LAN ADDRESS RESOLUTION

Document Number LAR-1

August 10, 1995

Walter Smith
Mark McCorry

Networking Systems Architecture
International Business Machines Corporation
Research Triangle Park, NC

Note:

This document contains material that is intended as an aid to the APPN Implementers Workshop (AIW) in defining agreements on procedures/protocols for the networking environment. It does not constitute any commitment on the part of IBM to implement, market or otherwise pursue business opportunities related to its contents. This document is furnished as is. IBM assumes no responsibility for the use of the functions described in any manner.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to use these patents

Trademarks

Ethernet™ is a trademark of Xerox Corporation.

Document Revisions

This document is currently at the A2-level (April 22, 1992) with additional revisions made to accommodate AS/400 high-level design decisions (June 19, 1992).

This document was revised April 11, 1995 to incorporate corrections and enhancements to ADVERTISE and NOTIFY flows.

Contents

1. DEFINITIONS 1
2. ADDRESS RESOLUTION REQUIREMENTS 3
2.1 Problem Statement 3
2.1.1 SNA APPN Networks 3
2.1.2 SNA Subarea Networks 4
3. FUNCTIONAL DESCRIPTION 5
3.1 Elements of the Connection Control Entity 5
3.2 CCE Functions 7
3.3 CCE Operation and Frames 9
3.4 CCE Timers and Counters 12
4. CCE FRAME FORMATS 13
4.1 SOLICIT Frame 14
4.2 ADVERTISE Frame 15
4.3 FIND Frame 16
4.4 FOUND Frame 17
4.5 QUERY Frame 19
4.6 NOTIFY Frame 20
4.7 Format of the LLC I-field in CCE Frames 21
4.7.1 Major Vector Format 21
4.7.2 Subvector Format 21
4.7.3 Netid, Name, and Group Name Subvector Formats 23
4.7.4 Connection Network ID Subvector Format 23
4.7.5 Routing Capabilities Subvector Format 23
4.7.6 MAC Address Subvector Format 23
4.7.7 LSAP Subvector Format 24
4.7.8 Correlator Subvector Format 24
5. CCE FINITE STATE MACHINES 25
5.1 Responding to Find_member Request 25
5.2 Responding to Find Request 27
5.3 Responding to SOLICIT Frames 29
5.4 Responding to FIND Frames 30
5.5 Responding to Search Request 31
5.6 Responding to QUERY Frames 33
5.7 Responding to Record and Erase Requests 34
6. SAMPLE CCE LINE FLOWS 37
6.1 ADVERTISE/SOLICIT Flow 37
6.2 FIND/FOUND Flow 40
6.3 QUERY/NOTIFY Flow 41
Appendix A. RELATED CONSIDERATIONS 43
A.1 Bridge Routing Considerations with CCE 43
A.2 CCE Management Operations 44
A.2.1 CCE Configuration Resources 44
A.2.2 CCE Protocol Resources 45

FIGURES

1. CCE in a LAN Station 6
2. CCE Primitives and Return Signals 9
3. Format of the SOLICIT Frame 14
4. Format of the ADVERTISE Frame 15
5. Format of the FIND Frame to the All CCE Functional Address 16
6. Format of the FIND to an Individual CCE MAC Address 17
7. Format of the FOUND Frame 18
8. Format of the QUERY Frame 19
9. Format of the NOTIFY Frame 20
10. CCE Major Vectors and Subvectors 22
11. Canonical Form for MAC Address 24
12. Responding to Find_member Request 25
13. Responding to Find Request 28
14. Responding to SOLICIT Frames 29
15. Responding to FIND Frames 31
16. Responding to Search Request 32
17. Responding to QUERY Frames 33
18. Responding to Record and Erase Requests 34
19. ADVERTISE/SOLICIT Line Flow at the Initiating CCE - Case 1 38
20. ADVERTISE/SOLICIT Line Flow at the Initiating CCE - Case 2 39
21. ADVERTISE/SOLICIT Line Flow at the Receiving CCE 40
22. FIND/FOUND Line Flow 41
23. QUERY/NOTIFY Line Flow at the Initiating CCE 42
24. QUERY/NOTIFY Line Flow at the Receiving CCE 42

1. 1.0 DEFINITIONS

The terms and definitions used in this document are generally consistent with their standard use in the telecommunications environment (ISO, CCITT, etc.) In addition, familiarity with terms and definitions used in the context of the IBM Token Ring Network (see IBM Token-Ring Network Architecture Reference) is assumed. Terms that have specific meanings in the context of local area network address resolution architecture are defined below.

Station
One instance of the Data Link Control (DLC) layer of a Local Area Network (LAN), including the Logical Link Control (LLC) and the Medium Access Control (MAC) sublayers and the Physical Layers controlled by the DLC.

Network Entity
An element of the layer of a communications network that uses the LAN DLC to obtain communications services. Multiple network entities may use a single station, and a single network entity may use multiple stations.

Network Entity Name (also Identifier)
A permanent identifier assigned to one and only one instance of some network entity. A permanent identifier assigned to one and only one instance of some network entity. For APPN network entities, the Control Point (CP) Name of an APPN node is an example of an Network Entity Identifier. For subarea SNA networks, Physical Unit (PU) Names and SSCP Names are examples of Network Entity Identifiers. In this paper, "Name" and "Individual Name" are often used synonymously with "Network Entity Name".

Group Name (also Identifier)
A permanent identifier assigned to each of a collection of Network Entities. For a Network Entity assigned both an Individual Name and a Group Name, the entity is often referred to as a member of the Group Name. Group names use the character set 01134 as defined in the "SNA Formats" (IBM Document Number GA27-3136), i.e. uppercase letters A through Z and numerals 0-9.

Subnetwork Point of Attachment (SNPA)
A logical point in a station at which a LAN offers communications services to network entities. A SNPA is a seven-byte quantity consisting of the concatenation of a one-byte LLC address (LSAP) and a six-byte MAC address. SNPA is used in this document to refer to the pair LSAP and MAC address and to abbreviate the writing of the longer phrase.

Connection Control Entity (CCE)
A function, resident in every station, that provides address resolution services for network entities. Each station includes one instance of CCE for each attached LAN. This document defines the formats, protocols, and frames required to implement the CCE function.

Routing Capabilities
A property of a network entity that identifies the communications routing services provided by the entity. Routing capabilities for a network entity are typically defined by identifying a set routing properties and determining whether a particular network entity exhibits those properties.

Communications Server
A station that includes a network entity that provides some communications routing service, i.e., that has Routing Capabilities defined. "Server" and "Intermediate System" are often used interchangeably with "Communications Server".

Non-server
A station that is not a communications server, i.e., no network entity at the station provides any communications routing service. "End System" is often used interchangeably with "Non-server". Examples of a non-servers in the context of this document are APPN End Nodes and LEN nodes.

2. 2.0 ADDRESS RESOLUTION REQUIREMENTS

2.1 Problem Statement

This document defines an architecture to meet requirements that are significant for customers that need to install and operate SNA networks that include Local Area Network (LAN) technology. SNA systems today require that users provide the local and remote subnetwork points of attachment of SNA control points before communications can be initiated between subarea SNA or APPN network entities. This requirement produces a large (and growing) configuration burden for the end user and/or network administrator. It also produces a very static configuration environment for SNA protocols operating over LANs. This is especially critical when systems support multiple attachment points and when multiple systems can provide end systems with intermediate routing services. This architecture will meet the following general requirements for systems running over IEEE 802.2 Data Link Services:

- to allow a Non-Server to determine, dynamically and without pre-configuration, the SNPA of a Server. This allows the Non-Server node to use the SNPA to establish, without pre-configuration, the appropriate connections.

- to allow a Server node to determine, dynamically and without pre-configuration, the SNPAs of other Server nodes on the LAN. The Server node can use the SNPAs to establish, without pre-configuration, the necessary connections to build the appropriate topology database for optimum network performance.

This architecture meets the following specific requirements for APPN and subarea SNA networks:

- to allow an APPN end node or a Low Entry Networking (LEN) node to determine, dynamically and without pre-configuration, the SNPA of an APPN network node control point that can provide network node services. This allows the end node to use the SNPA to establish, without pre-configuration, the CP-to-CP sessions necessary to exploit APPN capabilities.

- to allow an APPN end node or a LEN node in APPN networks without a network node to determine, dynamically and without pre-configuration, the SNPA of a control point with a known name.

- to allow an APPN network node to determine, dynamically and without pre-configuration, the SNPAs of other network node control points on the LAN. The network node can use the SNPAs to establish, without pre-configuration, the CP-to-CP sessions necessary to form a spanning tree of APPN network nodes.

- to allow a subarea SNA Type 2.1 physical unit to determine, dynamically and without pre-configuration, the SNPA of a Type 4.0/5.0 node that can provide SNA boundary node services. The Type 2.1 node can then use the SNPA to establish the PU-to-SSCP session for subarea communications.

- to allow a subarea SNA Type 4.0/5.0 node to determine, dynamically and without pre-configuration, the SNPA of a Type 2.1 physical unit with a known name. The SNPA can then be used to establish the SSCP-to-PU session for subarea communications.

2.1.1 SNA APPN Networks

The implementation in APPN end nodes and network nodes of the protocols defined in this paper will dramatically improve the usability and availability of APPN networks. Within APPN today, a significant amount of system definition is required for data link connections on a LAN. Each end node requires a link to be defined to its principal network node, and it is suggested that a second link be defined to an alternate network node for backup

service. These definitions are used to establish CP-CP sessions between the end node and its principal network node in order to use APPN services.

Currently, each network node requires a link to one adjacent network node, and for availability it is suggested that the network nodes have a series of links configured in a ring to provide alternate paths in case of failure. These links between network nodes are used to establish the CP-CP sessions necessary in order for APPN to provide services for the network.

The addition of the address resolution protocol will eliminate the system definitions of the MAC addresses and names of these links for CP-CP sessions. End nodes could determine the address of one of the network nodes that is currently active, and that address could be used to establish a CP-CP session. In the event of a failure of the network node serving a particular end node, the end node could dynamically establish a CP-CP session with an operable network node, freeing the EN of its previously static assignment to an NN. Furthermore, the network nodes could learn the addresses of other network nodes on the same LAN and could dynamically establish the data links for each of the CP-CP sessions to other NNs.

