

Программная модель для исследования энергоэффективности алгоритмов сжатия информации в системах общественного зондирования

С.А. Воробьев, Т.А. Приходько

Аннотация - Оснащенность людей смартфонами, а смартфонов множеством датчиков и различных протоколов связи позволяет использовать их в качестве узлов ad-hoc сетей для решения задач общественного зондирования. Технология общественного зондирования Public Sensing (PS) позволяет осуществлять обмен информацией в обществе по требованию. Однако энергоресурсы мобильных устройств существенно исчерпываются при организации таких сетей, поэтому задача энергоэффективной передачи информации остается актуальной.

Одним из способов экономии энергоресурсов мобильных устройств является применение алгоритмов сжатия передаваемой информации.

Целью данной работы является разработка программы для тестирования разных методов сжатия без потерь или их комбинаций для выяснения того, какой из вариантов сможет сэкономить большее количество энергии при использовании в режиме общественного зондирования.

Разработанная программа, имитирующая процесс работы мобильных узлов в режиме Public Sensing, дала возможность оценить целесообразность и эффективность применения различных методов сжатия передаваемой информации.

Ключевые слова - самоорганизующиеся сети, система общественного зондирования, энергоэффективные алгоритмы сжатия.

ВВЕДЕНИЕ

Технология общественного зондирования Public Sensing (PS) приобретает все большее практическое распространение и связана с развитием ad-hoc сетей [1,2]. Оснащенность смартфонов множеством датчиков и различных протоколов связи позволяет их использовать в качестве узлов ad-hoc сетей для решения задач общественного зондирования. Таким образом, можно построить систему общественного зондирования Public Sensing, описывающую некоторую географическую область по множеству параметров: местоположение, уровень шума, скорость передвижения узлов с основным вектором направления движения группы участников,

зафиксировать повышение температуры или давления в заданной местности и т.д. Однако передача информации в такой сети требует существенного расхода энергии аккумуляторных батарей. Для повышения энергоэффективности PS-сети применяются различные методы среди которых энергосберегающие протоколы маршрутизации [3,4], методы агрегации [5] и методы сжатия передаваемой информации [6-9].

Целью данной работы является исследование возможностей сжатия информации при передаче сообщений в сетях общественного зондирования.

ПОСТАНОВКА ЗАДАЧИ

В данной работе разработана программа имитации функционирования системы общественного зондирования Public Sensing, в ней реализованы алгоритмы сжатия информации без потерь, перечисленные в следующем разделе, а также проведено исследование, эмулирующее отправку пакетов информации с помощью предложенных способов сжатия. Результаты исследования представлены в виде графиков, показывающих следующие зависимости для заданных исходных данных:

- исходные данные;
- количество узлов сети N ;
- количество сообщений, отправляемых за 1 «день» эмуляции K ;
- возможность выбора периода отправки сообщений – по одному или пачкой.

Сообщения генерируются случайным образом, содержат в себе информацию о текущем положении узла, время отправки, а также уникальный идентификатор устройства.

Искомые параметры исследования: средняя нагрузка за 1 «день» эмуляции по каждому из алгоритмов сжатия (возможны их комбинации), наиболее эффективный алгоритм или комбинация, варианты для периодической отправки и разовой.

АЛГОРИТМЫ, ЗАДЕЙСТВОВАННЫЕ В ИССЛЕДОВАНИИ

В реализации программы имитационного моделирования функционирования системы общественного зондирования при отправке сообщений на сервер, задействованы следующие алгоритмы:

1. CJson – алгоритм, основанный на сжатии исходного json сообщения путем преобразования его структуры и избавления от повторяющихся названий полей в каждом объекте;
2. JSONPack¹ – также основывается на преобразовании исходного json сообщения в более компактный вид, имеет более сложную структуру, нежели cJSON, кроме избавления от повторяющихся имен полей – также минимизирует повторы в самих значениях;
3. LZ4² – алгоритм сжатия без потерь;
4. LZFSE3 – алгоритм сжатия без потерь, рекомендованный Apple к применению на их устройствах;
5. LZMA⁴ – цепной алгоритм Лемпеля – Зива – Маркова, относящийся к алгоритмам сжатия данных без потерь;
6. ZLIB⁵ – обобщение алгоритма Deflate, также является алгоритмом сжатия без потерь, он же RFC-1950;
7. GZIP⁶ – алгоритм линейки Deflate, он же RFC-1952;
8. ZIP⁷ – RFC-1951.

ПОСТАНОВКА ЭКСПЕРИМЕНТОВ И РЕЗУЛЬТАТЫ ИССЛЕДОВАНИЙ

Будем рассматривать архитектуру сети общественного зондирования, использующую координаты узла, идентификатор устройства, а также время отправки сообщения в качестве рассматриваемых данных. Форматом обмена данными рассматривается json.

На рисунке 1 представлен пример json-кода одного сообщения.

В случае передачи большего количества сообщений, пакет данных будет представлять собой массив нескольких однотипных объектов, со структурой как у объекта в примере.

```
[
  {
    "latitude": 1.223232,
    "longitude": 1.232323,
    "uid": "95370D8B-75BD-4EB9-825B-11C08FF07F8E",
    "time": "2000-01-01 12:34:00"
  }
]
```

Рис. 1. Пример json сообщения

В симуляции рассмотрены два варианта: 1) в течение «дня» отправляются одиночные сообщения (рис. 1), или же 2) отправляются единожды сразу все **K** сообщений одним пакетом.

