software development kit (SDK) in connection with the blockchain network for accessing and communicating with the blockchain network. The primary node 404 and one or more backup nodes 406 can also be referred to as consensus nodes or network nodes that achieve consensus and immutably record information in the blockchain network.

[00101] Process 400 starts at 408 where the client node 402 generates a transaction request. In some implementations of the present specification, the transaction request can include a request requesting a blockchain service from the blockchain network 102 and/or 212.

[00102] At 410, the client node 402 multicasts the transaction request to the primary node 404 of the blockchain network 102 and/or 212. In some implementations of the present specification, the primary node 404 assigns a sequence number to the transaction request to keep track of transaction requests after receiving the transaction request from the client node 402.

[00103] At 412, the primary node 404 generates a number of EC blocks after receiving the transaction request from the client node 402. In some implementations of the present specification, the primary node 404 generates the number of EC blocks according to an EC code using the transaction request. For example, referring to FIG. 5, the primary node 404 applies an EC code 504 on a transaction request 502 and transforms the transaction request 502 into an EC message 506 using the EC code 504. The EC code 504 is a forward error correction (FEC) code under the assumption of bit erasures. The EC code 504 transforms the original transaction request 502 into a longer EC message 506 such that the original transaction request 502 can be recovered from a portion or a fragment of the EC message 506.

[00104] In some implementations of the present specification, the EC code 504 is a near-optimal erasure code, such as, a Tornado code or a low-density parity-check code. In alternative implementations of the present specification, the EC code 504 is a near optimal fountain code, such as, a fountain code, an online code, a Luby transform (LT) code, or a raptor code. In alternative implementations of the present specification, the EC code 504 is an optimal erasure code, such as, a parity code, a Parchive code, a Reed-Solomon code, or a regenerating code. In some implementations of the present specification, the EC code 504 can be any suitable type of erasure code.

change process discussed herein provides liveness and reduces network latency by allowing the system to make progress when the primary node fails.

[00134] At 808, the backup node 802 determines a respective weight of the backup node 802 associated with each of phases of the consensus process in the current epoch. In some implementations, the consensus process includes three phases as described above with reference to FIGS. 3-6. The weight is a metric of a qualification of the backup node 802 to be the new primary node in a new epoch.

[00135] In some implementations of the present specification, the backup node 802 determines a weight of the backup node 802 for a first phase of the consensus process in the current epoch to be a first value. For example, the backup node 802 may be assigned an initial weight of 10% if the backup node 802 has entered a first phase of the consensus process (e.g., the first phase 310 of the consensus process 300). In alternative implementations of the present specification, the backup node 802 may assign any suitable weight value to the backup node 802 for the first phase of the current consensus process.

[00136] In some implementations of the present specification, the backup node 802 determines a weight of the backup node 802 for a second phase of the consensus process (e.g., the second phase 320 of the consensus process 300) in the current epoch based on a quorum verification process. The quorum verification process is performed by determining whether the backup node 802 receives a predetermined number (e.g., 2f+1) of ECHO messages from the other network nodes in the second phase of the consensus process.

[00137] In some implementations of the present specification, if the backup node 802 fails the quorum verification (e.g., the backup node 802 receives a number of ECHO messages that is lower than a predetermined threshold), the backup node 802 may determine the weight of the backup node 802 for the second phase of the consensus process to be a first value. If the backup node 802 passes the quorum verification (e.g., the backup node 802 receives a number of ECHO messages that equals to or exceeds a predetermined threshold), the backup node 802 may determine the weight of the backup node 802 for the second phase of the consensus process to be a second value. In some implementations of the present specification, the second value is determines as larger

implementations, the second value is determines as larger than the first value. For example, if the backup node 802 fails the quorum verification, the backup node 802 can be assigned a weight value of zero to the backup node 802 for the third phase of the consensus process. If the backup node 802 passes the quorum verification, the backup node 802 can be assigned a weight value of 45% to the backup node 802 for the third phase of the consensus process. However, in alternative implementations of the present specification, the backup node 802 may assign any suitable value to the backup node 802 for the third phase of the consensus process in the current epoch.

