

field 603 contains the length of the Record LRPDU message 600. The first two octets of a Record LRPDU message 600 data field is a My Portal Number field 609 that encodes the chassis ID, port ID, and appId of the transmitting Portal. This is the same value transmitted in the Hello LRPDU message 500 for the same Portal. Following this are zero or more records in a Record field 630.

5 **[00101]** Each record includes a Record Number field 631, a Sequence Number field 633, a Checksum field 635, and optionally a Data Length field 637 and an Application Data field 639. The Record Number field 631 contains data indicating a record at an ~~application~~-applicant database that has been updated. The Sequence Number field 633 contains data indicating a version number and/or number of changes associated with the corresponding record. The Checksum field 635 includes
10 error correction data, such as a two octet value computed based on the values included in the corresponding record (e.g., values in the Record Number field 631, Sequence Number field 633, Data Length field 637, and/or Application Data field 639). The Data Length field 637, when present, indicates the length of data in the corresponding record and/or the length of the Application Data field 639. The Application Data field 639, when present, contains the updated record under
15 transmission from the applicant database to the registrar database.

[00102] Accordingly, the Record LRPDU message 600 may include one or more record numbers indicating records updated at the applicant database and one or more a sequence numbers identifying updates included in the records updated at the applicant database. Further, the Type field 601 is one octet with an offset of zero octet, the Length field 603 is two octets with an offset of one octets,
20 the My Portal Number field 609 is four octets with an offset of three octets, and the Record field 630 is of variable length with an offset of seven octets. Also, each record contains a Record Number field 631 of four octets with an offset of zero octets, a Sequence Number field 633 of four octets with an offset of four octets, a Checksum field 635 of two octets with an offset of eight octets, a Data Length field 637 of two octets with an offset of ten octets, and an Application Data field 639
25 of zero to sixty five thousand five hundred twenty octets with an offset of twelve octets, where record offsets are measured from the start of the record.

[00103] FIG. 7 illustrates an example encoding of a Partial List LRPDU message 700, which may implement a Partial List LRPDU message 409. As such, Partial List LRPDU message 700 may be employed to maintain database pair synchronization in a native system 110 and/or 130, a proxy 210 and/or 230, a slave 240 and/or 250, and/or a database system 300. The Partial List LRPDU
30 message 700 contains a Type field 701 and Length field 703, which are similar to Type field 501 and Length field 503, respectively. The Type field 701 may

CLAIMS

What is claimed is:

1. A method implemented in a local node, the method comprising:
receiving, at a receiver at the local node, a hello message including an application identifier
5 (AppId) and a target link between a target port for a local node and target port for a neighbor node;
determining, by a processor at the local node, the AppId is associated with an application
for performing database synchronization;
setting up a local database in memory at the local node as part of a database pair for use by
the application, the database pair including a neighbor database at the neighbor node;
10 associating, by the processor, the database pair with the target link; and
controlling synchronization of the local database with the neighbor database via the target
link.
2. The method of claim 1, wherein controlling synchronization of the local database with the
15 neighbor database includes transmitting, by a transmitter at the local node, record information to
the neighbor database via the target link in link-local registration protocol data unit(LRPDU)
messages.
3. The method of any of claims 1-2, wherein the local database includes an applicant database
20 to store local records for transmission to a registrar database in the neighbor database.
4. The method of any of claims 1-3, wherein the local database includes a registrar database
to receive neighbor records from an applicant database in the neighbor database.
- 25 5. The method of any of claims 1-4, wherein the hello message is a hello LRPDU, and
wherein the hello LRPDU represents the target link as a My Chassis identifier (ID) identifying the
local node, a My Port ID identifying a target port on the local node, a neighbor chassis ID
identifying the neighbor node, and a neighbor port ID identifying a target port on the neighbor
node.
30
6. The method of claim 1, further comprising:

transmitting, by a transmitter at the local node, one or more record link-local registration protocol data unit(LRPDU) messages toward a registrar database at the neighbor node, the record LRPDU messages indicating updates to records stored in an applicant database at the local node;
receiving, by a receiver at the local node, one or more LRPDU messages acknowledging
5 the record LRPDU messages;
marking, in a memory at the local node, at least one updated record in the applicant database as acknowledged by the registrar database; and
notifying, via the processor, an application that the applicant database is synchronized with the registrar database.

