOR k := 0 TO 8 DO
    Delay_DSTATE[k] := Delay_DSTATE[k + 1];
END_FOR;
Delay_DSTATE[9] := LREAL#0.004535 *
DiscreteTransferFcn1_states;
(* End of Update for Delay: '<S1>/Delay' *)
(* Update for DiscreteTransferFcn: '<S1>/Discrete
Transfer Fcn' incorporates:
   * Update for Inport: '<Root>/bi_Valve' *)
DiscreteTransferFcn_states := bi_Valve - (LREAL#-
0.9957 * DiscreteTransferFcn_states);
(* Update for DiscreteTransferFcn: '<S1>/Discrete
Transfer Fcn1' *)

```

```
        DiscreteTransferFcn1_states := DiscreteTransferFcn2
- (LREAL#-0.9955 * DiscreteTransferFcn1_states);
        (* Update for DiscreteTransferFcn: '<S1>/Discrete
Transfer Fcn2' *)
        DiscreteTransferFcn2_states := DiscreteTransferFcn -
(LREAL#-0.9867 * DiscreteTransferFcn2_states);
```



## Перелік змінних

```
VAR_INPUT
    bi_Valve : LREAL ;
END_VAR
VAR_OUTPUT
    bo_Pressure : LREAL ;
END_VAR
VAR
    DiscreteTransferFcn_states : LREAL ;
    DiscreteTransferFcn1_states : LREAL ;
    DiscreteTransferFcn2_states : LREAL ;
    Delay_DSTATE : ARRAY [0 .. 9] OF LREAL ;
    k : DINT ;
    i : DINT ;
    DiscreteTransferFcn : LREAL ;
    DiscreteTransferFcn2 : LREAL ;
END_VAR
```

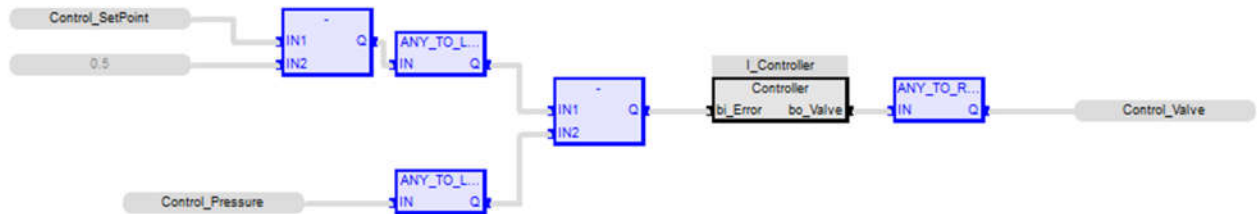
## 3 Програмне забезпечення регулятора

### Список програм

[FBD] Main (\* Основна програма \*)

[ST] Controller (\* Контролер \*)

### Основна програма



### Користувачький функціональний блок

```
(* Gain: '<S1>/íîðíóââííÿ ä³àîàçîííó1' incorporates:
 * Inport: '<Root>/bi_Error' *)
rtb_IntegralGain := LREAL#400.0 * bi_Error;

(* Gain: '<S2>/Filter Coefficient' incorporates:
 * DiscreteIntegrator: '<S2>/Filter'
 * Gain: '<S2>/Derivative Gain'
 * Sum: '<S2>/SumD' *)
rtb_FilterCoefficient := ((LREAL#9.52707581104004 *
rtb_IntegralGain) - Filter_DSTATE) * LREAL#0.05;

(* Sum: '<S2>/Sum' incorporates:
 * DiscreteIntegrator: '<S2>/Integrator'
 * Gain: '<S2>/Proportional Gain' *)
```

```

        rtb_DeadZone      :=      ((LREAL#1.35707510609239      *
rtb_IntegralGain) + Integrator_DSTATE) + rtb_FilterCoefficient;

        (* Gain: '<S2>/Integral Gain' *)
        rtb_IntegralGain :=      LREAL#0.0316047084829628      *
rtb_IntegralGain;

