

УДК: 004.052.2

## **Обеспечение надежной доставки по протоколу UDP с использованием кодов Рида-Соломона**

Д.С. Бардин, Е.О. Карпухин

### **Аннотация**

Предложена архитектура системы, построенной по технологии .NET Remoting с использованием кодов Рида-Соломона для обеспечения надежной доставки UDP-пакетов. Применение кодов Рида-Соломона призвано повысить эффективность работы приложений, использующих протокол UDP.

### **Ключевые слова**

Коды Рида-Соломона; протокол UDP; стек TCP/IP; распределенное клиент-серверное приложение.

### **Введение**

Помехоустойчивое кодирование является основой цифровой системы передачи информации. Оно повышает достоверность передаваемых данных путем внесения избыточности по некоторому математическому закону. Достижения теории помехоустойчивого кодирования нашли свое применение в системах информационного взаимодействия. Так, для защиты информации в заголовках протоколов на канальном, сетевом и транспортном уровнях, а также пользовательских данных, используются CRC-коды с различными алгоритмами формирования избыточности [1].

Малые накладные расходы, связанные с протоколом UDP (User Datagram Protocol) [2], который применяется на транспортном уровне модели ISO/OSI, а также отсутствие необходимости подтверждения получения пакета, делают его наиболее популярным при реализации приложений, где важна скорость передачи (потокковое видео, аудио, VoIP,

мобильное телевидение). UDP в отличие от «надежного» TCP не гарантирует проверку ошибок и доставку пакетов, поэтому велика вероятность искажения информации из-за потерь пакетов в канале. Для кодирования информации, передаваемой по этому протоколу, в работе были использованы коды Рида-Соломона, широко применяемые в цифровой технике и обладающие возможностью обнаружения и исправления как одиночных, так и пакетных ошибок из-за того, что для них разработаны относительно простые и конструктивные методы кодирования/декодирования.

Данная статья посвящена анализу применения кодов Рида-Соломона в системах с распределенной многоуровневой клиент-серверной архитектурой, построенной с использованием технологии .NET Remoting.

## **1. Использование технологии .NET Remoting**

Технология .Net Remoting [3] дает возможность осуществить межпроцессное взаимодействие, позволяя приложению создать объект на другом компьютере, соединенным сетью. Положительной стороной данной технологии является возможность передавать программный код через каналы в совершенно разные домены приложений. Каналы могут взаимозаменяться, благодаря этому значительно повышается безопасность. Для того, чтобы приложения не перепутали принцип передачи данных и формат присылаемых данных, существует форматировщик, который также определяет защищенность канала. Форматировщики бывают двух типов: бинарные форматировщики и SOAP-форматировщики. Все это служит средой для создания экспериментальной модели в виде распределенного приложения с последующим кодированием/декодированием информации. По умолчанию в .NET Remoting реализованы 4 канала: tcp-channel, http-channel, ipc-channel и IChannel. Последний как раз и предоставляет возможность реализации UDP-канала.

Для реализации канала UDP авторами были созданы следующие классы:

– UDPChannel - оболочка, которая предоставляет единый интерфейс к функциональности, реализованной классами UDPServerTransportSink и UDPClientTransportSink. Здесь задаются параметры канала: порт, приоритет и имя, создается цепочка приемников и переопределяются методы унаследованных интерфейсов IChannelSender и IChannelReceiver. Методы класса в основном переадресуют вызовы соответствующим методам клиентской и серверной части.

- UDPServerTransportSink - серверная реализация. Здесь переопределены методы унаследованного интерфейса IServerChannelSink: организован запуск сервера, управление цепочкой приемников, определен размер буфера (65535).
- UDPClientTransportSink - клиентская часть. Переопределены методы интерфейса IClientChannelSink, организована работа с цепочкой приемников и созданы условия для отправки пакетов.
- ChannelIO - в классе реализована работа с UDP-заголовком.