[00141] At 810, after determining the respective weights of the backup node 802 for the phases of the consensus process in the current epoch, the backup node 802 determines a weight sum of the backup node 802 for the consensus process based on the respective weights. In some implementations of the present specification, the weight sum is a sum of the respective sums of the backup nodes associated with each of the phases of the consensus process in the current epoch. For example, if the backup node 802 has determined a first weight value of the backup node 802 for the first phase to be 10%, a second weight value of the backup node 802 for the second phase to be 45%, and a third weight value of the backup node 802 for the third phase to be 45%, the backup node 802 determines the weight sum to be 100%. As another example, if the backup node 802 has determined a first weight value of the backup node 802 for the first phase to be 10%, a second weight value of the backup node 802 for the second phase to be 45%, and a third weight value of the backup node 802 for the third phase to be 0, the backup node 802 determines the weight sum to be 55%.

[00142] At 812, the backup node 802 sends an EPOCH_CHANGE message to the other network nodes 804 if the backup node 802 determines that the weight sum that was determined at 810 reaches or exceeds a predetermined threshold. For example, the backup node 802 may send an EPOCH_CHANGE message to the other network nodes 804 if the weight sum as determined at 810 reaches 100%. The EPOCH_CHANGE message indicates a request for a change from the current epoch with the current primary node to the new epoch with the backup node being the new primary node.

[00143] In some implementations of the present specification, referring to FIG. 9, the EPOCH_CHANGE message has a format of <weight, epoch+1, ECHO{}, ACCEPT{},

node to request ECHO messages from the other network nodes. These operations are performed such that the recovery process of the faulty network node does not interfere with the normal operation of the consensus process among the other non-faulty network nodes. This facilitates conserving computing and network resources for recovering the faulty network node by reducing the complexity of the recovery process.

[00173] Referring to FIG. 13, FIG. 13 is a diagram illustrating modules of a consensus apparatus 1300, according to an implementation of the present specification. The apparatus 1300 for achieving consensus can be applied to a consensus system based on a blockchain technology. For example, the apparatus 1300 can correspond to the implementations shown in FIGS. 1-6. The apparatus 1300 can be implemented in a primary node in blockchain network. The apparatus 1300 includes the following: a receiver or receiving unit 1302, configured to receive a transaction request; a generating unit 1304, configured to generate a number of erasure code (EC) blocks according to an EC code using the transaction request; a transmitter or transmitting unit 1306, configured to send a number of first messages to the one or more backup nodes, respectively, wherein each of the number of first messages includes a composite hash value associated with the number of EC blocks; the receiver or receiving unit 1302, further configured to receive at least one second message from at least one of the backup nodes, wherein the at least one second message includes one of the number of first messages and a signature of the at least one of the backup nodes associated with the one of the number of first messages; a verifying unit 1308, configured to verify whether the at least one second message is valid in response to receiving the at least one second message from the at least one of the backup node; a determining unit 1310, configured to determine whether a number of valid second messages exceeds a pre-determined threshold; a reconstructing unit 1312, configured to reconstruct the transaction request based on a subset of the number of valid second messages according to the EC code in response to determining that the number of valid second messages exceeds the pre-determined threshold; the transmitter or the transmitting unit 1306, further configured to send a third message, to the other network nodes in response to determining that the transaction request has been successfully reconstructed, wherein the third message includes a set of signatures that are in the valid second messages; the receiver or the receiving unit 1302, further configured

[00240] Optionally, the EPOCH_CHANGE message includes a weight sum associated with the backup node and a set of signatures associated with a set of network nodes out of the number of network nodes.

[00241] Optionally, the verifying whether the EPOCH_CHANGE message is valid includes verifying whether the weight sum in the EPOCH_CHANGE message is valid, and the verifying whether the weight sum in the EPOCH_CHANGE message is valid includes verifying whether the set of signatures are valid.

[00242] Optionally, the verifying whether the at least one NEW_EPOCH message is valid includes verifying whether the digest of the EPOCH_CHANGE message in the at least one NEW_EPOCH message is valid, and the verifying whether the digest of the EPOCH_CHANGE message in the at least one NEW_EPOCH message is valid includes verifying whether the set of signatures in the EPOCH_CHANGE message are valid.

