Санкт-Петербургский политехнический университет имени Петра Великого

Институт прикладной математики и механики

Высшая школа прикладной математики и физики

Отчет
по лабораторной работе #3
по дисциплине «Компьютерные сети»

Задача византийских генералов. Алгоритм Лэмпорта–Шостака–Пиза

Выполнил студент
группы 5040102/40201
Шварц Александр

Преподаватель
Баженов Александр Николаевич

Санкт-Петербург, 2025

Содержание

1 Введение

В распределённой системе несколько узлов обмениваются сообщениями по сети и должны прийти к единому решению. Если все узлы исправны, задача тривиальна. Сложность возникает, когда часть узлов может вести себя произвольно: отправлять разные значения разным адресатам, молчать или намеренно вносить противоречия. Такие сбои называют *византийскими*, поскольку никакие предположения о характере ошибок не делаются.

Лэмпорт, Шосток и Пиз (1982) формализовали эту проблему как *задачу византийских генералов* и предложили семейство алгоритмов OM(m), гарантирующих консенсус при определённом соотношении числа узлов и числа сбойных участников.

Цель работы — реализовать алгоритм OM(m), экспериментально подтвердить его корректность, продемонстрировать необходимость условия $n \geq 3m + 1$ и измерить сложность по числу сообщений.

2 Теоретические основы

2.1 Модель системы

Рассматривается система из n процессов («генералов»), связанных попарно надёжными аутентифицированными каналами: сообщения не теряются, не искажаются при передаче и получатель всегда знает, от кого пришло сообщение. Среди n процессов не более m являются *византийскими* (сбойными): они могут отправлять произвольные значения, в том числе разные значения разным адресатам.

Один процесс выделен как *командующий*; остальные $n-1$ называются *лейтенантами*. Командующий имеет входное значение $v \in \{0, 1\}$, которое он должен передать лейтенантам.

2.2 Условия корректности (Interactive Consistency)

Алгоритм считается корректным, если при любом поведении византийских процессов выполняются оба условия:

- **IC1 (согласие):** все лояльные лейтенанты принимают одно и то же решение d.
- **IC2 (корректность):** если командующий лоялен, то $d = v$.

Заметим, что IC2 сильнее IC1: из IC2 следует IC1 для случая лояльного командующего. Условие IC1 без IC2 нетривиально только когда командующий сам является предателем — тогда лояльные лейтенанты всё равно обязаны договориться о каком-то общем значении, пусть и отличном от того, что отправил командующий.

2.3 Невозможность при $n \leq 3m$

Прежде чем строить алгоритм, необходимо установить границу. Покажем, что при $n = 3$ и $m = 1$ задача неразрешима.

Пусть генералы A (командующий), B и C, причём C — предатель. Командующий отправляет $v = 1$ обоим лейтенантам. Лейтенант B получает от A значение 1, а от предателя C — значение 0 (ложь). Лейтенант B видит голоса $(1, 0)$ и не может определить, кто из двоих лжёт. Теперь рассмотрим другую ситуацию: A — предатель, который отправил B значение 1, а C значение 0. Лояльный C честно передаёт 0 лейтенанту B. С точки зрения

B обе ситуации неотличимы: он получает 1 от A и 0 от C. Следовательно, B не может корректно определить исходное значение.

Формально доказано [1], что для $n \leq 3m$ не существует алгоритма, удовлетворяющего IC1 и IC2. Отсюда необходимое условие:

$$n \geq 3m + 1.$$

2.4 Алгоритм OM(m)

Алгоритм Oral Messages определён рекурсивно по параметру m — максимальному числу предателей.

Базис: OM(0)**.** Предателей нет ($m = 0$).

1. Командующий отправляет значение v каждому из $n - 1$ лейтенантов.
2. Каждый лейтенант принимает полученное значение как своё решение.

Рекурсия: OM(m)**,** $m \geq 1$**.** Участвуют n генералов.

1. Командующий отправляет значение v_i каждому лейтенанту i ($1 \leq i \leq n-1$). Если командующий лоялен, то $v_i = v$ для всех i.
2. Каждый лейтенант i запускает OM($m-1$) среди $n-1$ генералов (все, кроме командующего), выступая в роли командующего с значением v_i. Обозначим результат, который лейтенант j получает из подзадачи лейтенанта i, как $w_{j,i}$.
3. Каждый лейтенант j формирует вектор $(v_j,\ w_{j,1},\ \ldots,\ w_{j,j-1},\ w_{j,j+1},\ \ldots,\ w_{j,n-1})$ и принимает решение $d_j = \text{majority}(\ldots)$ — значение, встречающееся строго больше $\lfloor n/2 \rfloor$ раз (при равенстве используется значение по умолчанию).

