group are not set up to support cloning, this is the only available option. As state transfer from the binary log is based on classic asynchronous replication, it might take a very long time if the server joining the group does not have the group's data at all, or has data taken from a very old backup image. In this situation, it is therefore recommended that before adding a server to the group, you should set it up with the group's data by transferring a fairly recent snapshot of a server already in the group. This minimizes the time taken for distributed recovery, and reduces the impact on donor servers, since they have to retain and transfer fewer binary log files.

18.5.3.1 Connections for Distributed Recovery

When a joining member connects to an online existing member for state transfer during distributed recovery, the joining member acts as a client on the connection and the existing member acts as a server. When state transfer from the donor's binary log is in progress over this connection (using the asynchronous replication channel `group_replication_recovery`), the joining member acts as the replica and the existing member acts as the source. When a remote cloning operation is in progress over this connection, the joining member acts as a recipient and the existing member acts as a donor. Configuration settings that apply to those roles outside the Group Replication context can apply for Group Replication also, unless they are overridden by a Group Replication-specific configuration setting or behavior.

The connection that an existing member offers to a joining member for distributed recovery is not the same connection that is used by Group Replication for communication between online members of the group.

- The connection used by the group communication engine for Group Replication (XCom, a Paxos variant) for TCP communication between remote XCom instances is specified by the `group_replication_local_address` system variable. This connection is used for TCP/IP messages between online members. Communication with the local instance is over an input channel using shared memory.
- For distributed recovery, up to MySQL 8.0.20, group members offer their standard SQL client connection to joining members, as specified by MySQL Server's `hostname` and `port` system variables. If an alternative port number is specified by the `report_port` system variable, that one is used instead.
- From MySQL 8.0.21, group members may advertise an alternative list of distributed recovery endpoints as dedicated client connections for joining members, allowing you to control distributed recovery traffic separately from connections by regular client users of the member. You specify this list using the `group_replication_advertise_recovery_endpoints` system variable, and the member transmits their list of distributed recovery endpoints to the group when they join the group. The default is that the member continues to offer the standard SQL client connection as in earlier releases.

Important

Distributed recovery can fail if a joining member cannot correctly identify the other members using the host name as defined by MySQL Server's `hostname` system variable. It is recommended that operating systems running MySQL have a properly configured unique host name, either using DNS or local settings. The host name that the server is using for SQL client connections can be verified in the `Member_host` column of the Performance Schema table `replication_group_members`. If multiple group members externalize a default host name set by the operating system, there is a chance of the joining member not resolving it to the correct member address and not being able to connect for distributed recovery. In this situation you can use MySQL Server's `report_host` system variable to configure a unique host name to be externalized by each of the servers.