Table of Contents

APPN Directory Services

Document ID: 12237

Introduction

This document details how Advanced Peer−to−Peer Networking (APPN) Directory Services add new network nodes (NNs) to a network, propagate new topology information to other nodes, and select routes between logical units (LUs).

Prerequisites

Requirements

Readers of this document should have knowledge of these topics:

- Systems Network Architecture (SNA)
- How APPN (an extension to SNA) defines logical structure, formats, protocols, and operational sequences to transmit information units

Components Used

This document is not restricted to specific software or hardware versions.

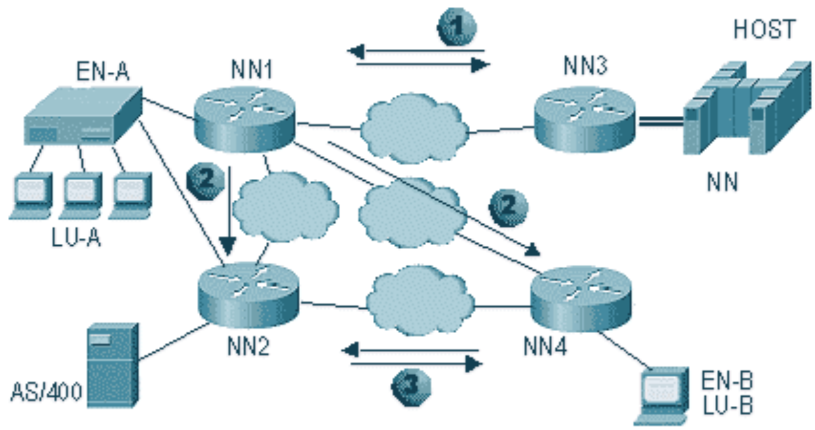
Conventions

For more information on document conventions, refer to the Cisco Technical Tips Conventions.

Sequence to Add a Node

Figure 1 demonstrates the sequence to add a network node to the network:

Figure 1



1. NN3 joins the network.
2. NN1 sends a topology update.
3. NN2 and NN4 propagate the update.

This is the general sequence when a new NN comes online in the topology:

1. When a node is added to the network, it sends its topology: its name and the transmission group (TG) that connects it to the adjacent nodes.

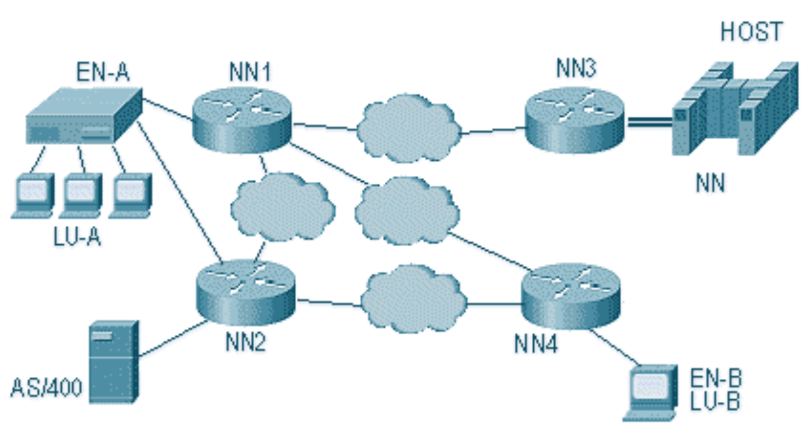
   The existing node to which the new node connects passes its topology database to the new node.
2. The existing NN passes information about the new node to adjacent NNs.
3. The adjacent NNs forward information to their adjacent nodes.

   Duplicate messages are discarded as the message is propagated through the network.

Sequence to Establish an LU−LU Session

Figure 2 demonstrates the sequence to establish an LU−LU session:

Figure 2



Once an NN has connected to the network, LUs on end nodes (ENs) that use NNs as their network server can establish LU−LU sessions with this sequence:

1. Directory Services is used to locate the destination LU.
2. A route is selected from the topology database, based on the class of service (CoS).