Рассмотрим следующую ситуацию для каждого из алгоритмов. Возьмем для исследования сеть с количеством узлов $N = 10$, число сообщений в день - 30, рассмотрим вариант А) отправка всех 30 сообщений одним пакетом, а также вариант В) с последовательной отправкой в течение дня, по одному сообщению.

Результат моделирования на каждом графике сопоставлен с размером оригинального сообщения.

Из графика (рис 2. а) видно, что сжатие на основе избавления от повторяющихся имен полей приводит нас к результату, с коэффициентом сжатия в 0.78.

1. JSONPack compression algorithm [Электронный ресурс]. URL: <https://github.com/rgcl/jsonpack/> (дата обращения 02.03.2020).
2. LZ4 v1.9.2 [Электронный ресурс]. URL: <https://github.com/lz4/lz4/releases/tag/v1.9.2> (дата обращения 19.02.20)
3. LZFSE [Электронный ресурс]. URL: <https://github.com/lzfse/lzfse> (дата обращения 18.02.20).
4. Igor Pavlov has asserted multiple times on SourceForge that the algorithm is his own creation. (2004-02-19). "LZMA spec?". Archived from the original on 2012-11-09. Retrieved 2013-06-16.
5. ZLIB DATA COMPRESSION LIBRARY [Электронный ресурс]. URL: <https://github.com/madler/zlib> (дата обращения 19.02.20).
6. GZIP file format specification version 4.3 [Электронный ресурс]. URL: <https://www.ietf.org/rfc/rfc1952.txt> (дата обращения 17.02.20).
7. DEFLATE Compressed Data Format Specification version 1.3 [Электронный ресурс]. URL: <https://www.ietf.org/rfc/rfc1951.txt> (дата обращения 18.02.20).

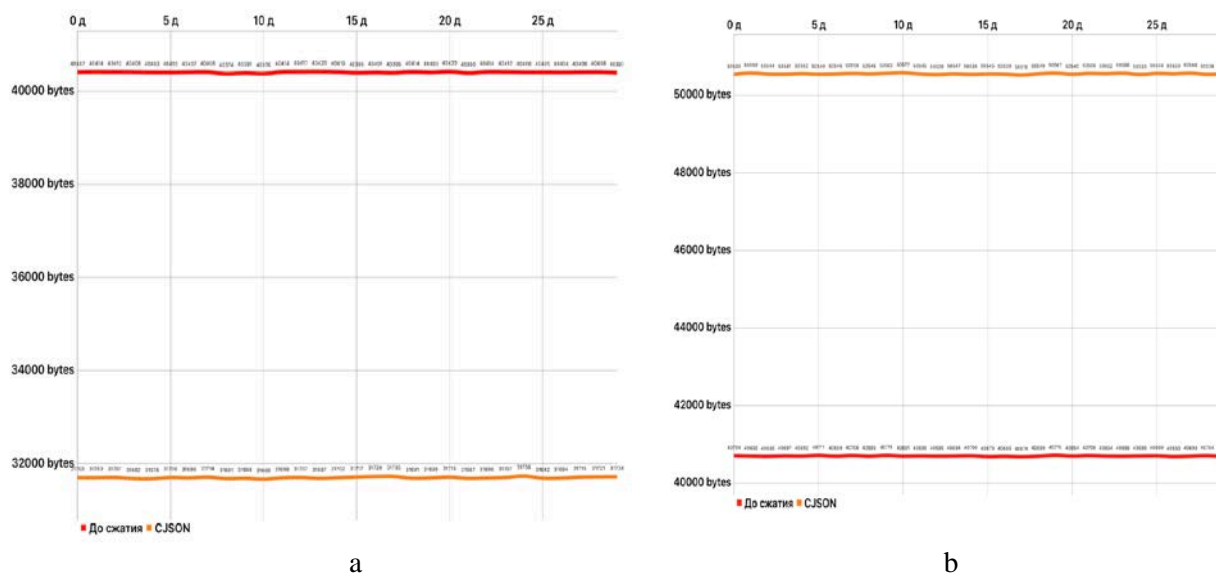


Рис. 2. Моделирование отправки на основе метода сжатия Cjson а) отправка одним пакетом, б) последовательная отправка 30 отдельных сообщений

В данном случае данные представляют из себя массив из 30 сообщений, у которых имеется большое количество повторений имен полей по каждому из пункту сообщения: широта, долгота, время отправки и уникальный идентификатор.

В случае В повторение полей в пакете отсутствует (рис 2. б), становится очевидным, что алгоритм cjson является неэффективным, при последовательной отправки сообщений: коэффициент сжатия составляет 1.24, - иными словами количество трафика,

передаваемого на сервер, увеличилось на 24% при использовании алгоритма Cjson.

Результат предсказуем, так как последовательная отправка не подразумевает передачу массивов информации с однотипными сообщениями, т.е. в сообщениях нет повторяющихся полей, а инфраструктура самого алгоритма добавила некоторое количество метаданных, которая и привела в сумме к увеличению объема сообщений.

Рассмотрим результаты моделирования отправки сообщений, применяя алгоритм jsonPack, результаты экспериментов представлены на рис. 3.

