Департамент образования и науки города Москвы

Государственное автономное образовательное учреждение

высшего образования города Москвы

«Московский городской педагогический университет»

Институт цифрового образования

Департамент информатики, управления и технологий

ДИСЦИПЛИНА:

Распределенные системы

**Лабораторная работа 4**

Выполнила: Селиверстова С.Н., группа: АДЭУ-221

Преподаватель: Босенко Т.М.

Москва

2024

**Лабораторная работа 4. Обнаружение отказов в распределенной системе**

Цель работы:

Изучить принципы обнаружения отказов в распределенных системах с

помощью симулятора Serf Convergence Simulator и проанализировать влияние

различных параметров на время конвергенции и использование полосы

пропускания.**Теоретическая часть**

Serf — это инструмент для управления кластером, который использует

протокол gossip для обнаружения узлов, обнаружения отказов и оркестрации

событий. Протокол gossip — это метод распространения информации в

распределенной системе, где узлы периодически обмениваются информацией с случайно выбранными соседями.

Основные понятия:

1. Gossip Interval - интервал между раундами gossip-протокола.

2. Gossip Fanout - количество узлов, с которыми каждый узел общается за

один раунд.

3. Конвергенция - состояние, когда все узлы в системе имеют согласованное

представление о состоянии системы.

4. Packet Loss - процент потери пакетов в сети.

5. Node Failures - процент отказавших узлов в системе.

**Ход работы**

1. Откройте Serf Convergence Simulator в веб-браузере.

2. Проведите серию экспериментов, изменяя следующие параметры:

- Gossip Interval (секунды): 0.2, 0.5, 1.0

- Gossip Fanout (узлы): 3, 5, 7

- Nodes (число узлов): 10, 50, 100

- Packet Loss (%): 0, 10, 25

- Node Failures (%): 0, 5, 10

3. Для каждого эксперимента зафиксируйте следующие показатели:

- Время до достижения состояния "Хотя бы один узел знает о сбое"

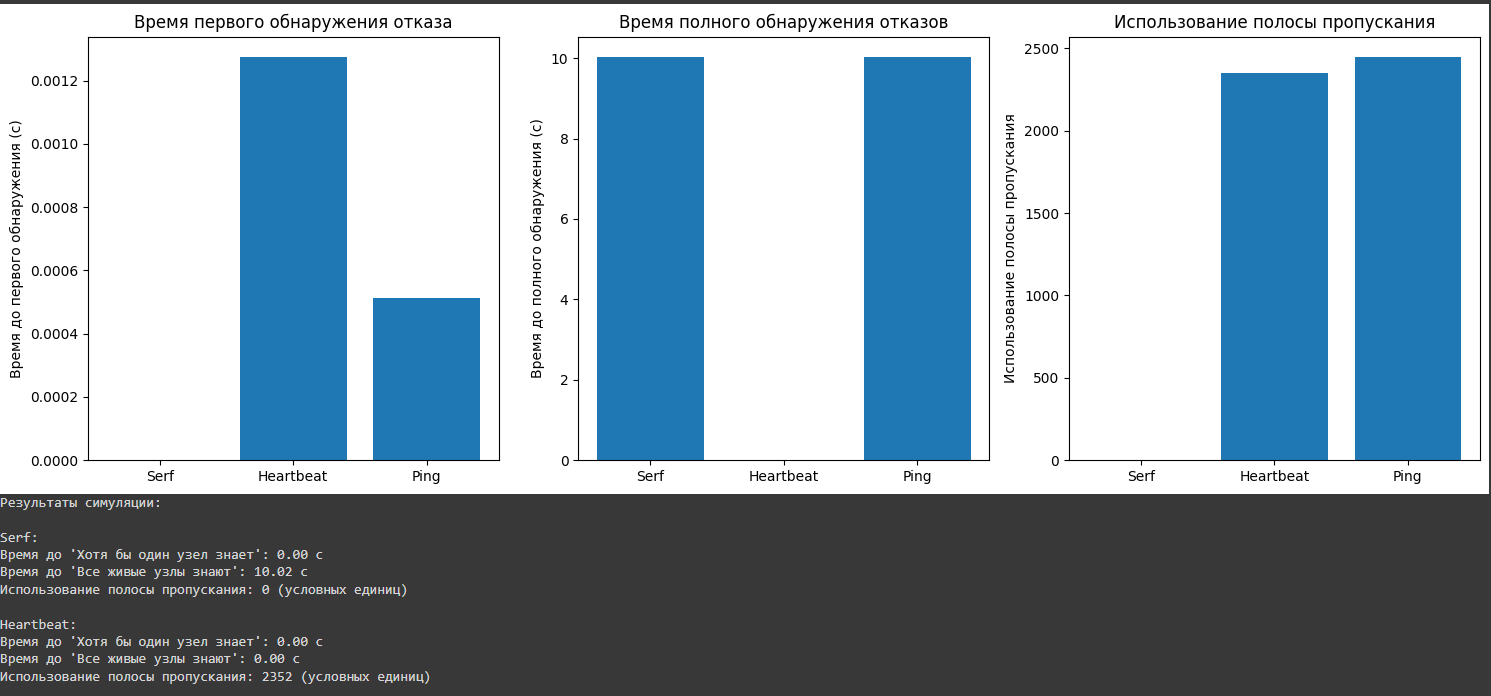
- Время до достижения состояния "Все живые узлы знают о сбое"