        (* Saturate: '<S2>/Saturation' *)
        IF rtb_DeadZone >= LREAL#100.0 THEN
            (* Output: '<Root>/bo_Valve' *)
            bo_Valve := LREAL#100.0;
        ELSIF rtb_DeadZone > LREAL#0.0 THEN
            (* Output: '<Root>/bo_Valve' *)
            bo_Valve := rtb_DeadZone;
        ELSE
            (* Output: '<Root>/bo_Valve' *)
            bo_Valve := LREAL#0.0;
        END_IF;
        (* End of Saturate: '<S2>/Saturation' *)

        (* DeadZone: '<S3>/DeadZone' *)
        IF rtb_DeadZone > LREAL#100.0 THEN
            rtb_DeadZone := rtb_DeadZone - LREAL#100.0;
        ELSIF rtb_DeadZone >= LREAL#0.0 THEN
            rtb_DeadZone := LREAL#0.0;
        END_IF;
        (* End of DeadZone: '<S3>/DeadZone' *)

        (* Signum: '<S3>/SignPreSat' *)
        IF rtb_DeadZone < LREAL#0.0 THEN
            y := LREAL#-1.0;
        ELSIF rtb_DeadZone > LREAL#0.0 THEN
            y := LREAL#1.0;
        ELSE
            y := rtb_DeadZone;
        END_IF;
        (* Signum: '<S3>/SignPreIntegrator' *)
        IF rtb_IntegralGain < LREAL#0.0 THEN
            y_0 := LREAL#-1.0;
        ELSIF rtb_IntegralGain > LREAL#0.0 THEN
            y_0 := LREAL#1.0;
        ELSE
            y_0 := rtb_IntegralGain;
        END_IF;
        (* Switch: '<S2>/Switch' incorporates:
        * Constant: '<S2>/Constant'
        * Logic: '<S3>/AND'
        * RelationalOperator: '<S3>/Equal'
        * RelationalOperator: '<S3>/NotEqual'
        * Signum: '<S3>/SignPreIntegrator'
        * Signum: '<S3>/SignPreSat' *)
        IF (LREAL#0.0 <> rtb_DeadZone) AND (y = y_0) THEN
            rtb_IntegralGain := LREAL#0.0;

```

```

        END_IF;
        (* End of Switch: '<S2>/Switch' *)

        (* Update for DiscreteIntegrator: '<S2>/Integrator'
*)
        Integrator_DSTATE := (LREAL#0.1 * rtb_IntegralGain)
+ Integrator_DSTATE;

        (* Update for DiscreteIntegrator: '<S2>/Filter' *)
        Filter_DSTATE := (LREAL#0.1 * rtb_FilterCoefficient)
+ Filter_DSTATE;

```

### **Перелік змінних**

