

Санкт-Петербургский государственный университет

Кафедра Системного программирования

Свитков Сергей Андреевич

Телеметрия роботов на базе контроллера
TRIK
Серверная часть

Курсовая работа

Научный руководитель:
старший преподаватель Литвинов Ю. В.

Санкт-Петербург
2016

Оглавление

Введение	3
1. Обзор	5
1.1. Обзор существующих решений	5
1.2. Итог	6
2. Реализация	7
2.1. UDP + TCP	7
2.2. Программное решение	7
3. Исследования	8
3.1. Сравнение времени задержки	9
3.2. Сравнение визуализации	9
3.3. Другие возможные исследования	10
3.4. Итог	12
4. Рефакторинг	13
Заключение	14
Список литературы	15

Введение

В последние годы область робототехники развивается очень быстро, появляется множество робототехнических конструкторов.

Одним из них является TRIK [7] — кибернетический конструктор с центральным процессором на базе ARM и операционной системой на базе ядра Linux. Модели роботов TRIK применяются как для обучения школьников, так и для серьезных соревнований, например, WRO (World Robot Olympiad), проходившей в Сочи (2014г); Ежегодный фестиваль робототехники Робофинист (2015г). Роботы моделей TRIK имеют ряд датчиков, таких как гироскоп, акселерометр, датчик заряда батареи. Так же робот предоставляет сетевой интерфейс Wi-Fi.

Но сам по себе робот, без программ на нем, представляет малый интерес. Следует сказать, что исполнение программ на роботе происходит в реальном времени, поэтому очень важной является возможность отслеживать показания датчиков робота во время работы приложений.

Кроме того, необходимо учитывать, что соединение Wi-Fi с роботом нестабильно, и его легко перегрузить, поэтому алгоритм обмена сообщениями между роботом и ПК пользователя¹ должен обеспечивать минимальную задержку между снятием показаний с датчиков робота и их визуализацией в клиентском приложении, но при этом обеспечивать минимальную потерю данных.

Поэтому задача разработки эффективного алгоритма для телеметрии данных, получаемых с датчиков робота, является актуальной и важной.

Постановка задачи

Поскольку задача телеметрии роботов модели TRIK была решена ранее Матвеем Брыксиным в рамках его курсовой работы, то целью данной работы является разработка нового алгоритма передачи данных с робота на ПК и последующее сравнение нового алгоритма с су-

¹В данном случае речь идет о клиентском приложении, запущенном на персональном компьютере пользователя. В дальнейшем — клиент

существующим, а так же рефакторинг уже написанного кода.

Для достижения данной цели были поставлены следующие задачи:

1. добавить поддержку передачи данных с использованием протокола UDP
2. провести сравнение эффективности существовавшего ранее решения и решения, полученного в ходе данной работы
3. провести рефакторинг кода

1. Обзор

1.1. Обзор существующих решений

Был проведен обзор существующих решений задачи телеметрии роботов в различных условиях.

A Comparison of Lightweight Communication Protocols in Robotic Applications [1]

В данной работе рассказывается о сравнении двух легковесных протоколов передачи данных с робота : CoAP [2] и MQTT-SN [4].

Статья, написанная по данной работе, была полезна тем, что в ней описан подход к сравнению двух протоколов.

A Data-Rate Aware Telemetry Scheduler [5]

В данной статье описан подход к реализации телеметрии в условиях очень плохого соединения с роботом.

Для решения задачи телеметрии в таких условиях автор статьи предлагает использовать алгоритм приоритизации сообщений в зависимости от их важности.

Но данный подход не может быть использован, поскольку это не соответствует требованиям системы реального времени.

Телеметрия моделей ТРИК [8]

Данная работа рассказывает о реализации телеметрии моделей ТРИК. Передача данных с робота осуществляется по протоколу ТСР с заранее объявленным интервалом.

Решение автора было взято в качестве основы с целью улучшить его.

1.2. Итог

Сформулирована гипотеза о том, что передача части данных по UDP уменьшит задержку при визуализации данных и при этом практически не повлияет на качество визуализации.