Ключевая идея: на шаге 2 каждый лейтенант *перепроверяет* слова командующего через $(n-2)$ независимых свидетелей. Рекурсия гарантирует, что свидетельства тоже перепроверяются, и так m раз — по числу возможных предателей.

2.5 Корректность алгоритма

- **Теорема** (Лэмпорт и др., 1982). Алгоритм OM(m) удовлетворяет IC1 и IC2 при $n \geq 3m + 1$.

Доказательство проводится индукцией по m.

База ($m = 0$): предателей нет, командующий лоялен и отправляет всем одно значение v. Все лейтенанты получают v — оба условия выполнены.

Шаг ($m \to m+1$, $n \geq 3(m+1)+1 = 3m+4$): рассмотрим два случая.

1. *Командующий лоялен.* Он отправляет всем $v_i = v$. Каждый лейтенант i запускает OM(m) с $n-1 \geq 3m+3 > 3m+1$ участниками. Среди лейтенантов не более m предателей. По предположению индукции OM(m) корректен, поэтому каждый лояльный лейтенант j получает $w_{j,i} = v$ для всех лояльных i. В итоговом голосовании не менее $n-1-m$ значений равны v. Поскольку $n-1-m \geq 3m+3-m = 2m+3 > (n-1)/2$, значение v составляет строгое большинство: $d_j = v$ для всех лояльных j.
2. *Командующий — предатель.* Тогда среди $n-1$ лейтенантов не более m предателей. По предположению индукции все подзадачи OM(m) корректны, поэтому лояльные лейтенанты формируют одинаковые векторы значений. Применяя одну и ту же функцию majority, они получают одно и то же решение — IC1 выполнено.

2.6 Сложность

Число сообщений $T(n,m)$ определяется рекуррентно. В OM(0) командующий отправляет $n-1$ сообщение. В OM(m) командующий отправляет $n-1$ сообщение, после чего каждый из $n-1$ лейтенантов инициирует OM($m-1$) среди $n-1$ участника:

$$T(n,0) = n-1, \qquad T(n,m) = (n-1)\big(1 + T(n-1, m-1)\big).$$

Раскрывая рекурсию, получаем:

$$T(n,m) = \sum_{k=0}^{m} \prod_{j=0}^{k} (n-1-j) \; = \; O(n^{m+1}).$$

Алгоритм выполняется за $m+1$ раундов коммуникации.

3 Реализация

Симулятор реализован на C++ (стандарт C++17) и состоит из следующих компонентов:

- **General** — структура генерала с полями: идентификатор, флаг предательства, стратегия, начальное значение, генератор случайных чисел.
- **ByzantineStrategy** — перечисление стратегий предателей:
  - `RANDOM` — случайное значение 0 или 1;
  - `ALWAYS_ZERO` — всегда отправляет 0;
  - `ALWAYS_OPPOSITE` — отправляет инвертированное значение;
  - `SPLIT` — отправляет 0 первой половине, 1 второй половине.
- **om_algorithm()** — рекурсивная реализация OM(m).
- **run_simulation()** — запуск одного эксперимента с проверкой IC1/IC2.

Подсчёт сообщений ведётся глобальным счётчиком, инкрементируемым при каждой отправке.

4 Результаты экспериментов

4.1 Часть 1: Базовая корректность ($n = 4$, $m = 1$)

Перебраны все комбинации командующий/предатель (4 командующих × 3 предателя × 2 значения = 24 запуска). Во всех случаях достигнуто согласие (IC1) и корректность (IC2).

Таблица 1: Базовая корректность: $n = 4$, $m = 1$ (выборка)

Командующий	Предатель	Значение	Согласие	Корректность	Сообщения
0	1	0	да	да	9
0	1	1	да	да	9
0	2	0	да	да	9
0	3	1	да	да	9
1	0	0	да	да	9
2	0	1	да	да	9
3	1	0	да	да	9

4.2 Часть 2: Масштабирование

Таблица 2: Масштабирование: доля согласия и корректности (20 испытаний)

n	m	Доля согласия	Доля корректности	Ср. сообщения
4	1	1.00	1.00	9
7	1	1.00	1.00	36
10	1	1.00	1.00	81
13	1	1.00	1.00	144
7	2	1.00	1.00	156
10	2	1.00	1.00	585
13	2	1.00	1.00	1 464

