Discover What's Possible™ Anritsu

TECHNICAL NOTE
- FEC Function -

MP1590A
Network Performance Tester

ANRITSU CORPORATION

1: FEC (Forward Error Correction)

1-1 Preface

The reliability for information which we handle usually is different by uses. For example, the information regarding money in bank ATMs requires an extremely high degree of reliability. (A serious problem will occur if the amount of money actually withdrawn differs from the information that is sent to the bank's database.) Moreover, high reliability is needed when exchanging data with a personal computer. (A file may not be opened if the error is contained in the copied file data.) On the other hand, audio and video do not require such a high degree of reliability. We will still be able to hear or see the audio and video correctly even if a few errors or noise occurs.

To increase the reliability when communicating with a personal computer, the sender is requested to resend the data if an error is detected in the received data. The method used to increase this reliability during such two-way exchange is referred to as ARQ (Automatic Repeat reQuest). Because resend requests cannot be issued for audio, any detected errors are corrected based on the received data. The method used to increase this reliability during such one-way exchange is referred to as FEC (Forward Error Correction).

A R Q　　　F E C

Figure 1 ARQ (Automatic Repeat Request) and FEC (Forward Error Correction)

Why are communication environments that use error correction techniques required for communication applications?

A communication environment where errors rarely occur can be constructed using large amounts of equipment and huge amounts of electrical power. However, such a communication environment will require an extensive installation space and will be very expensive. In the current business environment, where priority is placed on studying ways to reduce costs and the amount of equipment, signal error correction techniques that can bring about these reductions are extremely important.

For example, consider communications with an earth satellite. The main power-sources of earth satellites are solar batteries. In addition, earth satellites need to be small so they can be launched into orbit. At the same time, communication with satellites requires a high degree of reliability. With these conditions as the background, error correction techniques that offer a high degree of reliability are used for satellite communications.

The same is being studied for long-distance communications such as that using Submarine cables. If the power consumption is large or large amounts of equipment are used, signals can be transmitted using the minimum number of relays points. However, equipment costs will be higher. Reducing the electric power and avoiding large amounts of equipment necessitates the installation of numerous relay points to prevent signals from deteriorating and errors occurring. Equipment costs will of course be higher. If error correction techniques are used, however, data can be transmitted over a fairly long distance, the number of relay points reduced, and equipment costs curtailed.

1-2 Theory of Error Correction

To correct an error, the error must first be detected. Documents that describe error correction always handle the matter as Error Detection → Error Correction.

Here, an example of a conversation using cell phones is used to describe the flow of error correction.

During the conversation, noise occurs time and again or the connection is interrupted. At this time, the listener is probably thinking something like, “What did they say just now?” or “I didn’t catch that.” This is equivalent to error detection. When the connection is subsequently restored, the listener might then say to the talker, “Could you repeat that again?” (thinking that the missed conversation was important). Alternatively, the listener might simply continue the conversation thinking that, “Oh, the talker probably said this.” based on the conversation so far or the listener’s knowledge. Both actions are equivalent to error correction where the first action is equivalent to ARQ while the second action is equivalent to FEC.

We, as human beings, perform these actions naturally without being conscious of detecting and correcting errors.

Now, following explain FEC. Among the codes classified by code theory, following will touch upon systematic and non-systematic codes.

The signals used for error correction are called codes. Generating these codes is called coding. In addition, restoring received codes to their original form is called decoding.

To detect and correct errors, redundant data is required in addition to the actual data (in the cell phone conversation example, this redundant data is equivalent to the conversation so far or the listener’s knowledge). The send and receive sections both have the same redundant data, which is used for encoding and decoding.