2.1.2 SNA Subarea Networks

Within each SNA peripheral node attached to a LAN, the system administrator is required to configure the MAC address (and in some cases, the LLC address) of the upstream subarea node. In many cases, the use of locally administered addresses is encouraged in order to facilitate replacing adapters, thereby eliminating any benefit gained from having a universally administered address.

For SNA hosts that interface to a LAN through more than one communication controller, it is common practice to use the same MAC address on both communication controllers. This provides for backup in the case where the communication controllers are attached to different token ring segments. However, this configuration will not work in the presence of transparent bridges or source routing transparent bridges since the operation of these bridges assumes that all adapters have unique MAC addresses.

The recommendation to meet this availability requirement for multiple communication controllers is to adopt the concept of a group name. Multiple communication controllers attached to the same host machine would respond to a group name that identifies the host SSCP. The communication controllers would then be able to use their universally administered address instead of being assigned a locally administered address.

The addition of an address resolution protocol and the use of group names for the communication controllers will substantially reduce the system configuration burden. The system administrator would be able to reconfigure the host complex – both the number of communication controllers and the number of adapters – without affecting the numerous peripheral nodes that are supported.

3. 3.0 FUNCTIONAL DESCRIPTION

The remainder of this document defines a collection of functions, together with the associated formats and protocols, that meet the address resolution requirements identified in Section 2.0. When implemented in a Server (NN) or Non-server(EN), the aggregate of these functions is called the Connection Control Entity (CCE) of the node. CCE formats and protocols are defined in detail in Sections 4.0 and 5.0. This section presents an overview of CCE functions, CCE frames, and the timers and counters associated with CCE protocols.

3.1 Elements of the Connection Control Entity

A station may be attached to one or more LANs. For each attached LAN, a station implements a specified subset of the CCE functions, depending on whether the station is a server or a non-server on the attached LAN. The subset of functions implemented in servers and non-servers is specified in Section 5.0. The single instance (per LAN) of CCE provides address resolution services for all the LLC and MAC sublayers instances associated with the LAN at that station. The CCE will be identified to the LLC sublayers by an LSAP that is fixed, architected, and known to all stations implementing CCE. This LSAP is defined to be X'DC' and should be opened on all MAC addresses at the station. One of the MAC sublayers at the station will receive and forward CCE frames to the CCE. The MAC address of that MAC sublayer, initially known only at the local station, is called the MAC CCE address for the station. Figure 1 illustrates the relationships among CCE, the MAC sublayer, the LLC sublayer, and a network entity in a station that has implemented CCE on a LAN.

In addition to the individual MAC CCE address, each CCE will also be addressable at the MAC layer through one or more architected functional addresses.[1] Every station will be capable of receiving frames through a MAC functional address called the "All CCE functional address". This functional address is fixed, architected, and known to all CCE stations. The All CCE functional address is defined to be X'C0 00 00 00 00 04'. The All CCE functional address must always be opened and monitored whenever a LAN attachment is active and the corresponding CCE is available at the station.

In addition to the All CCE functional address that is open at all CCE stations, stations will also monitor other MAC functional addresses that are specific to the station's role in the network. Non-server stations will monitor an address called the "Non-server functional address". This functional address is fixed, architected, and known to all stations. The Non-server functional address is defined to be X'C0 00 00 01 00 00'. The conditions in which this address must be open are defined later in this document. Communications server stations will monitor an address called the "Server functional address". This functional address is also fixed, architected, and known to all stations. The Server functional address is defined to be X'C0 00 00 00 00 40'. The Server functional address must always be opened and monitored by communications servers whenever the server is functioning at the station.[2]

[1] On Ethernet, IEEE 802.3, IEEE 802.4, and FDDI LANs, the CCE should use the group addresses with values equal to the functional addresses defined for IEEE 802.5 LANs. The purposes of the group addresses are identical to those of the functional addresses defined in the succeeding paragraphs.

[2] When the network entity providing routing services becomes unavailable, the CCE may cease monitoring the server functional address until the entity is again available.

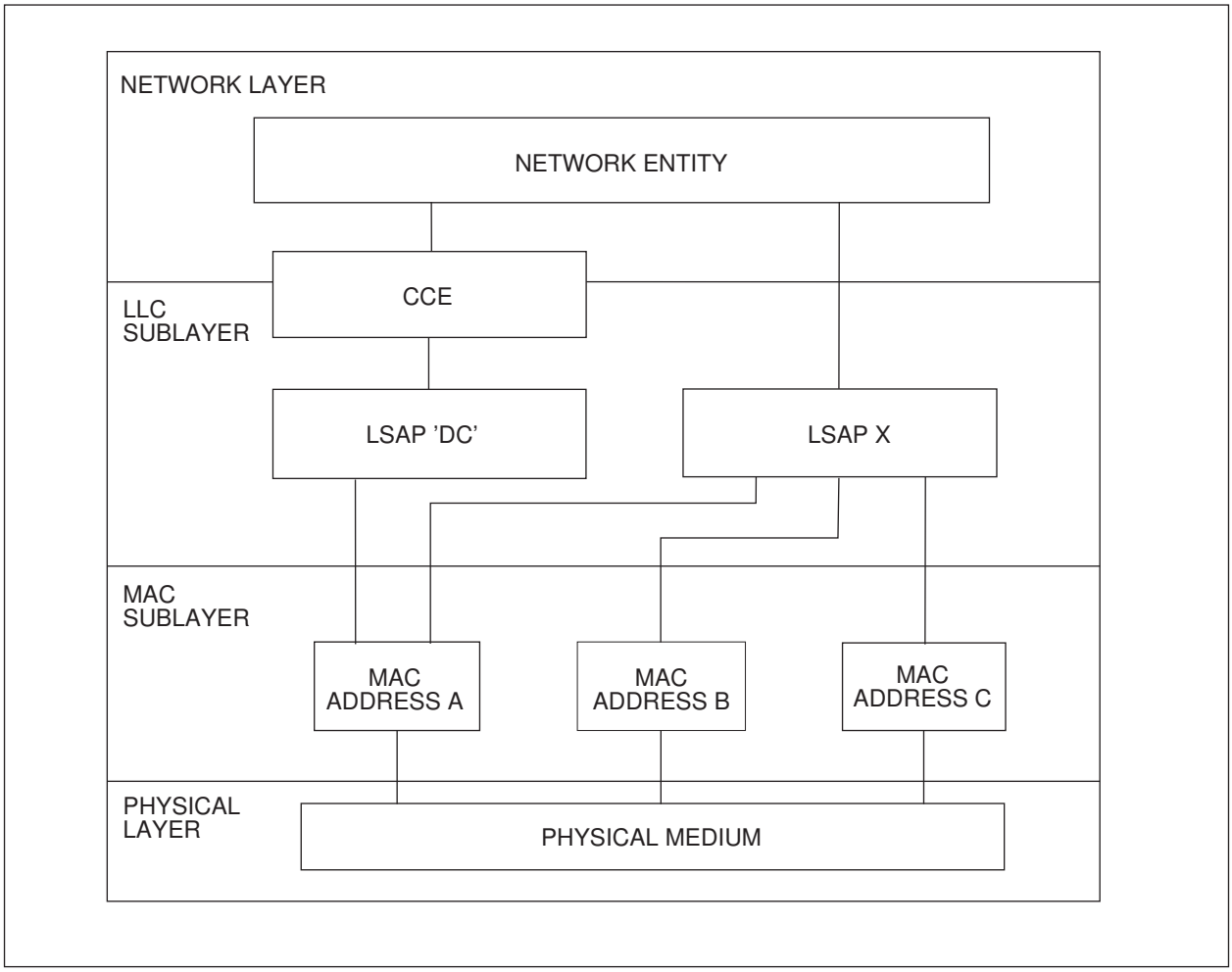


Figure 1. CCE in a LAN Station

In the context of the CCE architecture, communications servers are defined as stations at which a network entity provides some communications service ,i.e., routing capabilities are defined for a network entity. Possible routing capabilities for a network entity will be fixed, architected, and known to all stations. At the time of this release of the CCE architecture, three routing capabilities are defined. The first is that the network entity provide APPN network node services; the second is that the network entity provide subarea SNA boundary node services; the third is that the network entity provide APPC NameServer Services[3] . The routing capabilities of a network entity will be identified by a routing capabilities subvector for the network entity. Details of the format of routing capabilities subvectors can be found in Section 4.7.

As discussed in Section 3.2, the individual name and routing capabilities of the network entity will be recorded with the CCE at the server station. In addition, memberships of the network entity in various group names may also be recorded. An individual network entity at a server station (or, a "server resource") may be a member of one or more groups names. The group names may be user-defined. For example, user-defined group names could be used for any of the following purposes:

- partitioning networks of servers to assure that end systems receive services from a member of a specific group name. Such partitioning may be required for managerial, administrative, or technical reasons.

- reducing the reconfiguration required in end systems when server nodes are added or deleted from a group.

- reducing the reconfiguration required in end systems when subarea SNA Type 4.0 nodes are added or deleted to a single Type 5.0 node.

[3] Please refer to documentation on the APPC Application Suite for more information about the APPC NameServer.

- eliminating the need for the use of duplicate MAC addresses at Type 4.0 nodes.

Note that these uses of user-defined group names require that the appropriate group name (but not individual names or MAC addresses) be provided to end systems that seek services from members of a specific group name. There are fixed, architected group names that correspond to the architected routing capabilities that servers may belong to. The existence of these architected names will permit APPN and subarea SNA network entities to use CCE group name capabilities without the need for preconfiguration of server group names.

APPN and Subarea SNA servers will always be members of an "SNA Universal Group", the name of which is fixed, architected, and known to all stations. APPN Boundary Nodes will also belong to the group defined by the name IGO2HOST. This architected name is defined to be IROUTSNA (EBCDIC). APPC NameServers will always be members of a group defined by the name IGIVNAME.

3.2 CCE Functions

CCE provides the following functions to network entities in a communications network:

- finding the individual name and SNPA of one (or more) member of a specified group name that also has a specified routing capability.

- finding the individual names of all network entities that have a specified routing capability.

- finding the SNPA of a specified individual name.

The primitive associated with the first function is **Find_member**. The primitive that performs the second function is **Search**, and the **Find** primitive performs the third function.

