

УДК 004.7.057.4

Аспекты практической реализации протокола ARTCP на ядре Linux 2.6

Алексеев И.В., Меркулов С.А., Сивов А.А.

Ярославский государственный университет им. П.Г. Демидова

e-mail: aiv@yars.free.net;serg.merkulov@gmail.com;mm05@mail.ru

получена 22 марта 2010

Ключевые слова: протокол ARTCP, Linux, транспортные протоколы, TCP, опции TCP

Уточняется формат заголовка ARTCP и описывается алгоритм установки ARTCP соединения. Рассмотрены некоторые вопросы взаимодействия TCP и ARTCP. Приведен план работ по реализации ARTCP в Linux.

ARTCP (Adaptive Rate TCP) – коммуникационный протокол транспортного уровня модели OSI. Этот протокол является модификацией TCP, использующей темпоральные показатели потока в качестве входного параметра для алгоритма управления потоком и сочетающей оконный механизм контроля скорости с диспетчеризацией каждого сегмента. Подробно ARTCP описан в [1] и [2]. Данный протокол позволяет достичь отсутствующего в TCP логического разделения механизмов коррекции ошибок передачи и управления потоком.

В [1] описана имитационная модель протокола и представлены результаты испытаний данной модели. Согласно этим результатам, ARTCP в большинстве ситуаций превосходит TCP по основным показателям. По сравнению с TCP наибольшей эффективности ARTCP достигает при передаче данных по сети с ненулевой вероятностью ошибки, т.е. в случае, когда возникают потери пакетов.

В данной статье рассматриваются некоторые аспекты практической реализации протокола ARTCP для операционной системы Linux версии 2.6, приводится формализация некоторых моментов, не указанных или указанных нечетко в описании протокола. Авторы делают упор на возможность сосуществования в системе реализации ARTCP и TCP, в частности, в статье приводится описание установки TCP соединений в системах, реализующих ARTCP. Рассматривая аспекты реализации протокола, авторы пытались не ограничиваться выбранной операционной системой и привести наиболее общее решение. Тем не менее, Linux является операционной системой, выбранной для реализации протокола ARTCP, поэтому приведенный в данной статье план действий применим именно к ней.

1. Формат пакета

Протокол ARTCP использует для передачи данных пакеты, формат которых полностью совместим с форматом пакетов TCP. Отличие заголовка ARTCP от заголовка TCP заключается в наличии дополнительных полей, неформально определенных в [1] и названных PS и TI. С точки зрения TCP эти поля являются опциями заголовка TCP. Таким образом, заголовок ARTCP является заголовком TCP, полностью соответствующим RFC 793 [3].

Опция PS имеет размер 4 байта и используется для передачи предыдущего порядкового номера сегмента в направлении от отправителя к получателю. Каждый ARTCP пакет, содержащий полезные данные или флаг SYN, должен иметь в заголовке поле PS.

Поле TI используется для передачи значения скважности в направлении от получателя к отправителю. Описание протокола ARTCP предполагает, что получатель помещает в поле TI вычисленное им значение скорости потока $R=S/t$, где S – размер кадра канального уровня, содержащего сегмент, а t – временной промежуток, измеренный между последовательными моментами прибытия сегментов. Данное требование протокола вынуждает определять формат данных с плавающей точкой, передаваемых по сети. Предлагается использовать вместо размера кадра канального уровня, содержащего сегмент, размер полезных данных, содержащихся в сегменте. В этом случае в поле TI можно записывать не значение скорости, а значение временного промежутка t , представленного в виде целого числа, а размер поля установить равным 4 байтам. Отправитель, воспользовавшись значением поля TI, сможет самостоятельно вычислить скорость потока на стороне получателя с наибольшей точностью, возможной в реализации протокола на стороне отправителя.

Для корректной обработки значений интервалов времени протокол должен задать фактическую единицу измерения времени, не зависящую от машины, на которой запущен протокол. Такой единицей могут быть микросекунды. Для замера интервалов времени системе требуется монотонный таймер реального времени (т.н. wall clock), поддерживающий необходимую точность. Для платформы x86, начиная с Pentium, для этих целей может использоваться счетчик TSC, позволяющий измерять интервалы времени с точностью тактовой частоты процессора. ОС Linux предоставляет API, позволяющий преобразовывать размерность интервала в наносекунды.