For non-systematic codes (i.e., Hamming codes), assuming d(x) as the data to be sent, g(x) as the redundant data, and t(x) as the code to be sent, this is represented as follows:

$t(x) = d(x) \cdot g(x)$ ---(1)

Code t(x) is changed to data completely different from d(x). Using g(x), this is calculated as follows in the receive section:

$t(x) / g(x) = Q(x) + r(x)$ ---(2)

Q(x) is the quotient of division, and r(x) is equivalent to the remainder. For r(x), the following is judged and errors detected:

$r(x) = 0$ → No error

$r(x) \neq 0$ → Error

For an error ($r(x) \neq 0$), a judgment is made on the received code t(x). The judgment decides which code expected by the receiver is closest to the received code. The code t(x) is then replaced with the closest code. This replacement operation constitutes error correction.

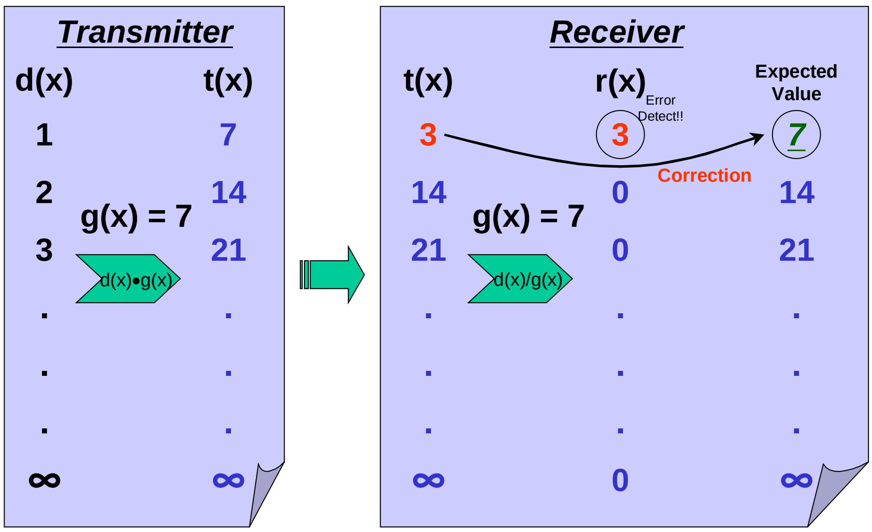

Figure 2 Non-systematic codes

For systematic codes (i.e., Reed-Solomon codes), assuming d(x) as the data to be sent, g(x) as the redundant data, t(x) as the code to be sent, and r(x) as the remainder of division, this is represented as follows:

$$r(x) = d(x) \cdot x^m \bmod g(x) \quad \text{---(3)}$$

$$t(x) = d(x) \cdot x^m + r(x) \quad \text{---(4)}$$

$d(x) \cdot x^m$ means d(x) plus an m bit.

When represented as a figure, formula (4) can be represented with the data and parity detection sections separate. The data section is assigned to d(x).

Using g(x), this is calculated as follows in the receive section:

$$t(x) / g(x) = Q(x) + r(x) \quad \text{---(5)}$$

Q(x) is the quotient of division, and r(x) is equivalent to the remainder. For r(x), the following is judged and errors detected:

$r(x) = 0$ → No error

$r(x) \neq 0$ → Error

For an error ($r(x) \neq 0$), r(x) is error location data called a syndrome and the error is corrected for code t(x).

