

## ПОДХОДЫ К МОДЕЛИРОВАНИЮ КАНАЛОВ ПЕРЕДАЧИ ДАННЫХ МЕЖДУ АВТОНОМНЫМИ РОБОТОТЕХНИЧЕСКИМИ КОМПЛЕКСАМИ

И.О. Шальнев, С.В. Кулешов (Санкт-Петербург)

Объектом моделирования являлся коммуникационная среда между автономными робототехническими комплексами (РТК), в том числе, между беспилотными летательными аппаратами (БПЛА), организованная с использованием лазерной оптической системы связи (коммуникационного конуса). Коммуникационный конус формируется оптико-электронным устройством ведущего РТК для обеспечения локального канала связи с ведомыми РТК. В связи с техническим ограничением ведомые РТК не имеют возможности прямой передачи сообщения между собой, а если это необходимо, то передача должна осуществляться через ведущего. Для программной реализации моделей использовались инструменты ROS [1] и Gazebo [2].

Реализация модели коммуникационного конуса предполагает моделирование следующих событий: потеря канала связи, задержка и потеря сетевых пакетов. Потеря канала связи происходит в момент, когда ведомый РТК выходит за геометрические пределы коммуникационного конуса. Как задержка, так и потеря сетевых пакетов может происходить по ряду обстоятельств, например, окружающая среда предполагает наличие внешних проблем для связи (погодные условия). В случае если предполагается использование протокола с гарантированной доставкой данных (TCP), то ошибки, возникающие в процессе передачи сетевых пакетов, накладывают дополнительную задержку в связи с повторной отправкой испорченных пакетов. Если используется протокол без гарантированной доставки (UDP), то сетевые пакеты с ошибками попросту отбрасываются, а это эквивалентно потере сетевых пакетов.

Для моделирования указанных выше свойств канала связи возможны несколько вариантов реализации в ROS. Рассмотрим пример, показанный на рис. 1. В рамках такой моделирующей сетевой структуры, как и в действительности, каждый отдельный РТК запущен на отдельной ОС (на отдельном физическом или виртуальном вычислительном устройстве, например, в контейнере). Пунктирной линией показано соединение с ROS Master'ом, который регистрирует все Publisher'ы и Subscriber'ы и осуществляет соединение публикуемых узлов с узлами подписчиками. Сплошной линией показано соединение различных РТК после того, как ROS Master их связал. Тут важно понимать, что без работы ROS master не будет возможности связать узлы друг с другом. Соединение РТК происходит вследствие того, что и «ведущий» (master), и «ведомый» (slave) РТК являются источниками и потребителями информации, осуществляющие передачу посредством механизма ROS topic.

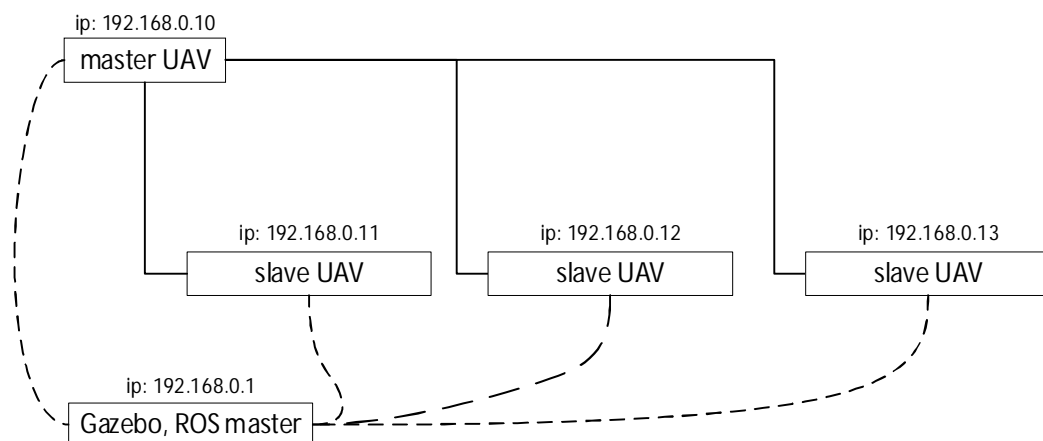


Рис. 1. Вариант сетевой структуры моделирующей среды

Несмотря на то, что ROS является удобным инструментарием для разработки робототехнических систем, он имеет ряд недостатков, которые не подходят для группового взаимодействия. На рис. 1 можно увидеть, что в рамках моделирования ROS master может быть загружен на отдельном вычислительном устройстве. В реальной системе вопрос о том, где должна располагаться эта связывающая сущность, непосредственно влияет на надежность системы. Так, если она будет располагаться на одном из ведущих РТК, то его устранение из группы повлечет неработоспособность всей группы РТК.