В связи с этим было принято решение переиспользовать существующий код, внося в него следующие изменения:

1. добавить возможность передачи данных с датчиков робота с использованием протокола UDP
2. разработать алгоритм передачи данных с использованием комбинации протоколов TCP и UDP
3. провести рефакторинг

2. Реализация

Основным моментом в реализации решения задачи являются использование UDP в комбинации с TCP.

2.1. UDP + TCP

UDP, в отличие от TCP, не гарантирует доставку всех пакетов, передаваемых по сети с одного устройства на другое. В связи с этим фактом UDP не может быть использован для передачи важных сообщений — в данном случае такими сообщениями будем считать сигналы о старте телеметрии и о ее завершении. Однако такие сообщения передаются редко в сравнении с частотой отправки основного типа сообщений, отправляемых с робота на клиент — сообщений, содержащих в себе показания датчиков робота.

Так как сообщение с показаниями датчиков отправляется каждые 10 мс, тот факт, что при передаче может быть утеряно некоторое количество пакетов, не является критичным. Следует отметить, что потеря пакетов, как показали исследования, случается довольно редко и в небольших количествах.

Поэтому UDP можно считать оптимальным для оптимизации протокола передачи данных, но, как было отмечено выше, использование только UDP может привести к потере важного сигнала, что может плохо отразиться на работе системы.

2.2. Программное решение

Был написан класс для реализации обмена сообщениями между сервером и клиентом по протоколу UDP. Кроме того, был добавлен интерфейс, в который вынесены общие методы для классов, реализующих UDP и TCP.

Итоговый вариант архитектуры [3]: 1

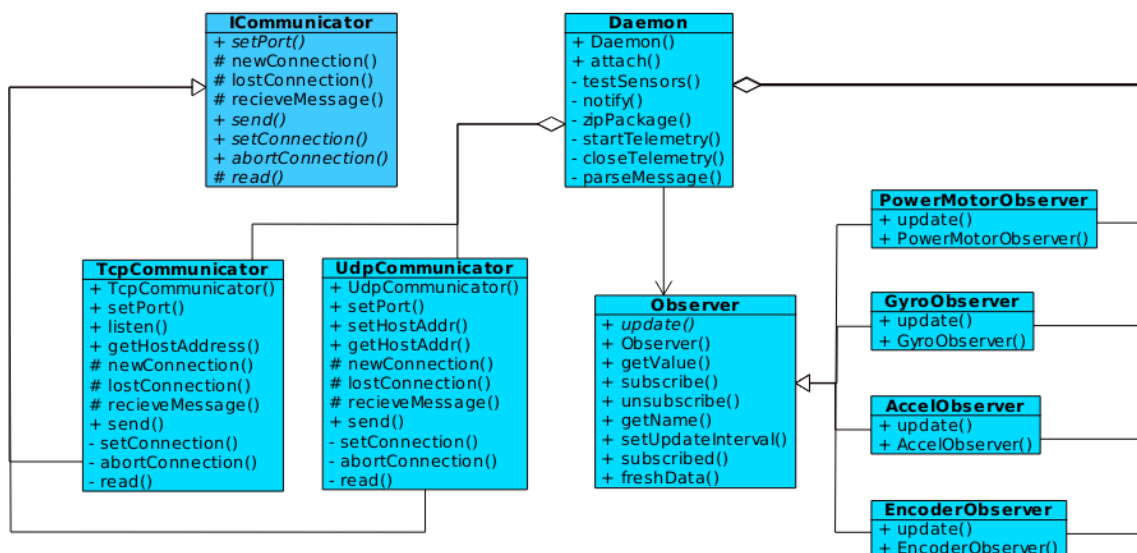


Рис. 1: Архитектура

3. Исследования

С целью проверки гипотезы, выдвинутой в начале данной работы, был проведен ряд исследований с целью сравнить исходный вариант протокола с разработанным в рамках данной работы.

Следует отметить, что радиус покрытия Wi-Fi сети робота — около 20 метров², поэтому было принято решение проводить исследования в условиях удаленности ПК с запущенным клиентом от робота на расстояние около 10 метров.

