Б. В. Шевкопляс

ВЕРОЯТНОСТНАЯ СИНХРОНИЗАЦИЯ В ТЕЛЕКОММУНИКАЦИОННЫХ СИСТЕМАХ

БИНОМ

Оглавление

Введение

Развитие телекоммуникационных систем постоянно выдвигает задачи улучшения их основных характеристик. К ним в первую очередь относятся повышение надежности и скорости передачи данных. Применение оптоволоконных сред передачи данных приблизило эти характеристики к некоторому теоретически достижимому пределу, так что поиск возможностей улучшения параметров систем следует искать не только в области физики, технологии и совершенствовании элементной базы, но и в области теории и логики построения систем, тем более что возможности последней далеко не исчерпаны. Данная работа посвящена ускорению передачи данных в телекоммуникационных системах с использованием того факта, что при некоторых условиях поток данных можно рассматривать как чисто случайную последовательность нулевых и единичных битов.

Термин «вероятностная синхронизация» («стохастическая синхронизация») используется в технике при описании некоторых процессов, протекающих в сложных системах. В данной книге этот термин, по нашему мнению, удачно отражает смысл основной идеи. Она состоит в том, что удаленные друг от друга телекоммуникационные устройства, например мультиплексоры, взаимно координируют свои действия в результате одновременной реакции на некоторые случайные события, являющиеся «побочными продуктами» передачи потока скремблированных данных.

Скремблирование — это шифрация потока данных, после которой он выглядит как поток случайных битов. Последовательности битов в исходном массиве данных, как регулярные, так и нерегулярные, обратимо разрушаются, так что вероятности появления логической единицы и логического нуля в каждой последующей битовой позиции скремблированного потока одинаковы и не зависят от предыстории. Известно, что применительно к телекоммуникационным системам скремблирование повышает надежность битовой синхронизации между устройствами, подключенными к противоположным сторонам канала связи, и уменьшает уровень помех, излучаемых на соседние линии многожильного кабеля.

Исследование ранее не рассмотренных в литературе функциональных возможностей систем со скремблированием данных показало, что в ре-

зультате некоторого их усовершенствования достигается весьма полезное свойство — самосинхронизация приемника с передатчиком.

В передаваемом по каналу связи скремблированном потоке данных с некоторой средней периодичностью можно обнаруживать любые заранее заданные сочетания битов. Так, при однократной выборке 30-разрядного кода из проходящего потока данных вероятность его совпадения с 30-разрядным эталоном составит $2^{-30} \approx 10^{-9}$. При скорости потока 10 Гбит/с средняя частота событий одновременного (с точностью до задержки передачи) обнаружения заданного кода на разных сторонах канала связи составляет 10 Гц. Это означает, что передатчик и приемник со средней периодичностью 10 Гц одновременно получают некие метки времени (импульсы вероятностной синхронизации), которые самопроизвольно порождаются «полезными» данными, а не создаются, как это делается в настоящее время, введением в поток служебной синхронизирующей информации, например, последовательностей флаговых кодов.

Спонтанно формируемые синхронизирующие метки времени можно использовать для нестандартного решения различных задач. В книге рассмотрен ряд таких решений, в которых эти метки позволяют вводить в поток данных командные вставки, разграничивать каналы в мультиплексированных потоках и т. п. Во всех описанных здесь системах, по сравнению с известными, уменьшена доля служебных битов в потоках данных и упрощено программное обеспечение.

Возможности применения идеи вероятностной синхронизации в телекоммуникационных системах далеко не исчерпаны и не ограничиваются рассмотренными далее примерами построения систем передачи данных.

Глава 1

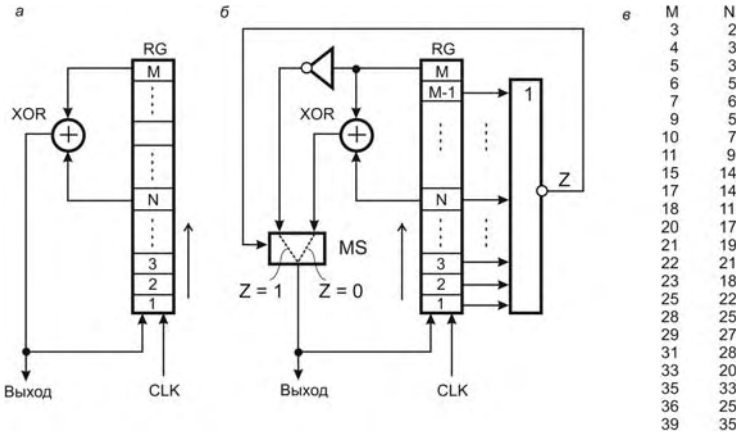
Скремблирование и вероятностная синхронизация

