УДК 621.391.251:681.3.06

В.И. Кубицкий (ГосНИИ «Аэронавигация», Россия)

МЕТОДЫ ДЕКОДИРОВАНИЯ ОШИБОК В КОМПЬЮТЕРНЫХ СИСТЕМАХ

Проанализированы методы и коды защиты информации от ошибок в компьютерных системах, выявлены их недостатки. Предложено использовать коды Лагранжа, для которых разработаны алгоритмы исправления стираний и одиночных ошибок. Определено количество операций в конечном поле при реализации этих алгоритмов.

Большие объёмы цифровых данных, циркулирующих в современных компьютерных системах, требуют защиты от ошибок. Появление ошибок вызвано тем, что практически во всех физических средах передачи и хранения данных присутствуют шумы, приводящие к искажениям и потерям данных.

Как известно, одним их основных способов защиты информации от ошибок является помехоустойчивое кодирование. В дуплексных каналах применяются коды, обнаруживающие ошибки, и это зачастую является достаточным, так как сигнализация об обнаруженной ошибке вызывает повторную передачу от источника. В таких каналах передатчик, прежде чем послать следующий пакет сообщения, должен получить подтверждение правильности приёма адресатом предыдущего пакета. Очевидно, что из-за ожидания подтверждений пакеты передаются с увеличенными задержками. Кроме того, запросы на пересылку, приводят к дополнительным нагрузкам на канал передачи данных, и тем самым уменьшается объём трафика в сети. К тому же, механизм обратной связи может дать положительный результат, если ошибки в пакетах или потери пакетов носят случайный характер. В противном случае, при систематических ошибках и потерях пакетов, восстановить потерянные пакеты в отведённое время с использованием обратной связи будет весьма затруднительно. Поэтому в современных компьютерных системах необходимо применять методы обмена без использования обратных каналов.

В симплексных каналах используют корректирующие коды, которые дополнительно к обнаружению ещё и исправляют ошибки.

Для обнаружения ошибок в пакетах используются контрольная сумма и коды CRC (Cyclic Redundancy Code). Для исправления ошибок в пакете используют корректирующий код Хэмминга, при этом обратный канал не используется. Однако применение кодов CRC, Хэмминга и т. п. для кодирования пакетов не решает проблему восстановления всего пакета без использования обратного канала. Не могут решить проблему другие классические помехоустойчивые коды: свёрточные коды исправляют отдельные битовые ошибки, многие блоковые коды способны исправлять пачки ошибок в отдельно взятом пакете [1].

Для восстановления целого пакета в случае его потери созданы новые помехоустойчивые стирающие коды (коды Лаби) [2], которые позволяют закодировать исходное сообщение конечной длины, состоящее из символов (или пакетов) любой одинаковой длины, потенциально неограниченным потоком кодовых символов (пакетов). В случае необходимости кодер стирающего кода по запросу всегда может добавить "на лету" (on-the-fly, on-line) небольшое число кодовых символов.

Следует отметить некоторые недостатки, которые препятствуют широкому внедрению кодов Лаби (кодов ЛТ).

Для кодов ЛТ, использующих случайную выборку соседей (исходные символы, использованные для генерации некоторого кодового символа), вероятность восстановления пакетов может существенно (и неконтролируемо) варьироваться. Некоторые пакеты, в случае неудачной выборки соседей, могут оказаться невосстановимыми даже при полном отсутствии потерь пактов.

Ввиду того, что в основе теории кодов Лаби (кодов ЛТ) лежит статистическая задача о мячах, бросаемых случайным образом в корзины, эффективность этих кодов проявляется при достаточно большом числе исходных символов (пакетов). Для кодов ЛТ количество информационных пакетов (K) разработчиком этих кодов рекомендуется порядка 10000, при

низких значениях K (порядка 100-500 пакетов) не работают статистические свойства, и эффективность кодов ЛТ может существенно снизиться.