Критерии, по которым проводилось сравнение протоколов:

1. величина задержки между моментом снятия показаний датчиков робота и моментом их визуализации на клиенте
2. количество теряемых пакетов
3. то, насколько негативно использование протокола влияет на визуализацию

²Выявлено опытным путем

Таблица 1: Результаты замеров

Используемый протокол	Среднее время, мс	Среднее количество потерянных пакетов, %	расстояние до робота
TCP	51100	0	10
TCP + UDP	44600	0	10
TCP	59200	0	15
TCP + UDP	43500	7	15

3.1. Сравнение времени задержки

Было решено сравнить количество времени³, за которое происходит считывание, отправка, парсинг и визуализация 1000 пакетов для протоколов TCP и комбинации TCP + UDP. Результаты представлены в таблице 1

Как можно заметить из результатов таблицы, средняя задержка для одного пакета на нормальном расстоянии (10 м) при использовании TCP составляет около 51 мс, а средняя задержка для одного пакета при использовании предложенного в статье алгоритма, соответственно, около 44 мс. Из данных результатов можно сделать вывод о том, что задержка при использовании TCP + UDP в среднем уменьшается на 14%.

Задержка же при расстоянии 15 м для протокола TCP становится больше, поскольку сигнал становится более слабым и TCP приходится досылать те пакеты, которые не были досланы до пользователя.

3.2. Сравнение визуализации

Поскольку потеря пакетов в нормальных условиях соединения практически не происходит, использование UDP в качестве протокола для передачи данных датчиков практически не окажет влияния на визуализацию.

