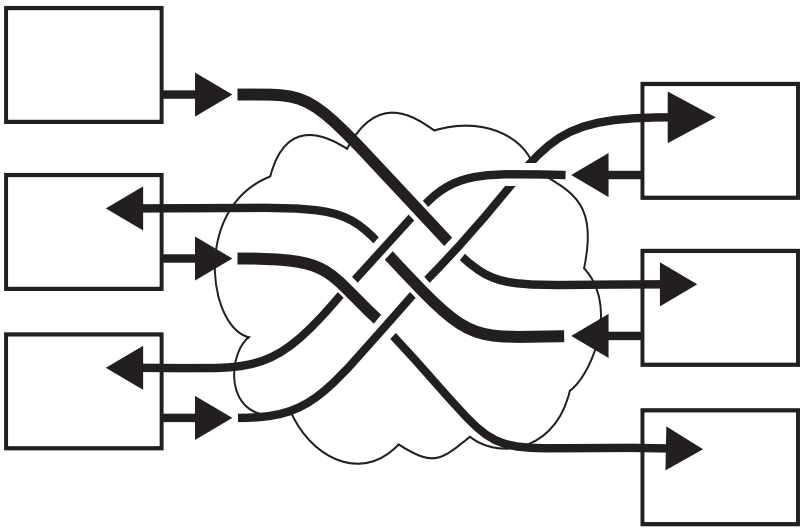
The Fibre Channel Consultant Series

Fibre Channel Switched Fabric



Robert W. Kembel

INC. TUCSON

Cover design by David Fischer, Fischer Graphic Services.

ISBN 0-931836-71-9

Other books in the Fibre Channel Consultant series:

Fibre Channel A Comprehensive Introduction, ISBN 0-931836-84-0
Fibre Channel Arbitrated Loop, ISBN 0-931836-82-4

Published by:

Northwest Learning Associates, Inc.
3061 N. Willow Creek Drive
Tucson, AZ 85712
520-881-0877, Fax: 520-881-0632
www.NLAbooks.com

email: inquiries@NLAbooks.com

Printed in the United States of America

Second Edition

10 9 8 7 6 5 4 3

28. Node Port Initialization

Fibre Channel standards do not provide a complete step-by-step definition of the actions a node port performs when it is first initialized or reinitialized. The standard does not provide this because the steps vary from one environment to another and depend on a number of implementation and functional factors.

This chapter discusses a number of disconnected topics from the standards and provides an initialization scenario that pulls together all of the various initialization-related actions.

NOTE – This scenario simply reflects the opinion of the author and is not required by the standards or necessarily implemented as described.

28.1 Node and Node Port Initialization

When a node initializes in a Fibre Channel fabric environment, there are a number of steps the node and node port may perform. The potential actions include:

- Link speed negotiation
- Link initialization and determination of the port's operating mode (N_Port or NL_Port)
- Performing fabric login (FLOGI)
- Registering to receive state change notifications
- Registering to receive link incident records
- Registering information with the Name Server
- Retrieving information from the Name Server
- Logging in with other node ports (PLOGI)
- Performing Process Login (PRLI), if required
- Performing protocol-specific initialization actions

A diagram illustrating these actions is shown in Figure 159 on page 362. Most of the actions are optional or depend on the configuration, node function, or protocols supported.

28.2 Link Initialization and Speed Negotiation

The first step of the initialization process is to initialize the port, perform power-on self-test, if supported, and acquire link synchronization. A node port may begin link initialization and speed negotiation as a result of one of the following events (there may be other events that also cause link initialization depending on the port design):

- a power-on reset
- an internal or external input requesting link initialization (e.g., a request from the port management interface)