```

VAR_INPUT
    bi_Error : LREAL ;
END_VAR
VAR_OUTPUT
    bo_Valve : LREAL ;
END_VAR
VAR
    Integrator_DSTATE : LREAL ;
    Filter_DSTATE : LREAL ;
    rtb_IntegralGain : LREAL ;
    rtb_DeadZone : LREAL ;
    rtb_FilterCoefficient : LREAL ;
    y : LREAL ;
    y_0 : LREAL ;
END_VAR

```

## 4 Програмне забезпечення людино-машинного інтерфейсу

### Драйвера

State	Identification	Description	File name
Filter text	Filter text	Filter text	Filter text
	Driver for internal variables		Intern
	Driver for mathematics variables		MATHDR32
	Driver for system variables		SYSDRV
	zenon Logic: Controller		STRATONNG
	VIPA		S7TCP32

### Перелік змінних

State	Name	Measurin...	Driver	Data type	Signal area max.	Signal area min.	Measuring ra...	Measuring range ...
Filter text	Filter text	Filter text	Filter text	Filter text	Filter text	Filter text	Filter text	Filter text
	Controller/Global/Control_Pressure		STRATONNG - zenon Logic: Contr...	REAL	1.000000	0.000000	1.000	0.000
	Controller/Global/Control_SetPoint		STRATONNG - zenon Logic: Contr...	REAL	1.000000	0.000000	1.000	0.000
	Controller/Global/Control_Valve		STRATONNG - zenon Logic: Contr...	REAL	100.000000	0.000000	100.0	0.0
	Pressure		S7TCP32 - VIPA	REAL	1.000000	0.000000	1.000	0.000
	Valve		S7TCP32 - VIPA	REAL	100.000000	0.000000	100.0	0.0

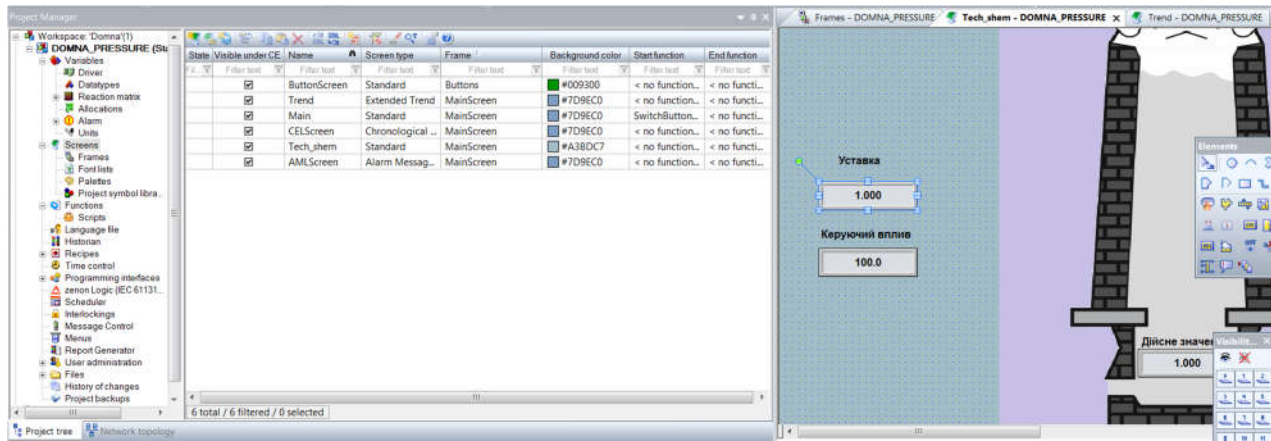
### Шаблони зображень

Sta	Always in L...	Display in Fra...	Name	Background co...	Freely defth	Left [pi...	Top [pixels]	Right [pixe...	Bottom [pi...
Filter text	Filter text	Filter text	Filter text	Filter text	Filter text	Filter text	Filter text	Filter text	Filter text
	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	ALARM STATUS LINE	#FF0000	<input type="checkbox"/>	0	0	1280	20
	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	MainScreen	#AAAAAF	<input type="checkbox"/>	0	20	1280	920
	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Buttons	#AAAAAF	<input type="checkbox"/>	0	920	1280	1024

### Функції

Sta	Name	Type	Parameter
Filter text	Filter text	Filter text	Filter text
	ExitRuntime	Exit Runtime	
	ReloadProject	Reload project online	changed objects
	SwitchAMLScreen	Screen switch	AMLScreen (Alarm Message List) - [*][*]-[T,Rel:0d,1h,0m,...
	SwitchButtonScreen	Screen switch	ButtonScreen (Standard)
	SwitchCELScreen	Screen switch	CELScreen (Chronological Event List) - [*][*]-[T,Rel:0d,1h,...
	SwitchMain	Screen switch	Main (Standard)
	SwitchProcessScreen	Screen switch	Tech_shem (Standard)
	SwitchTrendScreen	Screen switch	Trend (Extended Trend) - PROCESS[HD]

### Зображення



Система автоматичного управління перепадом тиску у нижній частині шахти доменної печі