Коммуникационная оптическая система связи имеет два типа приемника и передатчика, названные исходя из общепринятой практики: master и slave. Передатчики типа master формируют коммуникационный конус при помощи механической развертки лазерного луча.

В связи с этим перед нами стояла задача создания такой системы связи, которая позволила бы устранить этот недостаток и повысить отказоустойчивость всей группы. Такая система связи должна предполагать работу ROS master'a на каждом из РТК.

### Моделирование коммуникационного конуса при помощи утилиты `ifconfig`

Для моделирования потери соединения было предложено инициировать разрыв сетевого соединения в момент, когда ведомый выйдет из зоны покрытия или будет на его границе.

Для осуществления разъединения можно использовать отключение сетевого интерфейса посредством команды «`ifconfig`», что повлечет отключение всех сокетов и очистки TCP/IP буфера. В таком случае все сообщения будут утеряны, а не задержаны. Такая ситуация соответствует ручному разъединению в противовес логическому и не соответствует требованиям к моделирующему комплексу.

Решением этой проблемы может быть создание еще одного контейнера, интерфейс которого можно отключить, не опасаясь за работу сокетов на вычислительном устройстве, имитирующем РТК (рис. 2). Таким образом «посредник» будет имитацией конуса, который в момент вылета за пределы конуса будет отключать сетевой интерфейс. Однако в таком случае не будет возможности понять, в какой момент включать сетевой интерфейс, так как решение о нахождении РТК в конусе основано на информации, которая приходит по сети от ведущего и ведомого.

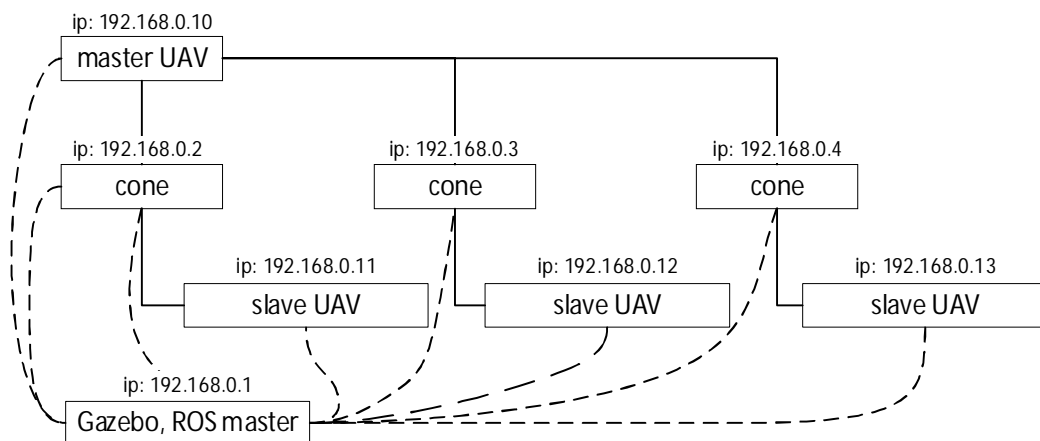


Рис. 2. Возможная сетевая структура с добавлением контейнера, моделирующим коммуникационный конус

### Моделирование конуса при помощи утилиты `tc`

Еще одним из возможных решений основано на конфигурации, показанной на рис. 1, и заключается в использовании системной команды `tc` (от англ. transport control). Она позволяет определять задержку и потерю пакетов в сетевом интерфейсе. Так как

логическое разъединение есть отсутствие возможности принимать и отправлять пакеты, то отсутствие логического соединения можно имитировать путем фильтрации пакетов. В таком случае на каждом из ведомых РТК должен работать узел, который в момент выхода из конуса начинает фильтровать принимаемые и отправляемые сетевые пакеты, кроме тех, что необходимы для определения границ конуса. А также в соответствии с конфигурацией моделирования устанавливать задержку и потерю пакетов.

Преимуществом такого подхода заключается в том, что мы рассматриваем всю систему как черный ящик, максимально имитируя реальную коммуникационную среду. Недостатком подхода является то, что фильтрация сообщений во время имитации разъединения будет отбрасывать пришедшие и отправляемые пакеты, что соответствует UDP соединению, но не соответствует работе TCP протокола.

