

Министерство образования и науки Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ
Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра дискретной математики и
информационных технологий

**Исследование сетевого кодирования с использованием среды
графического программирования LabVIEW**

АВТОРЕФЕРАТ МАГИСТЕРСКОЙ РАБОТЫ

студента 2 курса 271 группы
направления 09.04.01 «Информатика и вычислительная техника»
факультета компьютерных наук и информационных технологий
Аль-Хазраджи Али Тхаир Хамид

Научный руководитель

к. ф.- м.н., доцент

подпись, дата

А.Н. Савин

Зав. кафедрой

к. ф.- м.н., доцент

подпись, дата

Л.Б. Тяпаев

Саратов 2018

ВВЕДЕНИЕ

Актуальность работы. В последние годы был найден новый способ оптимизации ресурсов передачи информации. Этот новый способ позволяет осуществить прямое взаимодействие сигналов от нескольких пользователей на физическом уровне, формируя сигналы, которые объединяют информацию от нескольких пользователей. При этом увеличивается спектральная эффективность, уменьшается степень использования канала при обмене данными между сервером и пользователями. Объединение информации от нескольких пользователей обеспечивает новый способ устранения ошибок данных.

Предлагаемый подход, называемый сетевым кодированием, в основном требует промежуточных узлов для выполнения операций комбинирования входящих пакетов. Основная идея сетевого кодирования заключается в разрешении и поощрении смешивания данных на промежуточных сетевых узлах. Сама маршрутизация может рассматриваться как частный случай кодирования, в котором выходы узла являются перестановками входов.

Основное преимущество использования сетевого кодирования проявляется в сценариях многоадресной рассылки, где, учитывается пропускная способность сети. Проблема заключается в том, чтобы вычислить максимальную пропускную способность многоадресной передачи, между узлом отправителем и набором приемников. Сетевое кодирование позволяет лучше использовать ресурсы и может достичь максимального возможного потока, который является теоретической верхней границей использования сетевых ресурсов, позволяя сетевому узлу, например, маршрутизатору, кодировать полученные данные перед его отправкой. Каждый узел, реализованный с функцией сетевого кодирования, получает информацию от всех источников передачи данных, кодирует и отправляет информацию во все адреса назначения.

Кроме того, кодирование может использоваться для поддержки соединений после сбоев или при обрыве связи в сети. Использование сетевого кодирования доказало, что оно может быть устойчивым к возможным постоянным сбоям при передаче данных.

Также, если многоадресное соединение доступно при разных сценариях сбоев, один статический код может обеспечить надежность соединения при всех этих сценариях сбоев.

Таким образом, использование сетевого кодирования помогает эффективнее использовать общие ресурсы, такие как Интернет-соединения или беспроводную связь.

Более того, с теоретической точки зрения, это очень привлекательная междисциплинарная область исследований, представляющая интерес в различных областях, таких как теория информации, алгоритмы, теория алгебры, кодирования и теория графов кодирования.

Целью данной работы является разработка программного обеспечения, симулирующего отправку одновременной информации от нескольких источников передачи данных, с использованием сетевого кодирования, решающего проблему с задержкой информации при сбоях.

Задачи, решаемые для достижения цели:

- изучить основы сетевого кодирования, начиная с объяснения основной идеи концепции, а также его основные преимущества и недостатки;
- освоить программирование в среде LabVIEW;
- разработать программные модули, демонстрирующие процесс передачи информации с использованием сетевого кодирования в компьютерных сетях с помощью протокола TCP/IP;
- разработать модель, симулирующую передачу информации с помощью кодирования и декодирования данных, средствами программной среды LabVIEW;

- провести моделирование процессов обмена информацией в режиме сетевого кодирования между несколькими клиентами и одним сервером.

Работа состоит из введения, трех глав, заключения и списка литературы. Первая глава «Сетевое кодирование», вторая глава «Обзор LabVIEW», глава три «Программная реализация сетевого кодирования и результаты моделирования».

Основное содержание работы.

В первой главе «Сетевое кодирование» дается определение понятию «сетевое кодирование». Большинство работ по сетевому кодированию сконцентрировалось на определенной форме сетевого кодирования: линейное сетевое кодирование. Линейное сетевое кодирование было введено как простой рандомизированный метод кодирования, который поддерживает «вектор коэффициентов для каждого из исходных процессов», который «обновляется каждым узлом кодирования». Другими словами, случайное линейное сетевое кодирование требует, чтобы сообщения, передаваемые по сети, сопровождалась некоторой степенью дополнительной информации, в данном случае - вектором коэффициентов.

