

## Вероятностная синхронизация в телекоммуникационных системах: измерение параметров канала связи

*Для выбора оптимальных маршрутов передачи пакетов данных по компьютерной сети и управления информационными потоками необходимы сведения о её топологии и параметрах каналов связи между узлами. Эти сведения непрерывно отслеживаются и используются маршрутизаторами или иными устройствами. К параметрам каналов связи между каждой парой соседних узлов могут, в частности, относиться задержка распространения сигнала и уровень потерь пакетов, передаваемых в каждом направлении. В существующих сетях для контроля этих параметров используют специальные измерительные пакеты [1, 2]. Можно ли обойтись без столь громоздких средств измерения? В этой статье дан положительный ответ на этот вопрос, а именно, предложены способы измерений и схемные решения, уменьшающие вводимую в поток данных избыточность и основанные на вероятностной синхронизации [3].*

### Часть 1. Измерение задержки распространения сигнала в линии связи

Измерение обычно протекает по сценарию, сходному с отправкой письма по обычной почте и получением уведомления о его вручении. Даты отправки письма и получения уведомления известны отправителю; дата получения письма адресатом и дата отправки уведомления о вручении содержатся в уведомлении. Эти сведения позволяют отправителю подсчитать время задержки корреспонденции в транспортной системе.

В существующих компьютерных сетях в качестве письма и уведомления о его вручении используют измерительные пакеты [1], что связано с пересылкой по линиям связи относительно большого объёма служебной информации в ущерб полезной. Предлагаемый способ измерения задержки позволяет уменьшить этот объём благодаря применению вероятностной синхронизации.

В предлагаемой системе передачи данных (рис. 1), так же как и в известной, используется отправка “письма с уведомлением о вручении”, но и письмо, и уведомление представлены короткими командными словами (командами), а не громоздкими измерительными пакетами. Передача команд от устройства А к устройству В и обратно осуществляется “скрытно” на аппаратном уровне на фоне передачи полезных данных и не требует поддержки со стороны протоколов высоких уровней.

Система содержит два канала. Первый предназначен для передачи данных  $TxD1$  с входа устройства А через линию связи на выход устройства В (данные  $RxD1$ ). По второму каналу данные передаются в обратном направлении:  $TxD2 \rightarrow RxD2$ . Входные и выходные данные первого и второго каналов сопровождаются сигналами встречной и попутной синхронизации [4]:  $TxC1$  и  $RxC1$ ;  $TxC2$  и  $RxC2$ .

Прежде чем перейти к описанию общей идеи построения системы, рассмотрим её составные части.

Устройство А содержит генератор  $G$  синхросигнала, мультиплексор  $MUX1$ , скремблер  $SC1$ , дескремблер  $DSC2$ , анализаторы  $A1$  и  $A4$  битовых потоков, таймер  $Q1$ , процессор  $CPU1$ , генератор  $PLL2$  с фазовой автоподстройкой частоты, блоки управления  $C1$  и  $C4$ . Устройство В построено с использованием аналогичных элементов (исключая генератор  $G$ ).

Генератор  $G$  формирует непрерывную последовательность тактовых импульсов  $CLK1$ , определяющую скорость передачи данных по линии. Частота сигнала  $CLK1$  определяет точность измерения задержки распространения сигнала по линии связи.

Мультиплексор  $MUX1$  в зависимости от значения управляющего сигнала  $M1$  передаёт на выход сигнал с верхнего или нижнего входа: при  $M1 = 0$   $D1 = TxD1$ , на выход мульти-

плектора передаётся основной поток битов данных; при  $M1 = 1$   $D1 = CMD1$ , на выход мультиплектора пересылается командная вставка из нескольких битов.