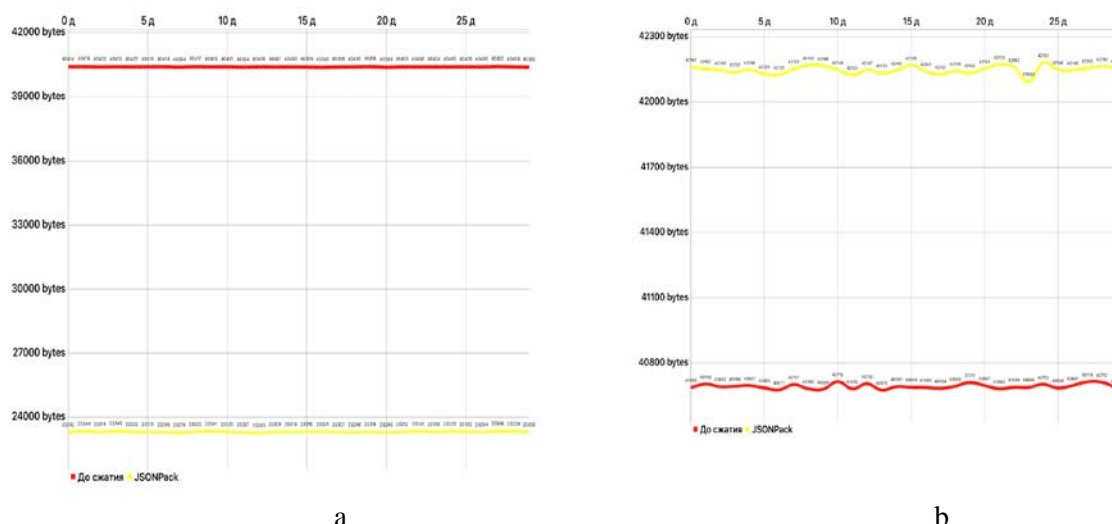


Рис. 3. Моделирование отправки на основе метода сжатия JSONPack а) отправка одним пакетом, б) последовательная отправка 30 отдельных сообщений

Алгоритм JSONPack при отправки одним пакетом дает коэффициент сжатия в 0.57, что явно лучше, чем алгоритм Cjson, который показал результат в 0.78.

В случае, когда отправка является последовательной, вновь видим увеличение количества отправляемых байтов информации на сервер с коэффициент сжатия 1.04, что является

менее критическим результатом, по сравнению с алгоритмом Cjson с коэффициентом 1.24.

Рассмотренные алгоритмы работают с исходными сообщениями на уровне json структуры, пытаются избавиться от повторяющихся названий полей в случае с Cjson, и с повторяющимися символами в случае алгоритма JSONPack.

Далее представлены результаты для алгоритмов сжатия без потерь, работающих с двоичным представлением данных, рассматривающие данные на более низком уровне.

Первым таким рассматриваемым алгоритмом является LZ4, результаты работы которого представлены на рис. 4.

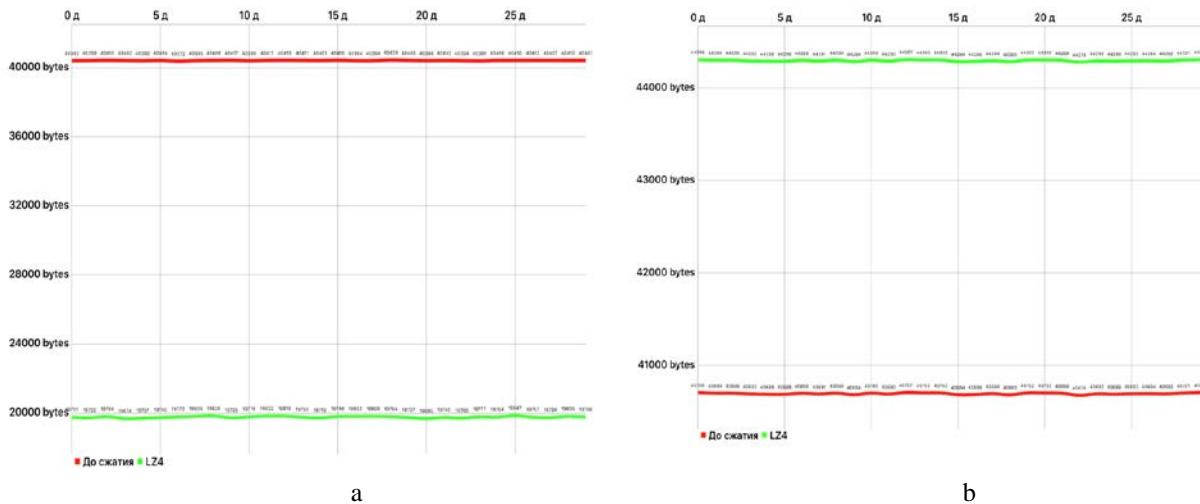


Рис. 4. Моделирование отправки на основе метода сжатия LZ4, а) отправка одним пакетом, б) последовательная отправка 30 отдельных сообщений

Fig.4. Simulating transmission based on the LZ4 compression method а) sending in a single packet, б) sending 30 separate messages sequentially

Как видно из рисунка 4 алгоритм LZ4 показывает результат коэффициента сжатия в 0.49 для отправки одним пакетом, что пока является лучшим результатом, чего не скажешь о последовательной передаче, где результат 1.08, что даже хуже, чем у алгоритма jsonPack, который работает не на таком низком уровне.