Для доказательства этого приводятся графики показаний акселеро-

³В таблице — среднее время

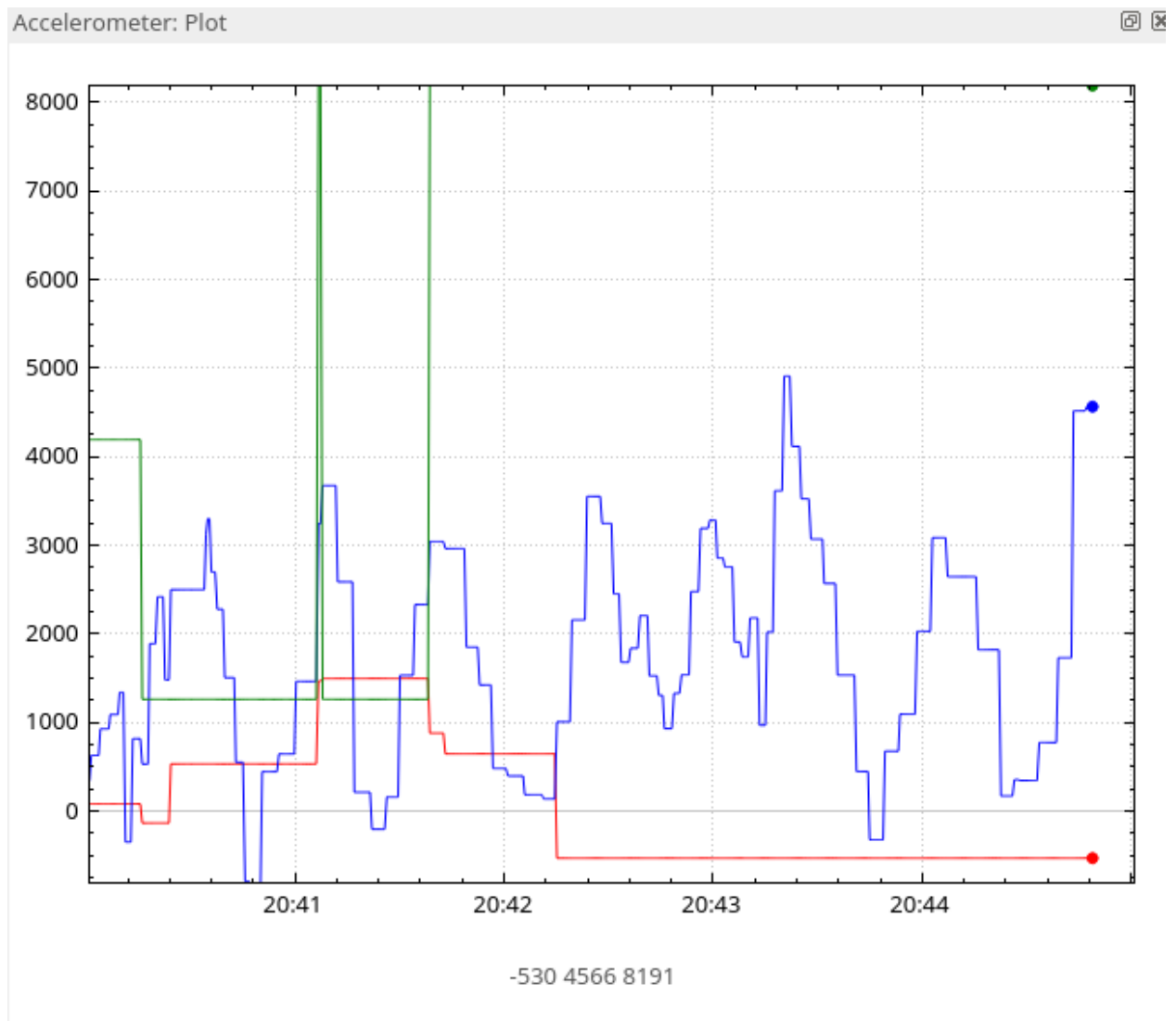


Рис. 2: Фрагмент графиков, TCP

метра 2 3

3.3. Другие возможные исследования

Следует отметить, что были предложены и другие варианты протоколов для сравнения с TCP + UDP, например:

1. протокол, использующий TCP, но при этом не посылающий каждый пятый пакет с данными
2. протокол, использующий TCP, но посылающий пакеты с данными в два раза реже