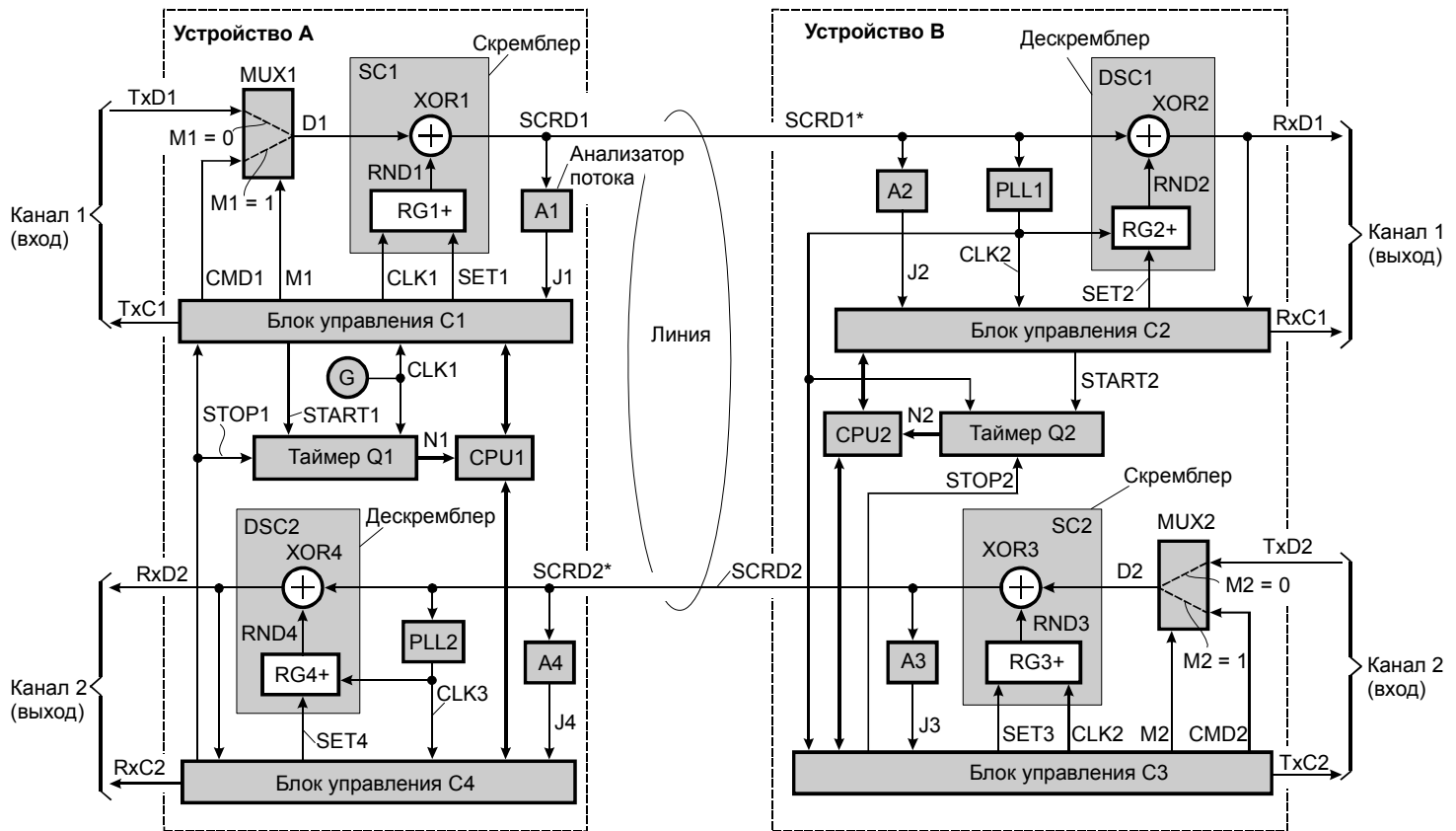


Рис. 1. Предлагаемая система передачи данных со встроенным измерителем задержки распространения сигнала по линии связи

Скремблер SC1 предназначен для преобразования данных  $D1$  к виду, при котором их можно рассматривать как случайные. Это означает, что вероятности появления в каждом такте скремблированных битов  $SCR1 = 0$  и  $SCR1 = 1$  одинаковы и не зависят от предыстории. Скремблирование достигается сложением по модулю два сигнала  $D1$  с сигналом  $RND1$  – последний формируется изолированным генератором псевдослучайной последовательности битов на основе регистра  $RG1+$ . Суммарный сигнал скремблированных данных  $SCR1$  передается в линию связи. Детальное описание применённых в системе скремблера SC1 (SC2) и дескремблера DSC1 (DSC2), выполняющего операцию восстановления исходных данных, приведено в [4, 5].

Линия связи может быть выполнена в виде одной или двух витых пар проводов, оптических волокон либо реализована с использованием иных сред передачи сигналов. Линия может содержать ряд ретрансляторов, поэтому задержка распространения сигнала между устройствами A и B может быть значительной и составлять десятки миллисекунд или более.

Генератор псевдослучайной последовательности битов выполнен на основе кольцевого сдвигового регистра ( $RG1+$ ) с элементом Исключающее ИЛИ в цепи обратной связи (этот элемент на рисунке не показан; подробности структуры генератора описаны в [5]). Сигнал SET1 устанавливает генератор скремблера SC1 в некоторое заранее известное начальное состояние. Синхронизация этого генератора осуществляется сигналом CLK1 от тактового генератора G. В каждом такте работы системы на выходе регистра  $RG1+$  формируется очередной псевдослучайный бит  $RND1$ .

Скремблированный поток данных SCRD1 просматривается анализатором A1 потока (последовательности битов) с целью отыскания в нём некоторого заранее заданного кода Z. Если разрядность этого кода равна M, то он обнаруживается в случайном потоке битов в среднем один раз в каждой цепи длиной  $2^M$  бит. Например, при  $M = 30$  длина цепи примерно равна  $10^9$  бит; при скорости передачи данных по линии, равной 1 Гбит/с (когда применён оптоволоконный кабель), средняя частота сигнала J1 обнаружения кода Z на выходе анализатора A1 составляет примерно 1 Гц. Структура анализатора описана в [4, 5].

Таймер Q1 регистрирует интервал времени между моментом выдачи устройством A команды начала измерения задержки и моментом получения ответной команды от устройства B. Таймер работает с частотой сигнала CLK1 синхронизации. Пуск таймера осуществляется сигналом START1, а его остановка – сигналом STOP1. Код N1 с выходов таймера Q1 после остановки его работы считывается процессором CPU1.

Таймер Q2 в устройстве B регистрирует интервал времени между моментом получения устройством B от устройства A команды начала измерения и моментом отправки ответной команды в устройство A. Таймер работает с частотой сигнала CLK2 синхронизации. Пуск таймера осуществляется сигналом START2, а его остановка – сигналом STOP2. Код N2 с выходов таймера Q2 после остановки его работы считывается процессором CPU2 и затем пересылается в устройство A.

Генератор PLL2 с фазовой автоподстройкой частоты предназначен для формирования высокостабильного синхросигнала CLK3 на основе непрерывного слежения за входным сигналом скремблированных данных SCRD2, поступающим из линии. Этот сигнал в данном примере представлен кодом NRZ [4]. Благодаря достаточной инерционности генератора сигнал CLK3 практически нечувствителен к “дрожанию фазы” сигнала SCRD2 и иным его кратковременным искажениям, вызванным помехами в линии связи. Некоторые схемные решения таких генераторов приведены в [4].

Процессор CPU1 и блоки управления C1 и C4 координируют действия, выполняемые при измерении задержки распространения сигнала по линии.