3. A BIND is sent between LUs. Each NN on the session path will build a connector for the session so that session traffic is routed efficiently.

Sequence to Locate a Destination LU

This is the sequence that LU−A (the originating LU in the network topology in Figure 2) uses to locate a destination LU:

1. LU−A requests a session with LU−B. EN−A sends a LOCATE to NN1, the NN server for EN−A.
2. NN1 locates LU−B:

   NN1 will search its directory entries first:

   - Because there is no CP−CP session between a low−entry networking (LEN) node and its NN, LEN nodes attached to the NN must be defined. You may also choose to define additional non−LEN entries to reduce broadcasts.
   - An EN can choose to register its resources with the NN. If registered, this information is sent on the CP−CP session and is stored in the directory.
3. NN1 acquires information:

   As resources are discovered through searches, the information is stored as cache entries in the directory of the NN.
4. If the NN finds the destination LU in its directory, a broadcast does not need to be sent to locate it. Instead, a directed search is sent to verify that it still exists in the same location.

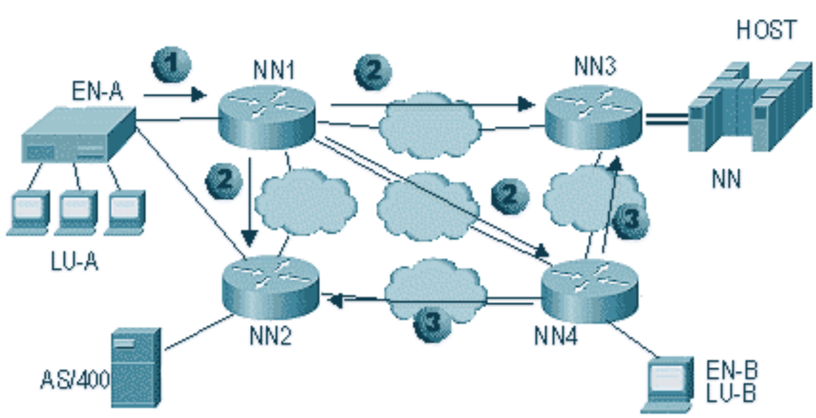
   If the NN *does not* find the destination LU in its directory, a broadcast is first sent locally and is then sent across the network, to locate the resource.

Optionally, you can define a Central Directory Server in virtual telecommunications access method (VTAM), to centralize the directory and reduce broadcasts.

Sequence of a LOCATE(FIND)

Figure 3 demonstrates the sequence of a LOCATE(FIND):

Figure 3



1. EN−A sends LOCATE(FIND) to NN1:

LU−A requests a session with LU−B, so a LOCATE(FIND) is sent from EN−A to NN1 for LU−B.
2. NN1 broadcasts LOCATE(FIND):

   After it determines that the resource is not local (by searching its local directory and sending directed searches to other local ENs), NN1 sends a broadcast to adjacent NNs, to request the location of LU−B.
3. Adjacent NNs propagate the request:

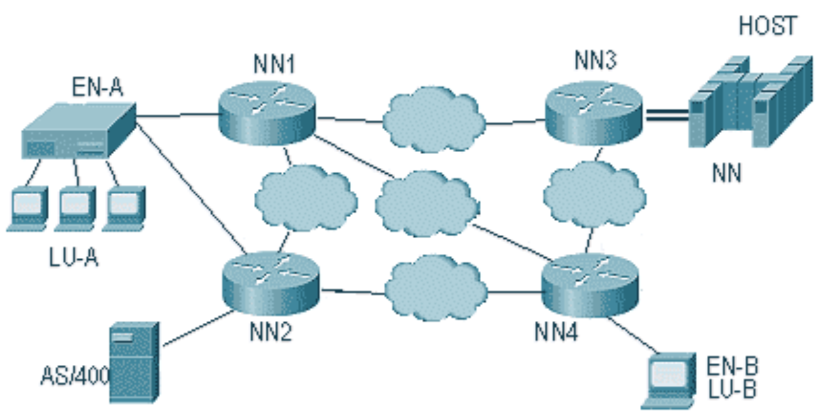
   The adjacent NNs search their domains and propagate the request to their adjacent NNs. Duplicate LOCATEs are discarded by NNs.
4. NN4 finds LU−B:

   NN4 finds LU−B when a directed search is sent to EN−B.