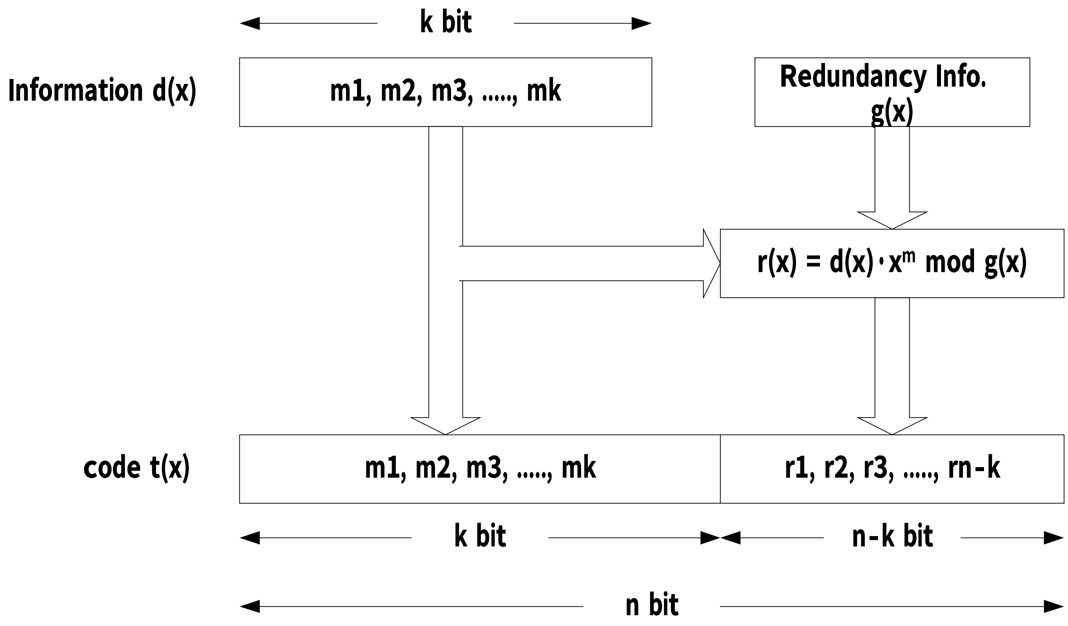

Figure 3 Systematic codes

The OTU frames standardized by ITU-T G.709 are systematic codes that utilize Reed-Solomon codes.

1-3 Reed-Solomon (RS: Reed-Solomon) Codes

Reed-Solomon codes consist of units called blocks (codewords) that consist of multiple symbols. Symbols are units used to separate data and consist of multiple bits.

Errors are detected and corrected using this block unit. Because Reed-Solomon codes are systematic codes, error correction is performed based the explanation given above. The r(x) obtained in formula (5) becomes the syndrome. This data is used to determine the presence or absence of an error for the symbol within the target block and then calculate the size of the error. Moreover, the symbol itself is corrected based on the size of the symbol.

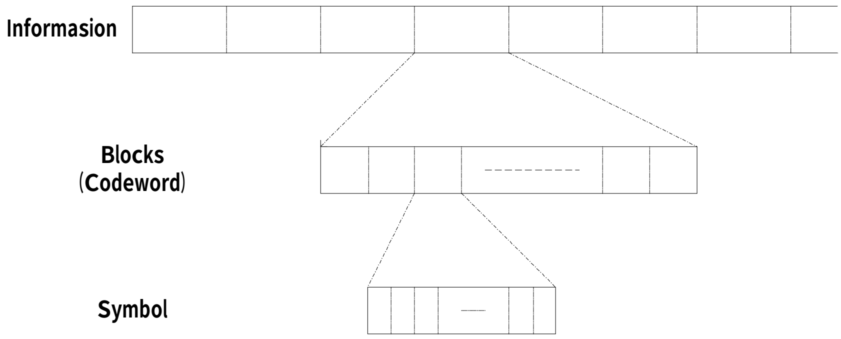

Figure 4 Code word

A graph like the one shown in Figure 5 is frequently used to represent the error correction capability of codes. This figure shows the performance when Reed-Solomon codes RS (255, 239) are used.