Architectural Note: The CCE Primitives and Return Signals described above are conceptual and do not necessarily constitute an API. They are used in the same context as the conceptual primitives in the ISO/IEC family of telecommunication standards to provide an entry point to the protocol flows and enhance understanding of the basic concepts.

The major customer requirements described in Section 2.0 can be satisfied using these functions, singly or in combination. For example, an APPN end node can determine the SNPA of an APPN network node control point using the first function, because all NNs will be members of the universal group name IROUTSNA and the EN knows the architected routing capabilities of network nodes. By issuing the **Find_member** primitive, using the well-known group name IROUTSNA and well-known network node routing capability, the SNPA of a network node will be returned to the end node. The EN can then initiate a CP-to-CP session with the found network node.

Similarly, an APPN network node can determine the SNPA of other network nodes by using a combination of the **Search** and **Find** primitives. In this case, the APPN NN issues the **Search** primitive, using the well-known network node routing capability. The individual names of the control points with that capability are returned to the originating network node control point. Upon deciding which of the network nodes are required for a CP-to-CP session, the network node control point issues the **Find** primitive and receives the required SNPA. Similar methods can be used to satisfy the other requirements described in Section 2.0. All CCE primitives and return signals are defined in Figure 2.

By issuing the **Find_member** primitive, a network entity requests that CCE return the name of one (or more) resource in a specified netid that is a member of a specified group and that also has specified routing capabilities. By issuing **Find_member** with the find_flag set to 1, the network entity can also request that CCE return the SNPA of the first such group member that CCE identifies. The CCE design permits the identification of all group members that respond within a given time interval. CCE uses the **Member_indication** and **Find_member_complete** return signals to respond to the network entity that issued the **Find_member**. When **Find_member** is issued with the find_flag set to 1, **Member_indication** is returned immediately upon identification of the SNPA of the first member of the group satisfying the specified routing capabilities. **Find_member_complete** is returned when the given time interval has elapsed, and matching members of the group have responded.

The **Find** primitive allows a network entity to request that CCE determine an SNPA to reach an entity with a specified individual name. The network entity must already know the name of the entity that is sought. **Find_complete** is the return signal that CCE uses to respond to **Find** requests. CCE issues **Find_complete** immediately upon determining the requested SNPA. The **Find_complete** specifies the SNPA that CCE has found to connect to the specified name.

Primitive	Description
Find_member(find_flag, netid, group name, routing capabilities mask)	Higher layer request CCE to determine the name of one (or more) member of the specified group with the specified routing capabilities and netid, and optionally, the SNPA of one such member
Find (netid, name)	Higher layer requests CCE to determine SNPA of a network entity with the specified netid and name
Search (netid, group name, routing capabilities mask)	Higher layer requests CCE to determine all network entities that are members of the specified group and that have the specified routing capabilities and netid
Record (netid, name, routing capabilities, connection network id, list of SNPAs, list of group names)	Higher layer requests CCE to record a resource identified by *netid.name* as available, addressable through each of the list of SNPAs, having the specified routing capabilities, belonging to each of the list of group names and to the specified connection network.
Erase (netid, name)	Higher layer requests CCE to remove the resource identified by *netid.name* from the list of network entities recorded at this station.

Return Signal	Description
Member_indication(reply code, netid, name, connection network id, group name, SNPA, routing capabilities)	CCE determined (reply code = 0) SNPA as the address of *netid.name*, which is a member of group name with the specified routing capabilities **OR** CCE did not determine (reply code = 1) the SNPA of a member of group name with the specified routing capabilities.
Find_member_complete (reply code, netid, group name, list of names (and optionally, SNPAs), routing capabilities mask)	CCE determined (reply code=0) list of names (optionally including SNPAs) as the members of group name with the specified routing capabilities and netid **OR** CCE could not find (reply code=1) any member of group name with the specified capabilities and netid
Find_complete (reply code, netid, name, SNPA)	CCE determined (reply code=0) SNPA as the address of *netid.name* OR CCE could not find (reply code=1) SNPA for *netid.name*

Search_complete (reply code, netid, list of names (and optionally, SNPAs), list of CCE MAC addresses, group name, routing capabilities mask)	CCE determined (reply code=0) list of names (optionally including SNPAs) as the members of the specified group, with the specified routing capabilities and netid and with the corresponding list of CCE MAC addresses **OR** CCE could not find (reply code=1) any node with the specified group name and routing capabilities
Record_complete(netid, name)	CCE has recorded the routing capabilities, SNPAs, and group membership of *netid.name*
Erase_complete (netid, name)	CCE has erased *netid.name* from the list of network entities recorded at this station

Figure 2. CCE Primitives and Return Signals

The **Search** primitive allows a network entity to request that CCE determine all[4] individual names that have specified routing capabilities and that are members of a specified group. The CCE design permits CCE to identify all such individual names that respond within a given interval of time. CCE issues **Search_complete** return signal to respond to the **Search** request. The return signal is issued when the given interval of time has elapsed since the invocation of the **Search** primitive. The CCE architecture permits wild cards in the specification of routing capabilities, group names, and netid, thus allowing searches that return individual names that meet any of the three specifications separately (see section 4).

The **Record** and **Erase** primitives do not produce any line flows on the LAN. These primitives are mechanisms for server resources to add and delete themselves from the list of entities using the CCE server functions at a station. **Record_complete** and **Erase_complete** are the return signals associated with **Record** and **Erase**, respectively. The required parameters in the **Record** primitive are the individual name of the network entity and the list of SNPAs through which the entity is addressable. A network entity that provides communications routing services also includes a routing capabilities subvector in the **Record** primitive, and an entity that is a member of one or more groups includes a list of all group memberships.

The connection network identifier (CNID) will be recorded only by APPN network nodes. In order for CCE to support the recording and exchange of connection network ids, it is required that system administrators provide the same CNID to every network node on the LAN. If this is not done, CCE can produce unpredictable results when end nodes discover their CNID through CCE flows. When more than one CNID is needed on a LAN, it is recommended that the proper CNID be preconfigured at each node.

Note: A CCE may serve as a *proxy* CCE for resources on nodes without CCE functionality. To achieve this, a CCE user program records all of the pertinent information for the server resource, including the resources (remote) SNPA. It is left to the implementer how to determine if and how the program supplying the proxy information to the CCE determine when to perform **Record** and **Erase** functions as the resource goes up and down.

3.3 CCE Operation and Frames

The CCE architecture defines six frames that are exchanged among two or more instances of CCE at different stations on a LAN. These frames are: SOLICIT, ADVERTISE, FIND, FOUND, QUERY, and NOTIFY. All CCE frames are LLC unnumbered information (UI) frames. Section 4.0 presents the MAC and LLC frame formats for each frame. This section presents an overview of the frames, the conditions under which they are generated, and the responses of CCE upon receipt of the frames.

[4] The Search primitive does not guarantee that exactly all possible matching resources will be returned, but it is reasonable to make this assumption in most cases.

The SOLICIT frame is transmitted following invocation of the **Find_member** primitive. The intended use of the **Find_member** primitive is to allow non-servers to determine the SNPA of a server that is a member of a specified group name with specified routing capabilities and netid. Upon invocation of the **Find_member** primitive, the CCE starts the Solicit Timer and begins to monitor the CCE non-server functional address, in anticipation of receiving, before expiration of the timer, an ADVERTISE frame containing the specified parameters in its I-field. Such an ADVERTISE frame could be received in response to invocations at other non-servers. (When several non-servers are IPLed at the same or nearly the same time, for example. Such an ADVERTISE frame could also be received when a server has been recently IPLed.) If an ADVERTISE frame is received, the CCE proceeds as described below. If no ADVERTISE frame is received, the CCE issues a SOLICIT frame. The SOLICIT frame is addressed to the server functional address (all CCE frames are addressed to LSAP X'DC') and carries the specified parameters in the LLC I-field of the frame.

CCEs at servers monitor the server functional address whenever the server is functioning at the station.[5] Upon receipt of a SOLICIT frame, a server determines whether an individual name is recorded at that server as a member of the group name specified on the SOLICIT and with the specified netid. The server CCE then takes one of two actions. If the Advertise Timer is not active, i.e., no ADVERTISE frame is pending (the SOLICIT frame is an "initiating SOLICIT"), the CCE starts the Advertise Timer and enters the group name specified on the SOLICIT frame as the first entry on a list of group names to be included on subsequent ADVERTISE frames. If an ADVERTISE frame is pending, the server CCE simply adds the group name to the list of group names to be included on subsequent ADVERTISE frames. At the expiration of the Advertise Timer, CCE transmits an ADVERTISE frame to the non-server functional address, restarts the Advertise Timer, and increments the Advertise Count, i.e., the number of times ADVERTISE frames have been sent since the initiating SOLICIT. The frame carries the netid, routing capabilities, the list of group names, and the CCE MAC address at the server. Any ADVERTISE frame may always include the full list of group names associated with the server resource. When an ADVERTISE frame has been transmitted a number of times equal to the Advertise Count Maximum, the server stops sending frames until it receives the next initiating SOLICIT frame. An ADVERTISE is transmitted for each individual name that is a member of the specified group name. In addition to responding to SOLICIT frames as described above, a CCE at a server may optionally transmit ADVERTISE frames whenever the CCE is initialized. In this case, the ADVERTISE for a particular individual name includes all group names of which the individual name is a member.

As mentioned above, the CCE does not issue a SOLICIT frame immediately upon the invocation of the **Find_member** primitive. Instead, the CCE first monitors the non-server functional address in anticipation of receiving an ADVERTISE frame that might have been issued in response to an earlier SOLICIT. Thus, it is possible that the CCE will receive ADVERTISE frames from only a subset of the members of the group name specified in the **Find_member**. This case occurs when some of the responding servers have transmitted ADVERTISE frames a number of times equal to Advertise Count Maximum, while other servers have transmitted fewer than that number of ADVERTISE frames. In this case, the non-server would not issue its own SOLICIT frame and would therefore not find all members of the group name specified on the **Find_member**.

[5] The invocations of **Record** and **Erase** primitives serve as the CCE's indicators of when server resources are available.