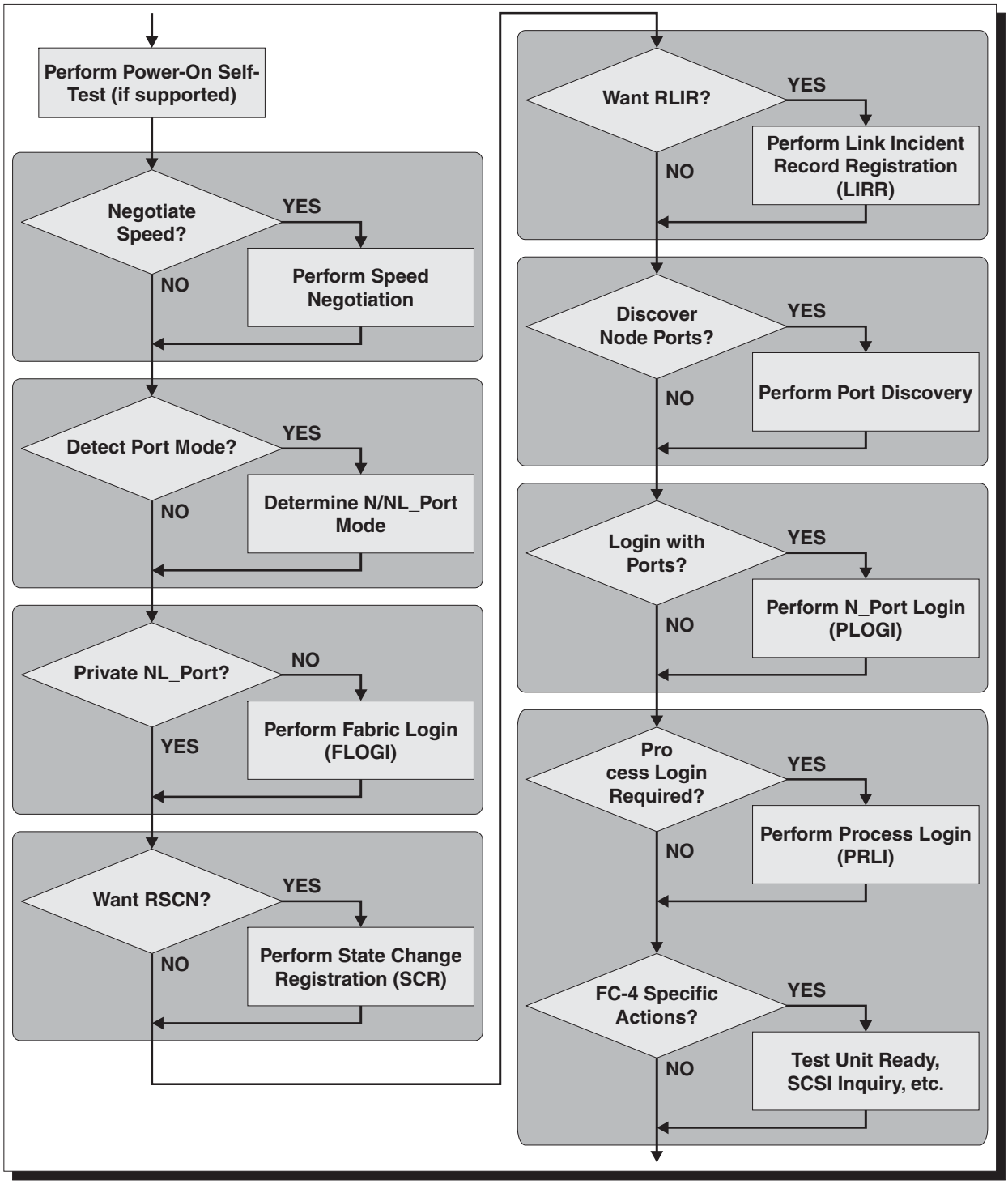


Figure 159. Node Port Initialization

- a transition to the Link Offline condition as defined in FC-PH and FC-FS
- loss of signal or loss of synchronization greater than R_T_TOV
- receiving an OLS, NOS, or LIP primitive sequence
- failure to complete a prior link initialization attempt

28.3 Speed Negotiation

If a Fibre Channel port is capable of operating at more than one link rate it may support auto-speed negotiation. When supported, speed negotiation allows the port to negotiate with the attached port to determine the highest mutually supported speed.

The procedure for speed negotiation is described in the Fibre Channel Framing and Signalling (FC-FS) standard (*reference 34 in the Bibliography on page 544*). Support for speed negotiation is optional in the standard, and does not apply to ports that only support one link speed.

Even when a port is capable of multiple link speeds and speed negotiation, it may be configured to operate at a specific speed. Just because the connected ports are capable of operating at a higher speed, other infrastructure components such as cables, arbitrated loop hubs, or disk enclosures may not be capable of reliably operating at the higher rate.

28.4 Determining the Port Operating Mode

Some Fibre Channel node ports may be capable of operating as either an NL_Port or N_Port. If the port is capable of operating in either mode, it may perform procedures to determine the correct operating mode for that port.

In some cases, the node port may only be capable of operating in one mode or the other, and even if the port is capable of operating in both modes, it may be configured to operated in a specific mode.

The port operating mode determination procedure can be started once a port has acquired link synchronization. A flowchart of this is shown in Figure 160 on page 364.

28.4.1 Determination of Arbitrated Loop or Point-to-Point Mode

If a node port is loop capable, it first attempts the arbitrated loop initialization procedure (for a description of this process the reader is referred to the companion book in this series, *Fibre Channel Arbitrated Loop*).

If the node port is not loop capable (or configured for N_Port mode), it performs non-loop link initialization as defined in the FC-PH and FC-FS standards. An example of this behavior is shown in Figure 161 on page 365.

In this example, Port A is capable of both arbitrated loop mode (NL_Port) and point-to-point mode (N_Port). Port B is only capable of point-to-point mode. Port A first attempts arbitrated loop initialization. Because Port B is not loop capable, it does not recognize the LIP ordered set, and treats it as an idle. Port A times out waiting to receive LIP and detects a loop initialization failure.