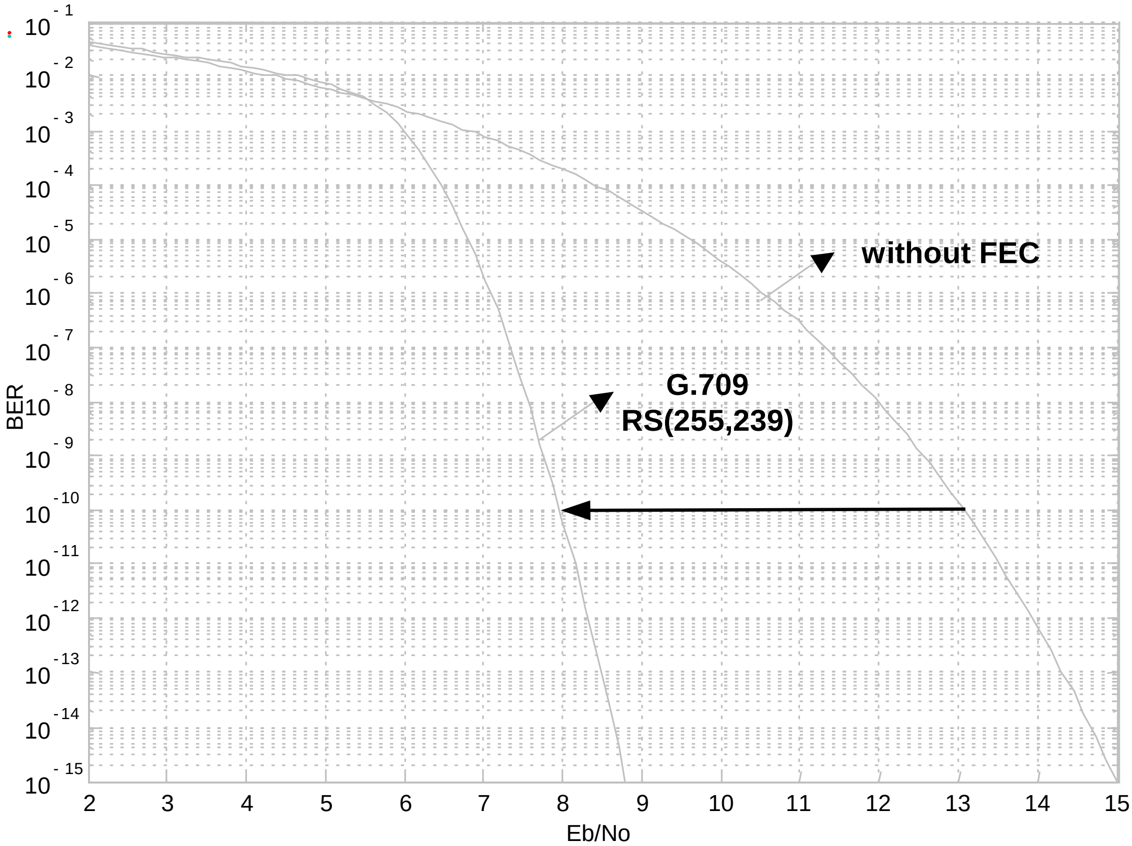

Figure 5 BER vs E_b/N_0

The horizontal axis represents the signal-to-noise ratio (Eb: Energy per bit, No: Noise power).
The vertical axis is the bit error rate.
As the signal energy increases against the noise, it becomes harder to generate errors. In contrast, as the noise increases against the signal energy, it becomes easier to generate errors.
The figure 5 shows a comparison of when errors are corrected and when errors are not corrected. The figure 5 shows that it is difficult to generate errors when errors are corrected. The figure 5 also shows that when Reed-Solomon codes are used, most errors cannot be corrected when the signal-to-noise ratio is very low.

2:ITU-T G.709 OTN FEC

2-1 Correction Capability of Reed-Solomon Codes RS (255, 239)

The FEC codes for OTU frames prescribed by ITU-T G.709 are Reed-Solomon codes defined by RS (255, 239). One block (one code word) consists of 255 bytes. One block is configured from

239 data bytes and 16 FEC parity check bytes.

This code can be used to correct errors of up to eight symbols (one symbol consists of eight bits) per block.