Соответствующие составные части устройства B функционируют аналогично.

Рассмотрим два режима работы системы.

#### *Передача данных без измерения задержки*

Под управлением синхросигнала TxС1 на вход устройства A поступает поток битов TxD1. Точнее, по фронту каждого импульса TxС1 на вход устройства A поступает очередной бит TxD1 и остаётся стабильным до следующего фронта. Сигнал TxС1 при обнаружении кода Z анализатором A1 фиксируется в состоянии лог. 0 на несколько тактов (интервал  $T_0 - T_1$  на рис. 2). При этом поток данных TxD1 кратковременно прерывается, и вместо него на выход мультиплексора MUX1 передаётся команда CMD1, состоящая из нескольких битов. Управление мультиплексором, как отмечалось, осуществляется сигналом M1 из блока управления C1. В данном режиме, когда задержка не измеряется, прерывание потока данных TxD1 сопровождается передачей однобитовой команды “нет операции” либо иной команды из нескольких битов, не относящейся к процессу измерения задержки, если в такой передаче есть необходимость.

В устройстве B генератор PLL1 с фазовой автоподстройкой частоты формирует синхросигнал CLK2, частота которого в точности совпадает с частотой синхросигнала CLK1. Анализатор A2 аналогичен анализатору A1, поэтому сигнал J2 распознавания кода Z формируется одновременно с сигналом J1 с точностью до задержки линии. Сигнал J2 устанавливает генератор псевдослучайной последовательности битов дескремблера DSC1 в то же начальное состояние, в которое установлен генератор скремблера SC1. Поэтому существует гарантия того, что дескремблирующая последовательность битов RND2 совпадает со скремблирующей

RND1 после первого же обнаружения кода  $Z$  в линии связи. Это обеспечивает самосинхронизацию системы “скремблер – дескремблер” в каждом канале (подробнее – см. [3, 5]).

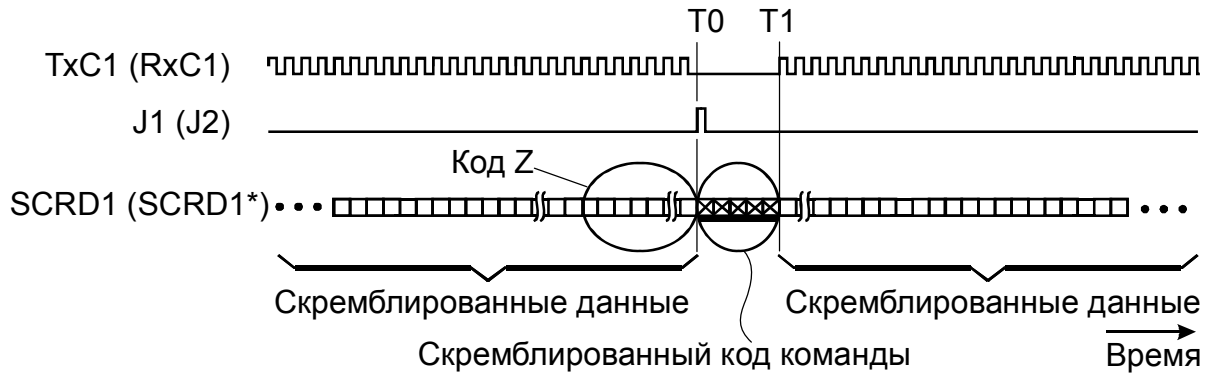


Рис. 2. Временные диаграммы передачи данных и команд из устройства А в устройство В

Сигнал  $RxC1$  повторяет сигнал  $TxC1$  с точностью до задержки передачи и сопровождает выходные данные  $RxD1$ . Сигнал  $RxC1$ , так же как и сигнал  $TxC1$ , в периоды передачи команды  $CMD1$  фиксируется в состоянии лог. 0, поэтому введённые в поток данных команды не воспринимаются внешним абонентом, подключенным к выходу канала 1 системы.

Передача данных и команд в обратном направлении происходит аналогично.

#### Передача данных с измерением задержки

Инициатором измерения задержки является процессор CPU1. Последовательность событий такова.

1. Процессор CPU1 настраивает блок управления C1 на выдачу в линию команды начала измерения задержки. Эта команда выдаётся в линию по сигналу J1 обнаружения анализатором A1 очередного кода  $Z$ . Команда помечена на диаграмме сигнала SCRD1 (рис. 3, а) чёрной точкой. Одновременно с её выдачей формируется сигнал START1 пуска таймера Q1. Команды, не имеющие отношения к измерению задержки, обозначены на диаграмме сигнала SCRD1 белыми точками.
2. По истечении интервала времени  $T(A \rightarrow B)$ , равного задержке распространения сигнала по линии связи, анализатор A2 обнаруживает в потоке скремблированных данных SCRD1\* код  $Z$ , к которому “прикреплена” команда начала измерения (помечена на диаграмме сигнала SCRD1\* чёрной точкой).
3. После дешифрации команды блок управления C2 формирует сигнал START2, по которому включается таймер Q2. Этот таймер регистрирует время  $T(B \rightarrow B)$  задержки ответа на полученную команду начала измерения. Ответ также представлен в виде некоторой команды.
4. Ответная команда выдаётся в линию по сигналу J3 обнаружения анализатором A3 очередного кода  $Z$  во втором канале (эти коды не коррелированы с кодами  $Z$  в первом канале). Ответная команда помечена на диаграмме сигнала SCRD2 чёрной точкой. В момент её выдачи блок управления C3 формирует сигнал STOP2, который приостанавливает работу таймера Q2. В нём фиксируется код N2, соответствующий интервалу времени  $T(B \rightarrow B)$ . Как и ранее, команды, не имеющие отношения к измерению задержки, обозначены на диаграмме сигнала SCRD2 белыми точками.