Скремблирование — это шифрация потока данных, в результате которой он выглядит как поток случайных битов. Последовательности битов в исходном массиве данных, как регулярные, так и нерегулярные, обратимо разрушаются, так что вероятности появления логической единицы и логического нуля в каждой последующей битовой позиции потока одинаковы и не зависят от предыстории. Применительно к телекоммуникационным системам скремблирование повышает надежность синхронизации устройств, подключенных к противоположным сторонам линии связи, и уменьшает уровень помех, излучаемых на соседние линии многожильного кабеля. Есть и иная область применения скремблеров — защита передаваемой информации от несанкционированного доступа; однако эта область здесь не рассматривается. В данной главе приведены схемы классических и модернизированных скремблеров и дескремблеров, описаны преимущества, связанные с их применением, рассмотрены меры защиты систем передачи скремблированных данных от злонамеренного пользователя. Описана предложенная нами концепция вероятностной синхронизации.

Генераторы псевдослучайных последовательностей битов

Скремблеры и дескремблеры (шифраторы и дешифраторы особого класса) обычно построены на основе генераторов псевдослучайных последовательностей битов. Генераторы чаще всего выполняются с использованием M-разрядных сдвиговых регистров RG с цепями обратной связи (рис. 1.1).

Приведенные схемы различаются длинами периодов генерируемых последовательностей битов. В схеме, показанной на рис. 1.1, *а* [1], регистр RG исходно установлен в некоторое ненулевое состояние (цепь начальной установки не показана). Под действием положительных фронтов синхросигнала CLK хранимый в регистре код непрерывно циркулирует в нем и одновременно видоизменяется благодаря преобразованию битов логическим элементом Исключающее ИЛИ (XOR). Генерируемая после-



в

M	N
3	2
4	3
5	3
6	5
7	6
9	5
10	7
11	9
15	14
17	14
18	11
20	17
21	19
22	21
23	18
25	22
28	25
29	27
31	28
33	20
35	33
36	25
39	35

Рис. 1.1. Генераторы псевдослучайных битовых последовательностей, формирующие: *а* — последовательность длиной 2^M–1 бит, *б* — последовательность длиной 2^M бит; *в* — таблица для выбора промежуточной точки N подключения цепи обратной связи

довательность битов снимается с выхода этого элемента или с выхода любого разряда регистра. Направление сдвига данных в регистре показано стрелкой. В полном цикле работы генератора в регистре однократно формируются все возможные M-разрядные коды, за исключением нулевого. Циклы следуют один за другим без пауз.

В общем случае при использовании M-разрядного регистра цепь обратной связи подключается к разрядам с номерами M и N (M > N). Для того чтобы на выходе генератора (рис. 1.1, *а*) формировалась псевдослучайная последовательность битов с периодом повторения, равным 2^M–1, следует выбирать точки подключения цепи обратной связи в соответствии с таблицей, приведенной на рис. 1.1, *в*, которая описывает ряд генераторов различной разрядности.

Псевдослучайная последовательность битов с периодом повторения, равным 2^M–1, обладает следующими свойствами.

1. В полном цикле (2^M–1 тактов) число лог. 1, формируемых на выходе генератора, на единицу больше, чем число лог. 0. Добавочная лог. 1 появляется за счет исключения состояния, при котором в регистре присутствовал бы нулевой код. Это можно интерпретировать так, что вероятности появления лог. 0 и лог. 1 на выходе генератора практически одинаковы.

2. В полном цикле (2^M–1 тактов) половина серий из последовательных лог. 1 имеет длину 1, одна четвертая серий — длину 2, одна восьмая — длину 3 и т. д. Такими же свойствами обладают и серии из лог. 0 с учетом пропущенного лог. 0. Это говорит о том, что вероятности появления «орлов» и «решек» не зависят от исходов предыдущих «подбрасываний».

Поэтому вероятность того, что серия из последовательных лог. 1 или лог. 0 закончится при следующем подбрасывании, равна S .

3. Если последовательность полного цикла (2^M–1 тактов) сравнивать с этой же последовательностью, но циклически сдвинутой на любое число тактов W (W не является нулем или числом, кратным 2^M–1), то число несовпадений будет на единицу больше, чем число совпадений.