2-2 Relation to OTU Frame Format

Each OTU frame has four OTU rows. Each OTU row has 16 FEC sub-rows. The FEC sub-rows are grouped by byte-interleaving the OTU rows.

One FEC sub-row consists of 255 bytes (239 data bytes and 16 FEC parity check bytes). This FEC sub-row is then defined as one block for a Reed-Solomon code.

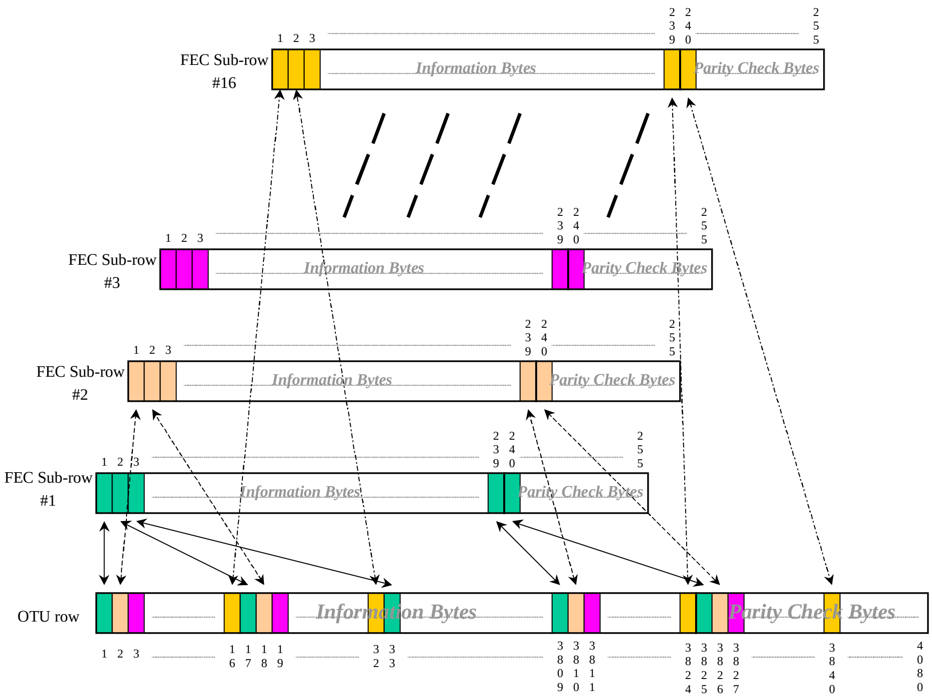

Figure 6 OTU frame interleave

Even if a burst error occurs for an OTU frame, byte-interleaving of the OTU rows can be used to distribute the error to the 16 FEC sub-rows and improve the efficiency of error correction.