Upon receipt of an ADVERTISE, the non-server CCE determines whether the group name, routing capabilities, and netid on the ADVERTISE match those specified in the **Find_member** primitive. If so, the CCE saves the associated information on the ADVERTISE and continues to monitor the non-server functional address for additional ADVERTISE frames. ADVERTISE frames may or may not contain the SNPA of the server resource. If the frame does not contain the resource's SNPA, a FIND/FOUND flow should be used to determine the connectable address of the resource (see description of the "Rotary" function in later in this section). Upon receipt of the first "matching" ADVERTISE, the non-server CCE may (find_flag = 1) return the **Member_indication** signal if the resource's SNPA was present on the ADVERTISE. If the resource's SNPA was not present, the CCE may transmit a FIND frame to the CCE MAC address from the ADVERTISE to determine the SNPA of the individual name from the ADVERTISE. If the FIND is transmitted, then upon receipt of the subsequent FOUND frame, the non-server CCE returns the information associated with the individual name to the network entity that invoked **Find_member**, using the **Member_indication** return signal. If no FOUND frame is received before the Find Timer expires, the non-server CCE signals that the FIND was not successful, using the negative reply code on the **Member_indication** return signal. In any case, the non-server CCE continues to monitor the non-server functional address for additional "matching" ADVERTISE frames until expiration of the Solicit Timer. At that time, the CCE returns the information gathered to the network entity that invoked **Find_member**, using the **Find_member_complete** return signal.

The FIND frame may be transmitted following the invocation of the **Find_member** primitive, as described above, and the **Search** primitive, as described later.. It may also be transmitted in response to the invocation of the **Find** primitive. The intent of the **Find** primitive is to allow a CCE at an initiating (non-server or server) station to determine the SNPA that can be used to address a specified individual name. Therefore the FIND frame carries the specified netid and name, the CCE MAC address of the initiator, and other information (including a correlator to be used for matching with responding frames). Upon invocation of the **Find** primitive, the CCE transmits a FIND frame (to LSAP X'DC') and starts the Find Timer. The FIND frame may be addressed to either the All CCE functional address or to an individual CCE MAC address. An individual CCE MAC address should be used whenever the specific address is known to the CCE. The specific address could be known, for example, because the address was previously cached upon receipt of a NOTIFY frame. When the individual CCE MAC address is not known, the FIND frame is addressed to the All CCE functional address. All CCE stations monitor their CCE MAC address and the All CCE functional address.

Upon receiving a FIND, a CCE determines whether the frame specifies a target resource that is recorded at that station. If so, the receiving CCE responds immediately with a FOUND frame that is addressed to the CCE MAC address from the FIND and carries the same correlator as the FIND. The FOUND frame carries the correlator and the SNPA of the individual name. The MAC address that is returned in the FOUND can be any of the MAC addresses in the resource's record with the CCE. The method for choosing which MAC address to return is left to the specific implementation. Basing the choice of MAC address on the load at individual MAC addresses is a reasonable implementation decision. The ability to choose which MAC address to return in the FOUND is called the Rotary function. Upon receipt of a FOUND with matching correlator, the initiating CCE returns the specified name and SNPA from the FOUND to the network entity that invoked the **Find** primitive. If the Find Timer expires before receipt of a matching FOUND, the initiating CCE retransmits the FIND for a specified number of times before returning a negative indication to the network entity that invoked the **Find** primitive.

The QUERY frame is transmitted by CCE following the invocation of the **Search** primitive by a network entity using CCE. The primary use intended for the **Search** primitive is to allow a CCE at a server station to determine individual names of other servers on the LAN with specified group name, netid, and routing capabilities. Upon invocation of the **Search** primitive, the CCE immediately transmits a QUERY frame to the Server functional address (and LSAP X'DC') that carries the routing capabilities, group name, and netid specified in the primitive, together with a correlator and the CCE MAC address of the initiating CCE. The initiating CCE also starts the Query Timer.

The CCE at a node monitors the Server functional address whenever server resources have been recorded with the CCE (either there is a server active on the node or a proxy record has been invoked). Upon receipt of a QUERY frame carrying routing capabilities, group name, and netid that match those of an individual name recorded at the server station, the server CCE responds with a NOTIFY frame addressed to the CCE MAC address from the QUERY. If the QUERY hits more than one record, then one NOTIFY is returned per resource. The NOTIFY frame carries the individual name, the CCE MAC address of the server, and the correlator from the QUERY. The NOTIFY frame can also optionally include the SNPA of the resource. Upon receipt of a NOTIFY frame with matching correlator, the initiating CCE saves the information from the NOTIFY and continues to monitor the CCE MAC address for additional NOTIFY frames from this and other servers. At the expiration of the Query Timer, the initiating CCE returns the saved names of servers to the network entity that invoked the **Search** primitive. The saved names and CCE MAC addresses may be cached by the CCE in order to determine specific addresses for subsequent FIND frames. Caching is optional and the decision is left to the implementation.

Architectural Note: Although either the **Find_member** or the **Search** primitives may be used to discover network resources that are members of a specified group and that have specified routing capabilities and netid, the two primitives use significantly different CCE frames to discover the network entities. As discussed above, **Find_member** uses the ADVERTISE/SOLICIT flows that are designed to allow one non-server to receive ADVERTISE frames that are transmitted by group members in response to SOLICIT frames from a separate non-server. The **Search** primitive does not allow a non-server to take advantage of "third-party" queries. In addition, **Search** uses the QUERY/NOTIFY flows, which require a group member to respond with a NOTIFY to every received QUERY. In cases in which many non-servers are attempting to identify appropriate group members, the use of **Search** rather than **Find_member** can therefore result in significantly increased traffic on the LAN and could have adverse effects on network performance. The possibly large number of non-servers in a network could further exacerbate this effect. The only benefit associated with the use of **Search** is that the non-server will receive responses from all group members, in contrast to the possibility that responses will be received from only a subset of group members when **Find_member** is used. As a result of these considerations, it is recommended that non-servers use **Find_member** to discover servers whenever possible, and that the use of **Search** be restricted to use by servers to discover other servers in the network.

3.4 CCE Timers and Counters

CCE includes the following timers and try counts:

- Find Timer. This is the time until repetition of FIND frames when no FOUND frame has been received. Find Timer takes values from 0 to Find Timer Maximum, which is a value from the range 0.5 to 20.0 seconds, defaulting to 9.0 seconds.

- Find Count. This is the number of times that a FIND frame is sent. Find Count takes values from 1 to Find Count Maximum, which is settable to 1,2,...6, defaulting to 2.

- Advertise Timer. This is the time until repetition of ADVERTISE frames. Advertise Timer takes values from 0 to Advertise Timer Maximum, which is a value from the range 0.5 to 60.0 seconds, defaulting to 5.0 seconds.

- Advertise Count. This is the number of times that an ADVERTISE frame is sent. Advertise Count takes values from 0 to Advertise Count Maximum, which is settable to 1, 2,...,20, defaulting to 10.

- Solicit Timer. This is the time the CCE gathers information from ADVERTISE frames before returning results to a network entity. The interval starts at the invocation of a **Find_member** primitive and again at the transmission of a SOLICIT frame (if no ADVERTISE frames were received during the first interval). Solicit Timer takes values from 0 to Solicit Timer Maximum, which is a value from the range 1 to 120 seconds, defaulting to 10 seconds.

- Query Timer. This is the time that a station waits after sending a QUERY frame to receive NOTIFY frames. Query Timer takes values from 0 to Query Timer Maximum, which is a value from the range 0.5 to 20.0 seconds, defaulting to 9.0 seconds.

4. 4.0 CCE FRAME FORMATS

This section describes the formats of CCE frames as transmitted on an IEEE 802.5 LAN. All CCE frames are LLC Unnumbered Information (UI) frames. The format of these CCE frames is shown below.[6]

Start-of-Frame Sequence (2 bytes)	Frame Control Field (1 byte)	Destination Address (6 bytes)	Source Address (6 bytes)	Routing Inform-ation (2 bytes)	LLC Inform-ation (variable)	Frame Check Sequence (4 bytes)	End-of-Frame Sequence (2 bytes)

The Start-of-Frame Sequence (SFS) consists of 2 one-byte fields: the starting delimiter and the access control field. The Frame Control (FC) is a 1-byte field defining the type of frame. The Frame Check Sequence (FCS) is a 4-byte cyclic redundancy check (CRC). The End-of-Frame Sequence (EFS) consists of two 1-byte fields: the ending delimiter and the frame status field.

The MAC Destination Address (MAC DA) is a 6-byte field identifying the receiving stations. The first bit of this field (the I/G bit) identifies whether the address is an individual or group MAC address. The second bit (the U/L bit) identifies whether the address is universally or locally administered. For the functional addresses used with CCE, the I/G bit is set to B'1' to indicate a group address, and the U/L bit is set to B'1' to indicate a locally administered address. Bit 0 of byte 2 is set to B'0' to indicate a functional address.

The MAC Source Address (MAC SA) is a 6-byte field identifying the sending ring station. The first bit, the routing information indicator (RII) bit, indicates whether the Routing Information Field (RIF) is present in the frame. For CCE frames on a source routing LAN, the RII bit is set to B'1' to indicate the presence of the RIF.[7] The remainder of the MAC SA provides the individual address of the sending station.

The Routing Information Field (RIF) consists of a two-byte Routing Control (RC) field and up to eight Route Designator (RD) fields. For CCE frames, the RD fields are added by bridges in the network, not by the sending station. (The RD fields record the route followed by CCE frame through the bridged LAN.) For CCE frames, the Routing Information Field (RIF) is a two-byte field specifying routing control (RC) information only. The first 3 bits of the RIF, the routing type (RT) bits, specify the type of route to be used. CCE frames will be Spanning Tree Explorer frames (sometimes called Single Route Broadcast or Limited Broadcast), so the RT bits should be set to B'110'. The next 5 bits of the RIF, the length (LTH) bits, identify the length of the RIF. These bits are set to B'00010' to indicate a length of 2 bytes. The next bit of the RIF, the Direction (D) bit, indicates how bridges should interpret the RIF. The D bit must be set to B'0'. The next 3 bits of the RIF, the Largest Frame (LF) bits, specify the largest information field that can be transmitted. The LF bits should be set to B'111' to indicate the largest frame that can travel on any path. The remaining 4 bits of the RIF are reserved and should be transmitted as B'0000'.[8]

The LLC Information Field has the following format:

Destination SAP (1 byte)	Source SAP (1 byte)	Control (1 byte)	Information (variable)

[6] Token-Ring Architecture Reference completely describes Token Ring MAC and LLC formats.

