

Все идеи и алгоритмы, описываемые в данной статье, являются результатом моей независимой и полностью самостоятельной интеллектуальной деятельности. Как автор, разрешаю свободно использовать, изменять, дополнять все идеи и алгоритмы любому человеку или организации в любых типах проектов при обязательном указании моего авторства.

© Балыбердин Андрей Леонидович 2019 Rutel@Mail.ru

Синхронная Символьная Иерархия
Принципы формирования синхронного канала
(устранение эффекта проскальзывания)

Автор : Балыбердин А.Л.

Новосибирск, 2022 г.

Передача голосового трафика практически сразу перешла на синхронные каналы (с постоянной скоростью и задержкой передачи), а компьютерные коммуникации до настоящего времени базируются на «телеграфном способе общения» (Асинхронная пакетная коммутация). Для конечного пользователя идеальной сетью является та, которая предоставляет точно запрашиваемую скорость передачи данных с заранее известной и неизменной задержкой передачи данных (мое личное мнение). У синхронных (плезиохронных) сетей есть много недостатков, не позволивших им выиграть соревнование у пакетной коммутации. Предлагаемая сеть, основанная на принципах Синхронной Символьной Коммутации или Иерархии (ССИ), представляет собой способ прозрачного перехода компьютерных сетей к синхронному способу связи и лишена практически всех недостатков присущих таким технологиям как PDH. В серии статей будут постепенно и достаточно подробно для того, чтобы являться технически заданием для инженера электроника, описываться принципы функционирования ССИ.

Устранение эффекта проскальзывания

У синхронного способа передачи данных есть неприятный эффект потери или дублирования единичных битов передаваемых данных. Появляется из-за отсутствия жесткой синхронизации генераторов всех передатчиков сети (имеется достаточно много материалов, описывающих эту проблему и методы борьбы с ней). В сетях PDH приняты сложные и дорогие методы борьбы с этим явлением, что в значимой степени и погубило данную технологию вместе с использующей ее телефонной индустрией (Мое мнение: Ее погубил монополизм и пренебрежительное отношение к потребителю).

В практически во всех современных сетях связи канальный уровень является синхронным. Передатчик синхронизируется собственным генератором, а приемник восстанавливает тактовую частоту из принимаемых данных. Участок сети между такими передатчиком и приемником можно считать синхронным каналом передачи данных. Скорости передачи в современных каналах связи достигают десятков и сотен гигабит, при частотах тактирования коммутаторов в единицы гигагерц. Поэтому единицей передачи являются не отдельные биты, а слова или символы (десятки и сотни бит). Канальный уровень полностью соответствует требованиям ССИ и может быть заимствован без изменений и дополнительных доработок.

Уровень потока символов

Эффект проскальзывания реализуется в двух вариантах как пропуск и как добавление лишнего символа. Вариант пропуска символа является самым «неудобным» поскольку требует повторной передачи утерянных данных, а это может занять много времени и потребует применения буферов большого размера. Буфер большого размера добавляет большую и переменную задержку,

что является неприемлемым для конечного пользователя (Требование к каналу в ССИ: Минимальная и заранее известная задержка). Коммутатор должен быть устроен так что бы время коммутации стремилась к четырем тактам синхронизации (прием символа, запись в буфер коммутатора, чтение буфера коммутатора, передача символа). В пределе буфер коммутатора (коммутационная матрица) должен выродиться в регистр, хранящий один символ.

Необходимо создать механизм гарантирующий, что будет реализовываться вариант удвоения символа (скорость на входе коммутатора меньше скорости на выходе) и сделать это с минимальными аппаратными и финансовыми затратами. Современные (бюджетные) кварцевые генераторы частоты имеют точность на уровне 50 ppm (расхождение от указанной частоты не превышает 50 миллионных частей) считаем, что взаимная ошибка гарантированно не превышает 500 ppm (для примера).

Самый простой способ реализовать такой механизм — это заменить в потоке пользовательских данных часть полезных символов символами «Компенсирующий символ». Полезная скорость потока сократится на 0.02%, но это не является значимой величиной.

Для того что бы отличать пользовательские символы от компенсирующих необходима какая-то метка или одна из битовых комбинаций должна быть зарезервирована. Теоретически можно использовать служебные комбинации из линейных кодов, но ранее декларировано отсутствие требований к каналному уровню. Кроме того, данный символ не является единственным, далее может быть развит в целое дерево различных служебных символов в том числе и пользовательских (структурирование передаваемых данных).

Бит ноль (младший бит передаваемого символа) должен указывать на тип символа. Если равен нулю пользовательский тип, если единица служебный тип (интерпретируемый коммутатором).

Сразу добавлю, что следующий за младшим бит равный нулю указывает что символ содержит данные, равный единице содержит «сигнал». Эффективность (КПД) использования первичного потока не пострадает (для 100 битового символа 98%).

Механизм вставки компенсирующих символов должен обеспечивать минимальный размер буфера. Нельзя производить единовременную вставку, например 500 компенсирующих символов один раз в секунду. Наиболее просто и эффективно равномерно (через одинаковые промежутки времени) добавлять их в поток пользовательских данных. При таком подходе в буфере приемника (в идеале) всегда будет ровно один символ, очень редко буфер будет пуст или содержать 2 символа. При чтении из пустого буфера, должен читаться компенсирующий символ.

Положительный эффект: Средняя задержка из-за размера буфера коммутатора

стремится к половине времени передачи одного символа в потоке.

Пример реализации:

Передатчик генерирует синхронизацию и сигнал запроса считывания следующего символа из FIFO коммутатора. Между ним и коммутатором находится блок, который один раз за установленное число тактов блокирует выборку символа, а вместо него вставляет компенсирующий символ (счетчик + мультиплексор).

На приемной стороне, при приеме компенсирующего символа блокируется уже запись в FIFO коммутатора.

Алгоритм устранения проскальзывания достаточно прозрачен и не нуждается в отдельном НИР. Можно только сравнить эффективность (КПД) использования первичного потока с уже существующими вариантами сетей.

Мое мнение: будет лучше даже чем у пакетной коммутации, в ней этот механизм заменен на необходимость передачи преамбулы.