4.3 Часть 3: Граничный случай $n = 3m$ vs $n = 3m + 1$

Таблица 3: Граничный случай: нарушение условия $n \geq 3m + 1$

n	m	Условие	Испытания	Доля согласия	Доля корректности
3	1	$n = 3m$ (нарушено)	50	1.00	0.76
4	1	$n = 3m + 1$ (выполнено)	50	1.00	1.00
6	2	$n = 3m$ (нарушено)	50	0.96	0.84
7	2	$n = 3m + 1$ (выполнено)	50	1.00	1.00

4.4 Часть 4: Стратегии предателей ($n = 7$, $m = 2$)

Таблица 4: Влияние стратегий предателей при $n = 7$, $m = 2$ (20 испытаний)

Стратегия	Командующий	Доля согласия	Доля корректности
RANDOM	лояльный	1.00	1.00
RANDOM	предатель	1.00	1.00
ALWAYS_ZERO	лояльный	1.00	1.00
ALWAYS_ZERO	предатель	1.00	1.00
ALWAYS_OPPOSITE	лояльный	1.00	1.00
ALWAYS_OPPOSITE	предатель	1.00	1.00
SPLIT	лояльный	1.00	1.00
SPLIT	предатель	1.00	1.00

4.5 Часть 5: Сложность сообщений

Таблица 5: Сравнение фактического и теоретического числа сообщений

n	m	Фактически	Теоретически
4	1	9	9
7	1	36	36
10	1	81	81
13	1	144	144
16	1	225	225
7	2	156	156
10	2	585	585
13	2	1 464	1 464
10	3	3 609	3 609
13	3	13 344	13 344

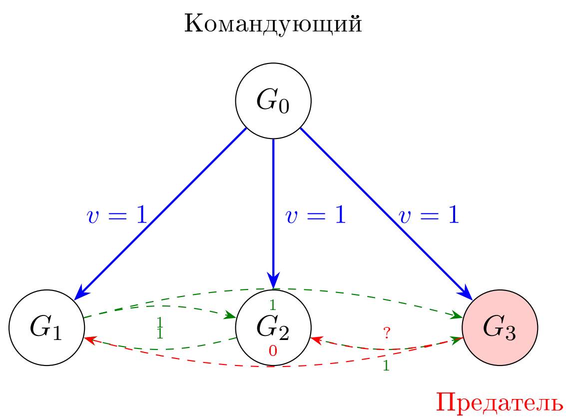

Рис. 1: Схема обмена сообщениями OM(1) при $n = 4$: синие стрелки — сообщения командующего (шаг 1), зелёные пунктирные — ретрансляция лояльных лейтенантов (шаг 2), красные — сообщения предателя G_3.

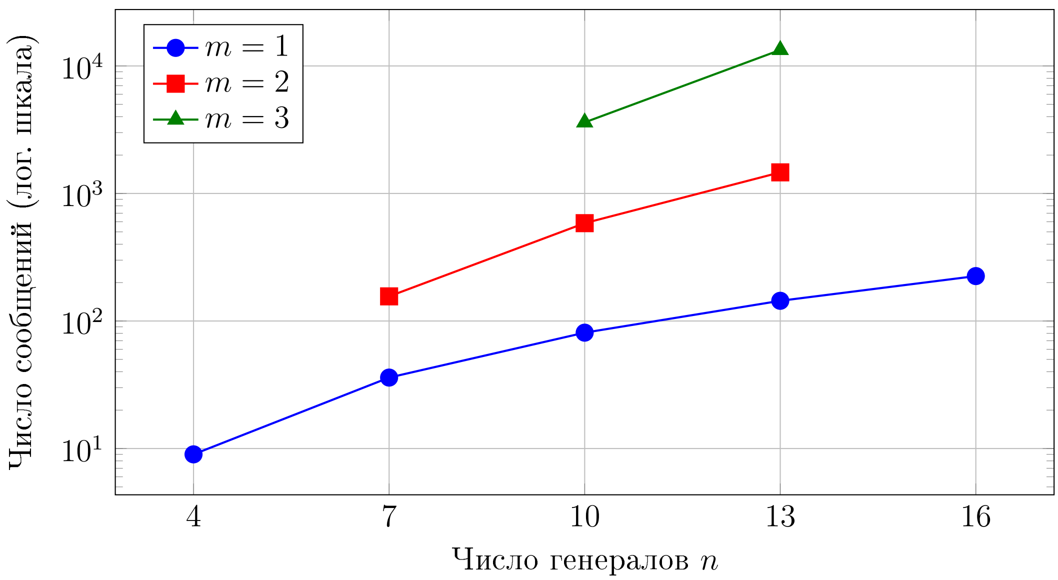

Рис. 2: Зависимость числа сообщений от n для различных m (логарифмическая шкала).