Усовершенствованная схема (рис. 1.1, *б* [2, 3]) формирует псевдослучайную последовательность битов с периодом повторения, равным 2^M. К этой схеме также применима таблица, приведенная на рис. 1.1, *в*. В регистре RG в определенном порядке формируются все возможные коды, включая нулевой. Схема дополнительно содержит элемент ИЛИ — НЕ, инвертор и мультиплексор MS. Сигнал Z на выходе элемента ИЛИ — НЕ задает направление передачи данных через мультиплексор. При Z = 0 на выход мультиплексора транслируется сигнал с выхода элемента Исключающее ИЛИ, а при Z = 1 — сигнал с выхода инвертора.

До тех пор пока на входах элемента ИЛИ — НЕ присутствует хотя бы одна лог. 1, на его выходе сформирован сигнал Z = 0. В этом случае мультиплексор MS в каждом такте передает в освободившийся (нижний) разряд сдвигового регистра бит с выхода элемента Исключающее ИЛИ так же, как и в схеме, показанной на рис. 1.1, *а*.

В некотором такте *i* в регистре фиксируется код, содержащий единственную лог. 1, размещенную в разряде M–1. Так как в разрядах M и N присутствуют лог. 0, то на выходе элемента Исключающее ИЛИ сформирован сигнал лог. 0, который к началу такта i + 1 поступает на вход регистра. В начале такта i + 1 лог. 1 перемещается из разряда M–1 в разряд M, на входах элемента ИЛИ — НЕ формируется нулевой код. Сигнал Z = 1 переводит мультиплексор MS в состояние, при котором на вход нижнего разряда сдвигового регистра поступает бит с выхода инвертора. В данном случае этот бит равен 0, поэтому в такте i + 2 в регистре фиксируется нулевой код.

К началу такта i + 3 на вход сдвигового регистра с выхода инвертора поступает лог. 1, поэтому по положительному фронту синхросигнала CLK в регистре фиксируется код, содержащий лог. 0 во всех разрядах, кроме первого. Сигнал Z вновь принимает нулевое значение, мультиплексор переключается в состояние передачи сигнала с выхода элемента Исключающее ИЛИ и т. д. Таким образом, регистр проходит через все состояния, включая нулевое. Возможны и иные варианты построения генераторов с числом состояний, равным 2^M [2, 3].

Классические системы «скремблер — дескремблер»

Наиболее распространены два вида систем «скремблер — дескремблер»: с неизолированными и изолированными (от линии связи) генераторами псевдослучайных последовательностей битов (рис. 1.2, 1.3) [4, 5].

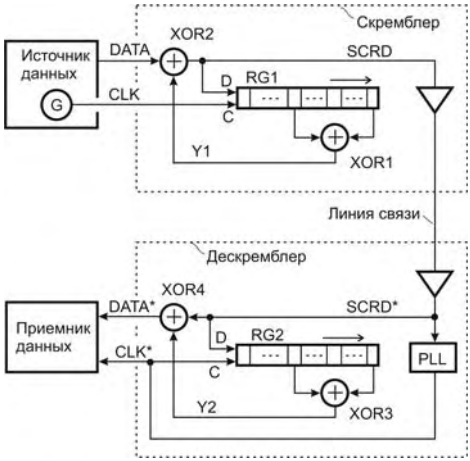


Рис. 1.2. Система «скремблер — дескремблер» с неизолированными генераторами псевдослучайных последовательностей битов

Система с неизолированными генераторами

В системе, показанной на рис. 1.2, скремблер и дескремблер содержат фрагменты рассмотренного ранее генератора (см. рис. 1.1, *а*) псевдослучайных последовательностей битов. В скремблере цепь обратной связи генератора на основе сдвигового регистра RG1 дополнительно содержит элемент Исключающее ИЛИ XOR2. В дескремблере применен аналогичный генератор на основе сдвигового регистра RG2 с разомкнутой цепью обратной связи.

Все процессы, протекающие в системе, синхронизируются от тактового генератора G, размещенного в источнике данных (возможно также его размещение в скремблере). Тактовый генератор формирует сигнал CLK — непрерывную последовательность тактовых импульсов со скважностью, равной двум. В каждом такте по положительному фронту сигнала CLK на вход скремблера подается «новый» бит передаваемых данных DATA, а код в его сдвиговом регистре RG1 продвигается на один разряд вправо, причем в этот же момент в освободившийся разряд заносится «старый» бит данных, просуммированный по модулю два со «старым» битом Y1 с выхода элемента XOR1.