[00243] Referring to FIG. 17, FIG. 17 is a diagram illustrating modules of a recovery apparatus 1700, according to an implementation of the present specification. The apparatus 1700 for recovery can be applied to a consensus system based on a blockchain technology. The apparatus 1700 can correspond to the implementations shown in FIGS. 10-12. For example, the apparatus 1700 can be implemented in a network node of a blockchain network. The apparatus 1700 includes the following: a broadcasting unit 1702, configured to broadcast, by a network node of a blockchain network, a state request message to a number of other network nodes of the blockchain network, wherein the network node is to recover a target transaction of a target sequence number; a receiver 1704 or a receiving unit 1704, configured to receive a number of state reply messages from the number of other network nodes, wherein each of the number of state reply messages includes a sequence number; an identifying unit 1706, configured to identify the target sequence number based on the target sequence number in response to determining that a number of state reply messages exceeds a pre-determined threshold, wherein each of the number of the state reply messages comprises a target sequence number; a transmitter 1708 or a transmitting unit 1708, configured to send a requesting message to the number of other network nodes, wherein the requesting message requests an ECHO message from each of the number of other network nodes, wherein the ECHO message is a message transmitted by the each of the number of other network nodes for

achieving a consensus among the number of other network nodes on the target transaction having the target sequence number, and the ECHO message includes a part of the target transaction and a signature of the each of the number of other network nodes; the receiver 1704 or the receiving unit 1704, further configured to receive a number of ECHO messages from the number of other network nodes; a determining unit 1710, configured to determine a number of valid ECHO messages out of the number of ECHO messages, wherein each of the number of valid ECHO messages includes the target sequence number; a recovering unit 1712, configured to recover the target transaction having the target sequence number at the network node based on the number of valid ECHO messages in response to determining that the number of valid ECHO messages exceeds a pre-determined threshold; and the transmitter 1708, further configured to send a message to the number of other network nodes indicating the network node has been recovered.

[00244] In an optional implementation, the number of network nodes includes a primary node and one or more backup nodes.

[00245] In an optional implementation, the network node is a primary node or a backup node.

[00246] In an optional implementation, the requesting message includes the target sequence number.

[00247] In an optional implementation, the recovery apparatus 1700 further includes the following: a verifying unit 1714, configured to verify, by each of the number of other network nodes other than the network node, the requesting message before sending the ECHO messages to the network node.

[00248] In an optional implementation, the verifying unit 1714, is further configured to verify whether each of the ECHO messages is valid, wherein the verifying whether each of the ECHO messages is valid includes verifying whether each of the ECHO messages is valid using a Merkel tree.

[00249] In an optional implementation, the verifying whether each of the ECHO message is valid further includes verifying whether the signature in the ECHO message is valid.

[00250] In an optional implementation, each of the ECHO messages further includes at

least one of a number of erasure code (EC) blocks associated with the target transaction, wherein the number of EC blocks are generated according to an EC code using the target transaction.

[00251] In an optional implementation, the recovering the target transaction having the target sequence number at the network node based on the number of valid ECHO messages comprises reconstructing the target transaction using a subset of the plurality of EC blocks that are in the number of valid ECHO messages.

[00252] In an optional implementation, the message to the number of other network nodes indicating the network node has been recovered include a set of signatures in the number of valid ECHO messages and the target sequence number.

[00253] The system, apparatus, module, or unit illustrated in the previous implementations can be implemented by using a computer chip or an entity, or can be implemented by using a product having a certain function. A typical implementation device is a computer, and the computer can be a personal computer, a laptop computer, a cellular phone, a camera phone, a smartphone, a personal digital assistant, a media player, a navigation device, an email receiving and sending device, a game console, a tablet computer, a wearable device, or any combination of these devices.

[00254] For an implementation process of functions and roles of each unit in the apparatus, references can be made to an implementation process of corresponding steps in the previous method. Details are omitted here for simplicity.

[00255] Because an apparatus implementation basically corresponds to a method implementation, for related parts, references can be made to related descriptions in the method implementation. The previously described apparatus implementation is merely an example. The units described as separate parts may or may not be physically separate, and parts displayed as units may or may not be physical units, may be located in one position, or may be distributed on a plurality of network units. Some or all of the modules can be selected based on actual demands to achieve the objectives of the solutions of the present specification. A person of ordinary skill in the art can understand and implement the implementations of the present application without creative efforts.

[00256] FIG. 17 is a schematic diagram illustrating an internal functional module and a structure of a recovery apparatus 1700. An execution body in essence can be an

electronic device, and the electronic device includes the following: at least one processor; and a memory configured to store an executable instruction of the at least one processor.