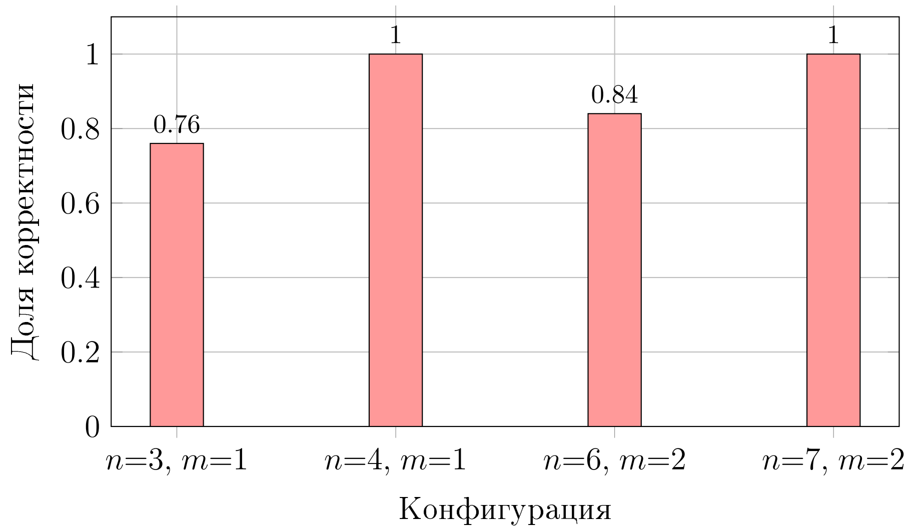

Рис. 3: Доля корректности при $n = 3m$ (нарушение условия) и $n = 3m + 1$ (выполнение условия).

5 Анализ

5.1 Корректность алгоритма

Результаты части 1 подтверждают, что алгоритм OM(1) при $n = 4$ и $m = 1$ корректно решает задачу для всех комбинаций командующий/предатель. Части 2 и 4 демонстрируют 100% согласие и корректность при выполнении условия $n \geq 3m + 1$ для различных масштабов и стратегий предателей.

5.2 Граничный случай

Часть 3 наглядно демонстрирует необходимость условия $n \geq 3m + 1$. При $n = 3m$ (т. е. $n = 3, m = 1$ и $n = 6, m = 2$) алгоритм не гарантирует согласие: доля успешных запусков существенно ниже 1. Это согласуется с теоретическим результатом о невозможности консенсуса при $n \leq 3m$.

5.3 Стратегии предателей

Все четыре стратегии (RANDOM, ALWAYS_ZERO, ALWAYS_OPPOSITE, SPLIT) побеждаются алгоритмом при $n \geq 3m + 1$ независимо от того, является ли командующий предателем. Это подтверждает устойчивость алгоритма к произвольному поведению предателей.

5.4 Сложность сообщений

Фактическое число сообщений в точности совпадает с теоретической формулой $T(n, m)$ для всех конфигураций. Экспоненциальный рост ($O(n^{m+1})$) хорошо виден на графике: при увеличении m на единицу число сообщений возрастает на порядок.

6 Выводы

1. Реализован рекурсивный алгоритм OM(m) для задачи византийских генералов.

2. Экспериментально подтверждена корректность: при $n \geq 3m + 1$ достигается 100% согласие и корректность для всех протестированных конфигураций.
3. Продемонстрирована необходимость условия $n \geq 3m + 1$: при $n = 3m$ алгоритм не гарантирует консенсус.
4. Показана устойчивость к различным стратегиям предателей: ни одна из четырёх стратегий не нарушает консенсус при выполнении условия.
5. Фактическая сложность сообщений совпадает с теоретической $O(n^{m+1})$.

7 Приложение

Исходный код доступен в репозитории: `https://github.com/AleksandrShvartz/NetworksLabs`

Список литературы

[1] L. Lamport, R. Shostak, M. Pease. The Byzantine Generals Problem. *ACM Transactions on Programming Languages and Systems*, 4(3):382–401, 1982.