[7] For frames on Ethernet™, IEEE 802.3, and IEEE 802.4 LANs, the first bit of the MAC SA has no special meaning and is set to B'0' because the source address is an individual address. There is no RIF field in the MAC header.

[8] Source Routing Supplement to IEEE 802.1d (MAC Bridges) is a draft standard on the Source Routing Transparent (SRT) Bridge. The document contains additional information on the setting of the LF bits in an SRT environment.

The Destination Service Access Point (DSAP) is a one-byte field identifying the service access point (SAP) for which the data unit is intended. For CCE frames, the DSAP is X'DC'. The Source Service Access Point (SSAP) is a one-byte field identifying the SAP that originated the data unit. For CCE frames, the SSAP is also X'DC'. The control field identifies the type of LLC frame. For CCE frames, the control field is X'03', identifying the frame as a UI frame with the poll/final bit set to B'0'.

The Information field (I-field) is a variable length field that carries the information generated by CCE. Sections 4.1 through 4.6 give the specific format of each of the CCE frames for a source routing (token ring) LAN. In those sections, only the content of the I-field is specified. The formats of the I-field for each frame are described in detail in Section 4.7.

4.1 SOLICIT Frame

The SOLICIT frame is typically sent by a non-server CCE in order to determine the names and SNPA of servers that are members of a particular group name. The format of the SOLICIT frame is shown in the figure below. Detailed formats of the LLC I-field are shown in Section 4.7.

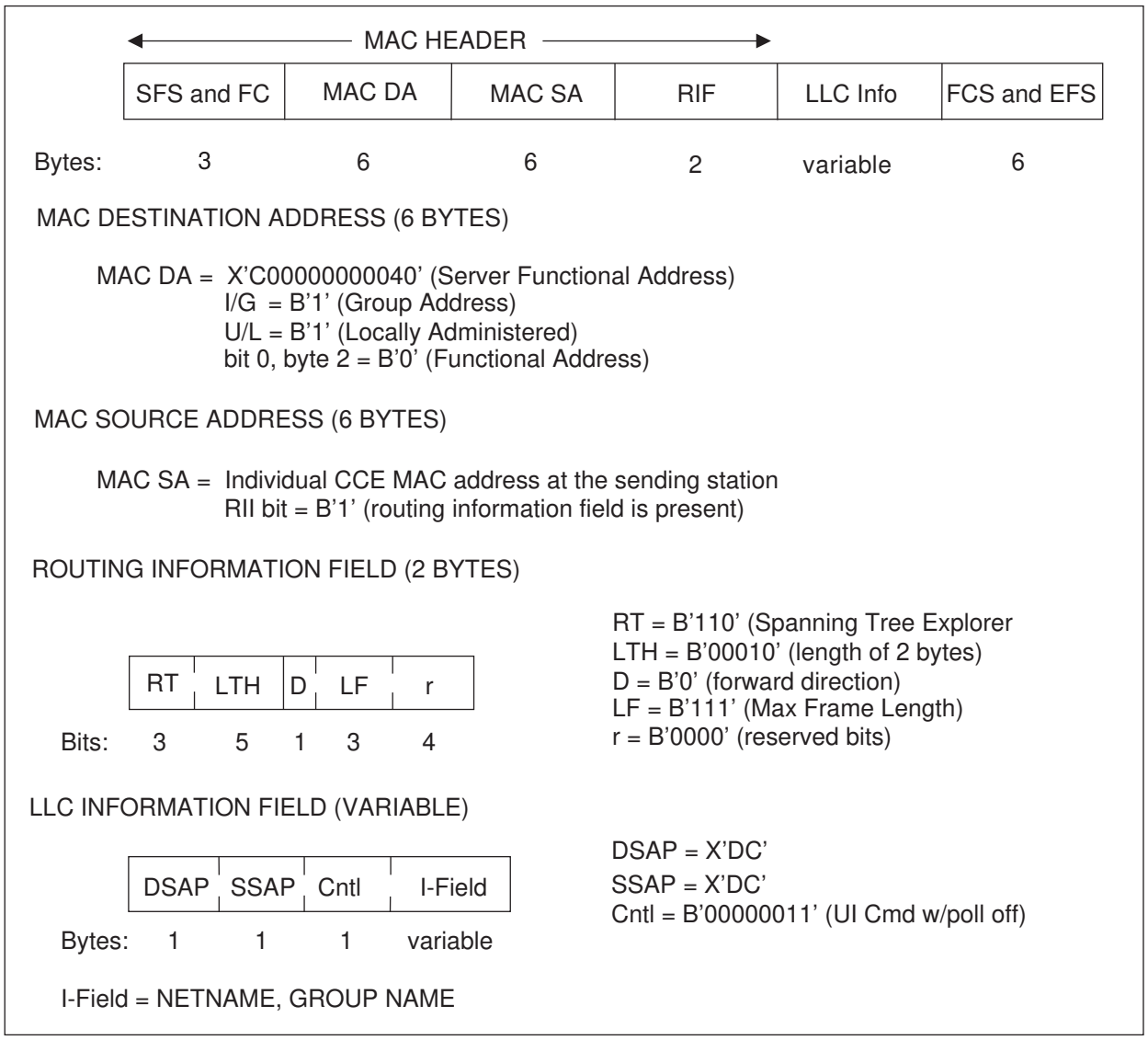


Figure 3. Format of the SOLICIT Frame

4.2 ADVERTISE Frame

The ADVERTISE frame is transmitted by a server CCE in response to the receipt of one or more SOLICIT frames. The ADVERTISE is transmitted whenever a received SOLICIT frame specifies a group name that matches one that has been recorded by a network entity at the station receiving the SOLICIT. The format of the ADVERTISE frame is shown in the figure below. Detailed formats of the LLC I-field are shown in Section 4.7

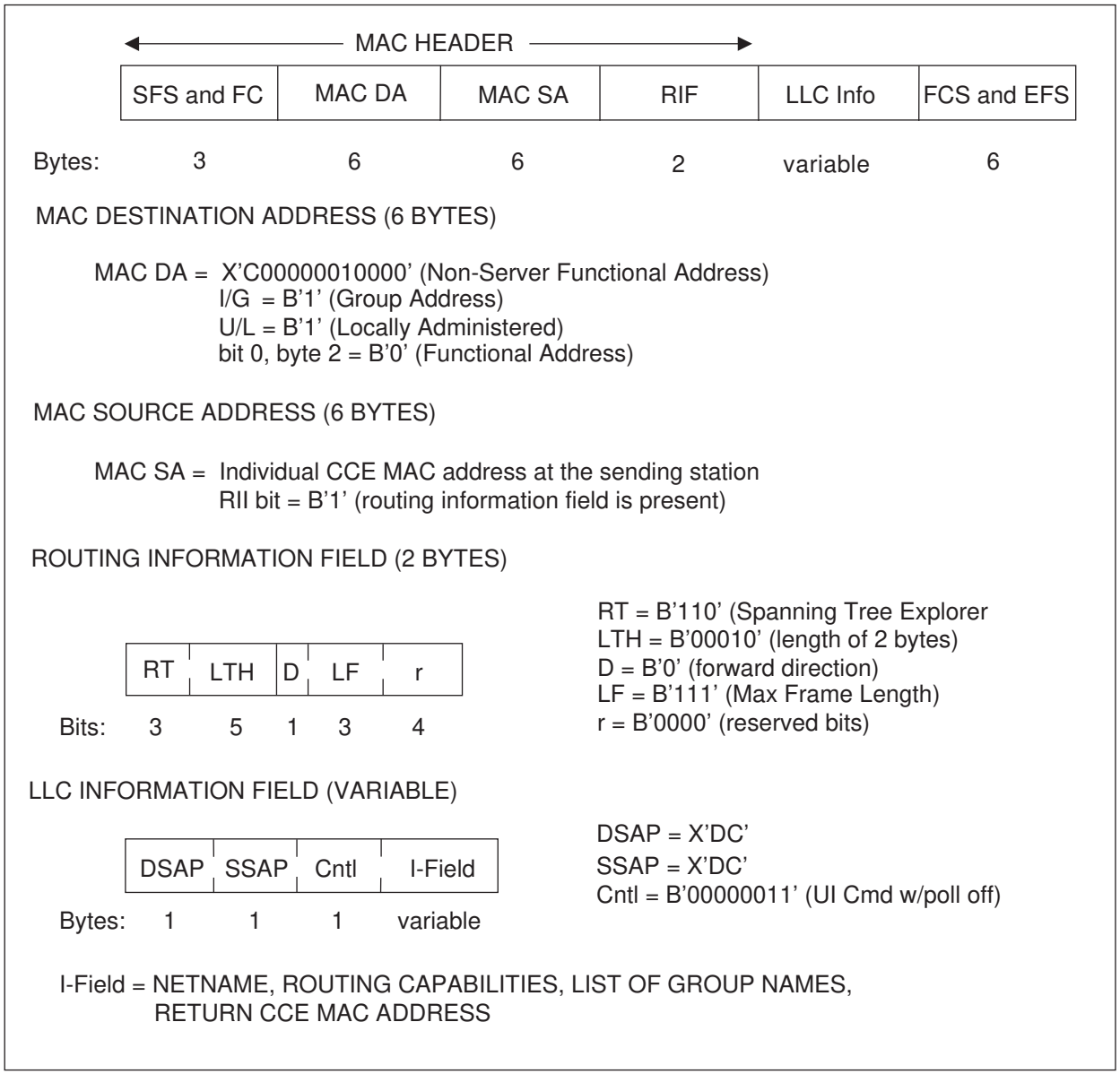


Figure 4. Format of the ADVERTISE Frame

4.3 FIND Frame

The FIND frame may be transmitted to either of two MAC destination addresses: the All CCE functional address or to the individual CCE MAC address that is contained in a received ADVERTISE or NOTIFY frame. The All CCE functional address is used when the CCE MAC address of the entity named on the **Find primitive** is not known. The figure below illustrates that form of the FIND frame. Detailed formats of the LLC I-field are shown in Section 4.7.