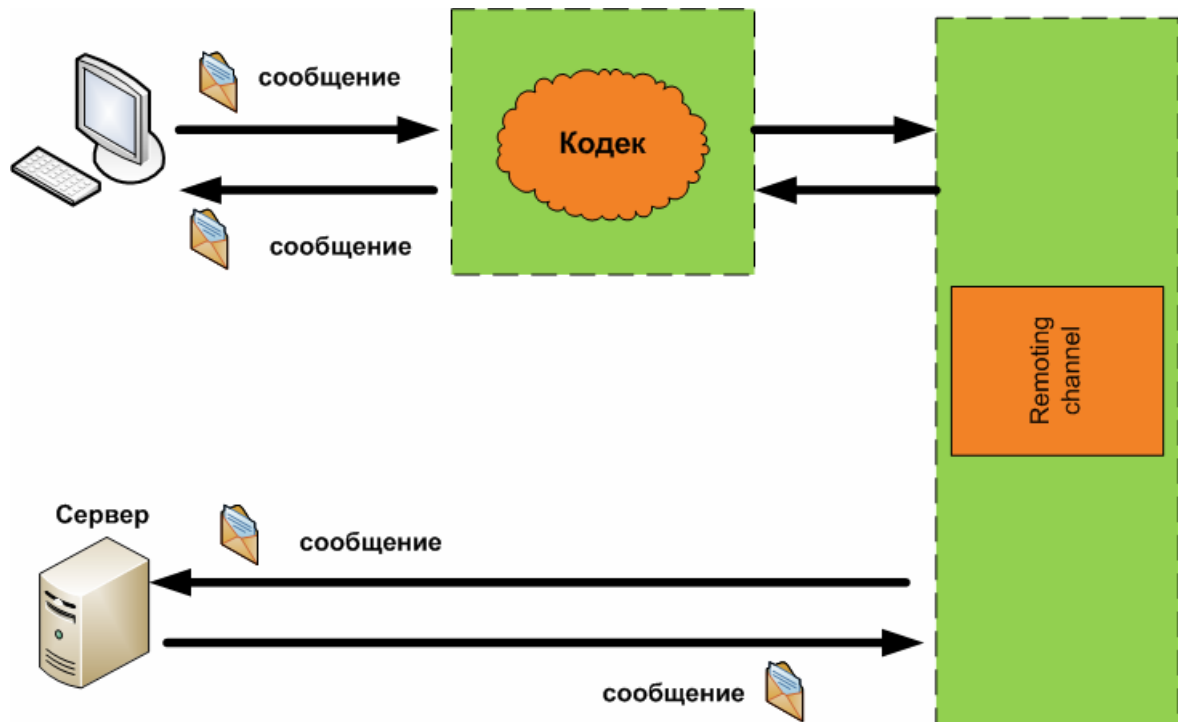


Рис. 1. Архитектура системы, построенной по технологии .NET Remoting с применением кодека Рида-Соломона.

## 2. Модель передачи данных по протоколу UDP

Протокол UDP проектировался для создания в компьютерных сетях с коммутацией пакетов режима передачи дейтаграмм клиента. Он предполагает, что нижестоящим протоколом является IP. К заголовку IP-пакета UDP добавляет поля порт отправителя и порт получателя, которые обеспечивают мультиплексирование информации между различными прикладными процессами, а также поля длина UDP-дейтаграммы и контрольная сумма, позволяющие поддерживать целостность данных [2]. Так как максимальная длина IP-дейтаграммы равна 65535 байтам, максимальная протяженность поля, которое содержит пользовательские данные, в UDP-дейтаграмме составляет 65507 байт. На практике, большинство систем работает с UDP-дейтаграммами длиной 8192 байта или менее.