### **Коммуникационная ROS система**

Другой способ осуществления задержки заключается в создании такой системы ROS узлов, один из которых выступает в качестве «посредника», моделирующего коммуникационный конус, а другой занимается сериализацией и десериализацией данных. Именно такой способ был выбран для реализации, как наилучший из выявленных. Одним из преимуществ такого решения является, то, что эту симуляцию можно запустить как на одном вычислительном устройстве в рамках localhost (для упрощения запуска симуляции модели), так и на различных устройствах. Главное преимущество при таком разделении заключается в том, что, убирая «посредника» из системы, в случае реальной эксплуатации – система ретрансляции топиков будет работать корректно, при условии наличия драйверов коммуникационных устройств. Это позволяет запускать ROS master на каждом из узлов, устраняя при этом брешь в отказоустойчивости ROS системы.

Узел, моделирующий конус, был назван «com\_sim». Для имитации выхода РТК из области коммуникационной среды конуса, необходимо прекратить отправку данных. В случае соединения без гарантированной доставки отбрасывать сообщения и случае соединением с гарантированной доставкой сохранять во *внутренний буфер* этого узла для отправки сообщения после переподключения. Для имитации задержек необходимо на каждое сообщение определять таймер в соответствии с вычисленной в модели задержкой. Потеря пакета в этой реализации может осуществляться при помощи отброса сообщений с вычисленной вероятностью.

Для осуществления передачи данных от ведущего РТК до ведомого РТК и наоборот, была придумана система ретрансляции сообщений пользовательских топиков и названа «uav\_com». На каждом РТК должен запускаться этот узел с namespace'ом «master» или «slave». Namespace в данном случае понимается механизм идентификации узлов в ROS. Так, например, узел «uav\_com» может быть запущен несколько раз с различным namespace'ом для обеспечения возможности отличить один экземпляр узла от другого. Если разработчику необходимо опубликовать в топик с именем «/slave0/topic», то он должен опубликовать в топик с префиксом «/master0/uav\_com», то есть «/master0/uav\_com/slave0/topic». Здесь «slave0» является бортовым номером, который идентифицирует принимаемый РТК, а «master0» идентифицирует РТК отправителя.

Узел uav\_com каждого РТК анализирует публикуемые в системе топики и подписывается на все топики с этим префиксом, сериализуя данные сообщения, перенаправляя их в топик output (рис. 3).

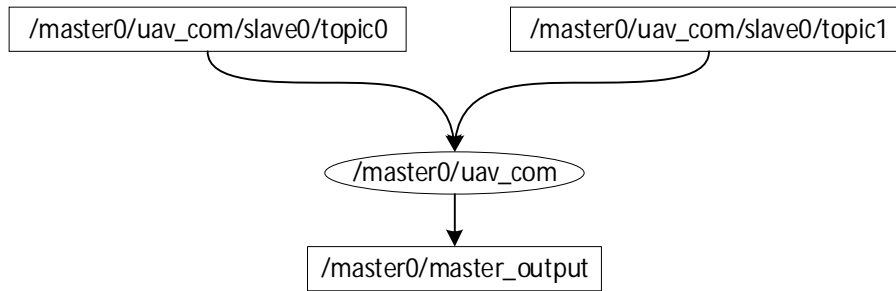


Рис. 3. Демонстрация работы сериализации узла uav\_com

Принимаемая информация в сериализованном виде приходит в топик input и после десериализации узел uav\_com публикует сообщения в указанный отправителем топик (рис. 4).

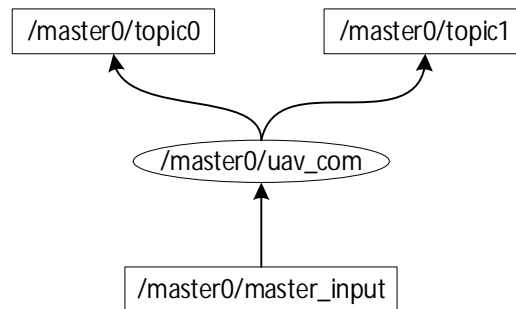


Рис. 4. Демонстрация работы десериализации узла uav\_com

Узел com\_sim анализирует принимаемые (input) и отправляемые (output) топики и перенаправляет данные всем РТК, находящемуся в конусе. Таким образом, происходит моделирование широковещательной сети.

В общем виде схема передачи данных представлена на рис. 5.