Уставка  
0.625

Керуючий вплив  
48.8

Дійсне значення  
0.626

Інформація    Технологічна схема    Параметри процесу    Список подій    Список тривоги    Перезавантажити    Вихід



Фільтр  
[\*]-[\*]-[T,Rel.Od,1h,0m,0s]

Профілі фільтрів

Зберегти    Імпорт    Експорт

Стоп

Кількість  
15

Час отримання	Текст	Ім'я змінної	Значе... Один...	Користувач - по...
14.12.2019 16:49:54	System was started			SYSTEM
14.12.2019 16:50:23	Modify spontaneous value: (0.6)	Controller/Global/Control_S0	625	SYSTEM
14.12.2019 16:50:34	System was stopped			SYSTEM
14.12.2019 16:54:38	System was started			SYSTEM
14.12.2019 16:54:52	System was stopped			SYSTEM
14.12.2019 16:55:23	System was started			SYSTEM
14.12.2019 16:56:44	System was stopped			SYSTEM
14.12.2019 17:00:22	System was started			SYSTEM
14.12.2019 17:02:47	Modify spontaneous value: (1.000)	Controller/Global/Control_S1	1.000	SYSTEM
14.12.2019 17:04:14	System was stopped			SYSTEM
14.12.2019 17:06:23	System was started			SYSTEM
14.12.2019 17:09:50	System was stopped			SYSTEM
14.12.2019 17:10:28	System was started			SYSTEM
14.12.2019 17:11:59	System was stopped			SYSTEM
14.12.2019 17:13:13	System was started			SYSTEM

Коментар

Віддалити    Друкувати

Информация    Технологічна схема    Параметри процесу    Список подій    Список тривог    Перезавантажити    Вихід

Історія функцій

Зберегти

Фільтр

Вартість

Не ідентифіковані

Alar...	Time received	Time cleared	Time acknowledged	Variable name	Value	Meas...
---------	---------------	--------------	-------------------	---------------	-------	---------

Коментарі

Функції

## **Додаток Б**

**Тексти програм реалізації математичної моделі комп'ютерної мережі як замкнутої системи масового обслуговування**



**Міністерство освіти і науки України  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
“ДНІПРОВСЬКА ПОЛІТЕХНІКА”**

**ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ  
ПРОГРАМА МАТЕМАТИЧНОЇ МОДЕЛІ МЕРЕЖІ  
КОМП'ЮТЕРНОЇ СИСТЕМИ**

Текст програми

804.02070743.20005-01 12 01

Листів 19

2020

## АНОТАЦІЯ

Даний документ містить ПЗ реалізації математичної моделі комп'ютерної мережі.

Тексти програм реалізовані в середовищі Matchsfdb середовищі операційної системи Windows 10..

Програма реалізує рекурентний метод Бузена для розрахунку параметрів комп'ютерної мережі як замкнутої системи масового обслуговування.

## ЗМІСТ

Стор.

Перелік використаних змінних

.

Текст програми

.

Результати розрахунку

.

7

## 1 Перелік використаних змінних

$N_n$  – кількість вузлів мережі.

$\tau$  – час обробки одного пакета у вузлі.