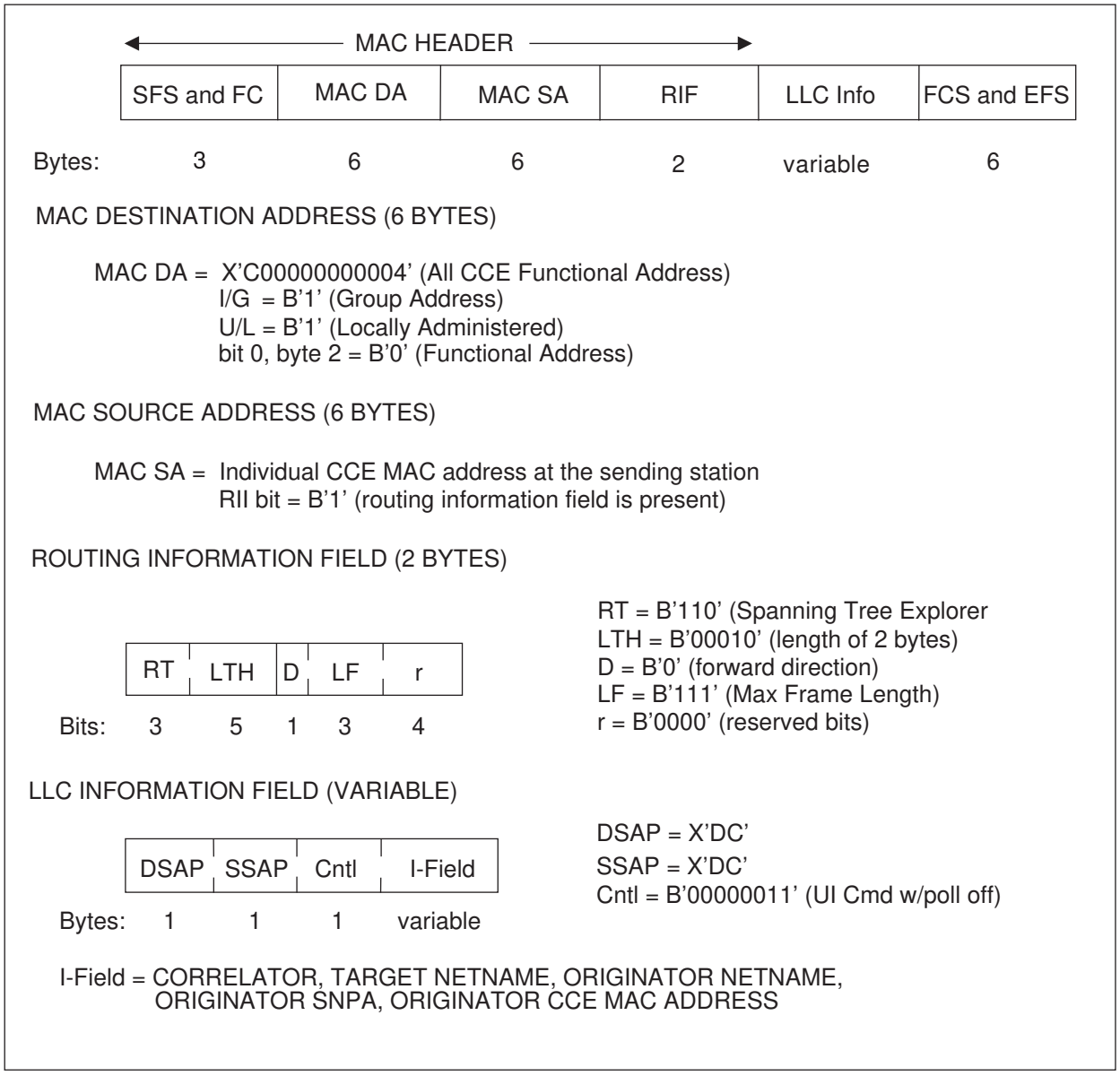


Figure 5. Format of the FIND Frame to the All CCE Functional Address

The FIND frame may also be transmitted to a known CCE MAC address. This form of the FIND is used whenever the **Find_member** primitive is invoked with find_flag set to 1 and the first matching ADVERTISE frame is received. The following figure illustrates that form of the FIND frame. Detailed formats of the LLC I-field are shown in Section 4.7.

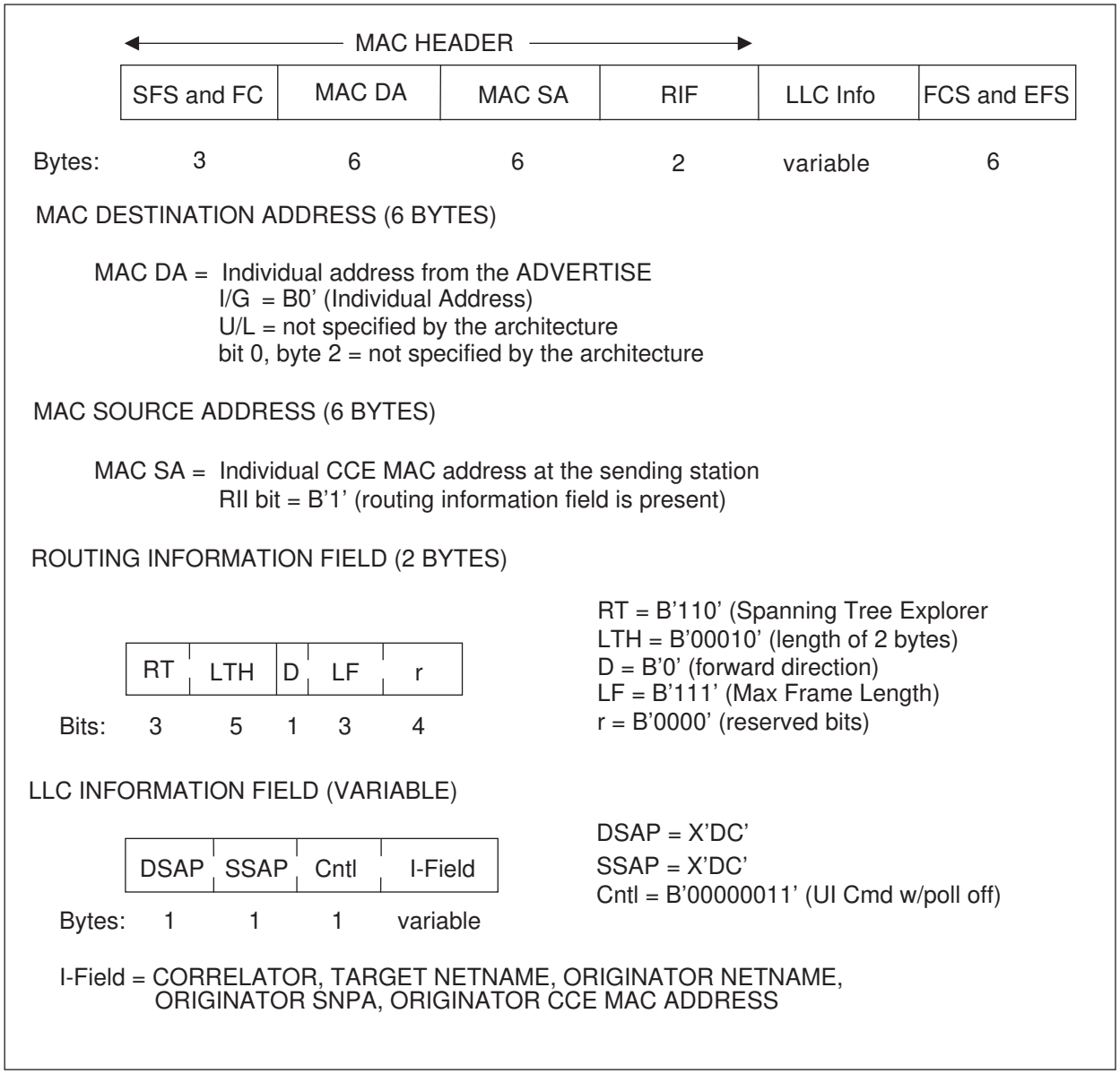


Figure 6. Format of the FIND Frame to an Individual CCE MAC Address

4.4 FOUND Frame

The FOUND frame is transmitted immediately upon receipt of a FIND frame that specifies a name matching a network entity recorded at the station receiving the FIND. The FOUND frame is transmitted to the CCE MAC address (individual address) that is specified on the FIND. The format of the FOUND frame is illustrated in the figure below. Detailed formats of the LLC I-field are shown in Section 4.7.

The MAC source address for the FOUND frame is the same MAC address that is specified in the LLC I-field of the frame (the target MAC address), and the FOUND should be transmitted from the target MAC address (and the corresponding physical connection to the LAN). For stations with multiple MAC addresses, the target MAC address may be different from the CCE MAC address that is used as the MAC source address for all other CCE frames.

The reason for this uniqueness of the FOUND frame among CCE frames is that the two addresses could conceivably reside on different segments of the LAN. A reliable (bridged) path could exist from the CCE MAC address to the receiver of the FOUND, while no reliable path exists from the target MAC address. The CCE station receiving the FOUND will attempt to use the MAC address in the LLC I-field for the establishment of any subsequent links. If the FOUND frame were transmitted and received from the CCE MAC address, then the receiver of the FOUND cannot be assured that a reliable path exists for the establishment of the data link. If the FOUND frame has been transmitted and received from the target MAC address, however, then the receiver of the FOUND can be reasonably assured that such a path exists.