Здесь ведомые РТК с бортовыми номерами «slave0» и «slave1» публикуют сообщения в топик с именем «/master0/topic0». Эти потоки данных представлены зеленым цветом. В свою очередь РТК «/master0» публикует сообщения в топик с названием «/slave1/topic1» в РТК с бортовым номером «/slave1». Этот поток данных представлен оранжевым цветом.

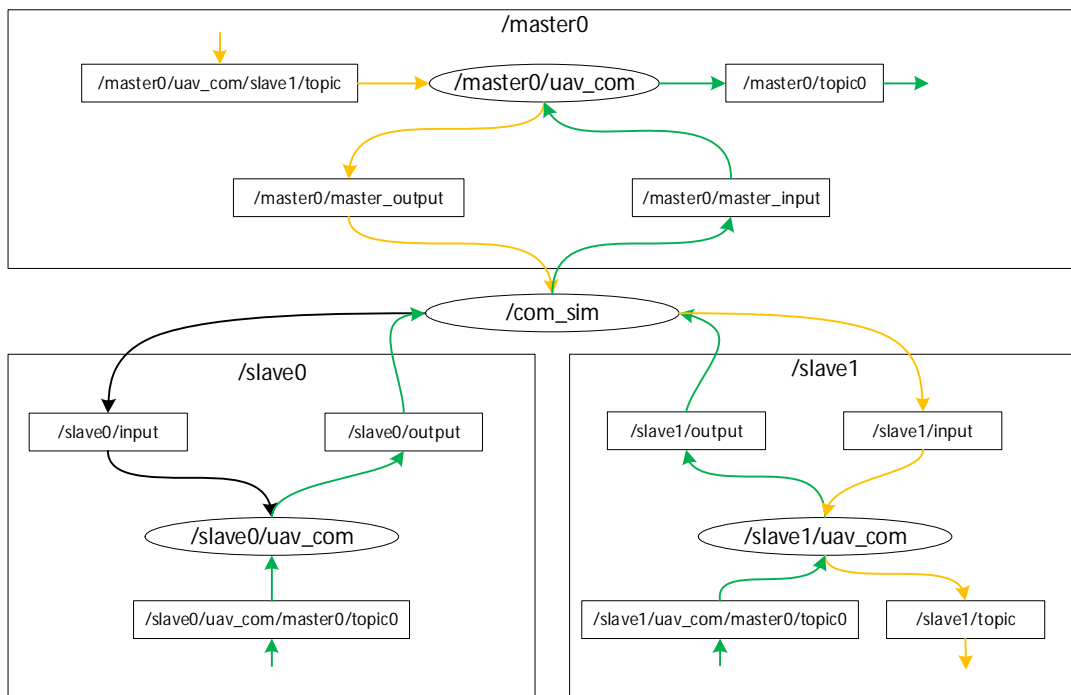


Рис. 5. Общая схема передачи данных

### **Заключение**

В общем виде моделирование специализированных каналов связи задача сложная и на текущий момент полностью не решенная. Такое моделирование нужно не только для полноценного тестирования самих коммуникационных устройств, но и для тестирования прикладного ПО, связанного с коммуникационными задачами. Инструментарии ROS и Gazebo не имеют средств моделирования таких проблем передачи данных как: задержка, потеря сетевых пакетов, разъединения и т.д., что потребовало искать иные решения поставленной задачи.

Стандартные средства сетевого администрирования оказались не совсем подходящими для задач моделирования коммуникационного взаимодействия. Некоторые из них невозможно использовать, другие сложны в использовании и недостаточно гибкие в реализации. В связи с этим для решения задачи было принято решение инкапсулировать моделирование проблем передачи данных в отдельную программную единицу, через которую можно пропускать весь трафик с учетом моделирования коммуникационного пространства, создаваемого конусом. Такое решение оказалось наиболее приемлемым: во-первых, такую модель легко выделить из программного комплекса для повторного использования наработанных программных решений. Во-вторых, оно является более гибким и адаптируемым, в задачах моделирования, за счет возможности настройки отдельных изменяемых параметров моделирования сети передачи данных.

### **Литература**

1. About ROS [Электронный ресурс] URL: <https://www.ros.org/about-ros/> (Дата обращения: 02.04.2021).
2. Beginner: Overview [Электронный ресурс] URL: [http://gazebosim.org/tutorials?cat=guided\\_b&tut=guided\\_b1](http://gazebosim.org/tutorials?cat=guided_b&tut=guided_b1) (Дата обращения: 05.04.2021).