$P_r$  – матриця перехідних ймовірностей.

$e$  – матриця перехідних коефіцієнтів.

$m$  – кількість конвеєрів у вузлах.

$N$  – кількість пакетів що циркулюють в мережі.

$B$  – матриця ймовірностей черги у вузлах.

$\lambda$  – середня інтенсивність запитів на вході у вузол.

$L$  – середня черга пакетів у вузлі.

$t$  – середній час перебування пакета у вузлі.

## 2 Текст програми

Кількість вузлів мережі

$N_n := 11$

$i := 0..N_n \quad j := 0..N_n$

$\tau_i :=$  Час обробки запиту в вузлах мережі

3
3
3
3
3
3
3
3
3
3
3
3
3

	0
0	3
1	3
2	3
3	3
4	3
5	3
6	3
7	3
8	3
9	3
10	3
11	3
12	

Розрахунок інтенсивності обробки запитів в вузлах мережі

$$\mu_i := \frac{1}{\tau_i}$$

	0
0	0.333
1	0.333
2	0.333
3	0.333
4	0.333
5	0.333
6	0.333
7	0.333
8	0.333
9	0.333
10	0.333
11	0.333

Матриця імовірностей передачі

$$Pr := \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.4 & 0 & 0 & 0.6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.25 & 0.25 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.15 & 0.15 & 0.2 & 0 & 0.25 & 0.25 \\ 0 & 0 & 0 & 0.2 & 0.2 & 0 & 0 & 0 & 0 & 0.3 & 0 & 0.3 \\ 0 & 0 & 0 & 0 & 0 & 0.4 & 0 & 0 & 0 & 0.3 & 0.3 & 0 \end{pmatrix}$$

Перевірка правильності заповнення передаточної матриці

$$\text{SumPr}_i := \sum_{j=0}^{Nn} Pr_{i,j}$$

	0
0	1
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1
11	1



	0	1	2	3	4	5	6	7	8
0	-1	0	0.4	0	0	0	0	0	0
1	0	-1	0	0.25	0	0	0	0	0
2	1	0	-1	0.25	0	0	0	0	0
3	0	1	0.6	-1	0	0	0	0	0
4	0	0	0	0	-1	0	0	0	0
5	0	0	0	0	0	-1	0	0	0
6	0	0	0	0	0	0	-1	0	0
7	0	0	0	0	0	0	0	-1	0
8	0	0	0	0	0	0	0	0	-1
9	0	0	0	0	0	0	1	1	1
10	0	0	0	0.5	1	0	0	0	0
11	0	0	0	0	0	1	0	0	...

P1 =

$$j := 1..Nn \quad i := 0..Nn$$

$$P2_{(j-1),i} := P1_{0,i} + P1_{j,i}$$

	0	1	2	3	4	5	6	7	8	9
0	-1	-1	0.4	0.25	0	0	0	0	0	0
1	0	0	-0.6	0.25	0	0	0	0	0	0
2	-1	1	1	-1	0	0	0	0	0	0
3	-1	0	0.4	0	-1	0	0	0	0	0
4	-1	0	0.4	0	0	-1	0	0	0	0
5	-1	0	0.4	0	0	0	-1	0	0	0.15
6	-1	0	0.4	0	0	0	0	-1	0	0.15
7	-1	0	0.4	0	0	0	0	0	-1	0.2
8	-1	0	0.4	0	0	0	1	1	1	-1
9	-1	0	0.4	0.5	1	0	0	0	0	0.25
10	-1	0	0.4	0	0	1	0	0	0	...

P2 =



$$j := 0..Nn - 1 \quad i := 0..Nn - 1 \quad PP2_{j,i} := P2_{j,i+1}$$

PP2 =

	0	1	2	3	4	5	6	7	8
0	-1	0.4	0.25	0	0	0	0	0	0
1	0	-0.6	0.25	0	0	0	0	0	0
2	1	1	-1	0	0	0	0	0	0
3	0	0.4	0	-1	0	0	0	0	0
4	0	0.4	0	0	-1	0	0	0	0
5	0	0.4	0	0	0	-1	0	0	0.15
6	0	0.4	0	0	0	0	-1	0	0.15
7	0	0.4	0	0	0	0	0	-1	0.2
8	0	0.4	0	0	0	1	1	1	-1
9	0	0.4	0.5	1	0	0	0	0	0.25
10	0	0.4	0	0	1	0	0	0	...

$$Q_{j,0} := P2_{j,0}$$

Q =

	0
0	-1
1	0
2	-1
3	-1
4	-1
5	-1
6	-1
7	-1
8	-1
9	-1
10	-1

$$E := \text{Isolve}(PP2, Q)$$

E =

	0
0	-1.5
1	-2.5
2	-6
3	-3
4	-6
5	-2.7