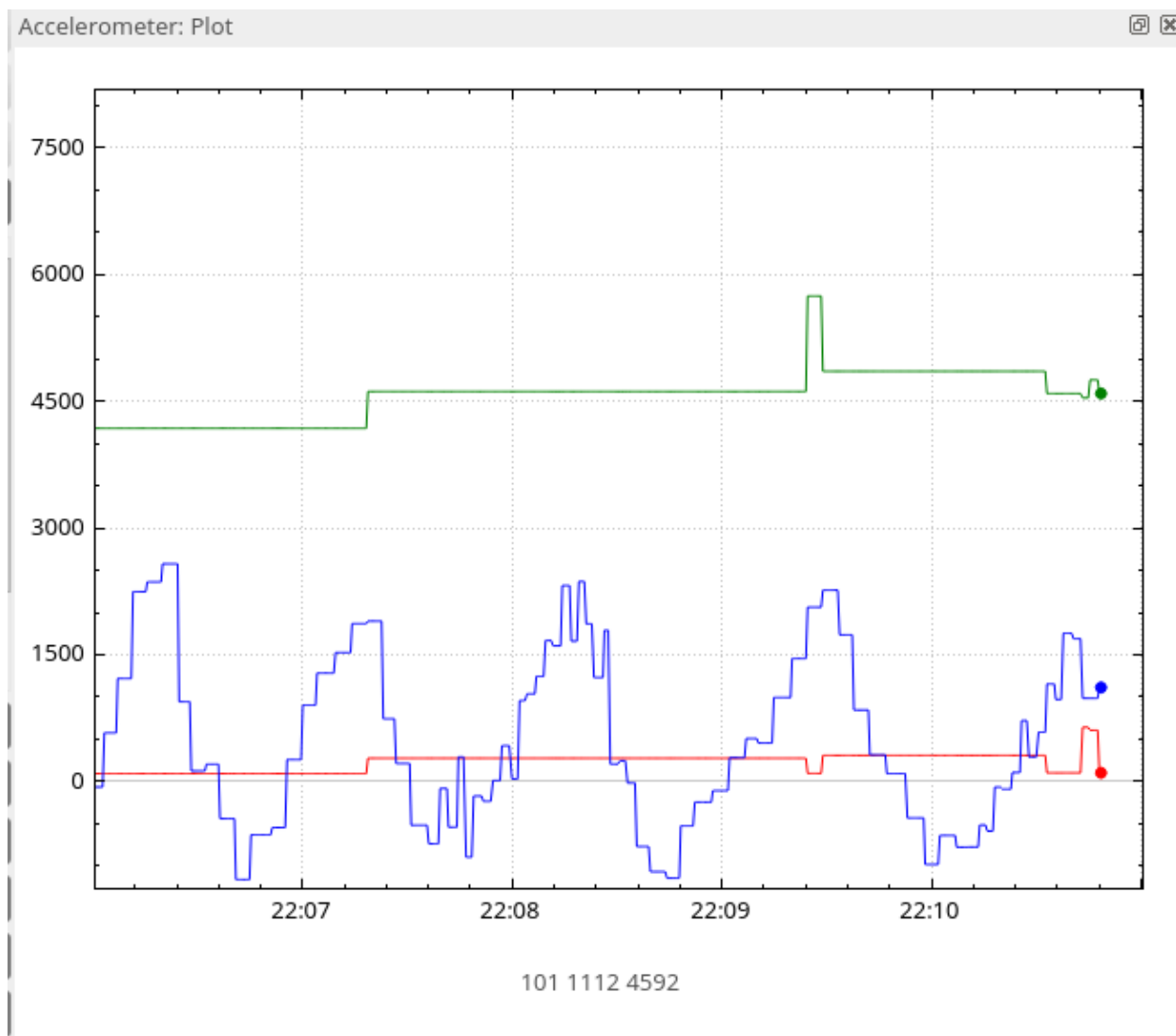


Рис. 3: Фрагмент графиков, TCP + UDP

Данные варианты не были рассмотрены, поскольку никак не могут повлиять на задержку и лишь негативно сказываются на визуализации данных.

3.4. Итог

По результатам сравнения можно утверждать, что разработанное в рамках данной курсовой работы решение является более эффективным чем то, что было предложено ранее.

4. Рефакторинг

Код, на основе которого проводилась данная работа, обладал рядом недостатков, которые были исправлены. Исправленные недостатки:

1. объявление нескольких классов в одном файле
2. cyclic dependency
3. неуместное совместное использование forward declaration и include в .h файлах
4. использование using namespace в .h файлах

Заключение

В ходе исследований, проведенных в рамках данной работы было выявлено, что предложенный протокол передачи данных, использующий TCP + UDP, является более эффективным в рамках конкретной задачи, поэтому результатом данной работы можно считать улучшение алгоритма обмена сообщениями между роботом и клиентом, выражающееся в виде уменьшения задержки между отправкой пакета с данными и визуализацией данных на 14% и рефакторинг кода.

По результатам работы был сделан доклад на конференции SEIM-2016.

Дальнейшие перспективы

В дальнейшем планируется закончить оформление кода, полученного в ходе данной работы, в соответствии со стайлгайдом [6] TRIK.

Список литературы

- [1] A Comparison of Lightweight Communication Protocols in Robotic Applications : Rep. / Universiti Teknologi MARA UiTM Shah Alam ; Executor: Muhammad Harith Amaran et al. : 2015.
- [2] CoAP.— URL: https://en.wikipedia.org/wiki/Constrained_Application_Protocol (online; accessed: 10.04.2016).
- [3] Erich Gamma Richard Helm Ralph Johnson John Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software.— Addison Wesley, 1994.— ISBN: 860-1419047741.
- [4] MQTT.— URL: <https://en.wikipedia.org/wiki/MQTT> (online; accessed: 10.04.2016).
- [5] A Data-Rate Aware Telemetry Scheduler : Rep. : CMU-RI-TR-01-12 / Robotics Institute ; Executor: Punitha Manavalan, Michael D Wagner . — Pittsburgh, PA : 2001. — May.
- [6] TRIK styleguide.— URL: <https://github.com/qreal/qreal/wiki/Стайлгайд> (online; accessed: 10.04.2016).
- [7] TRIK website.— URL: <http://www.trikset.com/> (online; accessed: 10.04.2016).
- [8] Телеметрия моделей ТРИК : Rep. / СПбГУ ; Executor: Брыксин Матвей Александрович : 2014. — Май.