[00257] The at least one processor is configured to broadcast, by a network node of a blockchain network, a state request message to a number of other network nodes of the blockchain network, wherein the network node is to recover a target transaction of a target sequence number; receive a number of state reply messages from the number of other network nodes, wherein each of the number of state reply messages includes a sequence number; identify the target sequence number based on the target sequence number in response to determining that a number of state reply messages exceeds a pre-determined threshold, wherein each of the number of the state reply messages comprises a target sequence number; send a requesting message to the number of other network nodes, wherein the requesting message requests an ECHO message from each of the number of other network nodes, wherein the ECHO message is a message transmitted by the each of the number of other network nodes for achieving a consensus among the number of other network nodes on the target transaction having the target sequence number, and the ECHO message includes a part of the target transaction and a signature of the each of the number of other network nodes; receive a number of ECHO messages from the plurality of other network nodes; determine a number of valid ECHO messages out of the number of ECHO messages, wherein each of the number of valid ECHO messages includes the target sequence number; recover the target transaction having the target sequence number at the network node based on the number of valid ECHO messages in response to determining that the number of valid ECHO messages exceeds a pre-determined threshold; and send a message to the number of other network nodes indicating the network node has been recovered.

[00258] Optionally, the number of network nodes include a primary node and one or more backup nodes.

[00259] Optionally, the network node is a primary node or a backup node.

[00260] Optionally, the requesting message includes the target sequence number.

[00261] Optionally, the at least one processor is further configured to verify, by each of the number of other network nodes other than the network node, the requesting message before sending the ECHO messages to the network node.

[00262] Optionally, the at least one processor is further configured to verify whether each of the ECHO messages is valid, wherein the verifying whether each of the ECHO messages is valid includes verifying whether each of the ECHO messages is valid using a Merkel tree.

[00263] Optionally, the verifying whether each of the ECHO message is valid further includes verifying whether the signature in the ECHO message is valid.

[00264] Optionally, each of the ECHO messages further includes at least one of a number of erasure code (EC) blocks associated with the target transaction, wherein the number of EC blocks are generated according to an EC code using the target transaction.

[00265] Optionally, recovering the target transaction having the target sequence number at the network node based on the number of valid ECHO messages includes reconstructing the target transaction using a subset of the number of EC blocks that are in the number of valid ECHO messages.

[00266] Optionally, the message to the number of other network nodes indicating the network node has been recovered includes a set of signatures in the number of valid ECHO messages and the target sequence number.

[00267] Implementations of the subject matter and the actions and operations described in this specification can be implemented in digital electronic circuitry, in tangibly-embodied computer software or firmware, in computer hardware, including the structures disclosed in this specification and their structural equivalents, or in combinations of one or more of them. Implementations of the subject matter described in this specification can be implemented as one or more computer programs, e.g., one or more modules of computer program instructions, encoded on a computer program carrier, for execution by, or to control the operation of, data processing apparatus. The carrier may be a tangible non-transitory computer storage medium. Alternatively, or in addition, the carrier may be an artificially-generated propagated signal, e.g., a machine-generated electrical, optical, or electromagnetic signal that is generated to encode information for transmission to suitable receiver apparatus for execution by a data processing apparatus. The computer storage medium can be or be part of a machine-readable storage device, a machine-readable storage substrate, a random or serial access memory device, or a

WHAT IS CLAIMED IS:

1. A computer-implemented method of performing a change of a primary node in a blockchain network that comprises a plurality of network nodes, wherein the plurality of network nodes comprises the primary node and one or more backup nodes, the method comprising:

determining, by a backup node, that an epoch change needs to be performed, wherein the epoch change causes a change from a current epoch with a current primary node to a new epoch with a new primary node, wherein the current epoch comprises a consensus process for obtaining agreement among the plurality of network nodes using the primary node, the consensus process comprising three phases, wherein the three phases comprises:

a first phase where the current primary node sends a first message to the one or more backup nodes,

a second phase where each backup node of the one or more backup nodes sends a second message to the current primary node, and

a third phase where each network node of the plurality of network nodes sends a third message to other network nodes;

determining, by the backup node, a respective weight of the backup node associated with each of the three phases of the consensus process in the current epoch, wherein the weight is a metric of a qualification of the backup node to be the new primary node;

determining, by the backup node, a weight sum for the backup node based on the respective weight of the backup node associated with each of the three phases in the current epoch;