Каждый ARTCP пакет, содержащий флаг ACK, должен иметь в заголовке поле TI.

С точки зрения TCP поля PS и TI являются опциями TCP, поэтому номера для этих опций должны соответствовать RFC 2780. Полный список существующих номеров для опций TCP доступен на сайте IANA (Internet Assigned Numbers Authority) [4]. RFC 4727 [5] предоставляет два номера для опций TCP, которые можно использовать в экспериментальных целях. Эти номера – 253 и 254. Они могут использоваться для полей PS и TI во время экспериментов, но не должны являться номерами, которые будут использоваться в конечной реализации. RFC 3692 [6] регламентирует использование данных номеров для экспериментов.

2. Взаимодействие ARTCP и TCP

Реализация ARTCP базируется на реализации TCP. ARTCP использует состояния TCP, механизм обработки ошибок TCP, но устанавливает собственный механизм управления потоком данных. Каждый ARTCP сокет должен быть способен принимать TCP соединения (как пассивные, так и активные). Для установленного ARTCP соединения, если принимаемый пакет является корректным TCP пакетом, но не является корректным ARTCP пакетом, то данное соединение переключается в режим TCP соединения (механизм управления потоком ARTCP выключается, вместо него включается механизм управления потоком TCP). Корректный TCP пакет не является корректным ARTCP пакетом, если:

1. Установлен флаг ACK, отсутствует поле TI.
2. Установлен флаг SYN, отсутствует поле PS.
3. Пакет содержит полезные данные, отсутствует поле PS.
4. Отсутствуют полезные данные, есть поле PS.

3. Установка соединения

Для установки соединения ARTCP использует механизм трехстороннего обмена TCP. Чтобы ARTCP сокет мог отличить ARTCP соединение от TCP соединения, при установке ARTCP соединения используются поля PS и TI. Алгоритм установки ARTCP соединения следующий:

1. Инициатор соединения отправляет SYN пакет, содержащий поле PS со значением 0.
2. Вторая сторона отправляет SYN ACK пакет с полями PS и TI со значениями 0.
3. В ответ первая сторона отправляет ACK пакет с полем TI со значением 0.

Если на каком-либо из шагов 1–3 указанные поля PS или TI отсутствуют или их значения не равны указанным, то механизм ARTCP выключается для устанавливаемого соединения и устанавливается TCP соединение.

4. Вычисления значений полей PS и TI для установленного соединения

Значение поля PS устанавливается при отправке пакета равным номеру предыдущего отправленного пакета. Это поле используется получателем для определения

ситуации повторной отправки пакетов вследствие их потери в сети. Определение таких ситуаций предотвращает некорректное вычисление скорости потока на стороне получателя.

Если получатель принимает пакет, поле PS которого равно порядковому номеру предыдущего принятого получателем пакета, то он записывает в поле TI ответного АСК пакета значение промежутка времени, прошедшего между получением этих двух пакетов. В противном случае в поле TI записывается значение, равное нулю, означающее, что отправитель не должен изменять скорость передачи.

5. Основные этапы реализации ARTCP в Linux

Мы планируем разбить работу над реализацией протокола ARTCP в Linux на три этапа. В рамках первого этапа предполагается формализация поведения протокола ARTCP, реализация его компонентов, анализ их работы. Основными задачами на этом этапе являются построение формального описания (модели) протокола, выделение программных компонентов, необходимых для реализации протокола, анализ их взаимодействия с реализацией TCP. На конец данного этапа должна быть создана реализация ARTCP для ОС Linux версии 2.6. На этом этапе мы рассматриваем в качестве целевой платформы компьютеры архитектуры x86, код планируется организовать в качестве опции компиляции ядра Linux.

На втором этапе планируется анализ полученной реализации протокола, выявление слабых и сильных сторон, внедрение поддержки аппаратных архитектур, отличных от x86 (например, ARM), рассмотрение возможности выделения реализации протокола в отдельный модуль ядра. Результатом данного этапа должен стать отчет о корректности и производительности реализации, выявление возможных проблем, создание максимально переносимой реализации ARTCP для Linux, по возможности максимальное отделение реализации ARTCP от реализации TCP. Оптимальным результатом этого отделения является модуль ядра, содержащий реализацию ARTCP.