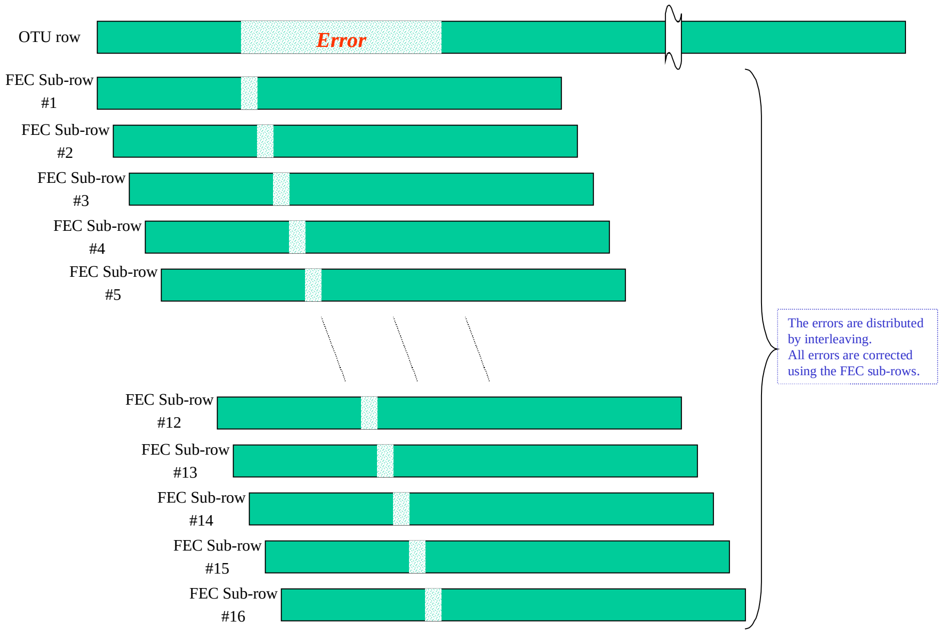

Figure 7 Distribution of burst errors to FEC Sub-rows

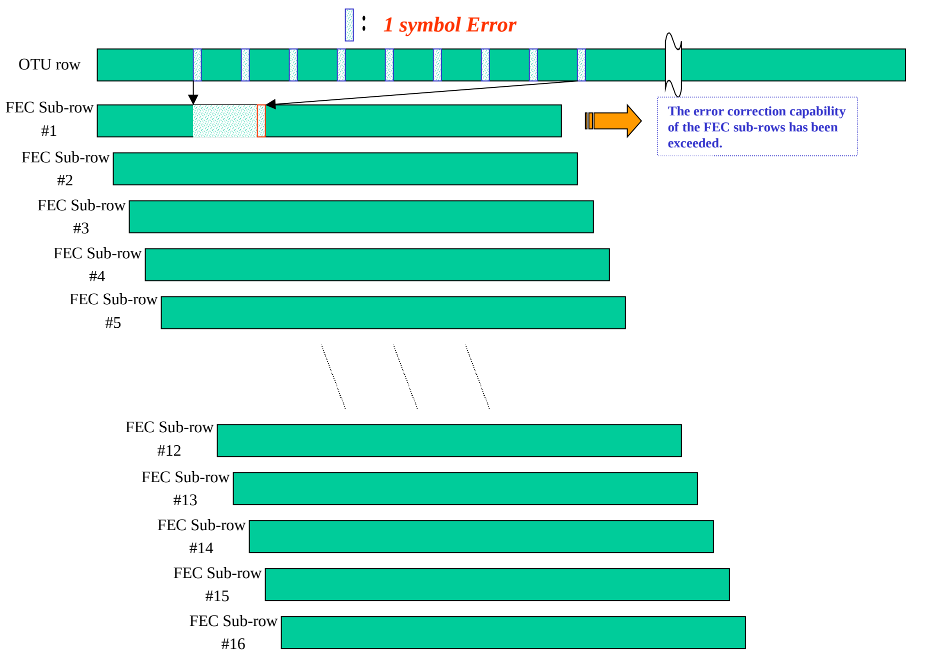

Figure 8 Uncorrectable error status

You might ask, "But won't an uncorrectable error occur if an error is inserted in the area of the FEC parity check bytes?"

The answer is that because Reed-Solomon codes are systematic codes as described above, when they are represented using a figure, they can only be divided into data bytes and FEC parity check bytes. In the receive section, error detection is performed using code as a single code without this distinction being made. Therefore, even if an error is inserted in the FEC parity check byte area, error correction can be performed correctly.

3 MP1590A FEC Function

The MP1590A provides an OTN FEC function based on ITU-T G.709.

The MP1590A supports an ON/OFF function separately for FEC encoding and decoding, which is useful when applying FEC codes unique to the DUT to an OTU frame. (The FEC parity detection area becomes all 0.)

The MP1590A also provides a random error insertion function for evaluating the FEC function. A random error based on Poisson distribution can be inserted for the entire OTU frame to evaluate the performance of the FEC decoder. Because the MP1590A supports error rates from 0.1×10^{-10} to 1.0×10^{-2}, evaluations that are close to the actual line can be performed. The insertion of random errors is essential for evaluating the FEC performance.