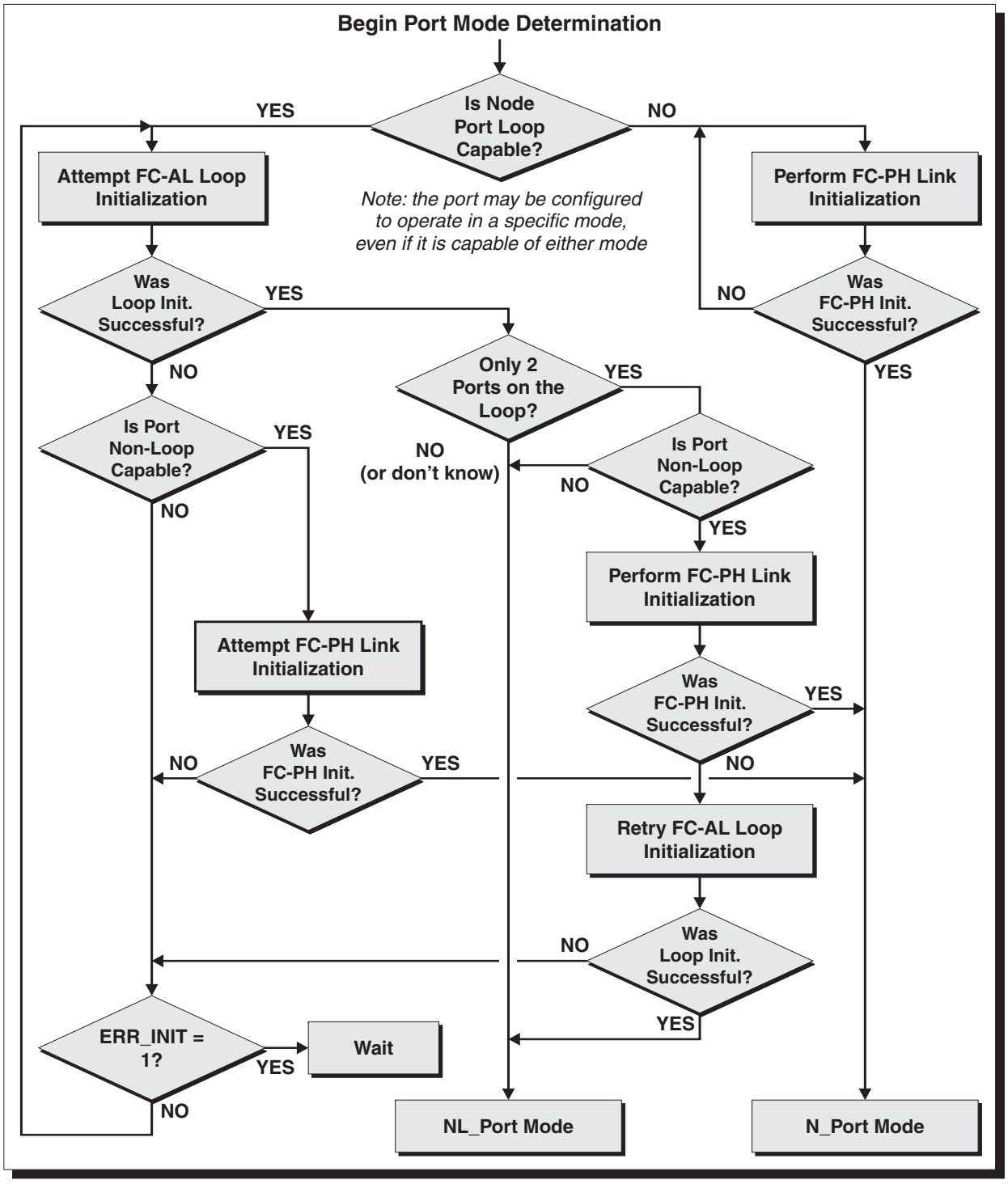


Figure 160. Node Port Mode Determination

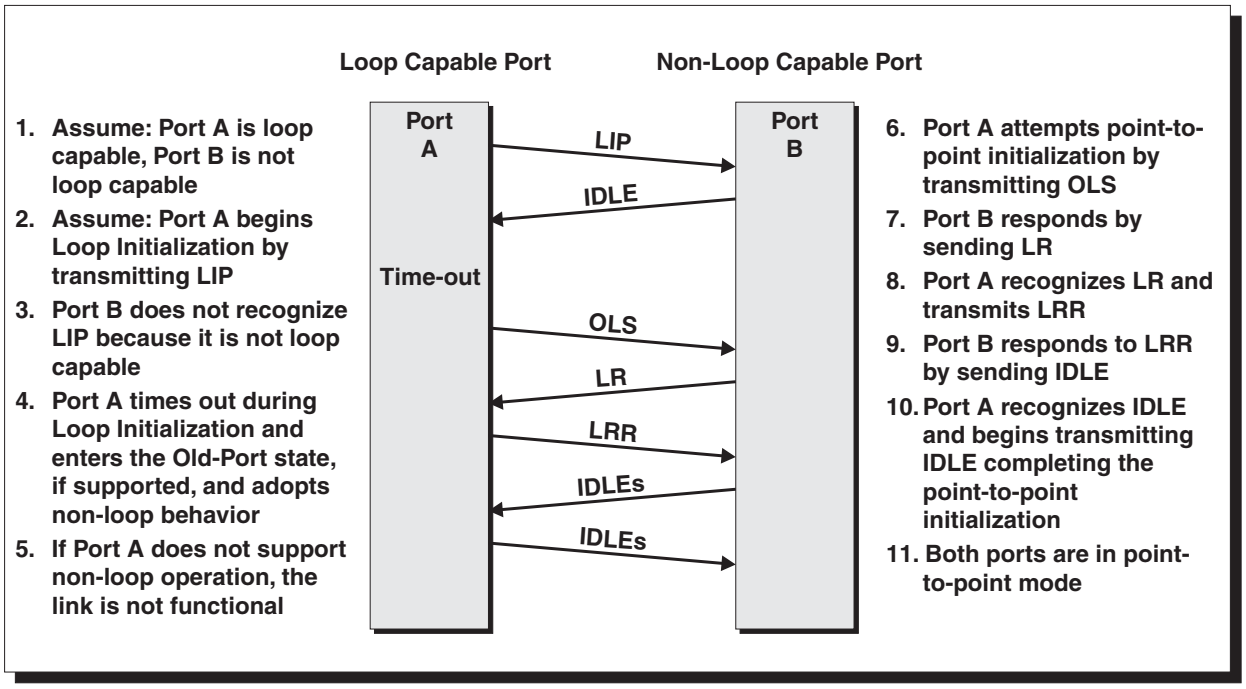


Figure 161. Loop Capable Port and Non-Loop Capable Port

Because Port A is also capable of point-to-point mode, it attempts non-loop initialization by transmitting the Offline Sequence (OLS). Port B recognizes the OLS and responds with Link Reset (LR). Port A then sends Link Reset Response (LRR). Port B sends idles and Port A responds with idles completing the point-to-point initialization.

28.4.2 Two-Ported Loop vs. Point-to-Point Mode

If the arbitrated loop initialization process is successful and there are only two ports on the loop, the two arbitrated loop ports may attempt to operate in N_Port mode. This is not required by the standards and the subject of controversy and debate.

The controversy stems from the overhead associated with the arbitrated loop protocols and the behavior NL_Ports exhibit when there are only two ports on the loop. If the NL_Ports repeatedly open and close loop circuits on a two-ported loop they may be incurring loop overheads needlessly. While this may not be a significant factor on short links, it may adversely affect performance on longer links.

If there are only two ports on the arbitrated loop and the NL_Ports behave appropriately in loop mode, there is no reason to ever close a loop circuit once it has been opened. If the loop circuit is left open indefinitely, there is no performance penalty associated with continuing to operate in arbitrated loop mode. If there are more than two ports on the loop, the loop circuit could still be left open until the current arbitration winner detects that another port is arbitrating. At that time, the loop circuit could be closed.

The problem with leaving the loop circuit open is that some NL_Port implementations may not behave as expected.

If the NL_Port is a half-duplex design, it normally cannot send and receive data frames during the same loop circuit. If the half-duplex port is opened, it prepares to receive frames and is unable to send any data frames during this loop circuit. If the loop circuit is left open indefinitely, the half-duplex port can never send any data frames. If a half-duplex port opens the loop circuit, it opens a half-duplex circuit preventing the open recipient from sending data frames.

Even if both NL_Ports are capable of full-duplex operation, they still may not behave as expected if the loop circuit is left open. While a port may be capable of sending or receiving data frames during a loop circuit, it may not be capable of supporting data transfers in both directions during the same loop circuit. This can occur if the port has single data-transfer function (such as a direct memory access, or DMA) that is initialized on a per-circuit basis.

