CODEWORD SYNCHRONIZATION METHOD, RECEIVER, NETWORK DEVICE, AND NETWORK SYSTEM

TECHNICAL FIELD

[0001] This application relates to the communication field, and in particular, to a codeword synchronization method, a receiver, a network device, and a network system.

BACKGROUND

[0002] With improvement of a production process level, channel loss and noise have become key factors that limit a data transmission rate and distance. Emergence of forward error correction (Forward Error Correction) provides error correction protection for data in transmission, thereby increasing a data transmission rate and distance of a channel. The FEC may be classified into a block code (block code) and a convolutional code (convolutional code) based on different processing manners for information sequences. The block code may be further sub-classified into a linear block code (linear block code) and a non-linear block code (non-linear block code). Because encoding and decoding implementation of the linear block code is simple, the linear block code is widely used at a physical layer (physical layer) and a media access control (media access control) sublayer of a data link layer (data link layer) in an Open Systems Interconnection model (Open System Interconnection Model, OSI) of Ethernet.

[0003] Error detection and error correction functions of the linear block code need to be implemented based on a complete codeword (codeword). Therefore, a codeword boundary needs

CLAIMS

1. A codeword synchronization method, wherein the method comprises:

step 1: receiving a data sequence, wherein the data sequence comprises a plurality of bits;

step 2: determining a candidate bit in the data sequence, wherein the candidate bit is comprised in the plurality of bits; and

step 3: determining a synchronization position based on the candidate bit, wherein the synchronization position indicates a start position of a codeword in the data sequence.

2. The method according to claim 1, wherein step 3 comprises:

performing verification on the candidate bit, and when the verification succeeds, determining that a position of the candidate bit is the synchronization position.

3. The method according to claim 2, wherein step 3 comprises:

obtaining at least one second test data block from the data sequence through division based on the candidate bit, wherein the position of the candidate bit is a start position of the at least one second test data block; and

performing verification on a characteristic value of the at least one second test data block, and when the verification succeeds, determining that the position of the candidate bit is the synchronization position.

4. The method according to claim 3, wherein

the performing verification on a characteristic value of the at least one second test data block comprises:

sequentially accumulating a characteristic value of each of the at least one second test data block to obtain a cumulative value, wherein when the cumulative value meets a synchronization condition, the verification succeeds.

5. The method according to claim 3, wherein

the performing verification on a characteristic value of the at least one second test data block comprises:

adding up characteristic values of all of the at least one second test data block to obtain a total value, wherein when the total value meets a synchronization condition, the verification succeeds.

6. The method according to claim 1, wherein step 3 comprises:

using a position of the candidate bit as the synchronization position.

7. The method according to claim 1, wherein step 2 comprises:

selecting a plurality of observed bits from the data sequence; and

selecting the candidate bit from the plurality of observed bits.

8. The method according to claim 7, wherein the selecting the candidate bit from the plurality of observed bits comprises:

determining a plurality of groups of first test data blocks in the data sequence based on the plurality of observed bits, wherein each of the plurality of groups of first test data blocks comprise at least one first test data block, and a position of each of the plurality of observed bits is a start position of each of the plurality of groups of first test data blocks; and

selecting one observed bit from the plurality of observed bits as the candidate bit based on characteristic values of the plurality of groups of first test data blocks.

9. The method according to claim 7 or 8, wherein

the selecting a plurality of observed bits from the data sequence comprises:

selecting one bit from the data sequence at an interval of T bits, and using the bit as the observed bit, wherein T is an integer greater than 0; or

selecting one bit from the data sequence at an interval of L×n+T bits, and using the bit as the observed bit, wherein L is a quantity of test data blocks in the interval, a length of the test data block is n bits, and L and T are integers greater than 0.

10. The method according to claim 7 or 8, wherein the data sequence is a modulated signal, and the data sequence comprises a plurality of modulated symbols; and

the selecting a plurality of observed bits from the data sequence comprises:

selecting a start bit of a modulated symbol from the data sequence at an interval of T modulated symbols, and using the start bit as the observed bit, wherein T is an integer greater than 0; or

selecting a start bit of a modulated symbol from the data sequence at an interval of L×m+T modulated symbols, and using the start bit as the observed bit, wherein L is a quantity of test data blocks in the interval, a length of the test data block is m modulated symbols, and L and T are integers greater than 0.

11. The method according to claim 7 or 8, wherein a quantity of observed bits is P, P is a

positive integer, and a length of the codeword is P bits.

12. The method according to any one of claims 3 to 5, wherein

the determining a candidate bit in the data sequence in step 2 comprises: determining a candidate bit in a first subsequence; and the obtaining at least one second test data block from the data sequence through division based on the candidate bit in step 3 comprises: obtaining at least one second test data block from a second subsequence through division based on the candidate bit, wherein the first subsequence and the second subsequence are comprised in the data sequence, and the second subsequence is the same as, partially the same as, or different from the first subsequence.

13. The method according to any one of claims 1 to 12, wherein after step 3, the method further comprises:

step 4: in response to that the data sequence is in an out-of-lock state, determining an updated synchronization position of the data sequence.

14. The method according to claim 13, wherein the determining an updated synchronization position of the data sequence comprises:

performing step 2 and step 3 again, and using a synchronization position determined in step 3 that is performed again as the updated synchronization position.

15. The method according to claim 13 or 14, wherein after step 3 and before step 4, the method further comprises:

obtaining a plurality of synchronization codewords from the data sequence through division based on the synchronization position, wherein the synchronization position is a start position of the plurality of synchronization codewords; and

performing verification on characteristic values of the plurality of synchronization codewords, and when the verification fails, determining that the data sequence is in the out-of-lock state.

16. The method according to claim 15, wherein

the performing verification on characteristic values of the plurality of synchronization codewords comprises:

sequentially accumulating a characteristic value of each of the plurality of synchronization codewords to obtain a cumulative value, wherein when the cumulative value meets an out-of-lock condition, the verification fails.

17. The method according to claim 15, wherein

the performing verification on characteristic values of the plurality of synchronization

codewords comprises:

adding up characteristic values of all of the plurality of synchronization codewords to obtain a total value, wherein when the total value meets an out-of-lock condition, the verification fails.

18. The method according to any one of claims 1 to 17, wherein the data sequence is a linear block code.

19. A communication device, wherein the communication device comprises a processor, a communication interface, and a memory, the memory may be configured to store program code, and the processor is configured to invoke the program code from the memory to perform the method according to any one of claims 1 to 18.

20. A chip, wherein the chip comprises a memory and a processor, the memory is configured to store a computer program, and the processor is configured to invoke the computer program from the memory and run the computer program, to perform the method according to any one of claims 1 to 18.