Строго говоря, на границах между битовыми интервалами на выходе элемента XOR2 могут формироваться короткие ложные импульсы в результате неодновременного формирования «новых» сигналов на его входах (сигнал Y1 приходит чуть позже сигнала DATA). Для устранения ложных импуль-

сов можно ввести в цепь сигнала SCRD D-триггер, синхронизируемый отрицательным фронтом сигнала CLK (триггер на рисунке не показан). Короткими ложными импульсами пока пренебрегаем для упрощения изложения основных идей построения систем «скремблер — дескремблер».

Если источник данных посылает в скремблер длинную последовательность сигналов лог. 0 (DATA ≡ 0), то элемент XOR2 можно рассматривать как повторитель сигнала Y1. Тогда регистр RG1 фактически оказывается замкнутым в кольцо и генерирует точно такую же псевдослучайную последовательность битов, как и в рассмотренной ранее схеме генератора, приведенной на рис. 1.1, *а*. Отметим, что в этой ситуации при неблагоприятном стечении обстоятельств есть опасность потери работоспособности скремблера, если в регистре RG1 к началу передачи последовательности сигналов лог. 0 зафиксирован нулевой код. (Об этом — позже.)

Если от источника данных поступает произвольная битовая последовательность, то она взаимодействует с последовательностью битов с выхода элемента XOR1. В результате формируется новая (скремблированная) последовательность битов данных SCRD, по структуре близкая случайной. Эта последовательность, в свою очередь, продвигается по регистру RG1, формирует поток битов Y1 на выходе элемента XOR1 и т. д.

Скремблированная последовательность битов SCRD проходит через передающий усилитель и по линии связи поступает в дескремблер, где проходит через приемный усилитель. Линия связи может быть выполнена, например, в виде витой пары проводов многожильного кабеля городской телефонной сети. С помощью генератора PLL (Phase Locked Loop) с фазовой автоподстройкой частоты из входного сигнала SCRD* выделяется тактовый сигнал CLK*, который передается на синхронизирующие входы регистра RG2 и приемника данных.

Генератор PLL с фазовой автоподстройкой частоты может быть построен по одной из известных схем (см., например, [6]). Он предназначен для формирования высокостабильного синхросигнала CLK* на основе непрерывного слежения за входным сигналом SCRD*. В данном случае отрицательный фронт сигнала CLK* привязан к моментам изменения сигнала SCRD* ($0 \to 1$ или $1 \to 0$), так что положительный фронт сигнала CLK* формируется в середине битового интервала сигнала SCRD*, что соответствует его установившемуся значению. Сдвиг данных в регистре RG2 и прием очередного бита SCRD* в его освободившийся разряд происходят по положительному фронту сигнала CLK*. Дескремблированные данные DATA* поступают в приемник данных и фиксируются в нем по положительным фронтам сигнала CLK*. Благодаря достаточной инерционности генератора PLL сигнал CLK* практически нечувствителен к «дрожанию фазы» сигнала SCRD* и иным его кратковременным искажениям, вызванным помехами в линии связи.

Потоки данных DATA и DATA* совпадают с точностью до задержки передачи. Действительно, в установившемся режиме в сдвиговых регистрах

RG1 и RG2 присутствуют одинаковые коды, так как на входы D этих регистров поданы одни и те же данные SCRD = SCRD* (с учетом задержки передачи), а тактовая частота одна и та же. Поэтому Y2 = Y1 и с учетом этого

$$DATA^* = SCRD^* \oplus Y2 = SCRD \oplus Y2 = (DATA \oplus Y1) \oplus Y2 =$$
$$= DATA \oplus Y1 \oplus Y1 = DATA \oplus 0 = DATA.$$

Рассмотренный способ скремблирования — дескремблирования данных не требует применения какой-либо специальной процедуры начальной кодовой синхронизации, после которой коды в обоих регистрах становятся одинаковыми и, следовательно, начинает выполняться условие Y2 = Y1. Синхронизация достигается автоматически после заполнения регистров одинаковыми данными. Это, пожалуй, единственное преимущество данной схемы перед классической схемой с изолированными генераторами (рис. 1.3).