The problem an NL_Port encounters on a two-ported loop is that there is no way to determine the capabilities of the other port. Consequently, most arbitrated loop ports default to opening and closing loop circuits as required.

Because of the ambiguity concerning the capabilities of the other port, a loop port may attempt to operate in point-to-point mode when there are only two ports on the loop. Therefore the decision to attempt point-to-point mode operation is more about determining the capabilities of the other port than the overhead of the loop protocols.

To determine if the other port is capable of non-loop mode operation, a node port may begin a new loop initialization by transmitting the Loop Initialization Primitive sequence (LIP). When it receives LIP, the port knows the other port has entered the initialization process. The node port may then transmit the Offline Sequence (OLS) to begin non-loop initialization (starting a new loop initialization is necessary because OLS is only recognized by a loop port during the loop initialization process).

If there is only one other port on the loop and that port is non-loop capable, it responds to OLS by transmitting Link Reset (LR). The node port responds to LR by transmitting Link Reset Response (LRR). The other port then transmits idles and the node port responds by also transmitting idles completing the point-to-point mode initialization. At this point the node port can assume N_Port behavior and perform fabric login (FLOGI).

If there is more than one other port on the loop, but that port was not detected for some reason (e.g., one or more loop ports in non-participating mode), the point-to-point initialization will not be successful.

When the loop port restarts loop initialization and transmits the Offline Sequence (OLS), the next port on the loop responds to OLS by sending Link Reset (LR), but the third port does not recognize LR while in the loop initialization process. The port that is transmitting OLS times out waiting for LR and assumes point-to-point mode is not supported by this configuration. It may then begin a third loop initialization to return to arbitrated loop mode. This initialization should complete successfully with the same results as the first loop initialization

28.5 Fabric Login (FLOGI)

After a node port completes link initialization and operating mode determination, it may perform fabric login. A flowchart of the fabric login process is shown in Figure 162 on page 368.