in response to determining that the weight sum reaches a first predetermined threshold, sending, by the backup node, an EPOCH_CHANGE message to the plurality of network nodes other than the network node, wherein the EPOCH_CHANGE message indicates a request for a change from the current epoch with the current primary node to the new epoch with the backup node being the new primary node, and the EPOCH_CHANGE message comprises the weight sum of the backup node;

receiving, by the backup node, at least one NEW_EPOCH message from at least one of the plurality of network nodes other than the backup node, wherein the NEW_EPOCH message indicates an acknowledgement of the backup node to be the new primary node;

verifying, by the backup node, whether the at least one NEW_EPOCH message is valid;

determining, by the backup node, whether a number of valid NEW_EPOCH messages out of the at least one NEW_EPOCH message exceeds a second predetermined threshold; and

in response to determining that the number of valid NEW_EPOCH messages exceeds the second predetermined threshold, determining, by the backup node, the backup node to be the new primary node in the new epoch,

wherein determining a respective weight of the backup node associated with each of the three phases of the consensus process in the current epoch comprises:

in response to determining a failure of a quorum verification in a third phase of the consensus process in the current epoch, determining a weight of the backup node for the third phase of the consensus process to be a third value, and

in response to determining a success of a quorum verification in the third phase of the consensus process in the current epoch, determining the weight of the backup node for the third phase of the consensus process to be a fourth value, wherein the third value is smaller than the fourth value.

2. The computer-implemented method of claim 1, wherein determining a respective weight of the backup node associated with each of the three phases of the consensus process in the current epoch comprises determining a weight of the backup node for a first phase of the consensus process to be a first value.

3. The computer-implemented method of claim 1, wherein determining a respective weight of the backup node associated with each of the three phases of the consensus process in the current epoch comprises:

in response to determining a failure of a quorum verification in a second phase of the consensus process in the current epoch, determining a weight of the backup node for the second phase of the consensus process to be a first value; and

in response to determining a success of a quorum verification in the second phase of the consensus process in the current epoch, determining the weight of the backup node for the second phase of the consensus process to be a second value, wherein the first value is smaller than the second value.

4. The computer-implemented method of claim 3, wherein the quorum verification in the second phase for the network node comprises receiving a predetermined number of ECHO messages from other network nodes.

5. The computer-implemented method of claim 1, wherein the quorum verification in the third phase for the network node comprises receiving a pre-determined number of accept messages from other network nodes, wherein each of the accept messages from other network nodes indicates each of the other network nodes has accepted a predetermined number of ECHO messages.

6. The computer-implemented method of claim 1, wherein the EPOCH_CHANGE message further comprises a set of signatures associated with a set of network nodes out of the plurality of network nodes, and wherein the NEW_EPOCH message comprises a digest of the EPOCH_CHANGE message.

7. The computer-implemented method of claim 6, wherein verifying whether the at least one valid NEW_EPOCH message is valid comprises verifying whether the digest of the EPOCH_CHANGE message in the at least one NEW_EPOCH message is valid, and wherein verifying whether the digest of the EPOCH_CHANGE message in the at least one NEW_EPOCH message is valid comprises verifying whether the set of signatures in the EPOCH_CHANGE message are valid.

8. The computer-implemented method of claim 1, wherein determining that an epoch change needs to be performed comprises determining that an epoch change needs to be performed in response to determining that the consensus has not been achieved in an old epoch within a predetermined time period.

9. The computer-implemented method of claim 1, further comprising operating in the new epoch with the new primary node, wherein the new epoch comprises a consensus process for achieving consensus among the plurality of network nodes using the new primary node.

10. A non-transitory, computer-readable medium storing one or more instructions executable by a computer system to perform operations comprising:

determining, by a backup node of a blockchain network comprising a plurality of network nodes, that an epoch change needs to be performed, wherein the plurality of network nodes comprise a primary node and one or more backup nodes comprising the backup node, wherein the epoch change causes a change from a current epoch with a current primary node to a new epoch with a new primary node, wherein the current epoch comprises a consensus process for obtaining agreement among the plurality of network nodes using the primary node, the consensus process comprising three phases, wherein the three phases comprises:

a first phase where the current primary node sends a first message to the one or more backup nodes,

a second phase where each backup node of the one or more backup nodes sends a second message to the current primary node, and

a third phase where each network node of the plurality of network nodes sends a third message to other network nodes;