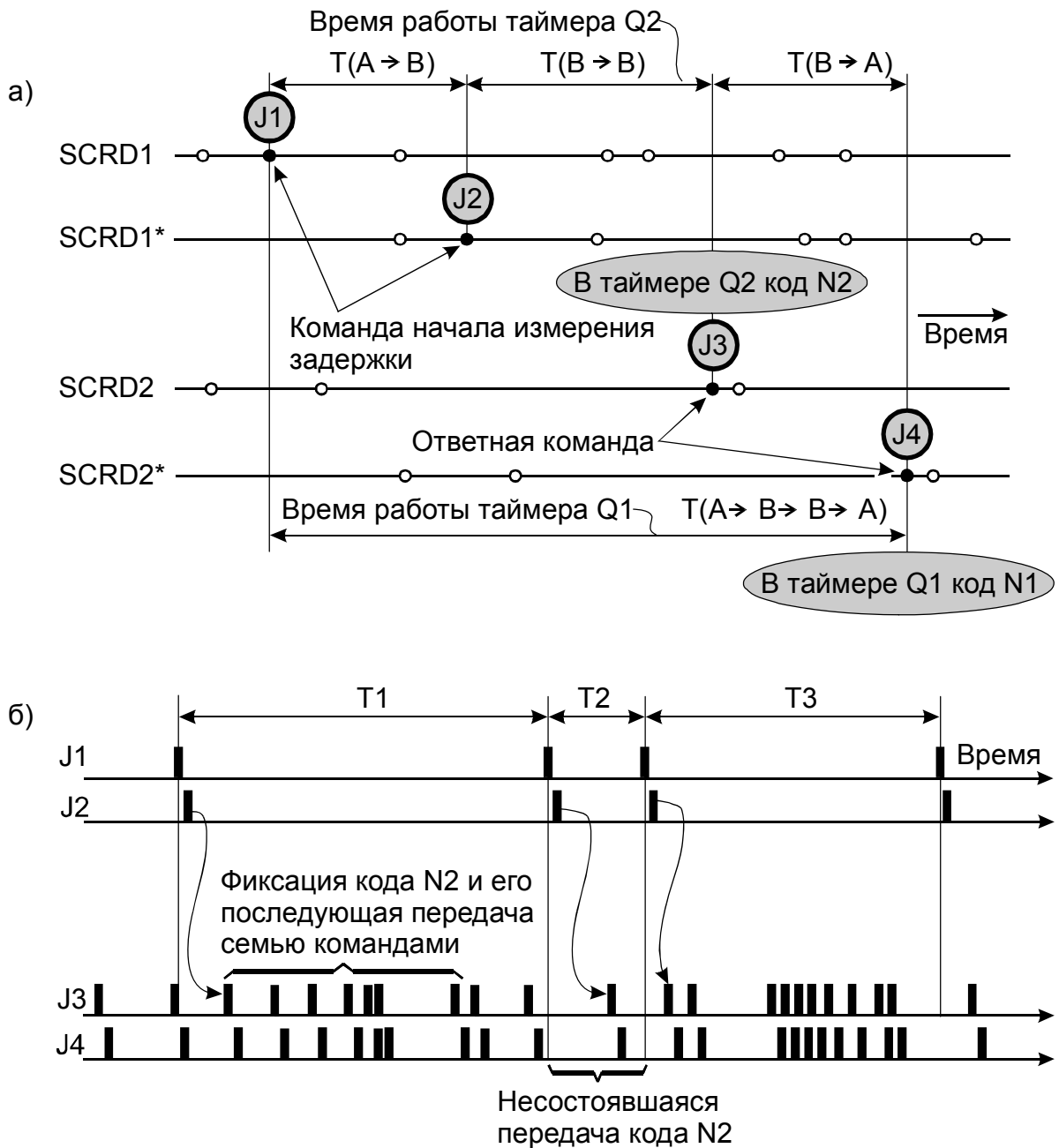


Рис. 3. Диаграмма работы системы в режиме совмещения передачи данных с измерением задержки: а – при передаче команд по обоим каналам; б – при передаче команд только по второму каналу

5. По истечении интервала времени  $T(B \rightarrow A)$ , равного задержке распространения сигнала по линии связи, анализатор А4 обнаруживает в потоке скремблированных данных SCRD2\* код Z, к которому прикреплена ответная команда (помечена на диаграмме сигнала SCRD2\* чёрной точкой).

6. После дешифрации команды блок управления С4 формирует сигнал STOP1, по которому таймер Q1 прекращает работу. В нём фиксируется код N1, соответствующий интервалу времени  $T(A \rightarrow B \rightarrow B \rightarrow A)$ .

7. Код N2 пересылается из устройства В в устройство А. Пересылка может осуществляться поэтапно (например, по одному биту кода N2 в каждой команде) с использованием командных вставок в данные второго канала по мере обнаружения в нём кодов Z.

8. Процессор CPU1, зная коды N1 и N2, вычисляет код  $(N1 - N2)/2$ , который соответствует искомой задержке  $T(A \rightarrow B) = T(B \rightarrow A)$  распространения сигнала по линии связи. При вычислениях учитываются поправки, которые определяются задержками сигналов в блоках управления C1 – C4. Эти задержки зависят от выбранных схемных решений блоков управления. Примеры построения подобных блоков приведены в [3].

Итак, предлагаемый способ измерения задержки распространения сигнала по линии связи позволяет уменьшить объём передаваемой служебной информации благодаря следующим факторам.

Во-первых, исключается необходимость формирования относительно громоздких измерительных пакетов (например пакетов, принятых в технологии асинхронной передачи (АТМ) длиной 53 байта). Вместо них на фоне передачи полезных данных пересылается серия команд малой разрядности в виде отдельных групп битов.