6	-2.7
7	-3.6
8	-18
9	-15
10	-15
11	



Розрахунок значень функції A

$$A_{i,j} := \begin{cases} j! & \text{if } m_i \geq N-1 \\ 1 & \text{if } m_i = 1 \\ j! & \text{if } 1 < m_i < N-1 \wedge j \leq m_i \\ m_i! \cdot (m_i)^{j-m_i} & \text{if } 1 < m_i < N-1 \wedge j > m_i \end{cases}$$

A =

	0	1	2	3	4	5
0	1	1	1	1	1	
1	1	1	1	1	1	
2	1	1	1	1	1	
3	1	1	1	1	1	
4	1	1	1	1	1	
5	1	1	1	1	1	
6	1	1	1	1	1	
7	1	1	1	1	1	
8	1	1	1	1	1	
9	1	1	1	1	1	
10	1	1	1	1	1	
11	1	1	1	1	1	

$$X_i := \frac{e_i}{\mu_i}$$

X =

	0
0	3
1	4.5
2	7.5
3	18
4	9
5	18
6	8.1
7	8.1
8	10.8
9	54
10	45
11	45

Обчислення матриці констант T

$$T_{i,j} := \frac{(X_i)^j}{A_{i,j}} \quad T_{i,0} := 1$$

T =

	0	1	2	3	4	5
0	1	3	9	27	81	
1	1	4.5	20.25	91.125	410.063	
2	1	7.5	56.25	421.875	3.164·10 <sup>3</sup>	
3	1	18	324	5.832·10 <sup>3</sup>	1.05·10 <sup>5</sup>	
4	1	9	81	729	6.561·10 <sup>3</sup>	
5	1	18	324	5.832·10 <sup>3</sup>	1.05·10 <sup>5</sup>	
6	1	8.1	65.61	531.441	4.305·10 <sup>3</sup>	
7	1	8.1	65.61	531.441	4.305·10 <sup>3</sup>	
8	1	10.8	116.64	1.26·10 <sup>3</sup>	1.36·10 <sup>4</sup>	
9	1	54	2.916·10 <sup>3</sup>	1.575·10 <sup>5</sup>	8.503·10 <sup>6</sup>	
10	1	45	2.025·10 <sup>3</sup>	9.113·10 <sup>4</sup>	4.101·10 <sup>6</sup>	
11	1	45	2.025·10 <sup>3</sup>	9.113·10 <sup>4</sup>	4.101·10 <sup>6</sup>	

Розраховуємо константи для другого і наступних вузлів

$$i := 1..Nn \quad k := 0..N-1$$

$$G_{0,j} := T_{0,j}$$

	0	1	2	3	4	5
0	1	3	9	27	81	
1	1	7.5	42.75	219.375	1.068·10 <sup>3</sup>	
2	1	15	155.25	1.384·10 <sup>3</sup>	1.145·10 <sup>4</sup>	
3	1	33	749.25	1.487·10 <sup>4</sup>	2.791·10 <sup>5</sup>	
4	1	42	1.127·10 <sup>3</sup>	2.502·10 <sup>4</sup>	5.043·10 <sup>5</sup>	
5	1	60	2.207·10 <sup>3</sup>	6.475·10 <sup>4</sup>	1.67·10 <sup>6</sup>	
6	1	68.1	2.759·10 <sup>3</sup>	8.709·10 <sup>4</sup>	2.375·10 <sup>6</sup>	
7	1	76.2	3.376·10 <sup>3</sup>	1.144·10 <sup>5</sup>	3.302·10 <sup>6</sup>	
8	1	87	4.316·10 <sup>3</sup>	1.61·10 <sup>5</sup>	5.041·10 <sup>6</sup>	
9	1	141	1.193·10 <sup>4</sup>	8.053·10 <sup>5</sup>	4.852·10 <sup>7</sup>	
10	1	186	2.03·10 <sup>4</sup>	1.719·10 <sup>6</sup>	1.259·10 <sup>8</sup>	
11	1	231	3.069·10 <sup>4</sup>	3.1·10 <sup>6</sup>	2.654·10 <sup>8</sup>	
12						
13						
14						
15						
16						

G =

$$B_{Nn,j} = \frac{T_{Nn,j}}{G_{Nn,N-1}} \cdot G_{Nn,N-1-j}$$

$$B_{Nn,0} := 1 - B_{Nn,1}$$

	0	1	2	3	4	5
0	0	0	0	0	0	
1	0	0	0	0	0	
2	0	0	0	0	0	
3	0	0	0	0	0	
4	0	0	0	0	0	
5	0	0	0	0	0	
6	0	0	0	0	0	
7	0	0	0	0	0	
8	0	0	0	0	0	
9	0	0	0	0	0	
10	0	0	0	0	0	
11	0.474	0.526	0.234	0.079	0.015	
12						
13						
14						
15						
16						