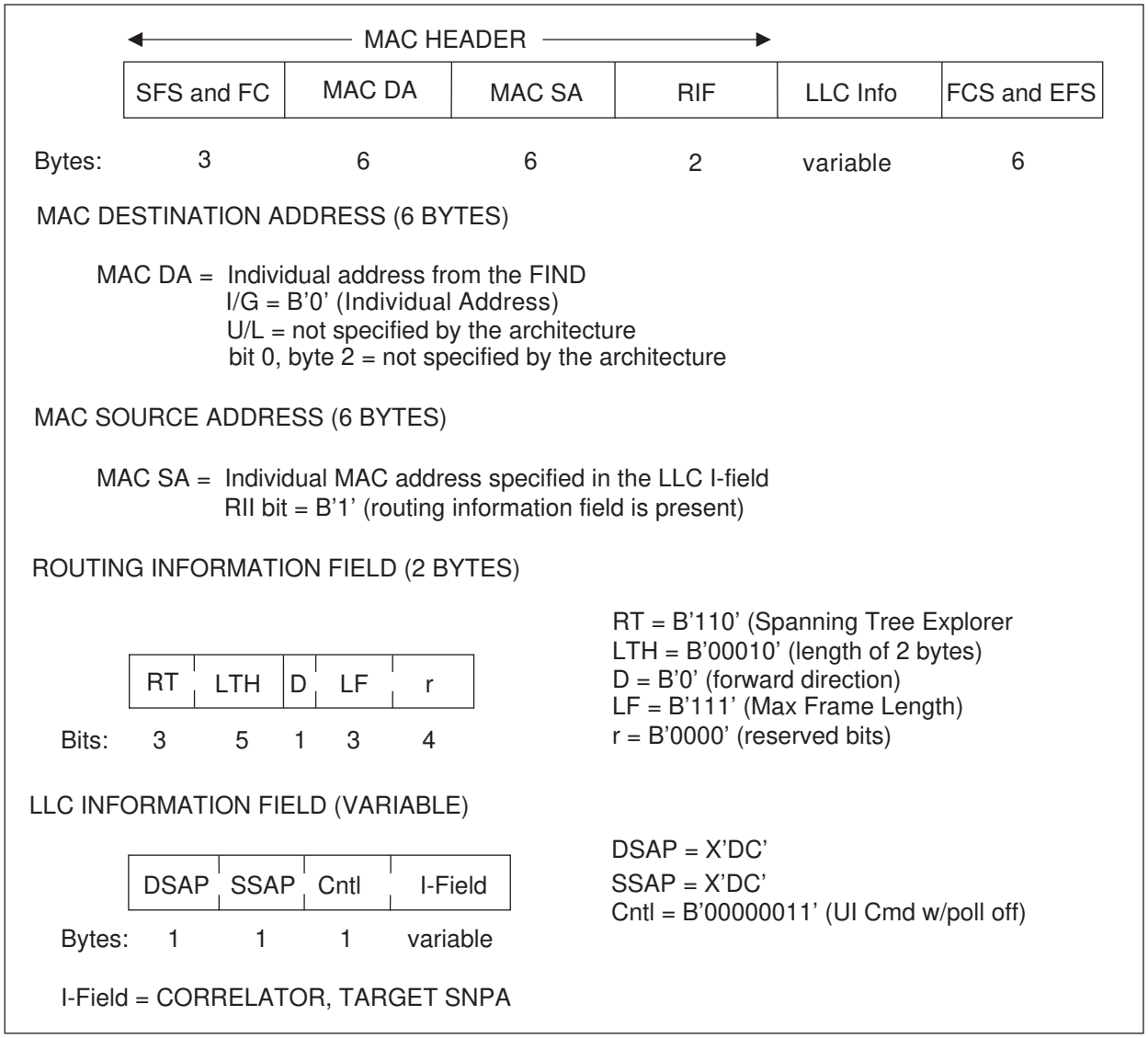


Figure 7. Format of the FOUND Frame

4.5 QUERY Frame

The QUERY frame is transmitted by the CCE upon invocation of the **Search** primitive by a network entity at the station. The QUERY frame is transmitted to the Server functional address so that all servers will receive the frame. The format of the frame is illustrated below. Detailed formats of the LLC I-field are shown in Section 4.7.

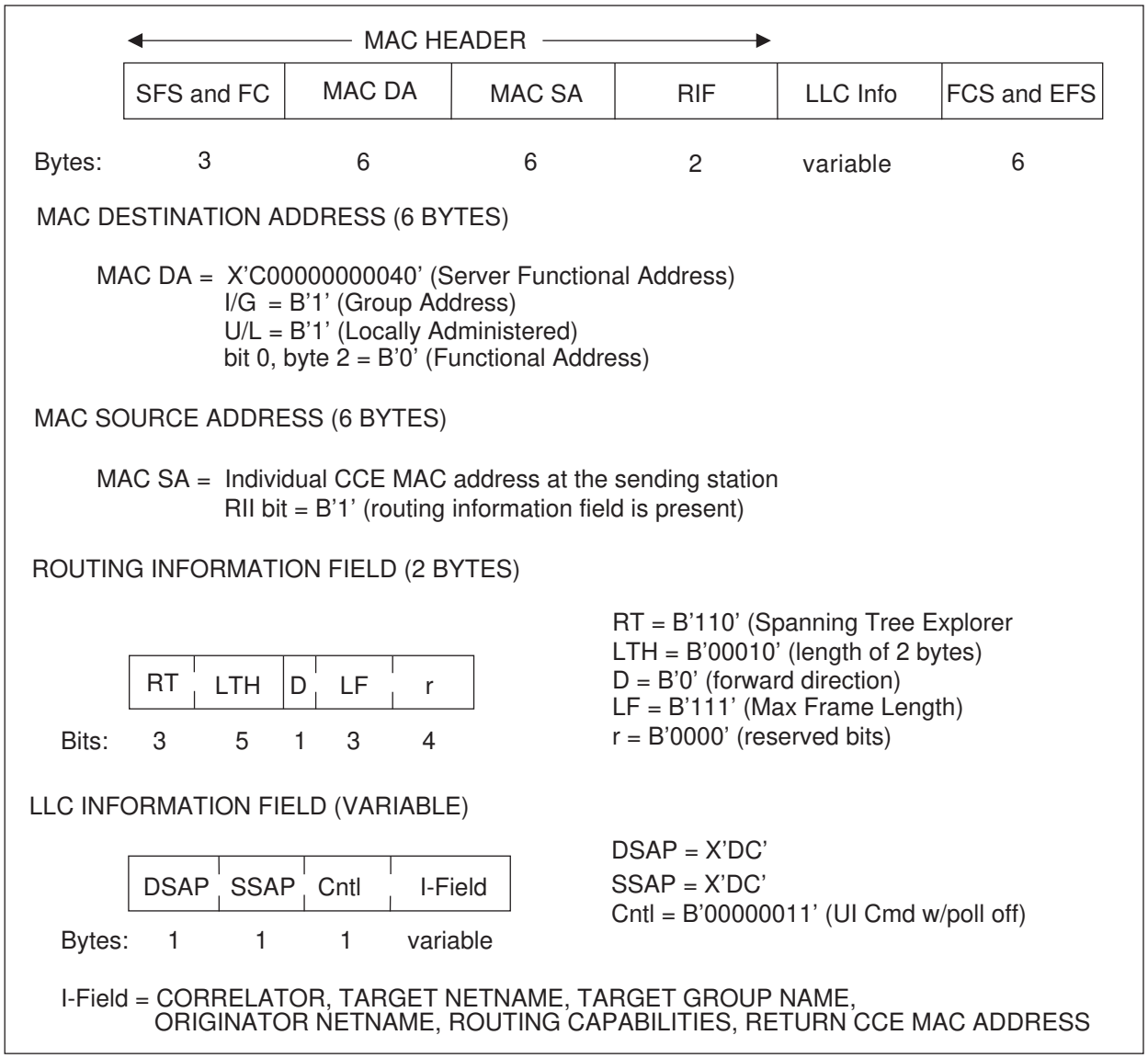


Figure 8. Format of the QUERY Frame

4.6 NOTIFY Frame

The NOTIFY frame is transmitted[9] by the CCE upon receipt of a QUERY specifying routing capabilities, group name, netid, and name that match those recorded by some network entity at the station receiving the QUERY. NOTIFY frames are transmitted to the CCE MAC address specified in the received QUERY and contains a correlator so that a NOTIFY frame can be matched with the QUERY that generated it. The format of the NOTIFY frame is illustrated below. Detailed formats of the LLC I-field are shown in Section 4.7.