Следующий алгоритм – LZFSSE, разработанный компанией Apple и рекомендованный к применению на собственных устройствах (рис.5). Алгоритм показывает хороший результат коэффициента сжатия в 0.42, что даже лучше, чем результат , представленный на рис. 4 для алгоритма LZ4.

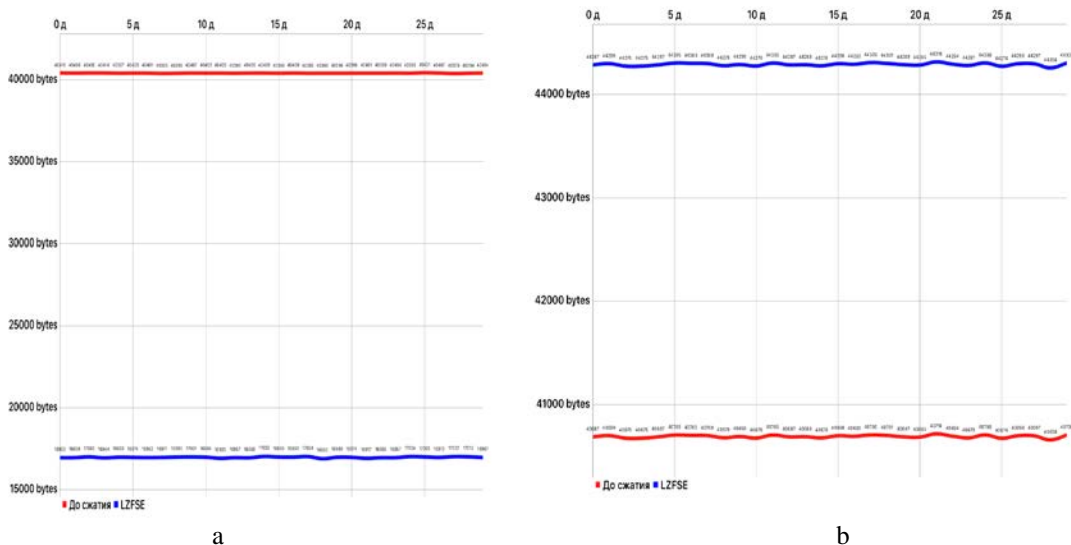


Рис.5. Моделирование отправки на основе метода сжатия LZFSSE, а) отправка одним пакетом, б) последовательная отправка 30 отдельных сообщений

Коэффициент сжатия при применении алгоритма LZFSSE с последовательной отправкой показывает

результат сжатия в 1.08, результат аналогичен алгоритму LZ4.

Далее на очереди алгоритм, разработанный Игорем Павловым – LZMA (Lempel-Ziv-Markov chain-Algorithm). Данный алгоритм является очень популярным и известен как основной алгоритм архиватора 7-zip (рис.6).

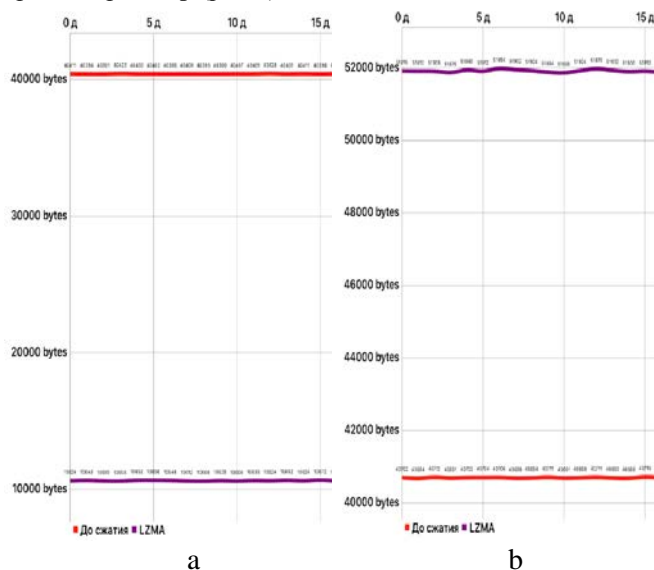


Рис. 6. Моделирование отправки сообщений на основе метода сжатия LZMA, а) отправка одним пакетом, б) последовательная отправка 30 отдельных сообщений

В первом случае наблюдается очень высокий, по сравнению с предыдущими алгоритмами, коэффициент сжатия – 0.26, т.е. алгоритм показывает наилучший результат для отправки в одном сообщении 30 информационных сообщений. Однако результат для последовательной отправки составляет 1.27, что является худшим результатом, из описанных выше, сообщение увеличивается в размере на 27%.

Далее представлен ряд deflate и inflate методов: zlib (RFC-1951), gzip (RFC-1952), zip (RFC-1950). На рисунке 7 показаны результаты их испытания.

Исследуемые алгоритмы показывают практически одинаковые результат сжатия, с коэффициентом в 0.28. Это очень высокий показатель уровня сжатия, что говорит об эффективности применения в случае отправки одним сообщением. Из рис. 7 В) видим, что алгоритмы, основанные на методе deflate, показывают некоторое сжатие даже в случае последовательной отправки сообщений. Так алгоритм zlib показал коэффициент сжатия в 0.83. gzip - в 0.97, zip - в 0.88. Это говорит о том, что данные алгоритмы можно использовать как для систем общественного зондирования, которые требуют постоянного обновления информации, так и для систем, этого не требующих. Таким образом, лучший результат среди этих алгоритмов показал zip, который дал лучшие коэффициенты сжатия как для единой отправки, так и для последовательной отправки.