determining, by the backup node, a respective weight of the backup node associated with each of the three phases of the consensus process in the current epoch, wherein the weight is a metric of a qualification of the backup node to be the new primary node;

determining, by the backup node, a weight sum for the backup node based on the respective weight of the backup node associated with each of the three phases in the current epoch;

in response to determining that the weight sum reaches a first predetermined threshold, sending, by the backup node, an EPOCH_CHANGE message to the plurality of network nodes other than the network node, wherein the EPOCH_CHANGE message indicates a request for a change from the current epoch with the current primary node to the new epoch with the backup node being the new primary node, and the EPOCH_CHANGE message comprises the weight sum of the backup node;

receiving, by the backup node, at least one NEW_EPOCH message from at least one of the plurality of network nodes other than the backup node, wherein the NEW_EPOCH message indicates an acknowledgement of the backup node to be the new primary node;

verifying, by the backup node, whether the at least one NEW_EPOCH message is valid;

determining, by the backup node, whether a number of valid NEW_EPOCH messages out of the at least one NEW_EPOCH message exceeds a second predetermined threshold; and

in response to determining that the number of valid NEW_EPOCH messages exceeds the second predetermined threshold, determining, by the backup node, the backup node to be the new primary node in the new epoch,

wherein determining a respective weight of the backup node associated with each of the three phases of the consensus process in the current epoch comprises:

in response to determining a failure of a quorum verification in a third phase of the consensus process in the current epoch, determining a weight of the backup node for the third phase of the consensus process to be a third value, and

in response to determining a success of a quorum verification in the third phase of the consensus process in the current epoch, determining the weight of the backup node for the third phase of the consensus process to be a fourth value, wherein the third value is smaller than the fourth value.

11. The non-transitory, computer-readable medium of claim 10, wherein determining a respective weight of the backup node associated with each of the three phases of the consensus process in the current epoch comprises determining a weight of the backup node for a first phase of the consensus process to be a first value.

12. The non-transitory, computer-readable medium of claim 10, wherein determining a respective weight of the backup node associated with each of the three phases of the consensus process in the current epoch comprises:

in response to determining a failure of a quorum verification in a second phase of the consensus process in the current epoch, determining a weight of the backup node for the second phase of the consensus process to be a first value; and

in response to determining a success of a quorum verification in the second phase of the consensus process in the current epoch, determining the weight of the backup node for the second phase of the consensus process to be a second value, wherein the first value is smaller than the second value.

13. The non-transitory, computer-readable medium of claim 12, wherein the quorum verification in the second phase for the network node comprises receiving a predetermined number of ECHO messages from other network nodes.

14. The non-transitory, computer-readable medium of claim 10, wherein the quorum verification in the third phase for the network node comprises receiving a pre-determined number of accept messages from other network nodes, wherein each of the accept messages from other network nodes indicates each of the other network nodes has accepted a predetermined number of ECHO messages.

15. The non-transitory, computer-readable medium of claim 10, wherein the EPOCH_CHANGE message further comprises a set of signatures associated with a set of network nodes out of the plurality of network nodes, and wherein the NEW_EPOCH message comprises a digest of the EPOCH_CHANGE message.

16. The non-transitory, computer-readable medium of claim 15, wherein verifying whether the at least one valid NEW_EPOCH message is valid comprises verifying whether the digest of the EPOCH_CHANGE message in the at least one NEW_EPOCH message is valid, and wherein verifying whether the digest of the EPOCH_CHANGE message in the at least one NEW_EPOCH message is valid comprises verifying whether the set of signatures in the EPOCH_CHANGE message are valid.

17. The non-transitory, computer-readable medium of claim 10, wherein determining that an epoch change needs to be performed comprises determining that an epoch change needs to be performed in response to determining that the consensus has not been achieved in an old epoch within a predetermined time period.

18. The non-transitory, computer-readable medium of claim 10, wherein the operations further comprise:

operating in the new epoch with the new primary node, wherein the new epoch comprises a consensus process for achieving consensus among the plurality of network nodes using the new primary node.