В современных сетях связи широко распространены способы, которые легко вмещают такую дополнительную информацию и, кроме того, состоят из безотказных соединений: пакетных сетей. С пакетами такая дополнительная информация или побочная информация может быть помещена в заголовки. В настоящее время размещение основной информации в заголовках пакетов является обычной практикой (например, номера последовательностей часто помещаются в заголовки пакетов, чтобы отслеживать порядок).

Третье определение сетевого кодирования — кодирование пакетов на узле сети (где данные делятся на пакеты, а сетевое кодирование применяется к их содержимому) или, в более широком смысле это называется кодированием на физическом уровне. Оно не похоже на теорию сетевой информации, которая обычно связана с кодированием на физическом уровне.

Во второй главе «Обзор LabVIEW» говорится о графической среде разработки LabVIEW. LabVIEW - это графический язык программирования, который использует значки вместо строк кода для создания приложений. В отличие от текстовых языков программирования, где инструкции определяют выполнение программы, LabVIEW использует программирование потока данных, где поток данных определяет выполнение. В LabVIEW вы создаете

пользовательский интерфейс с набором инструментов и объектов. Пользовательский интерфейс называется передней панелью. Управление объектами передней панели происходит путём добавления кода, использующего функции графического представления. Данный код содержится в Блок-диаграмме.

LabVIEW предлагает различные дополнительные программные наборы для разработки специализированных приложений. Все инструменты легко интегрируются с программой. Получить дополнительную информацию об этих инструментах можно на официальном сайте компании National Instruments.

LabVIEW (сокращение от Laboratory Virtual Instrumentation Engineering Workbench) – это платформа и среда разработки для визуального языка программирования от компании National Instruments. Графический язык называется «G». Первоначально выпущенный для Apple Macintosh в 1986 году, LabVIEW обычно используется для сбора данных, управления инструментами и промышленной автоматизации на различных платформах, включая Microsoft Windows, различные варианты UNIX, Linux и MacOSX. Последняя версия LabVIEW 2017. Файлы кода имеют расширение «.vi», которое является аббревиатурой для «Virtual Instrument». LabVIEW предлагает множество дополнительных надстроек и наборов инструментов.

Третья глава «Программная реализация сетевого кодирования и результаты моделирования» содержит сведения о Моделирование сетевого кодирования в графической программной среде LabVIEW. Графическая среда программирования LabVIEW является универсальной программой платформы, имеющей в своём составе много компонентов для решения задач в различных областях науки и техники. В NI LabVIEW есть встроенные модули для организации компьютерных сетей на основе семейства протоколов (TCP/IP). Эти средства используются в данной работе для разработки сетевого симулятора, моделирующего обмен информации между сервером и несколькими клиентами, используя сетевые кодирования для

обеспечения повышение скорости передачи информацией в широковещательном режиме (Broadcast). Рассмотрим функционирование модулей симулятора.

Основное внимание уделяется проблеме, замедления при просмотре видео файлов, которая связана с пропаданием пакетов информации. Если вы смотрите видео, а оно останавливается снова и снова, это говорит об отсутствии пакетов. Чтобы понять, почему так происходит, представим, что в сети Интернет существует только один единственный пользователь, наблюдающий за прямой трансляцией футбольного матча. Пакеты видео и аудио отправляются по одному и тому же пути, и проблем с задержкой в этой ситуации не возникает. Теперь добавим ещё одного пользователя в эту сеть. В случае, когда оба пользователя одновременно будут смотреть трансляцию, в сети будут наблюдаться проблемы с замедлением.

Рассмотрим следующий пример на рисунке 1, два человека хотят одновременно отправить данные двум другим людям.



Рисунок 1 – Лицевые панели модулей симулятора

Первый отправитель хочет отправить данные (a), которые могут представлять собой изображения, видео, документы и т.д. Порт сервера (serverport) используется для отправки данных на сервер, он должен соответствовать удаленному порту, расположенному на сервере (remote port),

а Port Send для получателя - это порт, используемый для отправки данных первому получателю, который должен соответствовать remote port первого получателя. Канал связи открывается с помощью Tcp listen на сервере для первого получателя. После того, как данные будут отправлены с помощью tcp write, соединение можно закрыть. Второй пользователь хочет отправить данные (b), которые могут быть изображениями, видео, документами и т.д.). Как и в предыдущем случае Server port, используемый для отправки данных на сервер, должен соответствовать remote port, а Port Send для получателя - это порт, используемый для отправки данных второму получателю, который должен соответствовать remote port второго получателя.

Канал связи открывается (Tcp listen) на сервере для второго получателя. Данные отправляются с помощью Tcp write, после чего, соединение закрывается.

Сервер получает данные (a) от первого отправителя и данные (b) от второго. Сервер отправляет данные на соответствующие его порту передачи (portsend) порты первого и второго получателей (remoteport).