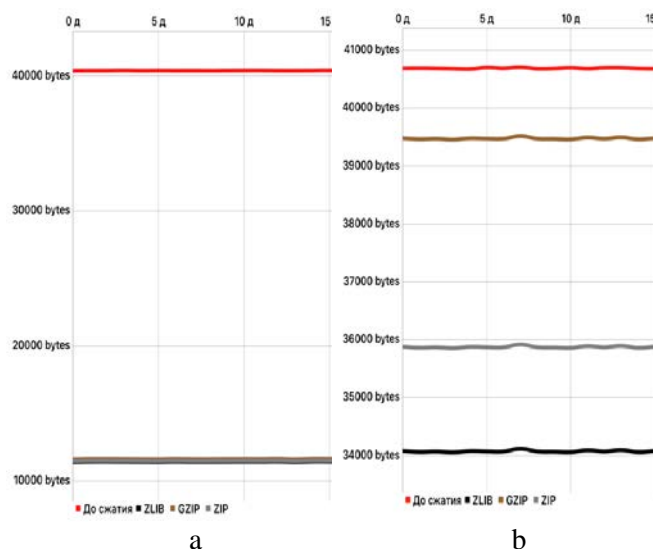


Рис. 7. Моделирование отправки сообщений на основе методов сжатия Zlib, gzip, zip, а) отправка одним пакетом, б) последовательная отправка 30 отдельных сообщений

Подводя итоги, рассмотрим результирующий рисунок, на котором изображены все исследованные алгоритмы для отправки одного пакета сообщений в течение дня.

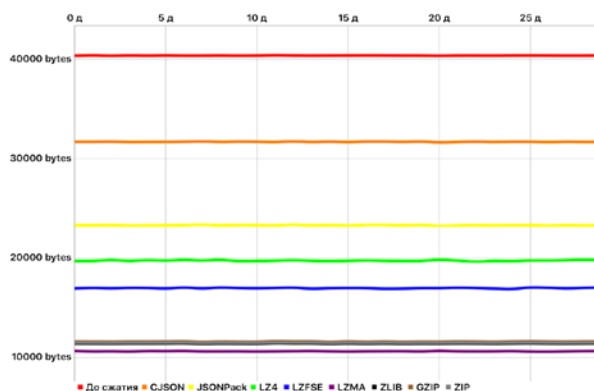


Рис. 8. Моделирование отправки сообщений на основе всех рассмотренных алгоритмов, отправка сообщения одним пакетом

Из графика видно, что наиболее эффективным алгоритмом является lzma, в случае отправки одним сообщением.

Рассмотрим также случай для последовательной отправки, сравнивая все алгоритмы (рис. 9).

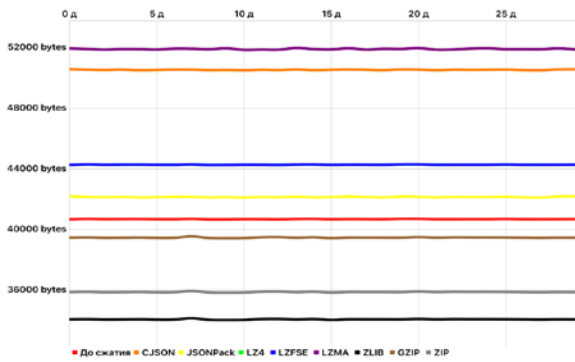


Рис. 9. Моделирование отправки сообщений на основе всех рассмотренных алгоритмов, последовательная отправка отдельных сообщений

Из графика видно, что большинство алгоритмов являются эффективными только на больших объемах данных, содержащих несколько сообщений сразу. Тем не менее, алгоритмы zlib, zip, gzip показали эффективность также и в этом случае.

Рассмотрим возможность применения алгоритма Cjson в качестве предварительного сжатия исходного json сообщения, в случае с одновременной отправкой, так и с последовательной отправкой сообщений (рис. 10).

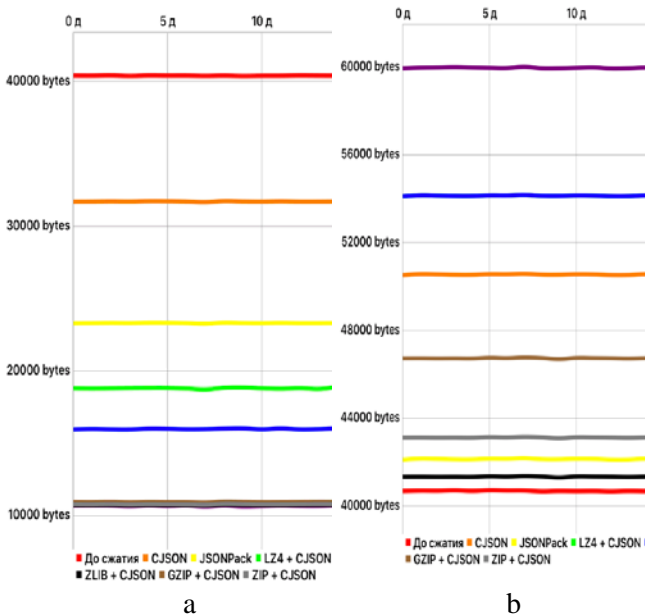


Рис. 10. Моделирование отправки сообщений на основе комбинации Cjson + все остальные алгоритмы, а) отправка одним пакетом, б) последовательный режим

При отправке сообщений с использованием всех алгоритмов одновременно, предварительное сжатие алгоритмом cjson позволило повысить эффективность некоторых алгоритмов, тем не менее lzma показал результат идентичный тому, который был без предварительного сжатия. Остальные же алгоритмы, позволили добиться еще большего сжатия по сравнению с исходным сообщением.

