

Метод устранения пустого трафика в технологии TDM

В.В. Очеретин

Современный вектор развития общества тесно связан с процессом информатизации и совершенствования инфокоммуникационных технологий. Одновременно с всеобщей информационной интеграцией стремительно возрастает количество информации, передаваемой с использованием сетевых технологий [1].

Одной из основных технологий передачи информации является мультиплексирование с разделением по времени (англ. TDM, Time Division Multiplexing). Эта технология предусматривает объединение нескольких входных низкоскоростных каналов в один составной высокоскоростной канал.

Принцип организации TDM выглядит следующим образом: на вход мультиплексора подается n двоичных последовательностей, мультиплексор поочередно отбирает из этих входных каналов заданную последовательность бит и составляет из них выходной фрейм. Каждая ячейка фрейма зарезервирована для определенного канала поступающих данных [2, 3]. Это позволяет распределять данные при приеме, но возникает проблема «пустого трафика».

Метод уменьшение трафика

Передача мультимедийного трафика реального времени с соблюдением требований по качеству обслуживания является одной из наиболее актуальных задач в современных телекоммуникациях [4]. Как было сказано выше, принцип мультиплексирование по времени заключается в кадровой передаче данных, другими словами данные из нескольких источников информации объединяются в кадр и передаются через высокоскоростной протокол. Для примера в протоколе передачи данных E1 используется 30 информационных ячеек, соответственно каждая ячейка

зарезервирована для определенного источника входных данных. Здесь возникает проблема «пустого трафика» – передача кадра будет происходить даже в том случае, если информация поступила только из 1 источника. В таком случае 29 из 30 информационных ячеек будут пусты, что создаст большое количество пустого трафика [5, 6]. В данной статье описано решение этой проблемы, представлен принцип кодирования информации перед передачей пакета данных и декомпиляции при получении.

Принцип построения передачи информации представлен ниже (рис. 1). На рисунке 1 показаны четырнадцать источников информации, разделенные на семь различных групп, каждая из которых соединена с блоком управляемой памяти или FIFO. Подобное разделение каналов необходимо для анализа поступающей информации с источников данных и вычислением указателя на меньший источник, который применяется при кодировке кадра перед передачей.

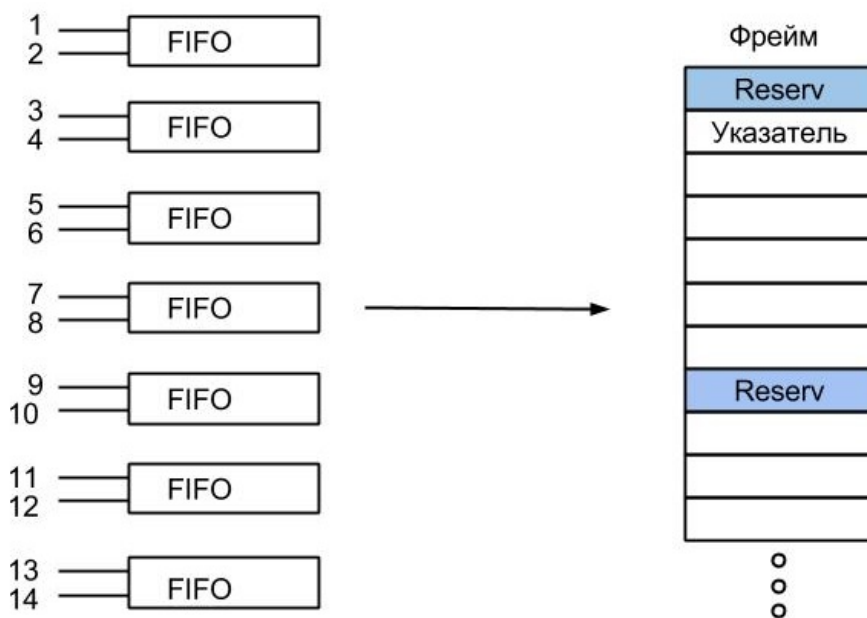


Рис.1. – Принцип построения передачи данных

В каждом FIFO формируется «информационная» часть кадра, то есть та информация, которая поступила с источников данных. Так как в протоколе

Е1 содержится 30 информационных ячеек и для кодирования передаваемой информации используется 8 бит или 1 ячейка, то в FIFO содержится 29 слотов памяти для буферизации поступивших данных. Принцип записи информации в блок управляемой памяти следующий: биты с первого источника записываются, начиная с первой ячейки, а со второго – начиная с последней. Для этого реализованы счетчики, которые отсчитывают количество заполненных слотов памяти определенным источником. Таким образом, можно просто вычислить, от какого источника пришло меньшее количество информации, и передать эти данные для кодирования в первый слот. Принцип кодирования кадра представлен на рисунке 2.