Для полного декодирования всех исходных пакетов коды ЛТ предполагают возможность досылки кодовых пакетов до тех пор, пока все исходные пакеты не будут восстановлены. Для компьютерных систем в режиме реального времени возможность ожидания досылки отсутствует. Каждое сообщение должно быть получено (обработано) в строго определённое время, и ждать дольше этого времени пока придёт дополнительный пакет для восстановления данных сообщения нельзя.

Необходимость буферизации большого числа исходных пакетов (т. к. эффективность кодов ЛТ начинает проявляться для большого числа исходных пакетов) вызывает длительную (до нескольких десятков секунд) задержку сигнала между приёмником и передатчиком.

Для доставки сообщений в канале со стираниями может быть использован блоковый код Рида-Соломона, если символы этого кода, состоящие из m бит, рассматривать как пакеты сообщения. Однако для кода Рида-Соломона существует проблема добавления «на лету» контрольных символов. При переходе к большему числу контрольных символов требуется повторное вычисление всех ранее вычисленных контрольных символов.

Отмеченные недостатки рассмотренных кодов приводят к необходимости разработки для компьютерных систем реального времени кодов, лишённых этих недостатков. Кроме того, такие коды должны поддерживать принцип обмена без использования обратных каналов.

Одним из таких кодов может быть код Лагранжа [4], в котором символы состоят из m бит. Если эти символы рассматривать как пакеты сообщения, то код может быть использован для доставки сообщений в канале со стираниями. Для кода Лагранжа, в отличие от кода Рида-Соломона, не существует проблемы добавления «на лету» контрольных пакетов (символов). При этом для правильного приёма сообщения из k пакетов (символов) в блоке из n пакетов (символов) могут быть стёртыми любые из $r=n–k$ пакетов (символов). Это означает, что в качестве контрольных узлов в случае стирания можно взять стёртые узлы (если их не больше, чем $n–k$). Расширяясь на эти узлы, т.е. выполняя кодирование кода Лагранжа с использованием последовательного алгоритма [5], получим значения пакетов (символов), которые соответствуют стёртым значениям:

$$f(v_\xi) = -\sum_{i=0}^{s'+\xi-1} f_i \prod_{l=\xi+1}^{e} \frac{x'_i - v_l}{v_\xi - v_l}\ , \qquad \xi = \overline{1,e},$$

где $\prod_{l=\xi+1}^{e} \frac{x'_i - v_l}{v_\xi - v_l} = -L^{(i)}_{S'_{\xi-1}}(v_\xi) = L^{(\xi)}_{V_\xi}(x'_i)$, $\quad S'_{\xi-1} = S' \cup \{v_1, \ldots, v_{\xi-1}\}$, $\quad V_\xi = V \setminus \{v_1, \ldots, v_\xi\}$,

$V_e = \varnothing$, $\quad x'_i \in S'_{\xi-1}$, $\quad v_\xi \in V_\xi$;

$S' = \{x'_o, \ldots, x'_{s'}\}$ – множество узлов мощности k', оставшихся после вычеркивания стёртых узлов;

$V = \{v_1, \ldots, v_e\}$ – множество стёртых узлов мощности e;

$f(v_\xi) = f_{s'+\xi}$ – вычисленное значение пакета (символа) в ξ-ом восстанавливаемом узле;

$f_{s'+\xi-1}$ – вычисленные значения пакета (символа) в предыдущих $(\xi-1)$-ых восстановленных узлах;

f_i – значение пакета (символа) в i-ом нестёртом узле ($i = \overline{0,\ s'}$).

Количество операций в конечных полях для восстановления стёртых символов будет следующим:

$N_\oplus = (k'+e)(2e-1) - e(e+1)$, $\quad N_\otimes = (k'+e-1)(e-1)$, $\quad N_{\ominus} = e-1$.