Но применение такого подхода может нивелироваться большой вычислительной сложностью, с которой приходится считаться в данном случае, что может плохо сказаться на длительности работы аккумулятора устройства пользователя на дальней дистанции.

В случае последовательной отправки сообщений получаем случай, когда предварительное сжатие только усугубило ситуацию. Ни один из алгоритмов не показал эффективного результата.

Рассмотрим в качестве предварительного сжатия алгоритм jsonPack, результаты на рис. 11.

В случае предварительного сжатия алгоритмом JSONPack для отправки одним сообщением эксперимент дал результаты хуже, чем для использования в качестве предварительного сжатия алгоритма Cjson.

Однако, в случае с последовательной отправкой три алгоритма показывают положительную динамику, а именно zlib, zip, gzip. Наибольшую эффективность показал zlib. Однако результат все равно чуть хуже, чем если бы мы не использовали предварительное сжатие вовсе.

Проведя ряд дополнительных испытаний было выяснено, что алгоритм lzma показывает лучший результат, если данные достаточно большие, в нашем случае хотя бы 20 сообщений в одном, что также очень важно.

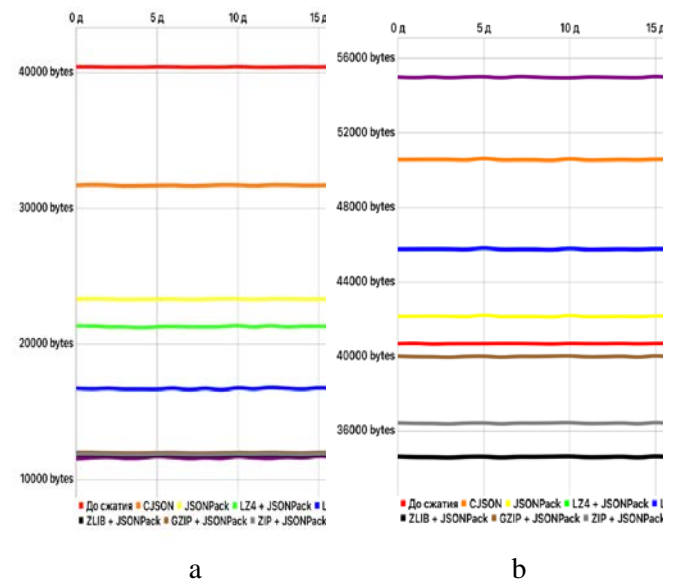


Рис. 11. Моделирование отправки сообщений на основе предварительного сжатия алгоритм jsonPack, а) отправка одним пакетом, б) последовательная отправка 30 отдельных сообщений

Рассмотрим более детально разницу между сжатием алгоритмов, основанных на использовании метода deflate.

На рисунке 12 продемонстрированы графики алгоритмов zip, gzip и zip.

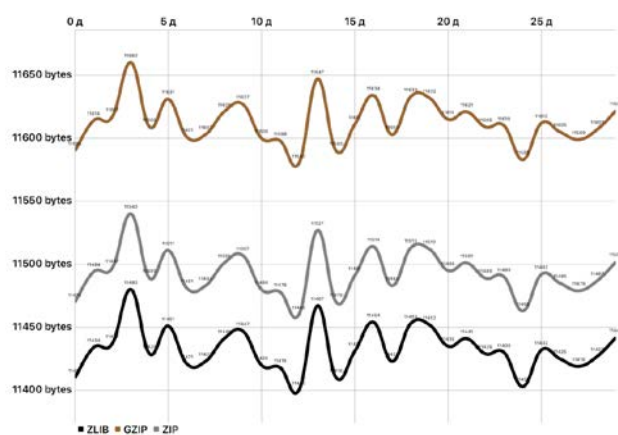


Рис. 12 – Моделирование отправки сообщений на основе deflate алгоритмов одним сообщением.

Как видно из графиков, алгоритмы ведут себя в целом одинаково и вся разница достаточно мала. Наиболее эффективный алгоритм является zlib, он показывает самый высокий коэффициент сжатия информации, тем не менее, разница совершенно незначительна.

В данном случае данные являются монотонными – это массив информации с однотипными сообщениями, именно поэтому алгоритмы показывают такой большой коэффициент сжатия. Стоит отметить, что чем больше данных будет накапливаться, тем более высокий коэффициент сжатия в итоге мы можем получить.

Однако, если сеть общественного зондирования является real-time сетью, т.е. требуется частое обновление информации во время функционирования, то устройство может не успевать накапливать нужное количество пакетов информации.

На рис.13 показан пример, когда данные представляют собой последовательно отправляемые сообщения.