Каналы связи открываются с помощью tcp open для первого и второго отправителей. Данные повторно обрабатываются с помощью tcp read, после чего они будут преобразованы в соответствующий код ASCII, а затем в двоичную систему (string to byte array), после чего данные будут преобразованы с помощью исключающего или (XOR). Затем результат будет преобразован в строку с помощью Byte Array To String. В результате проделанных операций кодирования a и b, полученные от первого и второго отправителя, получим L. После этого результат будет отправлен первому и второму ресиверу с помощью tcp write, после чего соединение будет закрыто.

Первый приемник, который получает данные от первого отправителя и сервера. Remoteport используется для приема данных (a) от первого отправителя и получения данных (L) от сервера. Данный порт должен

соответствовать `port send`, расположенному на устройстве первого отправителя на сервере.

Открывается соединительный канал с помощью `Tcp Open`, Данные могут быть получены посредством `Tcp write` от первого отправителя (исходные данные - `a`) и от сервера (зашифрованные данные - `L`). После чего они будут преобразованы в соответствующий код ASCII, а затем в двоичную систему посредством `string to byte array`, после этого над данными произведется преобразование операцией исключающее или (`XOR`). Затем результат выполнения операции преобразуется в строку с помощью `Byte Array To String`. В результате получается отсутствующая информации – `b`, и соединение закрывается. Второй приёмник получает данные от второго отправителя и сервера, `remoteport` используется для приема данных (`b`) от второго отправителя и данных (`L`) от сервера. Как и в случае с первым приемником, `remoteport` должен соответствовать `portsend`, расположенному на устройстве второго отправителя и на сервере.

Открывается соединительный канал с помощью `Tcp Open`. Данные могут быть получены посредством `Tcp write` от второго отправителя (исходные данные - `b`) и от сервера (зашифрованные данные - `L`). После чего они будут преобразованы в соответствующий код ASCII, а затем в двоичную систему посредством `stringtobytearray`, после этого над данными произведется преобразование операцией исключающее или (`XOR`). Затем результат выполнения операции преобразуется в строку с помощью `ByteArrayToString`. В результате получается отсутствующая информация – `a`, и соединение закрывается.

Как видно, первый отправитель передаёт данные (`a`), которые могут представлять собой изображения, видео, документы и т.д. для сервера и первому получателю.

Второй отправитель передаёт данные (`b`), которые могут быть изображениями, видео, документами и т.д. для сервера и второму получателю.

Сервер получает данные (a) от первого отправителя и данные (b) от второго, после чего данные будут преобразованы с помощью операции исключающего ИЛИ (XOR), после этого, результат (L) будет отправлен первому и второму получателям.

Первый получатель, который принимает данные (a) от первого отправителя и данные (L) от сервера, после этого над данными произведется преобразование операцией исключающее ИЛИ (XOR), В результате, получается отсутствующая информации – b.

Второй получатель, который принимает данные (b) от второго отправителя и данные (L) от сервера, после этого над данными произведется преобразование операцией исключающее ИЛИ (XOR), В результате получается отсутствующая информации – a.

Показано, что разработанные программные модели корректно моделируют обмен информацией в широковещательных сетях. Следовательно, они могут использоваться для поиска и моделирования новых конфигураций сетей, обеспечивающих увеличение пропускной способности и помехоустойчивости за счет сетевого кодирования.

ЗАКЛЮЧЕНИЕ

Сетевое кодирование (Network coding) – новое направление в теории информации, позволяющее увеличивать пропускную способность данных в сети. Сетевое кодирование предполагает, что узлы вместо обычной передачи пакетов могут комбинировать несколько входных пакетов в один или в несколько выходных. Это повышает производительность сети без радикальных изменений в ее инфраструктуре. Сетевое кодирование, особенно эффективно при многоадресных передачах.

В ходе выполнения выпускной квалификационной работы были изучены принципы сетевого кодирования, его основные преимущества. Рассмотрена возможность применения сетевого кодирования для повышения пропускной способности широковещательных сетей передачи информации.

С целью моделирования сетевого кодирования была освоена разработка сетевых приложений в среде графического программирования LabVIEW.

Были разработаны и реализованы программные модули, демонстрирующие процесс передачи информации с использованием сетевого кодирования в компьютерных сетях с помощью протокола TCP/IP при широковещательной передаче.

С помощью разработанной программы проведено моделирование передачи информации при наличии задержек отправки пакетов, показавшее, что применение технологии First Come, First Served (FCFS, FIFO) при сетевом кодировании решает проблему потери пакетов, и, следовательно, демонстрирует основное преимущество сетевого кодирования – возможность решения проблем с задержкой информации при сбоях.

Таким образом, все поставленные в работе задачи решены. Разработанные программные модули предполагается использовать для организации высокоскоростной широковещательной сети на основе сетевого кодирования.