N_Ports (ports not operating in arbitrated loop mode) are required to perform fabric login (FLOGI). If the N_Port is connected in a point-to-point configuration, the fabric login (FLOGI) extended link service will be accepted with an indication in the accept that the port is connected to another N_Port. In this case, the ports continue the initialization in point-to-point mode.

If the FLOGI is accepted with an indication that the other port is an F_Port, the N_Port is connected in a fabric environment and continues with fabric-related initialization activities.

If the fabric login is rejected with an indication that the class of service is not supported, the node port may retry fabric login using a higher-numbered class of service, if available. If no more classes of service are available, operation with the fabric is not possible because the port and fabric have no classes of service in common.

NL_Ports are not required to perform Fabric login (FLOGI). Whether to perform fabric login is an implementation decision, and not dictated by the standards.

If an NL_Port does not perform fabric login, it is referred to as a private NL_Port and is not part of the fabric address space.

If an NL_Port performs fabric login it is referred to as a public port. If a fabric is present and the fabric login is successful, the NL_Port becomes part of the fabric address space. If no fabric (FL_Port) is present, or the port is unable to complete the fabric login, the NL_Port behaves as a private NL_Port. Public NL_Ports attempt fabric login when they complete their power-on initialization processing, or following a loop initialization with the L_Bit (Login Required) bit set during the LISA loop initialization sequence.

A public NL_Port may determine that no fabric is present and bypass attempting fabric login. It may determine the presence of an FL_Port by examining the arbitrated loop position map (if available) or by detecting that an attempt to open the FL_Port failed. If the NL_Port does not attempt fabric login, it behaves as a private NL_Port.

28.6 State Change Registration (SCR)

Some ports may wish to receive notification when the login state of other ports in the fabric changes. Ports that want this type of notification can register to receive state change notifications. State changes occur when a new port logs in with the fabric (or performs a re-login), a port is disconnected from the fabric, the fabric receives the NOS or OLS primitive sequence, the fabric receives a state change notification from the node port, or other similar events.

A port that wants to receive state change notifications registers its interest by using the State Change Registration (SCR) extended link service. The SCR extended link service request is normally sent to the Fabric Controller at well-known address x'FF FF FD' (although the SCR request may also be sent directly to a specific port). If the Fabric Controller detects a state change event, it sends a Registered State Change Notification (RSCN) extended link service to all registered ports.