При использовании дейтаграммного сетевого уровня каждый раз, когда оконечная система хочет послать пакет, она указывает ему адрес получающей оконечной системы, а затем передает этот пакет в сеть. Эта процедура выполняется без предварительной установки виртуального канала. Коммутаторы пакетов в такой сети не содержат информации о состоянии виртуальных каналов. Вместо этого коммутаторы пакетов передают пакет по направлению к получателю, изучая адрес назначения пакета. При этом они ищут нужную им для этого информацию в своей таблице продвижения данных, используя адрес получателя в качестве индекса. Поскольку таблицы продвижения данных могут быть изменены в любое время, пакеты, относящиеся к одной серии пакетов, посланных одной оконечной системой другой оконечной системе, могут следовать по разным маршрутам и прибыть к получателю не в том порядке, в котором были посланы. Современная архитектура Интернета предоставляет единственную модель обслуживания с использованием дейтаграмм, также называемую обслуживанием по остаточному принципу (best-effort service) [4]. При обслуживании по остаточному принципу не дается гарантий сохранения временных интервалов между пакетами, не дается гарантий доставки пакетов с сохранением исходного порядка следования и даже не предоставляются гарантии доставки переданных пакетов. На сегодняшний день обслуживание по остаточному принципу расширяется, предлагая так называемое интегрированное обслуживание и дифференцированное обслуживание [5].

Модель передачи данных по протоколу UDP определяется характеристиками среды на сетевом, канальном и физическом уровнях между передающей и получающей оконечными системами. Ошибки, возникающие на этих уровнях, носят довольно сложный характер (коллизии, задержки в сетях, «проседания» пропускной способности каналов и т.п.), но проявляют себя на транспортном уровне потерей пакетов.

С учетом данных особенностей следует применять коды, которые позволяют исправлять пакетные ошибки. К таким кодам относятся коды Рида-Соломона. Данные коды применены авторами для кодирования пользовательских данных, не затрагивая заголовок UDP-пакета.

### 3. Коды Рида-Соломона

#### 3.1. Особенности рассматриваемых кодов

Коды Рида-Соломона являются частным случаем БЧХ-кода. Пусть  $\alpha$  — элемент поля  $GF(q)$  порядка  $n$ . Если  $\alpha$  — примитивный элемент, то его порядок равен  $q - 1$ , то есть  $\alpha^{q-1} = 1, \alpha^l \neq 1, 0 < l < q - 1$ . Тогда нормированный полином  $g(x)$  минимальной степени

над полем  $GF(q)$ , корнями которого являются  $d - 1$  подряд идущих степеней  $\alpha^{l_0}, \alpha^{l_0+1}, \alpha^{l_0+d-1}$  элемента  $\alpha$ , является порождающим полиномом кода Рида-Соломона над полем  $GF(q)$ :

$$g(x) = (x - \alpha^{l_0})(x - \alpha^{l_0+1})(x - \alpha^{l_0+d-1}), \quad (1)$$

где  $l_0$  — некоторое целое число (в том числе 0 и 1), с помощью которого удается упростить кодер. Обычно полагается  $l_0 = 1$ . Степень многочлена  $g(x)$  равна  $d - 1$ . Длина полученного кода  $n$ , минимальное расстояние  $d$  (минимальное расстояние  $d$  линейного кода является минимальным из всех расстояний Хемминга всех пар кодовых слов). Код содержит

$$r = d - 1 = \deg(g(x)) \quad (2)$$

проверочных символов, где  $\deg()$  обозначает степень полинома; а число информационных символов:

$$k = n - r = n - d + 1 \quad (3)$$

Таким образом,  $d = n - k + 1$ , и код Рида-Соломона является разделимым кодом с максимальным расстоянием. Кодовый полином  $c(x)$  может быть получен из информационного полинома  $m(x)$ , путем перемножения  $m(x)$  и  $g(x)$ :

$$c(x) = m(x)g(x). \quad (4)$$

Более подробно информацию о процессе кодирования и декодирования данных кодов можно найти в [6] и [7].