For the test function, the total "Correct" is displayed for the error-corrected bits and total "Uncorrected" is displayed for the blocks for which an error exceeding the correction capability was detected. The error for a block for which an error exceeding the correction capability was detected is displayed as a bit error. This value becomes the reference value for numerically comparing the theoretical value curve that represents the FEC performance.

The FEC function can be used in the same way even when the MP1590A is in through mode. When encoding and decoding are turned off, the signals from the DUT are passed through as is and output. When encoding and decoding are turned on, error correction is performed on the signals from the DUT. The signals that are passed through are encoded again and then output. Random errors can also be inserted regardless of whether encoding and decoding are turned on or off.

(In through mode, encoding and decoding cannot be turned on or off independently.)

4 FAQ

[1] Why are error-correcting codes necessary?
> Error correction techniques are used to reduce equipment costs and scale.

[2] Won't codes that have built-in redundancy have an adverse affect?
> This can occur depending on the conditions. The efficiency of data transmission will be reduced. However, sufficient error correction can be performed to make up for this.

[3] What will happen if the redundant bits themselves are incorrect?
> Error correction will be performed without any problem.

[4] What is the benefit of encoding?
> For observed error rates, the benefit is determined from the difference based on the presence or absence of error correction.

[5] Assuming that 1 dB is gained by encoding, how much would that be worth?
> When converted into money, it would be a significant amount.

[6] Won't error correction be performed by mistake if an error is overlooked or no error has occurred?
> If an error occurs and the error is converted to another value, the error will be overlooked. Error correction will not be performed if no error occurs.

[7] Will error correction itself be incorrect?
> This can occur sometimes. If an error exceeding the correction capability occurs, it is sometimes corrected to a different code.

[8] It is often said that digital is never incorrect. Is that true?
> That statement is false. Compared to analog, however, the number of errors is extremely low and the degree of reliability is higher.

[9] What is the difference between BCH and RS codes?
> In a strict sense, BCH codes are two-dimensional BCH codes. RS codes represent non-two-dimensional BCH codes.
In a wider sense, RS codes are also included as BCH codes.

[11] I know that there are block codes and convolutional codes, but what are their differences and characteristics?
> Block codes have a specific delimiter.
Convolutional codes are useful when the BER status is poor. Basically, they do not have a delimiter.

[12] I've heard that contiguous codes are used for systems for which performance is important, but why does performance improve when contiguous codes are used?
> Each code has a BER area, which is ideal for error correction. By combining and using codes, error correction for a wide range of BERs can be performed.

[13] Does the word code have any specific meaning?

> Yes. It means a data string that has a specific property.
In addition to its use in error-correction, the word is also used in other areas such as compression.

[14] I've heard that most codes are linear codes. What does linear mean?
> Codes that have linearity are represented in linear space.

[15] The terms "number of bits error-correcting code" and "number of symbols error-correcting code" are often used. What are their precise meanings?
> They refer to the number of bits or symbols that can be corrected with certainty.
However, if an error exceeding the correction capability occurs, the symbol will sometimes be corrected to a different symbol.

[16] For example, for a three-bit error-correcting code, is it possible for all errors of up to three bits to be corrected 100% of the time?
> Yes, it is.

[17] Is it possible for errors that exceed the correction capability to be corrected?
> It depends on the conditions.

[18] I'm not sure of the difference between symbols and bits. What do they mean?
> A symbol is a group of bits. In most cases, one symbol equals eight bits.
RS codes are used for detecting incorrect symbols and correcting the symbol based on the size (value) of the symbol.

[19] It is said that RS codes are powerful against burst errors. What is the reason for that?
> RS codes use the concept of symbols for error correction.
Therefore, a continuous error of up to the number of correctable symbols multiplied by the number of bits in one symbol can be corrected. In the RS (255, 239) example, errors of up to 64 bits can be corrected.