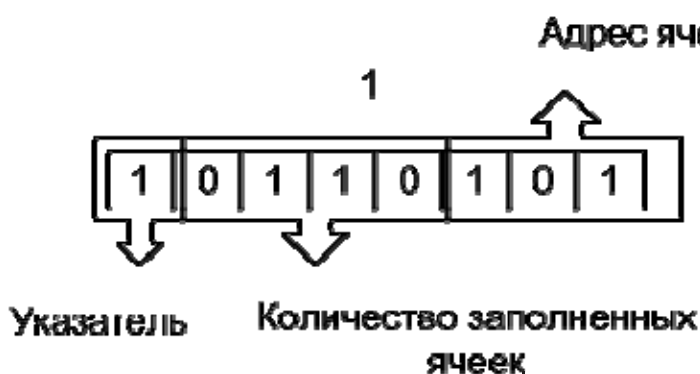


Рис.2. – Принцип кодирования кадра

На рисунке 2 «Указатель» – ссылка на источник, который передал меньшее количество информации в FIFO. 0 – первый источник, соответственно 1 – второй. «Количество заполненных ячеек» – двоичный код, который указывает на то, в скольких информационных ячейках была записана информация с меньшего источника. Так как в блоке управляемой памяти всего 29 информационных слотов, то количество используемых меньшим источником не будет превышать 14, значит достаточно 4 бит информации для кодирования. «Адрес ячейки» – двоичный код, указывающий номер блока управляемой памяти, с которого поступил кадр.

Процесс формирования кадра происходит следующим образом – когда информация с входных источников записывается во все 29 слотов FIFO, начинается подготовка кадра к передаче или его кодирования. Анализируется количество поступившей информации и определяется источник с меньшим информационным объемом. В зависимости от меньшего источника, записывается первый бит 0 или 1. Далее кодируется информация со счетчика, который регистрировал количество информационных ячеек с данными от меньшего источника, и записывается в следующие 4 бита. Последние 3 бита остаются для записи информации о номере источника, с которого сформировался кадр. Кодирование представляет собой преобразование типа данных `integer` в тип данных `std_logic_vector`. Это достигается использованием библиотеки `std_logic_arith` и функции `conv_std_logic_vector (arg: integer; size: integer)`. Затем формируется сам фрейм для передачи [7, 8]. Приоритет передачи фреймов с источников FIFO определяется первенством формирования, иначе говоря кадры передаются непрерывно по мере их формирования.

Когда фрейм поступает в устройство приема, происходит его декодирование, которое аналогично кодированию. Анализируется первый информационный слот данных и на его основании «разбирается» весь кадр. Первый бит указывает на то, каким образом будет распределена информация, а остальные биты являются указателями выходным сигналам на количество поступивших данных.

Формула «пустого трафика» или потери информации в протоколе E1 при классическом TDM представлена ниже (x – количество пустых информационных слотов):

$$Y = \frac{x}{30} * 100\%.$$

График, отображающий сравнение потерь трафика при различных методах представлен на рисунке 3.