К сожалению, есть и существенные недостатки. О первом из них уже вскользь упоминалось — это плохая устойчивость по отношению к некоторым неблагоприятным кодовым ситуациям, которые могут возникнуть как при нормальной работе системы, так и в результате злого умысла пользователя. (Этот недостаток и меры его устранения будут подробно рассмотрены далее.)

Второй недостаток состоит в размножении ошибок. При появлении одиночной ошибки в линии связи идентичность содержимого регистров RG1 и RG2 временно нарушается, но затем автоматически восстанавливается, как только правильные данные вновь заполнят регистр RG2. Однако в процессе продвижения ошибочного бита по сдвиговому регистру RG2, а именно, в периоды его попадания сначала на один, а затем на другой вход элемента XOR3 сигнал Y2 дважды принимает неправильное значение. Это приводит к размножению одиночной ошибки — она впервые появляется в сигнале DATA* в момент поступления из линии и затем возникает еще два раза при последующем двукратном искажении сигнала Y2.

Система с изолированными генераторами

В системе, показанной на рис. 1.3, применены изолированные от линии связи генераторы псевдослучайных битовых последовательностей. Преимущества этой системы перед предыдущей заключаются в том, что, во-первых, ошибки, поступающие из линии, не размножаются. Во-вторых, как будет показано, такая система более устойчива по отношению к неблагоприятным последовательностям битов, которые формируются либо в силу случайных стечений обстоятельств либо в результате преднамеренных действий пользователя.

Недостаток этой системы — сложность установления кодовой синхронизации. Отметим, что в конце главы предложено решение, лишенное этого недостатка — оно предусматривает автоматическое установление и

поддержание синхронной работы изолированных генераторов псевдослучайных битовых последовательностей.

Начальная кодовая синхронизация системы (рис. 1.3) осуществляется с использованием аппаратных средств дескремблера и программных средств источника и приемника данных. К аппаратным средствам относятся мультиплексор MUX и программно-управляемый выход приемника данных, на котором формируется управляющий сигнал F. При нормальной работе системы приемник данных постоянно поддерживает на выходе сигнал F = 0. На выход мультиплексора транслируется сигнал Z2 с выхода элемента Исключающее ИЛИ XOR3, генератор псевдослучайной битовой последовательности на основе регистра RG2 изолирован от внешних воздействий со стороны линии связи.

Предположим, что в исходном состоянии дескремблер не синхронизирован со скремблером. Такая ситуация может возникнуть, например, после включения напряжения питания аппаратуры приемной стороны, после ошибки в работе генератора PLL дескремблера из-за воздействия помех на линию связи или по иным причинам. В отсутствие кодовой синхронизации между скремблером и дескремблером содержимое регистров RG1 и RG2 не совпадает, поток принимаемых данных DATA* ошибочен и не совпадает с потоком передаваемых данных DATA.

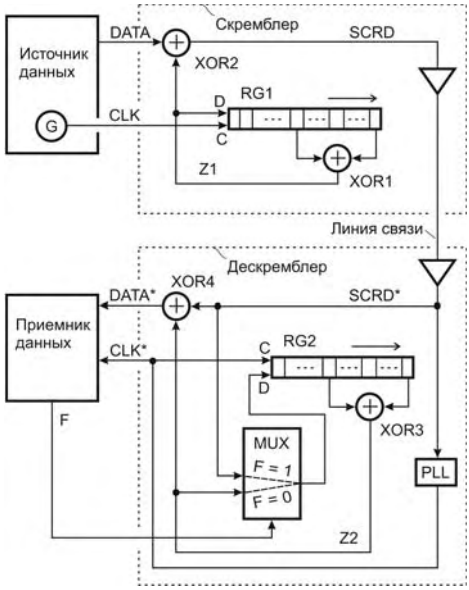


Рис. 1.3. Система «скремблер — дескремблер» с изолированными генераторами псевдослучайных последовательностей битов

При обнаружении устойчивого «хаоса» в данных DATA* (когда в потоке нет обусловленного протоколом обмена разделения на информационные кадры и т. п.) приемник формирует сигнал F = 1. Вследствие этого мультиплексор начинает транслировать на вход D регистра RG2 сигнал скремблированных данных SCRD*, как в ранее рассмотренной системе (см. рис. 1.2).