Третий этап посвящается поиску решений проблем, выявленных на втором этапе. На этом этапе рассматривается возможность модификации протокола ARTCP для улучшения его характеристик. По результатам данного этапа формируется план дальнейших работ.

6. Основные шаги первого этапа реализации ARTCP в Linux

Перечислим основные шаги, которые необходимо выполнить для успешного завершения первого этапа.

1. Описание формата пакета ARTCP, реализация формирования и обработки пакетов.

Этот шаг подразумевает формальное описание формата заголовка ARTCP, опций PS и TI, реализацию в коде Linux функционала для записи, чтения и обработки ARTCP пакетов.

2. Формализация и реализация функции диспетчеризации пакетов.

Данный шаг включает в себя измерение времени прибытия пакетов и отправку пакетов через строго заданные промежутки времени. Основными сложностями реализации данного шага является применение высокоточных таймеров реального времени. Таймеры применяются как для замера интервала времени между соседними полученными пакетами, так и для достижения необходимой скважности (реализации отправки пакетов через требуемые промежутки времени).

3. Включение функции адаптации ARTCP в качестве алгоритма управления потоком.

Этот шаг требует помимо реализации функции адаптации и ее включения в реализацию ARTCP выключение механизмов управления перегрузкой TCP.

По окончании этих трех шагов должна быть создана полностью функционирующая реализация ARTCP, корректно взаимодействующая с TCP. Дальнейшими шагами являются модельные эксперименты с протоколом ARTCP и проведение стендовых испытаний.

Модельный эксперимент нужен для того, чтобы проверить поведение функции адаптации на различных примерах и сравнивать с поведением протокола TCP. Для моделирования поведения протокола ARTCP из множества систем имитационного моделирования информационно-вычислительных сетей была выбрана система OmNET++ [8] по следующим критериям: свободное распространение; реализация моделей на C++; наличие графического режима; широкий диапазон применений; подробная документация. Система OmNET++ является совокупностью программных библиотек, в которых хранятся функции для работы с программами моделирования. Реализовав функцию адаптации ARTCP в этой системе, можно проверить работу протокола в различных режимах.

Список литературы

1. Алексеев И.В., Соколов В.А., Чалый Д.Ю. Моделирование и анализ транспортных протоколов в информационных сетях. Ярославль: Яросл. гос. ун-т, 2004.
2. Alekseev I.V., Sokolov V.A. Compensation Mechanism for Adaptive Rate TCP // 1-St International IEEE/Popov Seminar "Internet: Technologies A and Services". October 1999. P. 68–75.
3. Postel J. Transmission Control Protocol // RFC793 (STD7). 1981.
4. Transmission Control Protocol (TCP) Option Numbers // IANA. / <http://www.iana.org/assignments/tcp-parameters/>
5. Fenner B. Experimental Values in IPv4, IPv6, ICMPv4, ICMPv6, UDP, and TCP Headers // RFC4727. 2006.

6. Narten T. Assigning Experimental and Testing Numbers Considered Useful // RFC3692. 2004.
7. Jacobson V., Braden R., Borman D. TCP Extensions for High Performance // RFC1323. 1992.
8. OMNeT++ Community Site // <http://www.omnetpp.org/>

Aspects of the ARTCP implementation for the Linux kernel 2.6

Alexeev I.V., Merkulov S.A., Sivov A.A.

Keywords: ARTCP protocol, Linux, transport protocols, TCP, TCP options

The article specifies the ARTCP header format and describes the algorithm of the ARTCP connection establishment. Some aspects of TCP and ARTCP interaction are discussed. Presented a work plan to implement the ARTCP in Linux.

Сведения об авторах:

Алексеев Игорь Вадимович,

Ярославский государственный университет им. П.Г. Демидова,
канд. физ.-мат. наук, директор Университетского центра Интернет;

Меркулов Сергей Александрович,

Ярославский государственный университет им. П.Г. Демидова, аспирант;

Сивов Анатолий Александрович,

Ярославский государственный университет им. П.Г. Демидова, аспирант