Sequence of a LOCATE(FOUND)

This section explains the sequence of a LOCATE(FOUND):

Figure 4



1. NN4 returns LOCATE (FOUND):

   EN−B will respond to the LOCATE (FIND) from NN4 by sending a LOCATE (FOUND) to NN4. The local configuration of EN−B is included in that LOCATE (FOUND), together with all of the NNs to which EN−B is attached, as well as the TGs that are attaching them. NN1 will select the best session path based on this information from EN−B.
2. NN4 and NN1 cache the information:

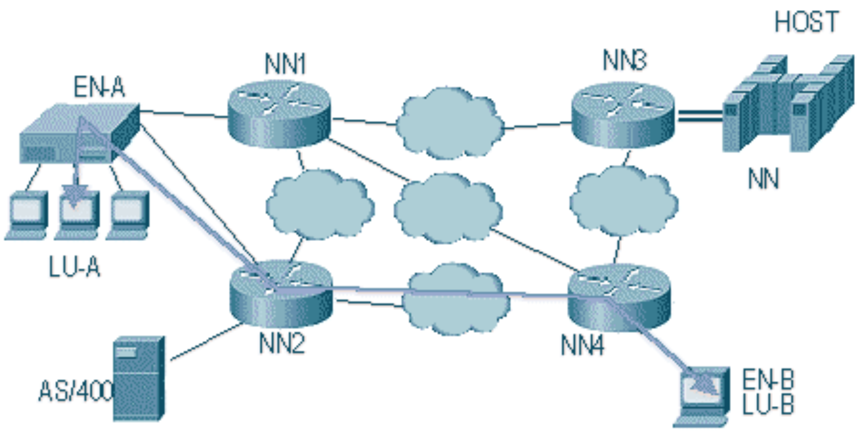
   NN4 sends the LOCATE (FOUND) back on the same path over which the original LOCATE (FIND) was received. NN1 eventually receives the LOCATE (FOUND); it then adds LU−B to its directory, to use with later session requests.
3. NN1 now has the complete topology:

   NN1 does not respond to EN−A with the LOCATE (FOUND) until a route is selected. The route is then included in the LOCATE (FOUND) sent to EN−A.

Sequence for Route Selection

Figure 5 shows the route selected by NN1 for the example network topology.

Figure 5



1. NN1 calculates the optimal route based on these factors:
    - network topology
    - NN and TG characteristics
    - CoS
2. Because NN1 maintains the network topology, it has information about all of the possible paths through the network. The LOCATEs provide the local topologies for EN−A and EN−B (their Tail Vectors the links that connect them to one or more adjacent NNs).
3. The LOCATE from EN−A also contains the CoS requested for this particular session. Through examination of its CoS table, NN1 determines the characteristics that the route must have, based on qualities of service (such as the line speed, the line costs, the security of the line, and up to three other user−defined qualities).
4. The NN then compares the qualities requested by CoS and maps these to the TGs and NNs on the possible paths through the network. Each NN and TG may be defined with the same qualities, along with a weight, which helps force sessions to a particular path. The qualities and weight of each NN and TG are compared to the qualities requested in the CoS. If a path meets the requirements of the CoS, the weights of each NN and TG are added and the path with the lowest cost is selected.

Related Information

- **Technology Support**
- **Product Support**
- **Technical Support and Documentation − Cisco Systems**

Updated: Jul 29, 2005 Document ID: 12237