### 3.2. Практическая реализация

Код Рида-Соломона над  $GF(q^m)$ , исправляющий  $t$  ошибок, требует  $2t$  проверочных символов, и восстанавливает произвольные пакеты длиной  $t$  и меньше. Код является одним из наиболее мощных кодов, исправляющих многократные пакеты ошибок. Он применяется в каналах, где пакеты ошибок могут образовываться так часто, что их нельзя исправить, применив коды, исправляющие одиночные ошибки. К такому каналу можно отнести передачу информации по протоколу UDP.

При передаче по UDP-каналу, используя технологию .NET Remoting, применен код Рида-Соломона со следующими параметрами:

- длина кодового слова - 255 бит
- длина информационного слова – 249 бит (6 бит – избыточность)
- количество корректируемых ошибок – 3
- коэффициенты порождающего полинома - { 1, 0, 1, 1, 1, 0, 0, 0, 1 }
- степень порождающего полинома – 8

Предположим, что возникла потеря одного UDP-пакета (**t = 512 байт**), при этом для восстановления исходных данных потребуются два UDP-пакета. В первый пакет от каждого кодового слова берутся первые три бита, во второй – следующие три бита и т.д. Всего получится 83 пакета с данными и 2 пакета с избыточной информацией. На приемной стороне происходит накопление и восстановление информации.

За перевод сообщения в индексную форму отвечает функция `divide_and_encodes`, работающая по рекурсивному алгоритму, которая затем передает управление функции `encode_rs` из подключаемой `.dll` с кодеком, разработанным авторами. Обратное преобразование сообщения осуществляют функции `decode_and_conquer` и `decode_rs`. Данные функции можно переработать так, что появится возможность исправлять не только небольшие пакеты ошибок, но и справляться с потерей больших объемов информации, например, потерей UDP-пакетов.

Вместо кодов Рида-Соломона, возможно применение одного из кодов, приведенных в [8]. Кроме плюсов, к которым стоит отнести относительную простоту реализации (в сравнении с другими кодами, исправляющими пакеты ошибок) жёсткая структура кода Рида-Соломона приводит к значительным вычислительным затратам при кодировании и декодировании.

#### **4. Возможность применения механизмов надежной доставки в протоколах, которые являются альтернативой UDP**

##### **4.1. Протокол DCCP**

Возможной альтернативой использования UDP-протокола, которая позволит избежать ряда его недостатков, является транспортный протокол DCCP (Datagram Congestion Control Protocol), использующий двунаправленные соединения с управлением перегрузкой для

ненадежной доставки дейтаграмм. В отличие от TCP, DCCP не занимается повторной передачей потерянной информации. Тем не менее, он способен обнаружить потерю информации и предотвратить её использование.

Протокол UDP исключает большие задержки, но приложения UDP, которым необходимо управление перегрузкой, вынуждены вносить задержки. Протокол DCCP имеет встроенную систему управления перегрузкой и обеспечивает надежное согласование параметров при установлении соединения. Одной из целей DCCP было максимальное облегчение для UDP приложений перехода на DCCP, когда он будет внедрен. Для этого, DCCP был спроектирован с минимальной избыточностью, как с точки зрения размера заголовка пакета, так и загрузки CPU оконечных устройств. В него была заложена минимальная функциональность, при сохранении возможности включения новых функций, таких как FEC (Forward Error Correction), псевдонадежность и множественные потоки, которые могут быть добавлены поверх основного протокола в случае необходимости [9].

## **4.2. Протокол UDP-Lite**

Другой альтернативой является UDP-Lite – «Облегченный протокол пользовательских дейтаграмм». Существует ряд приложений, которые получают преимущества в случае, если поврежденные данные будут доставляться, а не отбрасываться. Множество кодеков для аудио и видео относятся к этому классу (например, кодек речи AMR, кодек Internet Low Bit Rate Codec, устойчивые к ошибкам видеокодеки H.263+, H.264, MPEG-4).