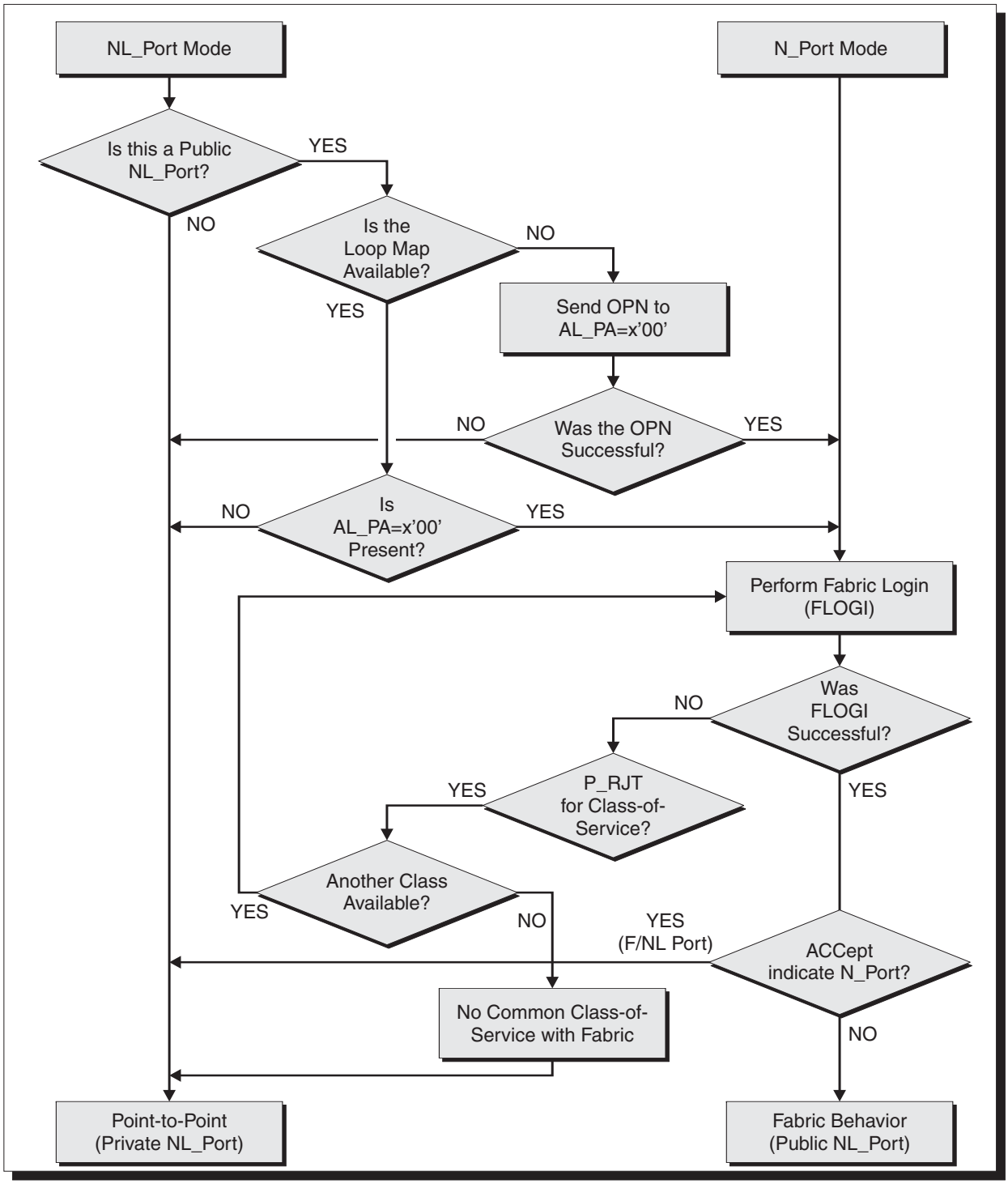


Figure 162. Fabric Login Flowchart

State change registration with the Fabric Controller is not port-specific — a port registers to be notified of all state changes. When a port receives an RSCN, the RSCN contains one or more addresses indicating ports that have had a state change event.

The RSCN recipient must examine the address of the affected port and determine if it is interested in the state of that port. If so, it may want to verify its current login state with the affected port, if possible, or release resources associated with that port if the port is no longer accessible. If the port in not interested in the affected port, the RSCN notification is accepted, but no further action is necessary.

28.7 Link Incident Record Registration (LIRR)

Some ports may wish to receive a notification when another port has a link incident record to report. A link incident record is error information that has been accumulated by a port and the port wishes to report that information to interested ports.

The fabric may contain one or more entities responsible for collecting and reporting error information. This may be a node running a management application.

If a port wishes to receive this type of information, it must register its interest by sending a Link Incident Record Registration (LIRR) extended link service request. The registration request is sent to the Management Server at address x'FF FF FA' or directly to a specific node port.

When a node supports link incident registration (i.e., it accepts the LIRR extended link service) and has a link incident record to report, it sends a Registered Link Incident Record (RLIR) extended link service to the Management Server and all registered node ports.

N_Port login is required prior to registering to receive link incident records from either the Management Server or a specific node port. The login session must be maintained for the entire time the port wishes to receive link incident records. If a port logs out with the Management Server or other port, the link incident registration is removed.

28.8 Port/Device Discovery

Many operating systems' input/output (I/O) architectures are based on the principle of device discovery. Discovery is needed when the I/O configuration of the system is not stored or predetermined. When the operating system is initialized, it discovers devices that are attached and assumes that those devices may be used. A flowchart showing how a port could discover available devices is shown in Figure 163 on page 370.

Device Discovery in a Point-to-Point Environment. Device discovery in the point-to-point topology is trivial as there can only be one other device. The address of that device is obtained during the PLOGI process that was used to determine the ports were connected in a point-to-point topology.

Device Discovery in an Arbitrated Loop. In an arbitrated loop topology, the loop initialization process may build a loop map (LILP and LIRP). When the loop map is available, it provides a list of ports on the loop.