Во-вторых, управление передачей команд осуществляется аппаратно, без поддержки со стороны протоколов высокого уровня.

В-третьих, распознавание команд на удалённой стороне линии не требует введения каких-либо флаговых кодов благодаря использованию принципа вероятностной синхронизации.

Предложенную систему передачи данных можно усовершенствовать. Например, можно принять условие, что все команды первого канала используются только как признаки начала измерения задержки (рис. 3, б). Тогда их можно не пересылать, т. е. не вставлять в передаваемые данные. Перезапуск таймера Q2 при этом осуществляется всякий раз при обнаружении сигнала J2. В этом случае средняя частота сигнала J1 (J2) выбирается, например, на порядок более низкой, чем частота сигнала J3 (J4), чтобы серия ответных команд успела проследовать в устройство А до начала очередного импульса J1 (J2). Так как надёжной гарантии этого нет (используются вероятностные процессы), процессор устройства А должен отслеживать временные соотношения между импульсами J1 и J4 и в некоторых ситуациях не принимать во внимание поступающую (неполную) информацию о результатах измерений.

Поясним сказанное. Предположим, что для передачи кода N2 нужны семь команд. В интервале T1 их девять, две последние команды пустые. В интервале T2 нет полноценных “транспортных средств” для последовательной передачи фрагментов кода N2 – из требуемых семи есть только одна команда. Поэтому начатая передача срывается. В интервале T3 пересылается очередной код N2 и т. д.

## Часть 2. Измерение уровня потерь пакетов, передаваемых между двумя узлами сети

Прежде всего, отметим, что скорость передачи пакетов – переменная величина, а скорость передачи битов по линии – постоянная. Скорость передачи битов выбирается максимально возможной для данной линии и не изменяется в процессе её эксплуатации. Эта скорость не зависит от характера передаваемой информации. В частности, в паузах между пакетами по линии передаются некоторые заранее оговоренные протоколом обмена последовательности битов. Средняя скорость передачи одинаковых по длине пакетов максимальна, если они следуют друг за другом непрерывно, без пауз. С увеличением средних длительностей пауз скорость передачи пакетов снижается. В частности, эта скорость на некотором отрезке времени может быть нулевой. Длина пакета может быть постоянной или переменной.

Потери пакетов при их передаче по каналу связи могут быть обусловлены двумя основными факторами. Первый – искажения принимаемого сигнала из-за помех, отражений,

асимметрии линии и т. п. Второй фактор – кратковременное или длительное ограничение способности многоканального приёмника обрабатывать прибывающий из N-го канала поток пакетов, например, в связи с получением более приоритетных потоков из других каналов. В этом случае буферная память N-го канала приёмника переполняется или блокируется, и вновь поступающие пакеты теряются.

Линии связи обычно обеспечивают высокое качество передачи сигналов. Поэтому потери пакетов обусловлены, в основном, вторым фактором – изменяющейся пропускной способностью приёмника. Именно о таких потерях и пойдёт речь.

Передачик должен регулировать скорость выдачи пакетов так, чтобы уровень их потерь (если потери имеются) находился около некоторой достаточно низкой отметки. Если уровень лежит ниже этой отметки, то это означает, что передачик может увеличить скорость передачи пакетов (если максимальная скорость ещё не достигнута) и, наоборот – при обнаружении заметных потерь передачик вынужден снизить скорость. Таким образом, передача пакетов осуществляется на скорости, близкой или равной максимально возможной. Альтернативой увеличению скорости выдачи пакетов может служить увеличение их длины.

Как измерить уровень потерь пакетов и снизить его до минимума? Чтобы ответить на эти вопросы, следует различать два типа систем передачи данных.

*Первый тип* характеризуется тем, что каждый удачно переданный пакет сопровождается ответным пакетом, в котором содержится подтверждение его получения. Альтернативный вариант оповещения предусматривает передачу ответных пакетов только в тех случаях, когда соответствующие пакеты данных не приняты приёмником в течение допустимого времени ожидания. Не принятые по тем или иным причинам пакеты передаются повторно один или несколько раз в зависимости от результата передачи.

В таких системах адаптация передачика к пропускной способности приёмника может выполняться так. Если к началу сеанса передачи потока пакетов максимально допустимая скорость их следования неизвестна передачику, он использует процедуру “мягкого старта” [6]. Она заключается в том, что на первом шаге скорость передачи пакетов выбирается весьма низкой (паузы между пакетами заполняются флаговыми кодами [4]). На втором шаге скорость увеличивается вдвое (паузы между пакетами соответственно уменьшаются) и проверяется уровень потерь пакетов. О наличии этих потерь передачик судит по содержанию ответных пакетов, поступающих по обратному каналу со стороны приёмника. Если уровень потерь по-прежнему низкий, на третьем шаге скорость вновь увеличивается вдвое и т. д. вплоть до обнаружения заметных потерь пакетов.

После обнаружения таких потерь достигнутая передачиком скорость уменьшается на 25% и вновь проводится очередная оценка их уровня. На следующем шаге, на основании этой оценки скорость увеличивается или уменьшается на 12,5% и т. д. В результате последовательного приближения к границе резкого возрастания уровня потерь пакетов передачик адаптируется к пропускной способности приёмника.

Полученные в процессе регулирования статистические данные о зависимости уровня потерь пакетов от скорости их передачи (или длины) могут использоваться при очередном старте сеанса передачи потока пакетов для уменьшения времени сходимости процесса адаптации.