19. A system, including:

one or more computers; and

one or more computer memory devices interoperably coupled with the one or more computers and having tangible, non-transitory, machine-readable media storing one or more instructions that, when executed by the one or more computers, perform one or more operations comprising:

determining, by a backup node of a blockchain network comprising a plurality of network nodes, that an epoch change needs to be performed, wherein the plurality of network nodes comprise a primary node and one or more backup nodes comprising the backup node, wherein the epoch change causes a change from a current epoch with a current primary node to a new epoch with a new primary node, wherein the current epoch comprises a consensus process for obtaining agreement among the plurality of network

nodes using the primary node, the consensus process comprising three phases, wherein the three phases comprises:

a first phase where the current primary node sends a first message to the one or more backup nodes,

a second phase where each backup node of the one or more backup nodes sends a second message to the current primary node, and

a third phase where each network node of the plurality of network nodes sends a third message to other network nodes;

determining, by the backup node, a respective weight of the backup node associated with each of the three phases of the consensus process in the current epoch, wherein the weight is a metric of a qualification of the backup node to be the new primary node;

determining, by the backup node, a weight sum for the backup node based on the respective weight of the backup node associated with each of the three phases in the current epoch;

in response to determining that the weight sum reaches a first predetermined threshold, send, by the backup node, an EPOCH_CHANGE message to the plurality of network nodes other than the network node, wherein the EPOCH_CHANGE message indicates a request for a change from the current epoch with the current primary node to the new epoch with the backup node being the new primary node, and the EPOCH_CHANGE message comprises the weight sum of the backup node;

receiving, by the backup node, at least one NEW_EPOCH message from at least one of the plurality of network nodes other than the backup node, wherein the NEW_EPOCH message indicates an acknowledgement of the backup node to be the new primary node;

verifying, by the backup node, whether the at least one NEW_EPOCH message is valid;

determining, by the backup node, whether a number of valid NEW_EPOCH messages out of the at least one NEW_EPOCH message exceeds a second predetermined threshold; and

in response to determining that the number of valid NEW_EPOCH messages exceeds the second predetermined threshold, determining, by the backup node, the backup node to be the new primary node in the new epoch,

wherein determining a respective weight of the backup node associated with each of the three phases of the consensus process in the current epoch comprises:

in response to determining a failure of a quorum verification in a third phase of the consensus process in the current epoch, determining a weight of the backup node for the third phase of the consensus process to be a third value, and

in response to determining a success of a quorum verification in the third phase of the consensus process in the current epoch, determining the weight of the backup node for the third phase of the consensus process to be a fourth value, wherein the third value is smaller than the fourth value.

20. The system of claim 19, wherein determining a respective weight of the backup node associated with each of the three phases of the consensus process in the current epoch comprises determining a weight of the backup node for a first phase of the consensus process to be a first value.

21. The system of claim 20, wherein determining a respective weight of the backup node associated with each of the three phases of the consensus process in the current epoch comprises:

in response to determining a failure of a quorum verification in a second phase of the consensus process in the current epoch, determining a weight of the backup node for the second phase of the consensus process to be a first value; and

in response to determining a success of a quorum verification in the second phase of the consensus process in the current epoch, determining the weight of the backup node for the second phase of the consensus process to be a second value, wherein the first value is smaller than the second value.

22. The system of claim 21, wherein the quorum verification in the second phase for the network node comprises receiving a predetermined number of ECHO messages from other network nodes.

23. The system of claim 19, wherein the quorum verification in the third phase for the network node comprises receiving a pre-determined number of accept messages from other network nodes, wherein each of the accept messages from other network nodes indicates each of the other network nodes has accepted a predetermined number of ECHO messages.

24. The system of claim 19, wherein the EPOCH_CHANGE message further comprises a set of signatures associated with a set of network nodes out of the plurality of network nodes, and wherein the NEW_EPOCH message comprises a digest of the EPOCH_CHANGE message.

25. The system of claim 24, wherein verifying whether the at least one valid NEW_EPOCH message is valid comprises verifying whether the digest of the EPOCH_CHANGE message in the at least one NEW_EPOCH message is valid, and wherein verifying whether the digest of the EPOCH_CHANGE message in the at least one NEW_EPOCH message is valid comprises verifying whether the set of signatures in the EPOCH_CHANGE message are valid.

26. The system of claim 19, wherein determining that an epoch change needs to be performed comprises determining that an epoch change needs to be performed in response to determining that the consensus has not been achieved in an old epoch within a predetermined time period.

27. The system of claim 19, wherein the operations further comprise:

operating in the new epoch with the new primary node, wherein the new epoch comprises a consensus process for achieving consensus among the plurality of network nodes using the new primary node.