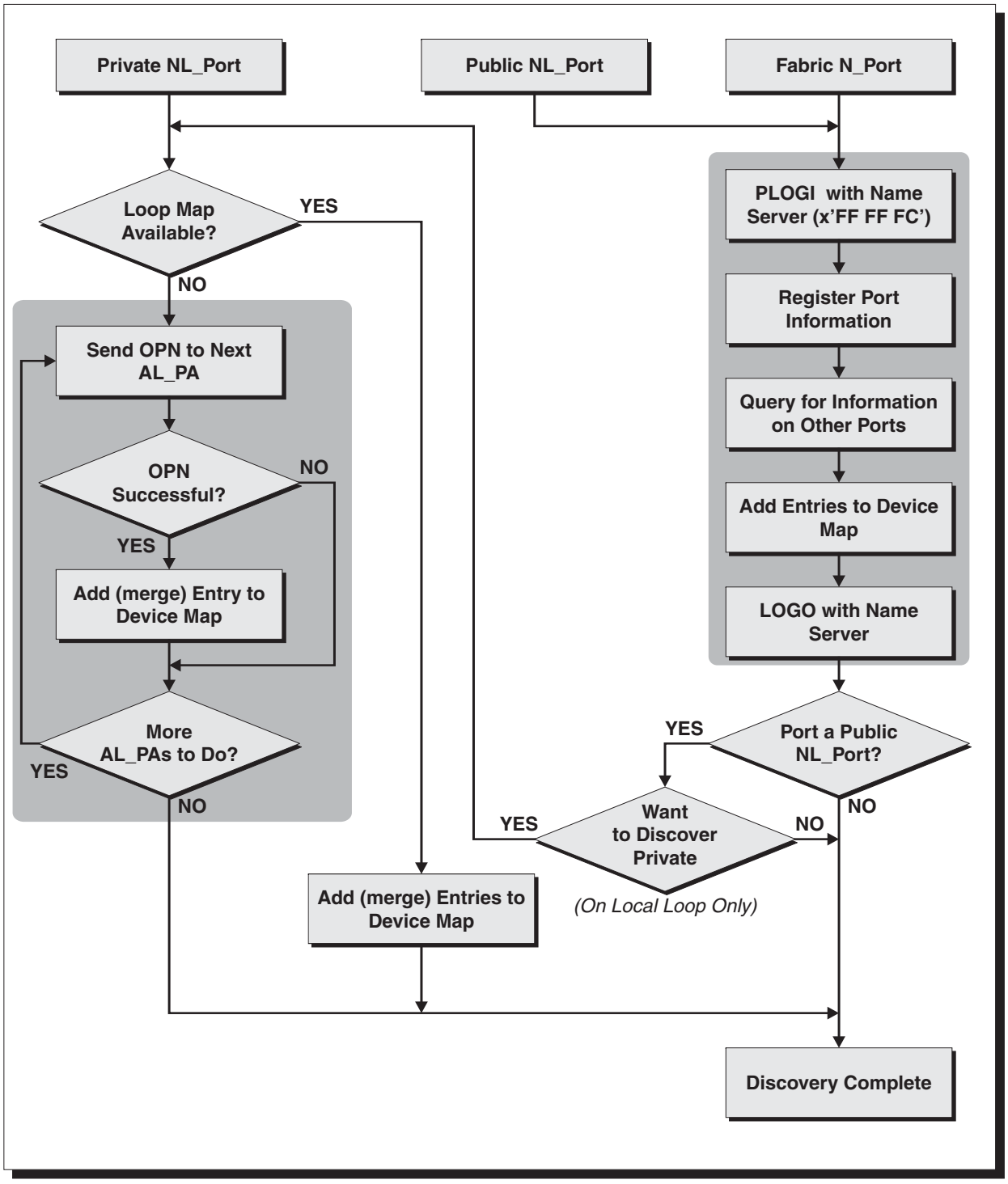


Figure 163. Port Discovery Flowchart

Some ports do not support building the loop map. If any port on the loop does not support this procedure, no loop map is available. In this case, a port may attempt to discover other ports on the loop by scanning all possible loop addresses (somewhat analogous to scanning a SCSI bus to detect attached devices).

The standard does not specify how a loop port discovers other ports on the same loop when no loop map is available. One possible approach is for a port to win arbitration and attempt to open a loop circuit with one of the AL_PAs. If the open is successful, a port has been discovered and represents a device that may be accessed. If the open is unsuccessful, no device exists at that AL_PA and scanning can proceed to the next AL_PA.

Ports operating in an environment based on device discovery need to make information regarding discovered ports available to the operating system. This is normally done by device driver software written specifically for that port and operating system combination.

Device Discovery in a Fabric Topology. Discovering other ports in a fabric topology is not as simple as looking at a loop map or scanning allowed addresses. The fabric provides a 24-bit address space with over 16 million possible addresses. Scanning is simply not practical. In this case, devices are normally discovered by querying the Fibre Channel Name Server (a sub-function of the Directory Server).

N_Ports and public NL_Ports login with the fabric and should register information with the Name Server. Because some NL_Ports implementations did not register information with the Name Server, many fabrics automatically register information during the fabric login process (this behavior is not required by the standards).

Because private NL_Ports do not login with the fabric, the Name Server may not contain information about private ports. In this case, public NL_Ports may need to use both the loop map, or scanning, and the Name Server to discover all devices. When both methods are used, the port must ensure it does not duplicate information when a port is discovered by both methods.

28.9 Name Server Registration

The Name Server provides a database that may contain information about node ports. Once information is registered with the Name Server, it can be retrieved or queried by other ports. The Name Server is described in *Directory Server* on page 249 and the commands used to access the Name Server are described in *Name Server Commands* on page 415.

Some information may be registered in the Name Server database by the Fabric Login Server when a node port logs in with the fabric. Other information must be explicitly registered by the node port. As a part of its initialization processing, a node port should register information with the Name Server so it is available to other ports (registering information is required by some Fibre Channel technical reports).

28.10 Name Server Query

A node port may want to interrogate the Name Server database to retrieve information about other ports. For example, a SCSI initiator may request a list of all devices that have registered

support of the SCSI protocol. If zoning is in effect (see *Zoning* on page 14), information returned by the Name Server is limited to other ports within the same zone(s) as the requestor.

Node ports must perform port login (PLOGI) with the Name Server before they can register or retrieve information from the Name Server database. When a port completes its registration or query operations, it should log out with the Name Server to free up login-related resources. Registered information is maintained in the Name Server after the logout as long as the associated port is still accessible by the fabric.

28.11 N_Port Login (PLOGI)

N_Port Login is required before performing operations using any protocol other than link services. N_Port login is performed using the PLOGI extended link service.

NOTE – Note: The standards allow for an implicit login. In this case, the port implicitly knows the service parameters of the other port and uses those parameters. Implicit login is not permitted in most open environments.

The standard allows either port to initiate the PLOGI operation, or both ports could simultaneously attempt to login with each other. If this occurs, one of the ports will accept the PLOGI and the other will send LS_RJT with a reason code of 'command already in progress'. The decision to accept or reject the login is based on the Port_Names of the two node ports.