Розрахунок допоміжних коефіцієнтів  $i := 0..Nn - 1$   $j := 1..N - 1$   $Gn_{i,0} := 1$

$$Gn_{i,j} := G_{Nn,j} - \sum_{k=1}^j (T_{i,k} \cdot Gn_{i,j-k})$$

	0	1	2	3	4	5
0	1	228	$3 \cdot 10^4$	$3.008 \cdot 10^6$	$2.561 \cdot 10^8$	
1	1	226.5	$2.966 \cdot 10^4$	$2.962 \cdot 10^6$	$2.514 \cdot 10^8$	
2	1	223.5	$2.896 \cdot 10^4$	$2.87 \cdot 10^6$	$2.421 \cdot 10^8$	
3	1	213	$2.654 \cdot 10^4$	$2.547 \cdot 10^6$	$2.096 \cdot 10^8$	
4	1	222	$2.862 \cdot 10^4$	$2.824 \cdot 10^6$	$2.375 \cdot 10^8$	
5	1	213	$2.654 \cdot 10^4$	$2.547 \cdot 10^6$	$2.096 \cdot 10^8$	
6	1	222.9	$2.882 \cdot 10^4$	$2.851 \cdot 10^6$	$2.403 \cdot 10^8$	
7	1	222.9	$2.882 \cdot 10^4$	$2.851 \cdot 10^6$	$2.403 \cdot 10^8$	
8	1	220.2	$2.82 \cdot 10^4$	$2.768 \cdot 10^6$	$2.319 \cdot 10^8$	
9	1	177	$1.822 \cdot 10^4$	$1.442 \cdot 10^6$	$9.797 \cdot 10^7$	
10	1	186	$2.03 \cdot 10^4$	$1.719 \cdot 10^6$	$1.259 \cdot 10^8$	
11						

$$i := 0..Nn-1 \quad j := 0..N-1$$

$$B_{i,j} := \frac{T_{i,j}}{G_{Nn,N-1}} G_{i,N-1-j}$$

	0	1	2	3	4
0	0.965	0.034	$1.018 \cdot 10^{-3}$	$2.32 \cdot 10^{-5}$	$3.052 \cdot 10^{-7}$
1	0.947	0.05	$2.263 \cdot 10^{-3}$	$7.778 \cdot 10^{-5}$	$1.545 \cdot 10^{-6}$
2	0.912	0.081	$6.139 \cdot 10^{-3}$	$3.553 \cdot 10^{-4}$	$1.192 \cdot 10^{-5}$
3	0.79	0.173	0.032	$4.681 \cdot 10^{-3}$	$3.956 \cdot 10^{-4}$
4	0.895	0.096	$8.735 \cdot 10^{-3}$	$6.099 \cdot 10^{-4}$	$2.472 \cdot 10^{-5}$
5	0.79	0.173	0.032	$4.681 \cdot 10^{-3}$	$3.956 \cdot 10^{-4}$
6	0.905	0.087	$7.126 \cdot 10^{-3}$	$4.464 \cdot 10^{-4}$	$1.622 \cdot 10^{-5}$
7	0.905	0.087	$7.126 \cdot 10^{-3}$	$4.464 \cdot 10^{-4}$	$1.622 \cdot 10^{-5}$
8	0.874	0.113	0.012	$1.045 \cdot 10^{-3}$	$5.127 \cdot 10^{-5}$
9	0.369	0.294	0.2	0.105	0.032
10	0.474	0.291	0.155	0.064	0.015
11	0.474	0.291	0.155	0.064	0.015
12					
13					
14					
15					
16					

B =

$$i := 0..Nn \quad j := 0..N-1$$

$$\text{SumB}_i := \sum_j B_{i,j}$$

	0
0	1
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1
11	1

SumB =

### 3 Результати розрахунку

$$\lambda_i := e_i \cdot \frac{G_{Nn-1, N-2}}{G_{Nn, N-1}}$$

Інтенсивність вхідного потоку

$$L_{wy} := \sum_{n=0}^{N-1} (n \cdot B_{i, n})$$

Середнє число пакетів в вузлах

$\lambda =$

	0
0	6.477·10 <sup>-3</sup>
1	9.715·10 <sup>-3</sup>
2	0.016
3	0.039
4	0.019
5	0.039
6	0.017
7	0.017
8	0.023
9	0.117
10	0.097
11	0.097

$L =$

	0
0	0.036
1	0.055
2	0.094
3	0.253
4	0.115
5	0.253
6	0.103
7	0.103
8	0.141
9	1.137
10	0.855
11	0.855

Середній час перебування пакета в вузлі

$$t_i := \frac{L_i}{\lambda_i}$$

$t =$

	0
0	5.575
1	5.66
2	5.836
3	6.516
4	5.927
5	6.516
6	5.872
7	5.872
8	6.039
9	9.755
10	8.797
11	8.797