10 7. The method of claim 1, further comprising:

generating, by the processor at the local node, a disconnected code for an application, the disconnected code indicating that an exchange of hello link-local registration protocol data unit (LRPDU) messages has failed;

15 receiving, at the processor, a command from the application to maintain an applicant database in memory at the local node despite the disconnected code;

receiving, by the receiver at the local node, a complete list LRPDU message from a registrar database at a neighbor node after a successful exchange of hello LRPDU messages, the complete list LRPDU message including record headers for all records at the registrar database;

20 and

comparing, by the processor, the record headers from the complete list LRPDU message with record headers in the applicant database to resynchronize the applicant database with the registrar database.

25 8. A local node comprising a memory, a transmitter, a receiver, and a processor coupled to the memory, the transmitter, the receiver, the local node configured to perform the method of any of claims 1-7.

30 9. A non-transitory computer readable storage medium comprising a computer program product for use by a local node, the computer program product comprising computer executable instructions stored on the non-transitory computer readable storage medium such that when executed by a processor cause the local node to perform the method of any of claims 1-7.

10. A local node comprising:

a receiving means for receiving a hello message including an application identifier (AppId) and a target link between a local node and a neighbor node;

a determination means for determining the AppId is associated with an application for performing database synchronization;

a database pair setup means for setting up a local database at the local node as part of a database pair for use by the application, the database pair including a neighbor database at the neighbor node;

an association means for associating the database pair with the target link; and

a synchronization control means for controlling synchronization of the local database with the neighbor database via the target link.

receiving, at a receiver at the local node, a Hello message including an application identifier (AppId) and a target link between a target port for a local node and target port for a neighbor node;

determining, by a processor at the local node, the AppId is associated with an application for performing database synchronization;

setting up a local database in memory at the local node as part of a database pair for use by the application, the database pair including a neighbor database at the neighbor node;

associating, by the processor, the database pair with the target link; and

controlling synchronization of the local database with the neighbor database via the target link.

~~2.——The method of claim 1, wherein controlling synchronization of the local database with the neighbor database includes transmitting, by a transmitter at the local node, record information to the neighbor database via the target link in Link Local Registration Protocol Data Unit (LRPDU) messages.~~

~~3.——The method of any of claims 1-2, wherein the local database includes an applicant database to store local records for transmission to a registrar database in the neighbor database.~~

~~4.——The method of any of claims 1-3, wherein the local database includes a registrar database to receive neighbor records from an applicant database in the neighbor database.~~

~~5.——The method of any of claims 1-4, wherein the Hello message is a Hello LRPDU, and wherein the Hello LRPDU represents the target link as a My Chassis identifier (ID) identifying the local node, a My Port ID identifying a target port on the local node, a Neighbor Chassis ID~~

identifying the neighbor node, and a Neighbor Port ID identifying a target port on the neighbor node.

6. — A method implemented in a local node, the method comprising:
- transmitting, by a transmitter at the local node, one or more Record Link Local Registration Protocol Data Unit (LRPDU) messages toward a registrar database at a neighbor node, the Record LRPDU messages indicating updates to records stored in an applicant database at the local node;
 - 5 — receiving, by a receiver at the local node, one or more LRPDU messages acknowledging the Record LRPDU messages;
 - marking, in a memory at the local node, at least one updated record in the applicant database as acknowledged by the registrar database; and
 - notifying, via the processor, an application that the applicant database is synchronized with
 - 10 the registrar database.
7. — The method of any of claims 1-6, further comprising determining, by a processor at the local node, that all updated records in the applicant database are acknowledged by the registrar database via LRPDU messages, wherein the notifying the application that the applicant database
- 15 is synchronized with the registrar database is initiated based on the determination that all updated records in the applicant database are acknowledged by the registrar database via LRPDU messages.
8. — The method of any of claims 1-7, wherein the LRPDU messages acknowledging the
- 20 Record LRPDU messages include a Partial List LRPDU message, the Partial List LRPDU message including at least one record header acknowledging the at least one updated record.
9. — The method of any of claims 1-8, wherein the record header includes a record number indicating the updated record and a sequence number identifying an update included in the updated
- 25 record.
10. — The method of any of claims 1-9, further comprising generating a failure notification to the application when a Record LRPDU message timer expires without receiving a corresponding Partial List LRPDU message.
- 30
11. — The method of any of claims 1-10, wherein the LRPDU messages acknowledging the Record LRPDU messages include a Complete List LRPDU message, the Complete List LRPDU message including record headers for all records at the registrar database.