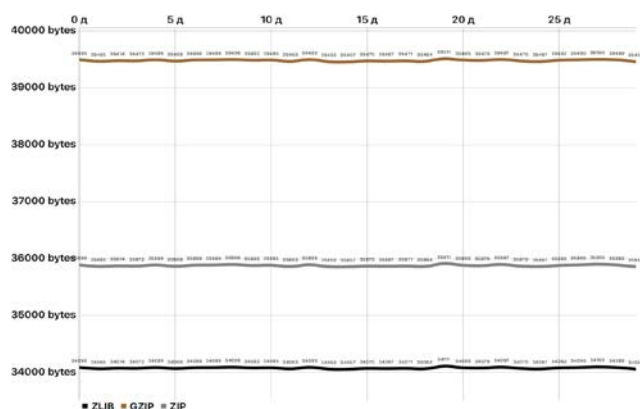


Рис. 13. Алгоритмы deflate, последовательная отправка

Очевидно, что отсутствие большого количества однородной информации привело к существенному снижению эффективности работы алгоритмов. Хотя и разница между ними сохранилась, по-прежнему наиболее эффективный алгоритм является zlib.

Тем не менее, глобальной разницы между, особенно между zlib и zip не наблюдается – можно использовать любой из них.

Также нужно помнить, что в отличие от алгоритмов, направленных на сжатие с помощью json, алгоритмы, основанные на сжатии битов информации – более сложные и требуют специальной настройки сервера для их поддержания.

ЗАКЛЮЧЕНИЕ

В статье представлены результаты работы программы, с помощью которой можно эмулировать функциональность Public Sensing сетей. Выполнено моделирование передачи сообщений на сервер, где для экономии трафика пользователя и снижения нагрузки на сервера используются следующие алгоритмы сжатия: cJSON, jsonPack, lz4, lzfs, lzma, zlib, gzip, zip, а также предварительное сжатие с помощью cJSON и с помощью jsonPack.

В результате проведенных исследований было выяснено, что наибольшую эффективность для случая, когда система общественного зондирования работает не в реальном времени и собирает данные одновременно один раз в день – алгоритм lzma показывает результат в 0.26 для коэффициента сжатия, что является очень хорошим результатом, с близкий к нему результат в 0.28 дают zlib, gzip, zip.

В случае работы алгоритмов на последовательной отправке сообщений имеем другую картину, алгоритм lzma показал свою непригодность с коэффициентом в 1.27, иными словами – он увеличил количество трафика, посылаемого на сервер. Однако алгоритмы zlib, gzip, zip показали свою эффективность и в данном случае, самый лучший результат у алгоритма zlib, а именно коэффициент сжатия 0.83.

Подход с применением предварительного сжатия показал следующие результаты: при использовании в качестве предварительного сжатия алгоритмов cJSON и JSONPack практически не принесло изменений.

В итоге можно сказать, что самые лучшие результаты для отправки одним сообщением – это применение алгоритма lzma без предварительного сжатия, но для большого количества сообщений, в нашем примере – от 20. Если сообщений меньше, наибольшую эффективность демонстрирует алгоритм zlib.

В случае отправки группы сообщений – лучший результат показывает алгоритм zlib, при этом любое предварительное сжатие не приносит положительных результатов.

СПИСОК ЛИТЕРАТУРЫ/REFERENCES

1. **Philipp D., Durr F., Rothermel K.** A Sensor Network Abstraction for Flexible Public Sensing Systems *IEEE Xplore*: 15 November 2011. DOI: 10.1109/MASS.2011.52

2. **Mayer R., Gupta H, Saurez E., Ramachandran U.** The Fog Makes Sense: Enabling Social Sensing Services with Limited Internet Connectivity, *SocialSens'17: Proceedings of the 2nd International Workshop on Social Sensing* April 2017 p 61–66 DOI:10.1145/3055601.3055614.
3. **Ogundile O.O., Alfa A.S.** A Survey on an Energy-Efficient and Energy-Balanced Routing Protocol for Wireless Sensor Networks. *Sensors*. 2017;17:1084. DOI: 10.3390/s17051084.
4. **Macho J.B., Montón L.G., Rodriguez R.C.** Context-and Template-Based Compression for Efficient Management of Data Models in Resource-Constrained Systems. *Sensors (Basel)*. 2017 Aug; vol. 17(8). DOI:10.3390/s17081755
5. **Pushpalatha S., Shivaprakasha K. S.** Energy-Efficient Communication Using Data Aggregation and Data Compression Techniques in Wireless Sensor Networks: A Survey. *Springer Nature Singapore Pte Ltd. 2020. Advances in Communication, Signal Processing, VLSI, and Embedded Systems* pp 161-179. DOI :10.1007/978-981-15-0626-0_14
6. **Sadler C.M., Martonosi M.** Data compression algorithms for energy-constrained devices in delay tolerant networks. *Proc. ACM Int'l Conf. Embedded Networked Sensor Systems*, p. 265–278, 2006. DOI: 10.1145/1182807.1182834
7. **Sheltamia T., Musaddiq M., Shakshukib E.** Data compression techniques in Wireless Sensor Networks. Elsevier. *Future Generation Computer Systems*. Volume 64, November 2016, Pages 151–162. DOI: 10.1016/j.future.2016.01.015.
8. **Marcelloni F., Vecchio M.** A simple algorithm for data compression in wireless sensor networks. *IEEE Commun Lett* 2008; 12(6): 411–413.
9. **Incebacak D, Zilan R, Tavli B,** et al. Optimal data compression for lifetime maximization in wireless sensor networks operating in stealth mode. *Elsevier Ad Hoc Netw* 2015; 24: 134–147.
10. **Sunyong K. , Chiwoo C. , Kyung-Joon P. Hyuk L.** Increasing network lifetime using data compression in wireless sensor networks with energy harvesting. *International Journal of Distributed Sensor Networks* 2017, Vol. 13(1) The Author(s) 2017 DOI: 10.1177/1550147716689682