*Второй тип* предусматривает “безответную” передачу пакетов, в том смысле, что в таких системах нет подтверждений получения или неполучения пакетов приёмником. Если, к тому же, в системе не предусмотрен мониторинг уровня потерь пакетов, то передачик лишён возможности оценивать правильность выбора скорости их отправки из-за отсутствия обратной связи.

Чтобы в таких системах оперативно отслеживать изменения пропускной способности приёмника, применяют измерительные пакеты [2]. Они периодически посылаются переда-

чиком в приёмник и затем возвращаются к нему по обратному каналу. По имеющимся в них показаниям счётчиков переданных и принятых пакетов передатчик вычисляет уровень потерь. Недостатком такого способа мониторинга уровня потерь является необходимость использования громоздких измерительных пакетов, “засоряющих” трафик.

В предлагаемой системе передачи данных (рис. 4) измерительные пакеты не нужны, взаимодействие передатчика с приёмником, как и в ранее рассмотренном решении, осуществляется на основе вероятностной синхронизации. Эта система по структуре близка к приведенной на рис. 1, поэтому описание общих для обеих систем составных частей далее опущено.

Система (рис. 4) предназначена для измерения уровня потерь пакетов при их передаче из устройства А в устройство В. На рисунке показаны только те элементы, которые имеют прямое отношение к такой передаче, поэтому система выглядит асимметричной. Схему можно привести к симметричному виду, дополнив её соответствующими элементами для измерения уровня потерь пакетов при их передаче из устройства В в устройство А.

Передаваемые в обе стороны данные упакованы в пакеты (их также называют фреймами, кадрами, ячейками [4]). Формирование и распознавание пакетов осуществляется внешними устройствами (эти устройства на рисунке не показаны). Система передачи данных не изменяет структуру передаваемых пакетов, а лишь ведёт их подсчёт при пересылке из устройства А в устройство В.

Процессор CPU1 содержит внутренний счётчик переданных в линию пакетов; этот счётчик устанавливается в нуль сигналом J1 и увеличивает текущее содержимое на единицу всякий раз при получении признака FR1-IN наличия очередного пакета, поступившего на вход канала 1. В качестве счётчика может использоваться ячейка памяти или регистр общего назначения процессора.

Процессор CPU2 содержит внутренний счётчик пакетов, успешно принятых с выхода канала 1 внешним устройством. Этот счётчик устанавливается в нуль сигналом J2 и увеличивает содержимое на единицу всякий раз при получении признака FR1-OUT наличия пакета, принятого внешним устройством. Если внешнее устройство отвергло пакет из-за имеющихся в нём ошибок или (что важнее) из-за снижения своей пропускной способности, то пакет теряется и признак FR1-OUT не формируется.

В отсутствие потерь содержимое счётчиков переданных и принятых внешними устройствами пакетов совпадает с точностью до некоторой задержки, которой пренебрегаем для упрощения описания (задержку можно вычислить при проектировании конкретной системы). Потери приводят к тому, что число переданных пакетов превышает число принятых.

Доля потерянных пакетов вычисляется процессором CPU1 в каждом цикле измерения после получения им сведений о содержимом счётчика принятых пакетов из процессора CPU2. Полученные в отдельных циклах результаты усредняются.

Рассмотрим процесс измерения потерь пакетов более подробно (рис. 5).

Циклы измерения задаются промежутками времени T1, T2, T3 и т. д. между импульсами J1 (J2) вероятностной синхронизации в канале прямой передачи данных. Импульсы J3 (J4) вероятностной синхронизации в канале обратной передачи данных задают моменты передачи команд из устройства В в устройство А. В некоторых из этих команд содержится информация о числе пакетов, принятых и зарегистрированных процессором CPU2 в ближайшем предыдущем цикле измерения.

Как отмечалось ранее, импульсы вероятностной синхронизации возникают спонтанно с некоторой средней частотой. Эта частота выбирается на этапе проектирования системы и зависит от скорости передачи битов по линии и разрядности кодов, которые отыскиваются анализаторами А1 – А4 в скремблированных потоках битов.



Средний период между импульсами J1 (J2) задаёт среднее число пакетов, передаваемых через систему в каждом цикле измерения. Это число может быть равно, например, 500. Отметим, что, в отличие от существующих систем, среди этих пакетов нет служебных, т. е. тех, которые вводились бы специально для обслуживания процесса измерения потерь.