Кроме потери пакетов (символов) пользователь может получить пакеты (символы) с ошибками. В этом случае код Лагранжа обнаружит и исправит t ошибочных пакетов (символов) при количестве контрольных пакетов (символов) равном $r=2t$.

Рассмотрим алгоритмы декодирования одиночной ошибки кодом Лагранжа.

Пусть имеем кодовое слово, определяемое полиномом

$$f(x)=\sum_{i=0}^{s+2} f_i\, L_{S\cup T}^{(i)}(x),$$

где f_i - символ кодовой комбинации; $L_{S\cup T}^{(i)}(x)$ – фундаментальные полиномы Лагранжа; $S=\{x_0,\ldots,x_s\}$ – множество информационных узлов мощности k; $T=\{\beta_1,\beta_2\}$ – множество контрольных узлов мощности 2.

Введём ошибку δ_{i_1} в i_1-ю позицию кодового слова. Тогда искажённый полином будет иметь вид:

$$\widetilde{f}(x)=f(x)+\delta_{i_1}(x)=\sum_{i=0}^{s+2} f_i\, L_{S\cup T}^{(i)}(x)+\delta_{i_1}\, L_{S\cup T}^{(i_1)}(x)=\sum_{i=0}^{s+2}\widetilde{f}_i\, L_{S\cup T}^{(i)}(x).$$

Исправить ошибку можно с помощью предлагаемого ниже алгоритма.

Последовательный алгоритм декодирования А3.

1) Вычисляются значения искажённого полинома в контрольных узлах.

Для 1-го контрольного узла имеем:

$$f^*(\beta_1)=\sum_{i=0}^{s}\widetilde{f}_i\, L_S^{(i)}(\beta_1)=-\sum_{i=0}^{s}\widetilde{f}_i\,(x_i-\beta_2)/(\beta_1-\beta_2).$$

При вычислении значения искажённого полинома $\widetilde{f}(x)$ во 2-ом контрольном узле кроме информационных символов $\widetilde{f}_i$ принятой кодовой последовательности используются принятые значения $\widetilde{f}(\beta_1)$ этого полинома в 1-ом контрольном узле:

$$f^*(\beta_2)=\sum_{i=0}^{s+1}\widetilde{f}_i\, L_{S_1}^{(i)}(\beta_2)=-\sum_{i=0}^{s+1}\widetilde{f}_i=-\sum_{i=0}^{s}\widetilde{f}_i-\widetilde{f}(\beta_1),$$

где $S_1=S\bigcup\{\beta_1\}$.

2) Определяются значения невязок:

$$R_1=\widetilde{f}(\beta_1)-f^*(\beta_1),\qquad R_2=\widetilde{f}(\beta_2)-f^*(\beta_2).$$

Анализируя величины невязок R_1 и R_2, делаем выводы о месте и величине ошибки:

- если $R_1=0,\quad R_2=0$, то ошибки нет;

- если $R_1\neq 0,\quad R_2\neq 0$ и $R_1\neq R_2$, то ошибка имела место в информационной части кодового слова;

- если $R_1=R_2\neq 0$, то ошибка произошла во 1-ом контрольном узле и величина ошибки равна R_2 (или R_1);

- если $R_1=0,\quad R_2\neq 0$, то ошибка произошла во 2-ом контрольном узле и величина ошибки равна R_2;

- если $R_1\neq 0,\quad R_2=0$, то имеют место 2 или более ошибки, сумма величин которых равна 0.

3) Вычисляются величины синдромов:

$$Q_0=R_2,\qquad Q_1=R_1(\beta_1-\beta_2)+Q_0\beta_2=R_1(\beta_1-\beta_2)+R_2\beta_2.$$

При $\beta_1=1,\quad \beta_2=0$: $\quad Q_1=R_1$.