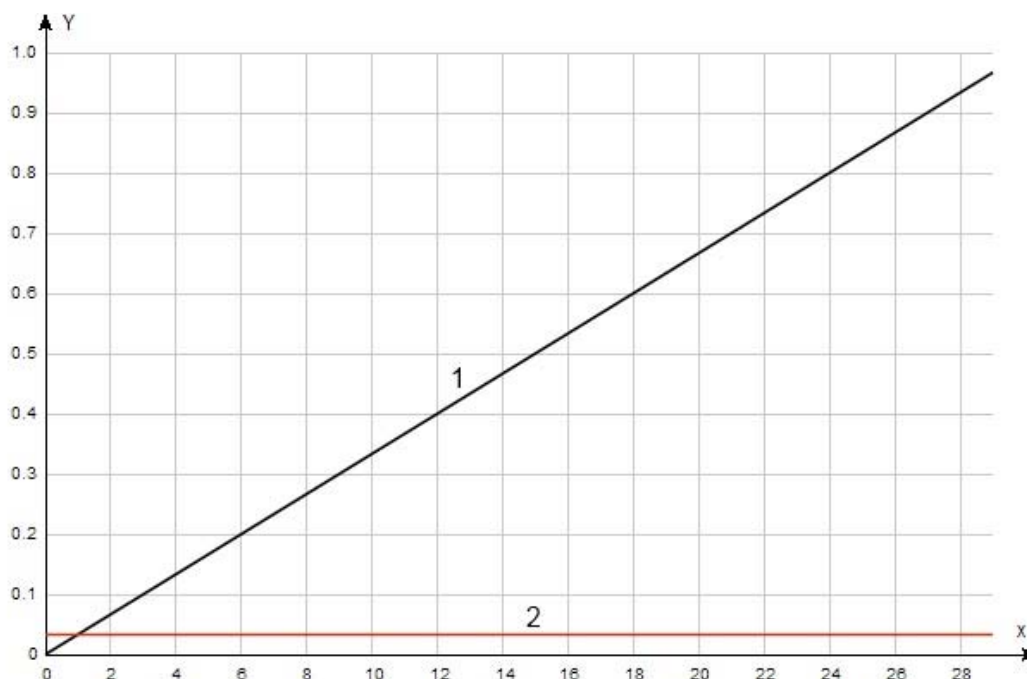


Рис. 3. – Потеря информации при различных методах кодирования (1 – классический TDM, 2 – разработанный метод кодирования)

На рисунке 3 видно, что при классическом TDM потеря информации при передаче зависит от количества пустых ячеек и может достигать порядка 96.6%. В разработанном методе потеря информации является константой и составляет 3.3%. Это дает основание говорить, что разработанный метод передачи данных является более эффективным и экономичным по сравнению с классическим мультиплексированием по времени.

Сравнение со статистическим временным мультиплексированием

При статистическом (асинхронном) временном мультиплексировании каждое устройство может пересылать данные по мере его готовности вне всякого предварительно установленного порядка. Вместе с данными идет информация, показывающая принадлежность сообщения тому или иному устройству [9, 10].

При статистическом мультиплексировании к каждому тайм-слоту добавляются биты идентификации. Таким образом, тайм-слот может состоять из $8+n$ бит (где n – количество бит идентификации, зависящее от источников информации). Если для сравнения, взять количество источников равных 14, то тайм-слот в этом случае будет равен 12 (8 бит информации плюс 4 бита идентификатора). Таким образом, процент полезной информации составляет 66%. Сразу стоит сказать, что эффективность кода зависит также от скорости источников. Чем больше разница между скоростями поступления данных, тем меньше эффективность кода, так как возрастает количество пустого трафика. Разработанный метод полностью устраняет проблему пустого трафика. Количество полезной информации составляет около 97% и не зависит от скорости источников информации. На рисунке 4 показана зависимость полезной информации от количества источников для статистического и для разработанного методов.