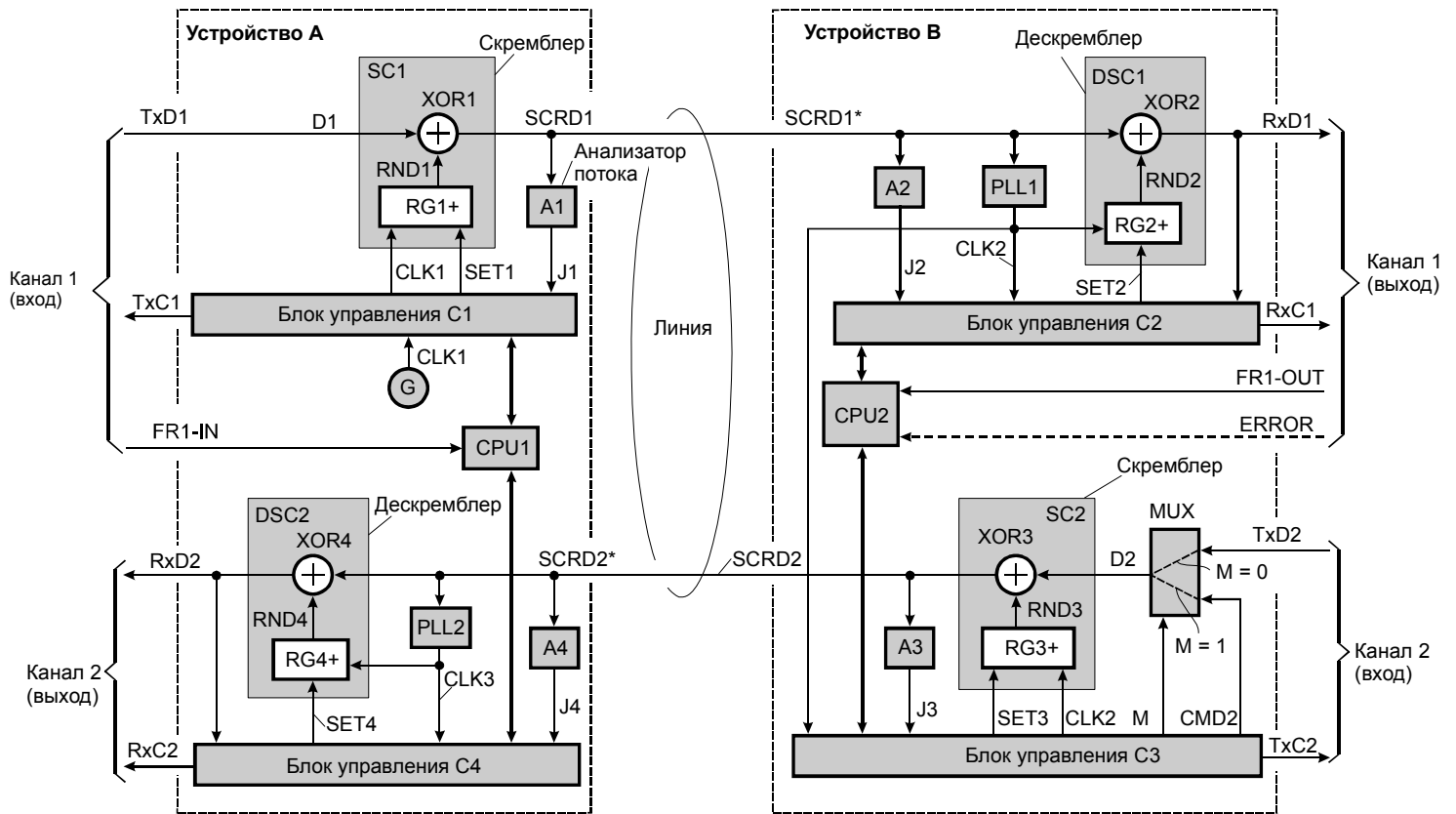


Рис. 4. Предлагаемая система передачи данных со встроенным измерителем уровня потерь пакетов, передаваемых из устройства А в устройство В

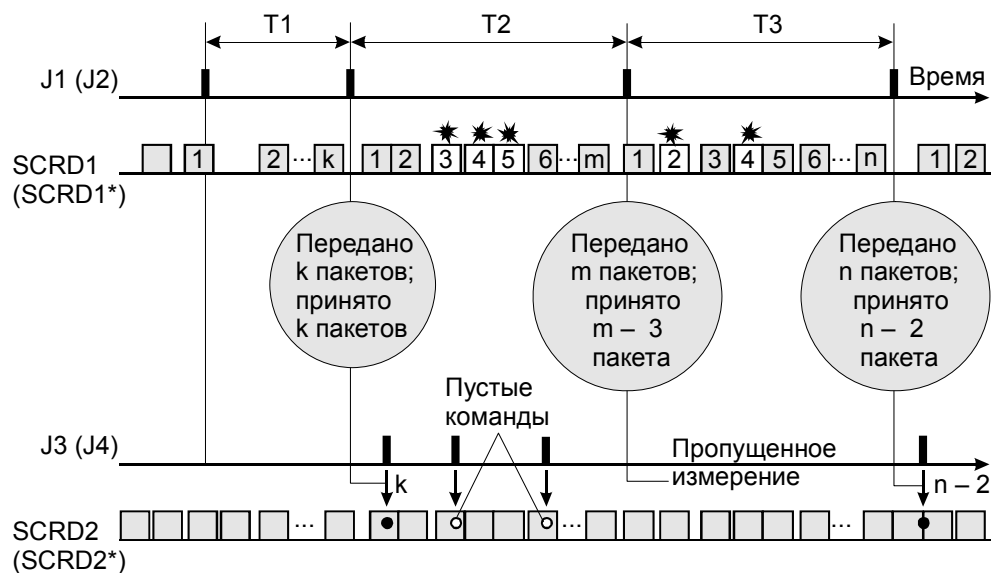


Рис. 5. Временные диаграммы измерения уровня потерь пакетов в системе передачи данных, показанной на рис. 4

В начале интервала  $T_1$  накопленный в счётчике переданных пакетов код запоминается, после чего счётчик устанавливается в нуль. По окончании прохождения текущего пакета к содержимому этого счётчика прибавляется единица. После прохождения через устройство А серии из  $k$  пакетов в счётчике накапливается код  $k$ . Так как в данном случае потерь пакетов нет, то аналогичный счётчик принятых пакетов в процессоре CPU2 также накапливает код  $k$ .

В начале интервала  $T_2$  накопленные в счётчиках переданных и принятых пакетов коды  $k$  запоминаются, после чего эти счётчики устанавливаются в нуль и начинают накопление новых кодов. Первый импульс  $J_3$ , зарегистрированный после окончания предыдущего интервала измерения ( $T_1$ ), вызывает вставку группы командных битов в поток данных SCRD2 (SCRD2\*). В этой группе битов, помеченной на рис. 5 крайней левой чёрной точкой, содержится код  $k$  – ранее сохранённый результат измерения числа полученных в интервале  $T_1$  пакетов.