Анализ величин синдромов Q_0 и Q_1 даёт следующее:

- если $Q_0=Q_1=0$, то ошибки нет;

- если $Q_0\neq 0,\quad Q_1=0$, то ошибка произошла в узле со значением равным 0;

- если $Q_0=Q_1\neq 0$, то ошибка произошла в узле со значением равным 1;

- если $Q_0 \neq Q_1 \neq 0$, то ошибка имеет место в узле, значение которого не равно 0 или 1;

- если $Q_0 = 0, \quad Q_1 \neq 0$, то имеют место 2 или более ошибки, сумма величин которых равна 0.

Переходим к определению места и величины ошибки. Ключевые уравнения декодирования одиночной ошибки имеют вид:

$$Q_0 = \delta_{i_1} \widetilde{x}_{i_1}^0, \qquad Q_1 = \delta_{i_1} \widetilde{x}_{i_1} = Q_0 \widetilde{x}_{i_1}.$$

4) Определяется номер позиции искажённого символа:

$$\widetilde{x}_{i_1} = Q_1 / Q_0 = [(R_1(\beta_1 - \beta_2) + R_2\beta_2)] / R_2 = R_1^{'} / R_2 + \beta_2,$$

где $R_1^{'} = R_1(\beta_1 - \beta_2)$.

При $\beta_1 = 1, \quad \beta_2 = 0$: $\widetilde{x}_{i_1} = R_1 / R_2$.

5) Вычисляется величина ошибки:

$$\delta_{i_1} = Q_0 = R_2.$$

6) Производится коррекция искажённого символа:

$$f_{i_1} = \widetilde{f}_{i_1} - \delta_{i_1}.$$

Если ошибка произошла в одном из контрольных узлов, то

$$\widetilde{f}(\beta_j) = f(\beta_j) + \delta_j = \sum_{i=0}^{s} f_i L_{S_{j-1}}^{(i)}(\beta_j) + \delta_j = f^*(\beta_j) + \delta_j,$$

и правильное значение контрольных символов определяется из выражения

$$f(\beta_j) = \widetilde{f}(\beta_j) - \delta_j = f^*(\beta_j).$$

Алгоритм декодирования требует выполнения операций в конечном поле в количестве (с учётом этапа исправления):

$$N_{\oplus} = \begin{cases} 2(n-1) & - \text{при} \quad \beta_1 = 1, \quad \beta_2 = 0, \\ 2n & - \text{при} \quad \beta_1 \neq 1, \quad \beta_2 \neq 0; \end{cases} \qquad N_{\otimes} = \begin{cases} n-1 & - \text{при} \quad \beta_1 = 1, \quad \beta_2 = 0, \\ n & - \text{при} \quad \beta_1 \neq 1, \quad \beta_2 \neq 0; \end{cases} \qquad N_{\ominus} = 1.$$

Выводы

Предложенные для исправления ошибок коды Лагранжа имеют простые алгоритмы декодирования. Так при исправлении стираний декодирование сводится к вычислению стёртых символов посредством алгоритмов кодирования. Разработанные алгоритмы декодирования дают возможность выполнить их программную и аппаратную реализацию.

Список литературы

1. *Питерсон У., Уэлдон Э.* Коды, исправляющие ошибки: Пер. с англ./Под ред. Р.Л. Добрушина и С.И. Самойленко. – М.: Мир, 1976. – 596 с.

2. *Luby* M. LT Codes // Proc. of the 43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS). – 2002. – P. 271-282.

3. *Амербаев В. М.* Теоретические основы машинной арифметики. – Алма-Ата: Наука, 1976. - 324 с.

4. *Кубицкий В.И.* Процедуры декодирования в каналах с ошибками и стираниями. – Проблеми інформатизації та управління: Збірник наукових праць: Випуск 4 (15). – К.: НАУ, 2005. - С. 118-122.

5. *Кубицкий В.И.* Процедуры кодирования и декодирования для полиномиальных кодов. – Сб. научных трудов «Эксплуатация программного обеспечения систем реального времени, построенных на базе микро- и мини-ЭВМ». – Киев: КИИГА, 1989. - С. 67-71.