Some protocols may specify which node port initiates the N_Port login (PLOGI) extended link service. For example, the SCSI-FCP protocol assumes the SCSI initiator will originate the PLOGI and the SCSI target waits for that to occur. This allows the initiator to determine which ports it wishes to communicate with and initiate login with only those ports.

28.12 Process Login (PRLI)

Some protocols require Process Login (PRLI). Process Login may be required when the FC-4 protocol mapping contains optional behaviors that must be negotiated or agreed upon by the two ports. (The SCSI-3 FCP protocol requires process login to negotiate the use of optional FC-4 information units and behaviors before commands will be accepted.) Process Login may also be required when Initial Process Associators are used between the ports.

28.13 Protocol-Specific Initialization

After the preceding steps have been completed, a node port may need to perform additional protocol-specific actions. The actions taken depend on the protocol(s) being used by the node and node port. In addition, the actions may be different for different operating systems, even if they are using the same protocol. Finally, different actions may be taken depending on the function of the node (e.g., SCSI initiator vs. target, FC-SB-2 channel vs. FC-SB-2 control unit).

A SCSI initiator may send a TEST UNIT READY command to clear a pending unit attention condition at a target. If may then send a REPORT LUNs command to determine which logical unit numbers (LUNs) are supported by the target. The SCSI initiator may then send an INQUIRY command to obtain information about each logical unit (LUN).

28.14 Chapter Summary

Node Port Initialization

- Node port initialization consists of a number of different steps:
  - Link Initialization and port mode determination
  - Fabric login (FLOGI)
  - Registering to receive state change notifications (SCR)
  - Registering to receive link incident records
  - Registering with the Name Server
  - Querying the Name Server
  - Logging-in with other node ports (PLOGI)
  - Performing Process Login (PRLI)
  - Performing protocol-specific actions

Link Initialization

- If a port is capable of multiple link speeds, it may perform speed negotiation
  - Determine the highest mutually supported speed
- A port may be manually configured to operate at a specific speed
  - By setting configuration options or backplane wiring in a disk enclosure
- Once word synchronization is acquired, the port operating mode can be determined

Determine Port Mode

- If a port is capable of both N_Port and NL_Port modes, it must determine which mode to use
- Attempt loop initialization first
  - If unsuccessful, attempt non-loop initialization
- If loop initialization is successful, the port may want to attempt non-loop initialization if there is only one other port on the loop
  - Point-to-point mode may avoid overhead associated with the loop protocols
  - The port may also remain in loop mode and never close the loop circuit (also avoids the overhead)

Fabric Login

- Fabric login (FLOGI) gives a port access to the fabric
- Fabric login also assigns a port's address
  - All 24-bits for an N_Port
  - Upper 16-bits for an NL_Port (least significant 8 bits are the AL_PA acquired during loop initialization)
- Fabric login is mandatory for N_Ports
- Fabric login is optional for NL_Ports
  - NL_Port that does not perform fabric login is called a 'private' NL_Port
  - NL_Port that does perform fabric login is called a 'public' NL_Port

State Change Registration

- Some ports may wish to be notified when the login state of other ports change
  - For example: SCSI initiators, FICON channels, nodes with maintenance functions, etc.
- To receive state change notifications, a port must register using the State Change Registration (SCR) ELS
  - SCR is sent to the Fabric Controller or directly to the other node port
  - Port receives Registered State Change Notification (RSCN) when a state change occurs

Link Incident Records

- Some ports may wish to be notified when other ports have a link incident record to report
  - A record containing error information
- To receive link incident records, a port must register using Link Incident Record Registration (LIRR) extended link service
  - LIRR is sent to the Management Server or directly to the other node port
  - The port will then receive a Registered Link Incident Records (RLIR)

Port/Device Discovery

- In some environments, a system discovers the available devices
  - Most SCSI environments use a discovery process
- The means to discover devices depends on the Fibre Channel topology
  - In point-to-point discovery is trivial
  - In an arbitrated loop ports may use the loop map (if available) or scan the loop
  - In a fabric environment, ports are discovered by querying the Name Server

Name Server Registration

- Information must be registered with the Name Server to be available to other ports
- Some information may be registered as a result of Fabric login (FLOGI)
  - Port Address
  - Port_Name and Node_Name
  - Port type (N_Port or NL_Port)
  - Classes of service supported
- Other information must be explicitly registered
  - FC-4 protocols supported
  - FC-4 features and descriptors
- A node port must login with the Name Server in order to register information

Name Server Query

- A node port may query the Name Server to obtain information about other ports
  - Get a list of addresses of all ports that have registered support of the SCSI-FCP protocol
  - Get the address of the port with the designated Port_Name
- A node port must login with the Name Server in order to query the database
  - The port should logout when it is done to free up login resources
- Name Server operations use the Fibre Channel Common Transport (FC-CT) protocol

Process Login (PRLI)

- Some protocols or applications require the use of process login (PRLI)
- Process login is used to communicate FC-4 specific information
  - For example, support for use of optional information units
- Each FC-4 specifies if Process Login is used, and if so, for what purpose
  - Process Login is required by the SCSI_FCP protocol mapping

Protocol-Specific Actions

- Some protocols require additional initialization actions
  - SCSI initiators may send an INQUIRY command to determine the device class
  - FICON channels establish logical paths with the attach control units