Протокол UDP-Lite обеспечивает поддержку контрольных сумм с опциональным частичным покрытием. При использовании этой опции пакет делится на две части — чувствительную к ошибкам (к которой применяется CRC-код) и нечувствительную к ним (не защищается контрольной суммой). Ошибки в нечувствительной к ним части не будут приводить к отбрасыванию пакетов транспортным уровнем принимающей стороны. Когда контрольная сумма обеспечивает защиту всего пакета, UDP-Lite семантически идентичен протоколу UDP. Для IPv4 контрольная сумма UDP покрывает пакет целиком или просто не используется. Для IPv6 контрольная сумма UDP является обязательной и ее использование не может быть отключено. Заголовок IPv6 не включает контрольной суммы самого заголовка и считается необходимым всегда защищать адресную информацию IP с помощью обязательной контрольной суммы UDP. Но по сравнению с UDP, неполные контрольные суммы UDP-Lite обеспечивают дополнительную гибкость для приложений, которые определяют часть своих данных в пакетах, как нечувствительные к ошибкам. В остальном

протоколы UDP и UDP-Lite похожи семантически. Схожими будут и механизмы обеспечения надежности для данных протоколов [10].

### **Заключение**

Применение кодов Рида-Соломона для кодирования сообщений, передаваемых по UDP протоколу, отличается простотой в реализации. Данное решение можно использовать при разработке приложений, осуществляющих передачу мультимедийной информации в сетях, построенных и функционирующих в условиях, приводящих к потере и искажению информации. Кроме того, архитектура, построенная с использованием .NET Remoting и протокола UDP позволит повысить эффективность и надежность работы уже созданных приложений при минимальных временных и материальных затратах на модификацию кода.

Другой возможный вариант повышения надежности информационного взаимодействия в распределенных системах – реализация и использование на базе .Net Remoting протоколов DCCP или UDP-Lite, лишенных ряда недостатков UDP и семантически максимально к нему приближенных.

### **Библиографический список**

1. RFC3385- Internet Protocol Small Computer System Interface (iSCSI) Cyclic Redundancy Check (CRC)/Checksum Considerations <http://www.ietf.org/rfc/rfc3385.txt>
2. RFC768 - User Datagram Protocol <http://www.faqs.org/rfcs/rfc768.html>
3. Маклин С., Нафтел Дж., Уильяме К. Microsoft .NET Remoting.-Пер. с англ. — М.: «Русская Редакция», 2003. — 384 с.: ил.
4. RFC1633- Integrated Services in the Internet Architecture <http://tools.ietf.org/html/rfc1633>
5. RFC-2475-An Architecture for Differentiated Services <http://tools.ietf.org/html/rfc2475>
6. Крис Касперски. «Коды Рида-Соломона в практических реализациях, или Информация, воскресшая из пепла» ,«Системный администратор» №11 2003, <http://av5.com/journals-magazines-online/1/35/308>
7. М.Вернер. Основы кодирования. М.: «Техносфера», 2006.-288с.:ил.



8. В.Варгаузин. «Помехоустойчивое кодирование в пакетных сетях»,  
«Телемультимедиа» № 4 (32) 2005 с.10-16; <http://www.telemultimedia.ru/art.php?id=56>
9. RFC4340- Datagram Congestion Control Protocol <http://www.rfc-editor.org/rfc/rfc4340.txt>
10. RFC3828- The Lightweight User Datagram Protocol  
<http://www.ietf.org/rfc/rfc3828.txt>

### **Сведения об авторах**

Дмитрий Бардин, студент Московского авиационного института (государственного технического университета). E-mail: [personligen@gmail.com](mailto:personligen@gmail.com)

Евгений Олегович Карпухин, аспирант Московского авиационного института (государственного технического университета). E-mail: [ret1987@rambler.ru](mailto:ret1987@rambler.ru)