Протокол обмена предусматривает пересылку данных в виде последовательности кадров. Группы обычных кадров перемежаются со служебными кадрами. Например, после группы из 1000 обычных кадров следует один служебный. Он, в частности, содержит синхронизирующую последовательность из некоторого числа (например, 256) нулевых битов. При выдаче этих битов (DATA = 0) в скремблер элемент XOR2 выполняет функцию повторителя сигнала Z1 с выхода элемента XOR1. Поэтому в данном случае скремблированный сигнал SCRD представляет собой фрагмент «истинной» псевдослучайной битовой последовательности, в том смысле, что она не смешана с потоком произвольных данных DATA и порождается только генератором скремблера.

Эта последовательность загружается в регистр RG2 и проходит через него, так как F = 1. После того как содержимое регистров RG1 и RG2 оказывается одинаковым, сигнал Z2 начинает повторять сигнал Z1. Кодовая синхронизация достигнута. На вход приемника данных подается непрерывная последовательность лог. 0, так как DATA* = DATA ≡ 0. После уверенного обнаружения достаточно длинной (например, содержащей 220 битов) последовательности лог. 0 приемник данных формирует сигнал F = 0 и тем самым возвращает генератор псевдослучайной последовательности битов дескремблера в режим изолированной работы. Теперь кодовая синхронизация не только достигнута, но и «сохранена» благодаря логической изоляции регистра RG2 от линии связи. После окончания передачи служебного (синхронизирующего) кадра источник данных приступает к передаче группы из 1000 обычных кадров согласно принятому в системе протоколу обмена.

Таким образом, в рассмотренной системе для поддержания синхронной работы сдвиговых регистров скремблера и дескремблера (в случае нарушения синхронизации или при первоначальном включении приемной части системы) необходимо периодически прерывать передачу полезных данных и передавать по линии связи служебные информационные кадры, содержащие достаточно длинные цепочки синхронизирующих битов (DATA ≡ 0). В результате уменьшается эффективная скорость передачи данных по линии, усложняется протокол обмена. Кроме того, с увеличением интервалов между служебными кадрами (это способствует более эффективной передаче пользовательских данных) увеличивается время ожидания этих кадров дескремблером в случае потери кодовой синхронизации. В течение времени ожидания передача полезных данных невозможна.

Прежде чем предложить решение, лишенное этих недостатков, рассмотрим преимущества, которые достигаются благодаря применению скремблирования данных, а также обсудим вопросы устойчивости скремблеров по отношению к нежелательным последовательностям битов.

Первое преимущество скремблирования — повышение устойчивости синхронизации

При передаче данных по линии связи применяют различные способы кодирования. В частности, широкое распространение получило NRZ-кодирование и его модификации [5]. Для передачи нулевых и единичных битов выделяются одинаковые интервалы времени — «битовые интервалы». В каждом битовом интервале в зависимости от значения передаваемого бита (лог. 1 или лог. 0) между проводами медной витой пары (двухпроводной линии) присутствует положительное или отрицательное напряжение. Применительно к оптоволоконным линиям, в каждом битовом интервале по оптическому волокну передается или не передается световой поток.

Такое кодирование неприменимо в тех случаях, когда по линии могут передаваться очень длинные цепочки из одинаковых битов. Тогда состояние линии будет оставаться неизменным на протяжении длительного интервала времени и синхронизация приемника с передатчиком может нарушиться. Действительно, приемник надежно восстанавливает синхросигнал только тогда, когда паузы между изменениями сигнала в линии не слишком велики. Изменение сигнала в линии после незначительной паузы позволяет всякий раз корректировать «ход часов» приемника. С увеличением паузы надежность «службы времени» падает. Например, после передачи серии из 10 тыс. нулей приемник, вероятнее всего, не сможет с уверенностью определить, находится ли последующая единица на позиции 9999, 10000 или 10001. То же относится и к передаче длинных цепочек лог. 1. Другими словами, при передаче достаточно большой последовательности нулей или единиц приемник теряет синхронизацию с передатчиком.

На практике данные группируют в кадры постоянной или переменной длины. Каждый кадр содержит некоторую служебную информацию, например флаговый код начала кадра, причем эта информация заведомо не является последовательностью одинаковых битов. Это облегчает поддержание синхронизации, так как возможная однородность пользовательских данных периодически нарушается заведомо неоднородными служебными данными. Тем не менее, для повышения надежности синхронизации желательно исключить из потока пользовательских данных длинные последовательности нулевых или единичных битов. Это и делается с помощью скремблирования — тогда цепочки нулевых или единичных битов (и не только они) преобразуются в псевдослучайные битовые последовательнос-