- Максимальное использование полосы пропускания на узел (Кбит/с)

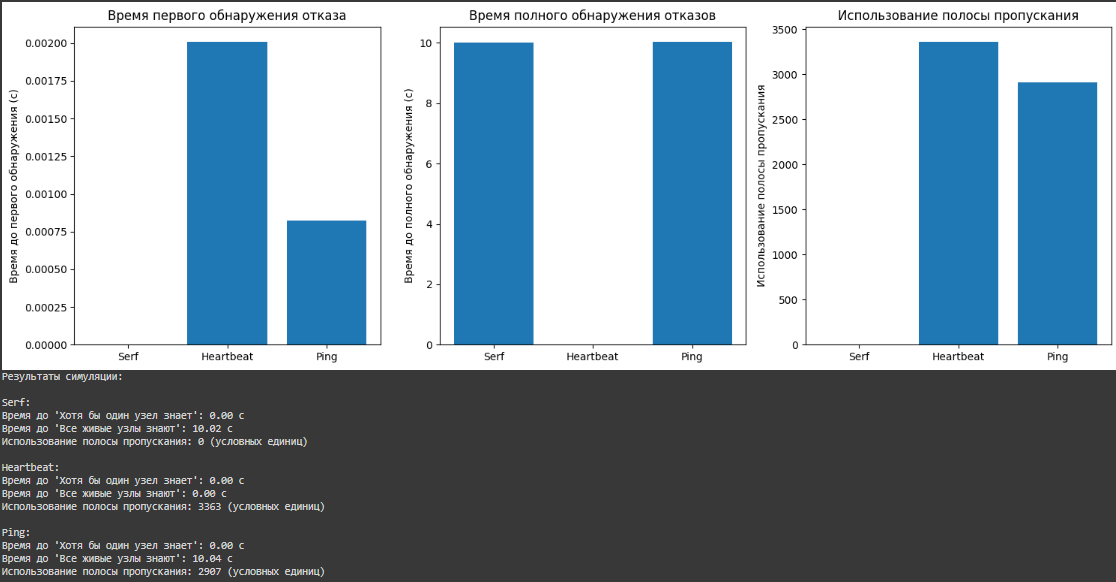
4. Заполните таблицу результатов (пример):

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Gossip  interval | Gossip  fanout | Nodes | Packet  loss | Nodes  failure | Ширина  Полосы пропускания | Время конвергенции |
| 0.2 | 3 | 50 | 5% | 5% | 6,653,952.00 | 14.20 |
| 0.2 | 3 | 60 | 5% | 5% | 7,984,742.40 | 17.20 |
| 0.2 | 3 | 70 | 5% | 5% | 9,315,532.80 | 20.00 |
| 0.2 | 3 | 80 | 5% | 5% | 10,646,323.20 | 23.00 |
| 0.2 | 3 | 90 | 5% | 5% | 11,977,113.60 | 25.80 |
| 0.2 | 3 | 100 | 5% | 5% | 13,307,904.00 | 28.80 |