- 12 — The method of any of claims 1-11, wherein the Complete List LRPDU message includes a first record number field indicating the first record stored at the registrar database and a last record number field indicating the last record stored at the registrar database, and wherein the record headers include record numbers indicating the records stored at the registrar database and sequence numbers identifying updates included in the records stored at the registrar database.
13. — The method of any of claims 1-12, wherein the Record LRPDU messages include one or more record numbers indicating records updated at the applicant database and one or more sequence numbers identifying updates included in the records updated at the applicant database.
14. — The method of any of claims 1-13, further comprising:
- transmitting, by the transmitter, a Request Complete List LRPDU message toward the registrar database; and
 - receiving, by the receiver, a Complete List LRPDU message from the registrar database in response to the Request Complete List LRPDU message.
15. — A method implemented in a local node, the method comprising:
- generating, by a processor at the local node, a disconnected code for an application, the disconnected code indicating that an exchange of Hello Link Local Registration Protocol Data Unit (LRPDU) messages has failed;
 - receiving, at the processor, a command from the application to maintain an applicant database in memory at the local node despite the disconnected code;
 - receiving, by a receiver at the local node, a Complete List LRPDU message from a registrar database at a neighbor node after a successful exchange of Hello LRPDU messages, the Complete List LRPDU message including record headers for all records at the registrar database; and
 - comparing, by the processor, the record headers from the Complete List LRPDU message with record headers in the applicant database to resynchronize the applicant database with the registrar database.

16. — ~~The method of any of claims 1-15, further comprising resetting notify timers at the applicant database after an unsuccessful exchange of Hello messages to prevent transmission of Record LRPDU messages until the Complete List LRPDU message is received.~~
- 5 17. — ~~The method of any of claims 1-16, further comprising:~~
~~determining, by the processor, a mismatch between one or more record headers in the~~
~~applicant database and one or more record headers from the Complete List LRPDU message; and~~
~~transmitting a Record LRPDU message toward the registrar database, the Record LRPDU~~
~~message containing updated record headers addressing the mismatch.~~
- 10 18. — ~~The method of any of claims 1-17, further comprising:~~
~~determining, by the processor, no mismatch between record headers in the applicant~~
~~database and record headers from the registrar database; and~~
~~based on the determination of no mismatch, notifying, via the processor, the application~~
15 ~~that the applicant database is synchronized with the registrar database.~~
19. — ~~The method of claim 1, further comprising the method of any of claims 6 and 15.~~
20. — ~~The method of claim of claim 6, further comprising the method of claim 15.~~
- 20 21. — ~~A local node comprising a memory, a transmitter, a receiver, and a processor coupled to~~
~~the memory, the transmitter, the receiver, the local node configured to perform the method of any~~
~~of claims 1-20.~~
- 25 22. — ~~A non-transitory computer readable medium comprising a computer program product for~~
~~use by a local node, the computer program product comprising computer executable instructions~~
~~stored on the non-transitory computer readable medium such that when executed by a processor~~
~~cause the local node to perform the method of any of claims 1-20.~~

23.—— A local node comprising:

a receiving means for receiving a Hello message including an application identifier (AppId) and a target link between a local node and a neighbor node;

a determination means for determining the AppId is associated with an application for performing database synchronization;

a database pair setup means for setting up a local database at the local node as part of a database pair for use by the application, the database pair including a neighbor database at the neighbor node;

an association means for associating the database pair with the target link; and

a synchronization control means for controlling synchronization of the local database with the neighbor database via the target link.

24.—— A local node comprising:

—— a transmitting means for transmitting one or more Record Link Local Registration Protocol Data Unit (LRPDU) messages toward a registrar database at a neighbor node, the Record LRPDU messages indicating updates to records stored in an applicant database at the local node;

—— a receiving means for receiving one or more LRPDU messages acknowledging the Record LRPDU messages;

—— a memory means for marking at least one updated record in the applicant database as acknowledged by the registrar database;

—— a determination means for determining that all updated records in the applicant database are acknowledged by the registrar database via LRPDU messages; and

—— a notification means for notifying an application that the applicant database is synchronized with the registrar database based on the determination that all updated records in the applicant database are acknowledged by the registrar database via LRPDU messages.

25.—— A local node comprising:

—— a disconnection means for generating a disconnected code for an application, the disconnected code indicating that an exchange of Hello Link Local Registration Protocol Data Unit (LRPDU) messages has failed;

—— a command means for receiving a command from the application to maintain an applicant database in memory at the local node despite the disconnected code;

—— a receiving means for receiving a Complete List LRPDU message from a registrar database at a neighbor node after a successful exchange of Hello messages, the Complete List LRPDU message including record headers for all records at the registrar database; and

—— a record comparison means for comparing the record headers from the Complete List LRPDU message with record headers in the applicant database to resynchronize the applicant database with the registrar database.

- 5 26.—— The local node of any of claims 23-25, further comprising means for performing the method of any of claims 1-20.