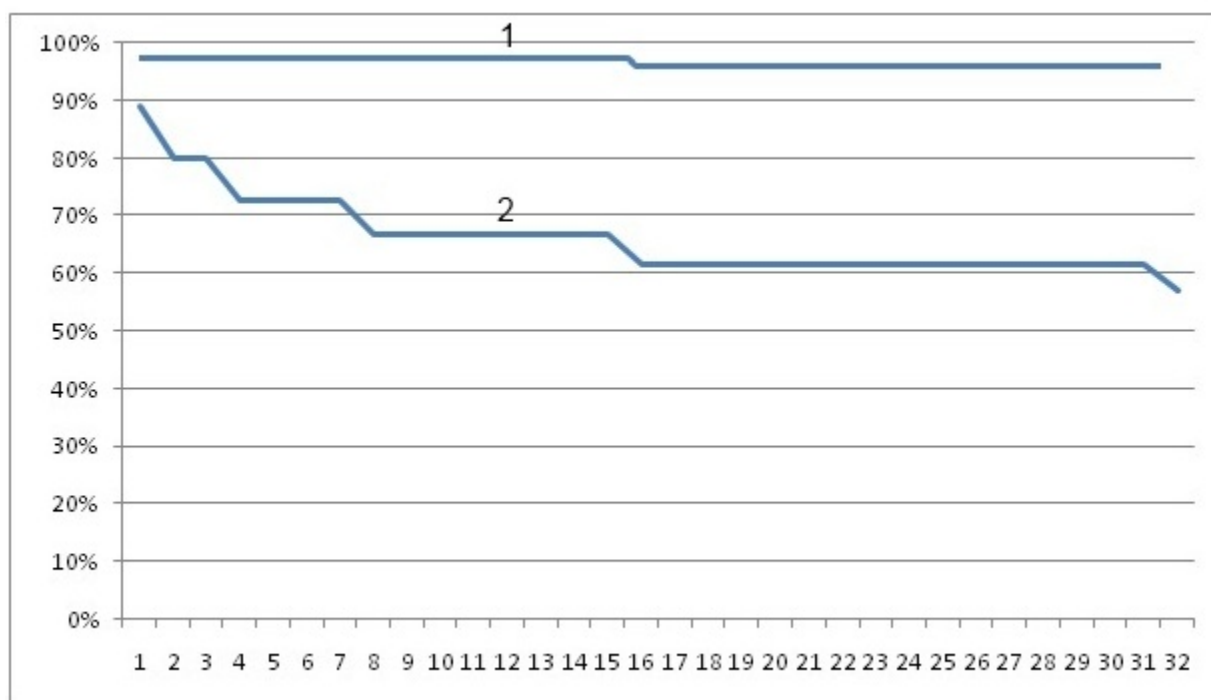


Рис. 4. – Зависимость полезной информации от количества источников для статистического (1) и для разработанного (2) методов

Главное отличие принципа указателей в статистическом методе от разработанного заключается в том, что при статистическом методе кодируется каждый тайм–слот, тогда как в разработанном кодируется сам фрейм, что существенно уменьшает количество бит, используемых для кодирования.

Литература:

1. Бабенко Г.В., Белов С.В. Анализ трафика TCP/IP на основе методики допустимого порога и отклонения [Электронный ресурс] // «Инженерный вестник Дона», 2011, №2. – Режим доступа: <http://www.ivdon.ru/magazine/archive/n2y2011/446> (доступ свободный) – Загл. с экрана. – Яз.рус.
2. Сергиенко А.М. VHDL для проектирования вычислительных устройств [Текст]: Монография / ЧП «Корнейчук», ООО «ТИД «ДС», 2003 – 208 с.
3. Гребешков А. Ю. Стандарты и технологии управления сетями связи [Текст]: Монография / М.: Эко–Трендз. 2003. – 288 с.
4. Трещановский П.А. Оптимизация стохастической модели трафика для мультисервисных сетей [Электронный ресурс] // «Инженерный вестник Дона», 2011, №3. – Режим доступа: <http://www.ivdon.ru/magazine/archive/n3y2011/496> (доступ свободный) – Загл. с экрана. – Яз.рус.
5. Гойхман В.Ю., Гольдштейн Б.С., Сибирякова Н.Г. Протоколы стека ОКС7. Подсистема MAP. Книга 10 [Электронный ресурс]: Монография / БХВ–Петербург, 2014 г. – Режим доступа: <http://kocha.co/KWZrO>

(предварительный просмотр; общее число показываемых страниц ограничено) – Загл. с экрана. – Яз.рус.

6. Одом У. Компьютерные сети. Первый шаг [Электронный ресурс]: Монография / Вильямс 2006 г. – Режим доступа: <http://kocha.co/NPyhG> (предварительный просмотр; общее число показываемых страниц ограничено) – Загл. с экрана. – Яз.рус.

7. Mealy B., Tappero F. Free Range VHDL // URL: http://www.freerangefactory.org/dl/free_range_vhdl.pdf

8. Andrew Rushton VHDL for Logic Synthesis, 3rd Edition // URL: <http://it-ebooks.info/book/2757>

9. Зотов В.Ю. Проектирование встраиваемых микропроцессорных систем на основе ПЛИС фирмы XILINX [Текст] / Горячая линия – Телеком, 2006 – 520 с.

10. Советов Б. Я. Применение микропроцессорных средств в системах передачи информации [Текст] / Под ред. Б. Я. Советова. – М.: Высшая школа, 1987 – 256 с.