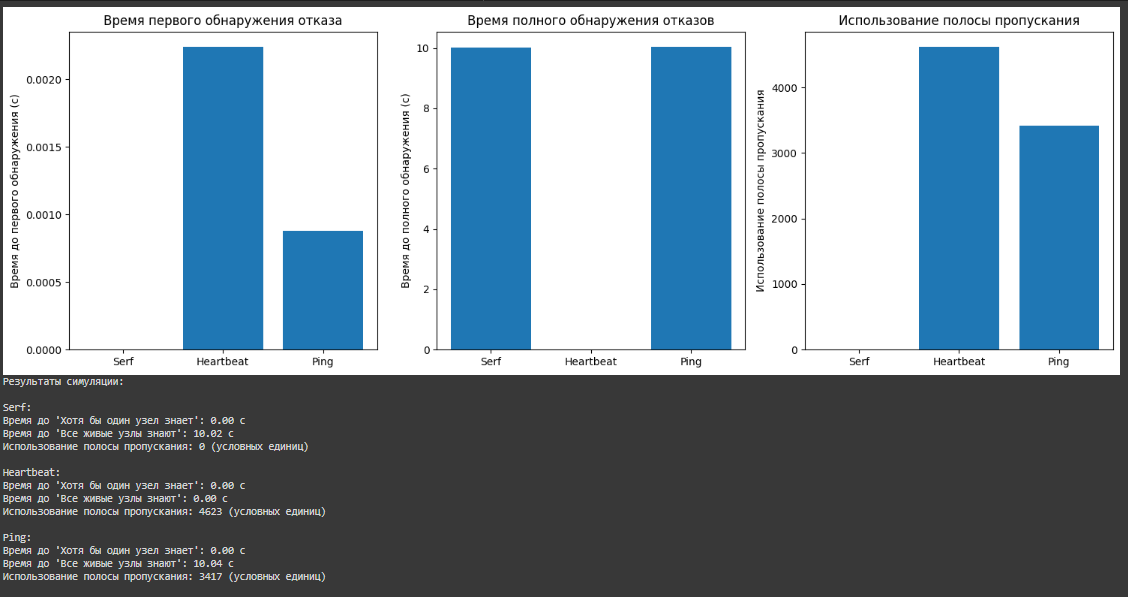
Для nodes = 50:



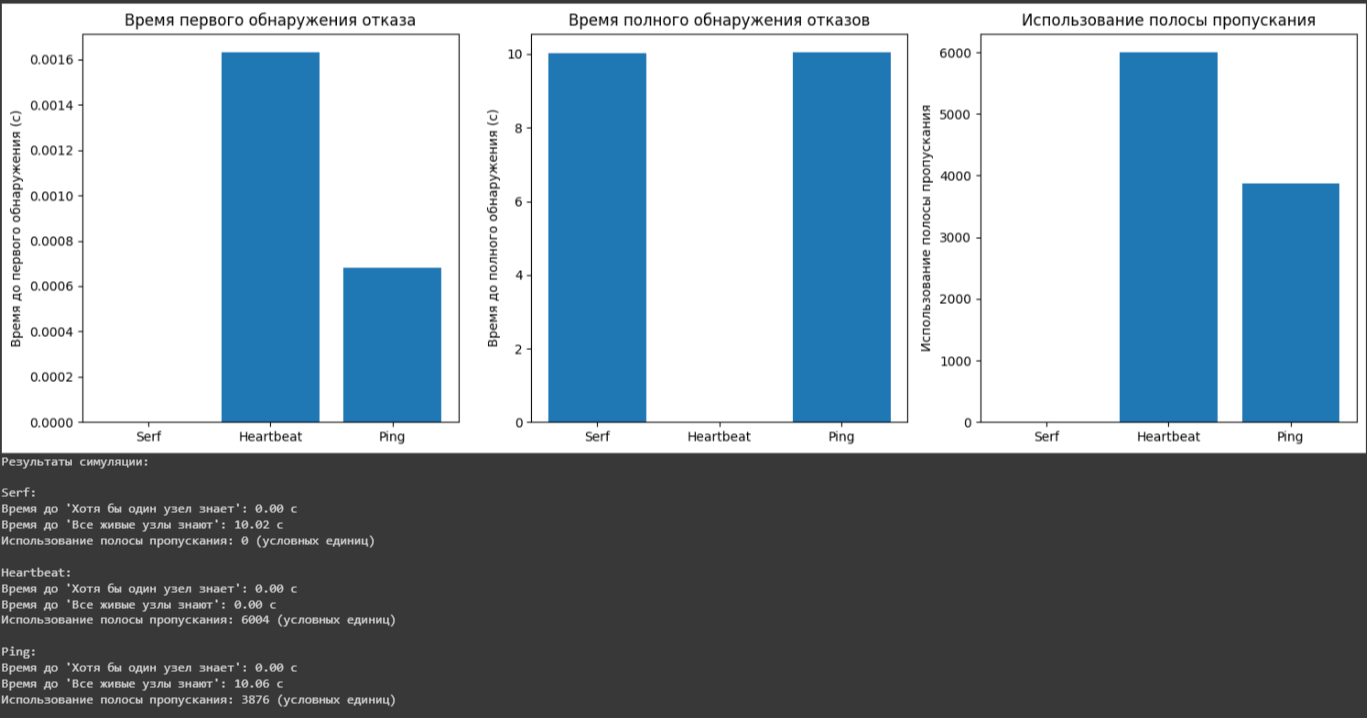
Для nodes = 60:



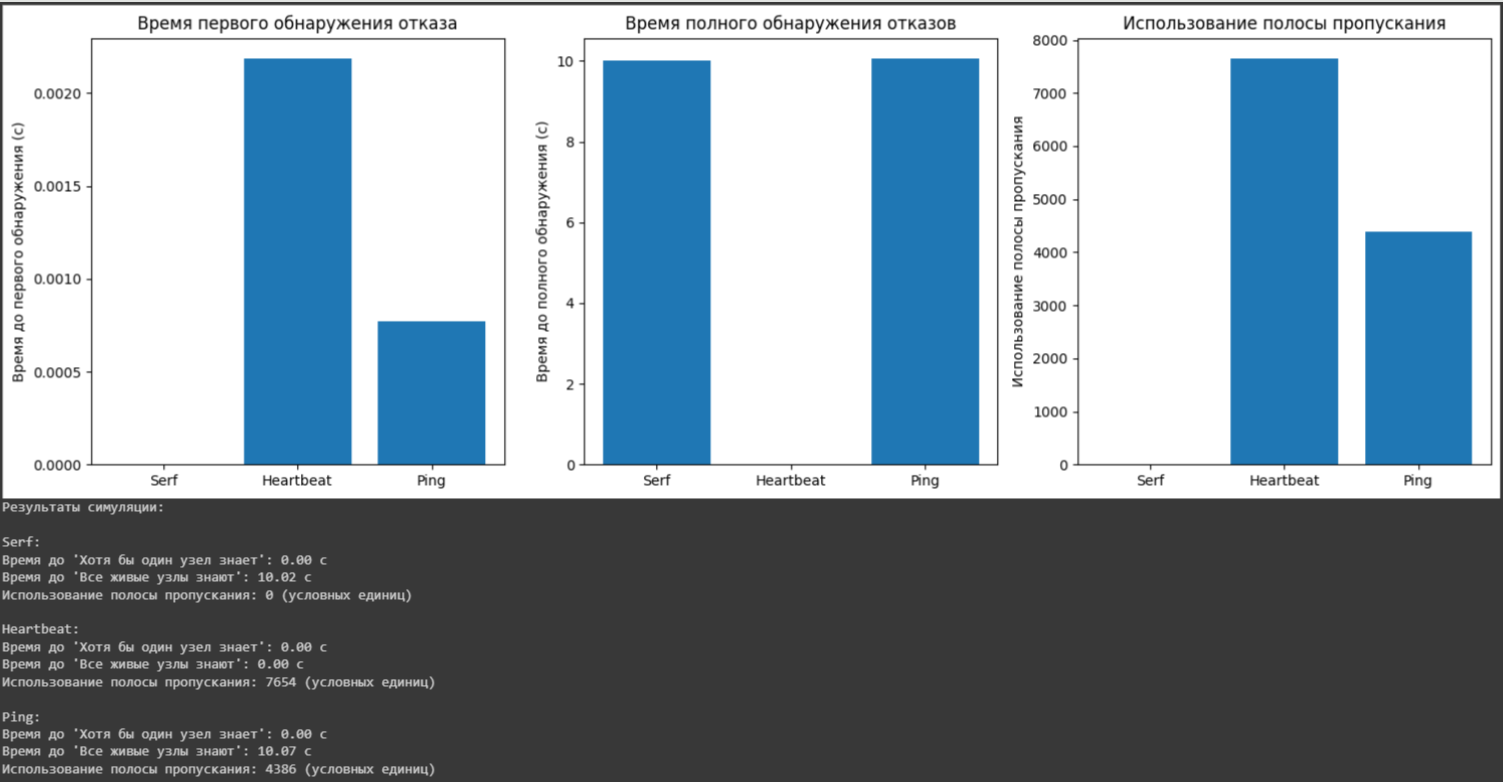
Для nodes = 70:



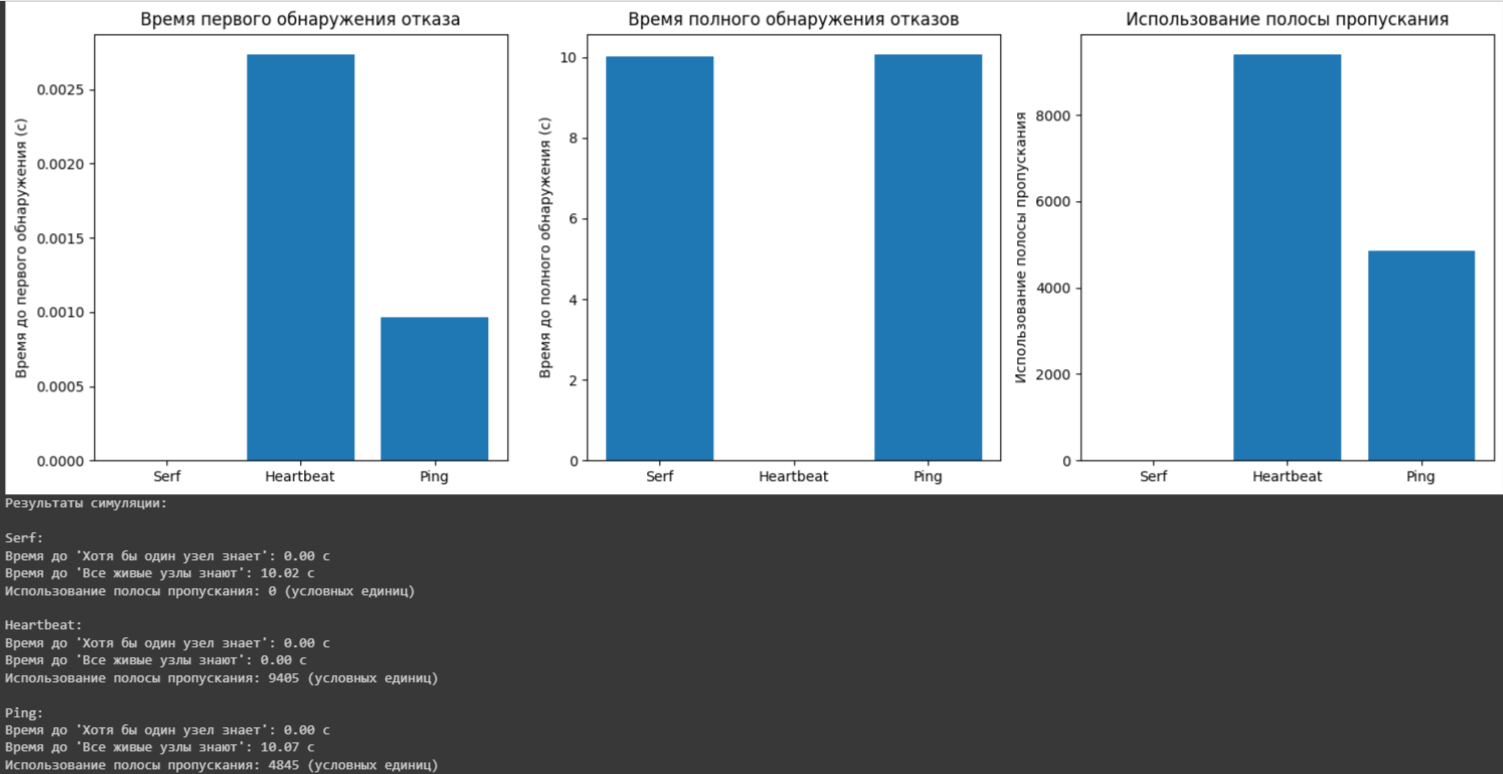
Для nodes = 80:



Для nodes = 90:



Для nodes = 100:



5. Проанализируйте полученные результаты и ответьте на следующие вопросы:

в) Как масштабируется система при увеличении количества узлов?

При добавлении узлов в распределенные системы с использованием алгоритмов gossip обычно наблюдается линейное масштабирование. Время конвергенции и ширина полосы пропускания увеличиваются, то есть система становится менее эффективной в реагировании на изменения, но также она способна обрабатывать более широкий диапазон частот, что может привести к улучшению качества передачи сигнала или увеличению скорости передачи данных.