Процессор CPU1 считывает принятый в составе команды код  $k$  и сверяет его с ранее сохранённым результатом подсчёта переданных пакетов. В данном случае число переданных пакетов в точности совпадает с числом принятых. Полученный результат усредняется с предыдущими в некотором “скользящем окне” длительностью, например, 10 с.

В примере, показанном на рис. 5, до момента окончания текущего цикла измерения ( $T_2$ ) спонтанно формируются ещё два импульса  $J_3$ , которые в соответствии с программой, выполняемой процессором CPU2, порождают две пустые команды, так как код  $k$  уже передан в процессор CPU1. Эти команды помечены белыми точками. Каждая пустая команда может быть представлена одним битом, например, нулевым.

В измерительном интервале  $T_2$  устройство А выдаёт в линию связи  $m$  пакетов. В этот же период, как предполагаем, внешнее устройство, подключенное к устройству В, в течение некоторого времени находится в состоянии переполнения и поэтому неспособно принять три пакета: 3, 4 и 5 (эти пакеты помечены на рис. 5 “кляксами”). Пакеты теряются, сигнал FR1-OUT в период их прохождения стабилен, поэтому в счётчике процессора CPU2 накапливается на три единицы меньше (код  $m - 3$ ), чем в счётчике процессора CPU1 (код  $m$ ).

Этот результат, однако, теряется, так как у устройства В в интервале  $T_3$  не оказалось “транспортного средства” в виде импульса  $J_3$ , чтобы передать его в процессор CPU1. Иными словами, один цикл измерения пропущен, что несущественно, так как в усреднении результатов такие циклы не участвуют.

В измерительном интервале  $T_3$  передано  $n$  пакетов, а принято на два пакета меньше. Код  $n - 2$  передаётся в процессор CPU1 и т. д.

Таким образом, процессор устройства А может постоянно оценивать ситуацию и корректировать скорость передачи пакетов или их длину.

Напомним, что речь идёт о предотвращении потерь пакетов в связи с перегрузкой приёмника. Однако, как уже отмечалось, потери пакетов могут быть также обусловлены искажениями принимаемого сигнала. С увеличением доли таких потерь ошибка адаптации возрастает и приводит к тому, что скорость выдачи пакетов передатчиком неоправданно снижается, так как с его точки зрения приёмник перегружен, что не соответствует действительности. Для устранения таких ошибок адаптации следует при мониторинге потерянных пакетов выделять из них те, которые отвергнуты внешним устройством в связи с неправильной контрольной суммой, а не из-за перегрузки приёмника.

Сортировку потерянных пакетов можно осуществить при введении в систему дополнительной линии, по которой в процессор CPU2 передаётся сигнал ERROR (штриховая линия на рис. 4).

Идея сортировки такова. Внешнее устройство, подключенное к устройству В, “знает” причину уничтожения пакета. Если пакет не принят из-за некорректируемой ошибки (обнаружена неправильная контрольная сумма, и ошибку исправить невозможно), то сигнал

FR1-OUT не формируется, так как пакет не принят, но взамен него формируется сигнал ERROR. Этот сигнал поступает в процессор CPU2 и вызывает прибавление единицы к дополнительно введённому в него внутреннему счётчику ошибочных пакетов. Начальная установка и передача содержимого обоих счётчиков процессора CPU2 в устройство А выполняются совместно.

Если пакет не принят из-за перегрузки приёмника, то не формируются оба сигнала FR1-OUT и ERROR.

Предположим, что в течение некоторого интервала измерения оправлено 250 пакетов, получено 240, причём в счётчике ошибочных пакетов зафиксировано число 3. Из этих сведений процессор CPU1 заключает, что семь из десяти потерянных пакетов не приняты внешним устройством из-за перегрузки, а оставшиеся три пакета отвергнуты им в связи с наличием в них некорректируемых ошибок. При адаптации скорости в расчёт принимаются только семь указанных пакетов.

### Заключение

В статье рассмотрена задача измерения некоторых параметров канала связи между узлами компьютерной сети. К ним относятся, в частности, задержка распространения сигнала и уровень потерь передаваемых информационных пакетов. При мониторинге этих параметров узлы сети обмениваются служебной информацией. В отличие от известных, предлагаемые системные и схемные решения позволяют уменьшить объём служебной информации, передаваемой по каналам связи, благодаря использованию вероятностной синхронизации [3, 5].

### ЛИТЕРАТУРА:

1. Патентная заявка США US 2005/0123003. <http://www.uspto.gov>
2. Пат. США № 7.016.309 B1 <http://www.uspto.gov>
3. Б.В.Шевкопляс. Вероятностная синхронизация в телекоммуникационных системах: вставка команд в поток данных без использования избыточных битов — Схемотехника, 2005, № 5, с. 23 — 25, № 6, с. 23 — 26. Электронная версия статьи: [http://lit.lib.ru/s/shewkopljias\\_b\\_w/](http://lit.lib.ru/s/shewkopljias_b_w/)
4. С.М.Сухман, А.В.Бернов, Б.В.Шевкопляс. Синхронизация в телекоммуникационных системах. Анализ инженерных решений. — М.: Эко-Трендз, 2003. Электронная версия книги: [http://lit.lib.ru/s/shewkopljias\\_b\\_w/](http://lit.lib.ru/s/shewkopljias_b_w/)
5. Б.В.Шевкопляс. Скремблирование передаваемых данных — Схемотехника, 2004, №12, с. 24 — 27, 2005, №1, с. 29 — 32, 2005, №2, с. 32 — 35, 2005, №3, с. 30 — 33. Электронная версия статьи: [http://lit.lib.ru/s/shewkopljias\\_b\\_w/](http://lit.lib.ru/s/shewkopljias_b_w/)
6. Пат. США № 7.035.214 B1 <http://www.uspto.gov>

Б. Шевкопляс