СВЕДЕНИЯ ОБ АВТОРАХ:

Приходько Татьяна Александровна, кандидат технических наук, доцент кафедры вычислительных технологий факультета компьютерных технологий и прикладной математики.
Кубанский государственный университет г. Краснодар, Россия
(350040. Россия, г. Краснодар, ул. Ставропольская, д.149.)
pr.tatyana@gmail.com
(+7)967-314-55-68
ORCID: <https://orcid.org/0000-0001-5137-2064>

Воробьев Семен Александрович, магистрант кафедры вычислительных технологий факультета компьютерных технологий и прикладной математики, Кубанский государственный университет г. Краснодар, Россия
(350040. Россия, г. Краснодар, ул. Ставропольская, д.149.) ORCID: ORCID: <https://orcid.org/0000-0001-8925-4709>

babaev.danil.vs@gmail.com

Все авторы прочитали и одобрили окончательный вариант рукописи.

Information about the Authors:

Tatyana A. Prikhodko, Candidate of Science (Techniques), associate Professor of the Department of computing technologies, Faculty of Computer technologies and Applied Mathematics. Kuban state University Krasnodar, Russia
(350040. 149, Stavropol str., Krasnodar, Russia.)
pr.tatyana@gmail.com
(+7)967-314-55-68
ORCID: <https://orcid.org/0000-0001-5137-2064>

Semyon A. Vorobyov, undergraduate student of the Department of computing technologies, Faculty of Computer technologies and Applied Mathematics, Kuban state University Krasnodar, Russia
(350040. 149, Stavropol str., Krasnodar, Russia.)
ORCID: <https://orcid.org/0000-0001-8925-4709>
babaev.danil.vs@gmail.com

All authors read and approved the final version manuscripts.

Programming model for investigation of Energy-Efficient Compression algorithms in public sensing systems

Vorobyov S.A., Prikhodko T.A.

Abstract - The fact that people are equipped with smartphones and smartphones are equipped with a variety of different sensors and various communication protocols allows them to be used as ad-hoc network nodes for solving public sensing problems. Public sensing (PS) technology allows the information exchange in society on demand. However, the energy resources of mobile devices are significantly exhausting participating in such networks, so the task of energy-efficient transmission of information remains relevant and actual. One of the ways to save energy resources of mobile devices is to use compression algorithms for information transition. The purpose of this work is to develop a program for testing different lossless compression methods or their combinations to find out which of possible variant can save more energy being used in public sensing mode. The developed program simulate the mobile nodes transmitting information in Public Sensing mode. As a result, authors evaluated the feasibility and effectiveness of various compression methods.

Keywords - self-organizing networks, public sensing systems, energy-efficient compression algorithms.

References

1. Philipp D., Durr F., Rothermel K. A Sensor Network Abstraction for Flexible Public Sensing Systems IEEE Xplore: 15 November 2011. DOI: 10.1109/MASS.2011.52
2. Mayer R., Gupta H, Saurez E., Ramachandran U. The Fog Makes Sense: Enabling Social Sensing Services with Limited Internet Connectivity, SocialSens'17: Proceedings of the 2nd International Workshop on Social Sensing April 2017 p 61–66 DOI:10.1145/3055601.3055614.
3. Ogundile O.O., Alfa A.S. A Survey on an Energy-Efficient and Energy-Balanced Routing Protocol for Wireless Sensor Networks. Sensors. 2017;17:1084. DOI: 10.3390/s17051084.
4. Macho J.B., Montón L.G., Rodriguez R.C. Context-and Template-Based Compression for Efficient Management of Data Models in Resource-Constrained Systems. Sensors (Basel). 2017 Aug; vol. 17(8). DOI:10.3390/s17081755
5. Pushpalatha S., Shivaprakasha K. S. Energy-Efficient Communication Using Data Aggregation and Data Compression Techniques in Wireless Sensor Networks: A Survey. Springer Nature Singapore Pte Ltd. 2020. Advances in Communication, Signal Processing, VLSI, and Embedded Systems pp 161-179. DOI :10.1007/978-981-15-0626-0_14
6. Sadler C.M., Martonosi M. Data compression algorithms for energy-constrained devices in delay tolerant networks. Proc. ACM Int'l Conf. Embedded Networked Sensor Systems, p. 265–278, 2006. DOI: 10.1145/1182807.1182834
7. Sheltamia T., Musaddiqa M., Shakshukibc E. Data compression techniques in Wireless Sensor Networks. Elsevier. Future Generation Computer Systems. Volume 64, November 2016, Pages 151 162. DOI: 10.1016/j.future.2016.01.015.
8. Marcelloni F., Vecchio M. A simple algorithm for data compression in wireless sensor networks. IEEE Commun Lett 2008; 12(6): 411–413.
9. Incebacak D, Zilan R, Tavli B, et al. Optimal data compression for lifetime maximization in wireless sensor networks operating in stealth mode. Elsevier Ad Hoc Netw 2015; 24: 134–147.
10. Sunyong K. , Chiwoo C. , Kyung-Joon P. Hyuk L. Increasing network lifetime using data compression in wireless sensor networks with energy harvesting. International Journal of Distributed Sensor Networks 2017, Vol. 13(1) The Author(s) 2017 DOI: 10.1177/1550147716689682