[20] Recently, I've been hearing the terms “Turbo codes” and “repetitive decoding.” What do these terms mean?
> There are no error-correcting codes referred to as Turbo codes.

[21] Are FEC techniques being used for other networks beside OTN networks?
> FEC techniques are also being used for wireless links and W-LANs.

(Blank Page)

ANRITSU CORPORATION
1800 Onna, Atsugi-shi, Kanagawa, 243-8555 Japan
Phone: +81-46-223-1111
Fax: +81-46-296-1264

• U.S.A.
ANRITSU COMPANY
TX OFFICE SALES AND SERVICE
1155 East Collins Blvd., Richardson, TX 75081, U.S.A.
Toll Free: 1-800-ANRITSU (267-4878)
Phone: +1-972-644-1777
Fax: +1-972-644-3416

• Canada
ANRITSU ELECTRONICS LTD.
700 Silver Seven Road, Suite 120, Kanata,
ON K2V 1C3, Canada
Phone: +1-613-591-2003
Fax: +1-613-591-1006

• Brasil
ANRITSU ELETRÔNICA LTDA.
Praca Amadeu Amaral, 27 - 1 andar
01327-010 - Paraiso, Sao Paulo, Brazil
Phone: +55-11-3283-2511
Fax: +55-11-3886940

• U.K.
ANRITSU LTD.
200 Capability Green, Luton, Bedfordshire LU1 3LU, U.K.
Phone: +44-1582-433280
Fax: +44-1582-731303

• Germany
ANRITSU GmbH
Grafenberger Allee 54-56, 40237 Düsseldorf, Germany
Phone: +49-211-96855-0
Fax: +49-211-96855-55

• France
ANRITSU S.A.
9, Avenue du Québec Z.A. de Courtaboeuf 91951 Les Ulis Cedex, France
Phone: +33-1-60-92-15-50
Fax: +33-1-64-46-10-65

• Italy
ANRITSU S.p.A.
Via Elio Vittorini, 129, 00144 Roma EUR, Italy
Phone: +39-06-509-9711
Fax: +39-06-502-2425

• Sweden
ANRITSU AB
Fagelviksvagen 9E S145 84 Stockholm, Sweden
Phone: +46-853470700
Fax: +46-853470730

• Singapore
ANRITSU PTE LTD.
10, Hoe Chiang Road #07-01/02, Keppel Towers,
Singapore 089315
Phone: +65-6282-2400
Fax: +65-6282-2533

• Hong Kong
ANRITSU COMPANY LTD.
Suite 923, 9/F., Chinachem Golden Plaza, 77 Mody Road, Tsimshatsui East, Kowloon, Hong Kong, China
Phone: +852-2301-4980
Fax: +852-2301-3545

• P. R. China
ANRITSU COMPANY LTD.
Beijing Representative Office
Room 1515, Beijing Fortune Building, No. 5 North Road, the East 3rd Ring Road, Chao-Yang District Beijing 100004, P.R. China
Phone: +86-10-6590-9230

• Korea
ANRITSU CORPORATION
8F Hyun Juk Bldg. 832-41, Yeoksam-dong,
Kangnam-ku, Seoul, 135-080, Korea
Phone: +82-2-553-6603
Fax: +82-2-553-6604

• Australia
ANRITSU PTY LTD.
Unit 3/170 Forster Road Mt. Waverley, Victoria, 3149, Australia
Phone: +61-3-9558-8177
Fax: +61-3-9558-8255

• Taiwan
ANRITSU COMPANY INC.
7F, No. 316, Sec. 1, NeiHu Rd., Taipei, Taiwan
Phone: +886-2-8751-1816
Fax: +886-2-8751-1817

031113

Printed on 100% Recycled Paper

No. MP1590Afec-E-E-1-(1.00) 公知 Printed in Japan 2004-3 AKD