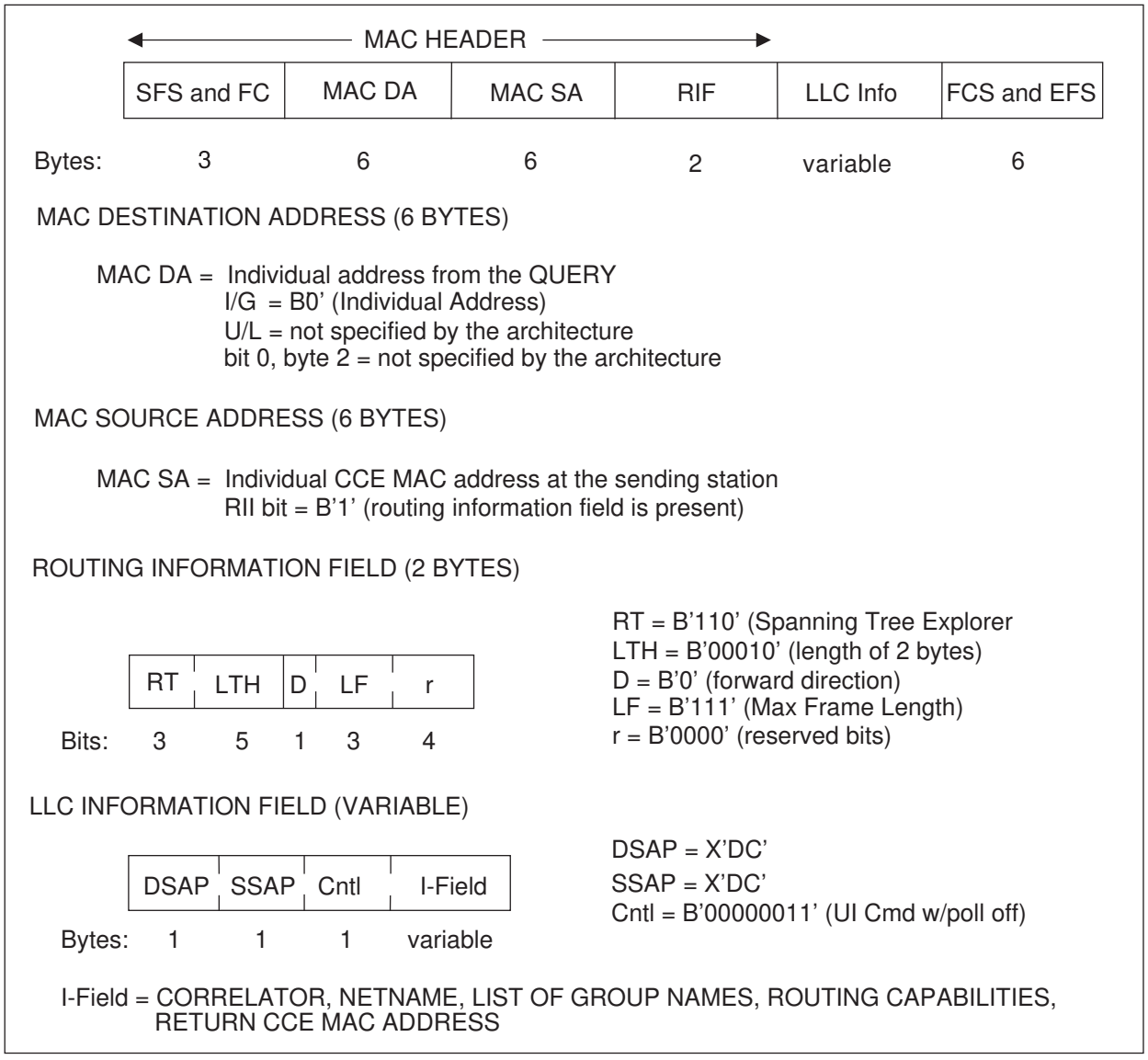


Figure 9. Format of the NOTIFY Frame

[9] If the search is used by non-servers it is recommended that the server randomize the time before sending the first NOTIFY to a time between zero and the Query Timer Maximum value minus the round trip delay. Generation of the random times is a local matter.

4.7 Format of the LLC I-field in CCE Frames

As described in the preceding sections, information is shared between CCEs in the LLC I-field of CCE frames. For all CCE frames, this information field, called the CCE information field, is a list of zero or more elements. The list of elements is called a major vector, and each element in the list is called a subvector. The format of major vectors and subvectors for CCE frames (defined below), permits the CCE information field to be self-contained and to expand as new requirements for information transfer are identified. The maximum length of the LLC I-field in any CCE frame is 480 bytes, and expansion of the field would be permissible up to this limit.

4.7.1 Major Vector Format

The format of major vectors is shown below:

MVL (2 bytes)	MVID (1 byte)	Subvector (variable)	.....	Subvector (variable)

A major vector consists of a major vector length (MVL), a major vector ID (MVID), and zero or more subvectors. The MVL is a 2-byte field that gives the length, in bytes, of the specific major vector, including the MVL field itself. This field is "big Endian", with the most significant byte first. Implementations on "little Endian" machines must swap the 2 bytes.

The MVID is a 1-byte field that identifies the specific CCE frame that is carrying the major vector.

4.7.2 Subvector Format

Subvectors may have either of the two formats shown below:

SVL (1 byte)	SVID (1 byte)	Value (variable)

SVL (1 byte)	SVID (1 byte)	Subvector (variable)

A subvector of the first type (simple subvector) consists of a subvector length (SVL), a subvector ID (SVID), and a single value associated with the subvector. A subvector of the second type (complex subvector) consists of a subvector length, subvector ID, and another subvector. The SVL is a 1-byte field that gives the length, in bytes, of the specific subvector, including the SVL field itself. The SVID is a 1-byte field. Bit 0 of the SVID field is the Simple/Complex (S/C) Indicator Bit, distinguishing between a Simple (Bit 0 = 0) or a Complex (Bit 0 = 1) Subvector. The remaining seven bits identify the specific subvector. Figure 10 identifies the major vectors and subvectors used by CCE. Only the ADVERTISE (MVID = X'02') and NOTIFY (MVID = X'06') frames include a complex subvector, the list of group names (SVID = X'81'). This complex subvector (SVID = X'81') is defined in the bottom portion of Figure 10.

Subvectors do not have to be in any specific order on a major vector. Also, if multiples of the same subvector appear on a major vector, the last one will be used. Subvectors not recognized are ignored. The specific implementation should log unexpected occurrences like multiple subvectors, missing subvectors, etc., but should not terminate.

Certain subvectors may be wildcarded on SOLICIT, QUERY, and FIND frames. Name subvectors may be identified as wildcards by being omitted or specifying a value of all blanks. The routing capabilities subvector is wildcarded by specifying a value of all zeros (X '00000000').

MVID	MV	SVID	Size (Bytes)	Wildcard Allowed	Optional	Value or Subvector
X'01'	SOLICIT Frame	X'0D'	1-8	Yes	No	Target Netid
		X'00'	1-8	Yes	No	Group Name
X'02'	ADVERTISE Frame	X'0E'	1-8	No	No	Resource Netid
		X'01'	1-8	No	No	Resource Name
		X'0B'	1	No	Yes*	Resource LSAP
		X'0C'	6	No	Yes*	Resource MAC Address
		X'02'	4	No	No	Routing Capabilities
		X'03'	3-17	No	Yes	Connection Network ID
		X'81'		No	No	Subvector of Group Names
		X'04'	6	No	No	Return CCE MAC Address
X'03'	FIND Frame	X'05'	4	No	No	Correlator
		X'0D'	1-8	Yes	No	Target Netid
		X'06'	1-8	Yes	No	Target Name
		X'0E'	1-8	No	No	Originator Netid
		X'07'	1-8	No	No	Originator Name
		X'08'	1	No	No	Originator LSAP
		X'09'	6	No	No	Originator MAC Address
		X'0A'	6	No	No	Originator CCE MAC Address
X'04'	FOUND Frame	X'05'	4	No	No	Correlator
		X'0B'	1	No	No	Resource LSAP
		X'0C'	6	No	No	Resource MAC Address
X'05'	QUERY Frame	X'05'	4	No	No	Correlator
		X'0D'	1-8	Yes	No	Target Netid
		X'00'	1-8	Yes	No	Group Name
		X'02'	4	Yes	No	Routing Capabilities
		X'04'	6	No	No	Return CCE MAC Address
X'06'	NOTIFY Frame	X'05'	4	No	No	Correlator
		X'0B'	1	No	Yes*	Resource LSAP
		X'0C'	6	No	Yes*	Resource MAC Address
		X'0E'	1-8	No	No	Resource Netid
		X'01'	1-8	No	No	Resource Name
		X'81'		No	Yes	Subvector of Group Names
		X'02'	4	No	No	Routing Capabilities
		X'04'	6	No	No	Return CCE MAC Address

Figure 10 (Part 1 of 2). CCE Major Vectors and Subvectors

*If Subvector X'0B' is returned, then Subvector X'0C' should be returned also.

SVID	Subvector	SVID	Value
X'81'	Subvector of Group Names	X'00'	Group Name

Figure 10 (Part 2 of 2. CCE major Vectors and Subvectors

4.7.3 Netid, Name, and Group Name Subvector Formats

Several subvectors in CCE frames specify a netid, the name of an individual network entity, or the name of a group of network entities. Note that these name subvectors are variable length fields and can be padded out to specific lengths if required by the specific implementation. The fields are left justified.

In addition to the architected group names defined in Section 3, all group names of the form Ixxxxxxx are reserved for future use.

Architectural Note: A group name will always be given when information is recorded on a CCE Server.

4.7.4 Connection Network ID Subvector Format

The Connection Network ID subvector (X'03') in the ADVERTISE frame is a variable length field made up of a 1-8 character-sized name field, a period, and a second 1-8 character-sized name field. The total length of the data in this subvector must be between 3 and 17 characters, including the period.

4.7.5 Routing Capabilities Subvector Format

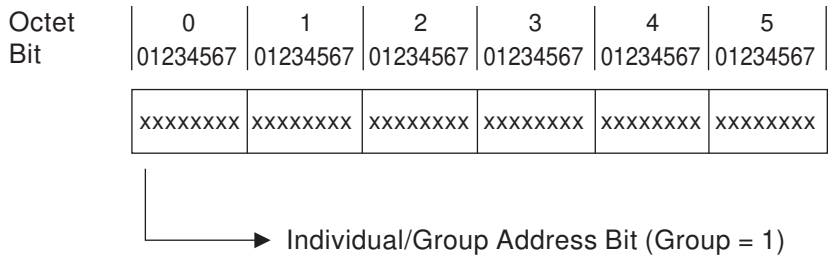
The Routing Capabilities Subvector is a 4-byte subvector of bit flags. The following code points have been architected:

- Byte 0, bit 0 - APPN Network Node Capability
- Byte 0, bit 1 - Subarea SNA Boundary Node Capability
- Byte 0, bits 2 through 7 - reserved for future use
- Byte 1, bits 0 through 7 - reserved for future use
- Byte 2, bit 0 - APPC NameServer Capability
- Byte 2, bits 1 through 7 - reserved for future use
- Byte 3, bits 0 through 7 - reserved for future use

The Routing Capabilities Subvector should be used only to identify the architected communications capabilities. The establishment of reserved bits of the subvector to identify additional communications routing capabilities will be handled through agreements reached in the AIW.

4.7.6 MAC Address Subvector Format

Subvectors containing MAC addresses are specified in the LLC I-field of several CCE frames. MAC addresses are selected from a 48-bit address space defined by the IEEE Project 802 standards committee on LANs. Within this space, the first bit in an address (called the Individual/Group, or I/G, bit) specifies whether the address is a group MAC address (I/G = 1), as illustrated below:



For example, the MAC address B'10000000 00000000 01011010 11110000 00101110 00010101' *(shown as it would be transmitted in the DA field of a MAC header) is a group address.

There are two possible methods to represent MAC addresses in the MAC Address Subvector of CCE frames. The first is to treat each octet in the address as most significant bit first (or MSB). The second method is to treat each octet as least significant bit first (LSB or canonical form). **The canonical form will be used to represent MAC addresses in the MAC Address Subvector of CCE frames.**[10] Using this representation, the above address would be represented as X'01-00-5A-0F-74-A8' in the value field of subvectors as shown in Figure 11.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
0000 0001	0000 0000	0101 1010	0000 1111	0111 0100	1010 1000

Figure 11. Canonical Form for MAC Address

4.7.7 LSAP Subvector Format

Subvectors containing Logical Service Access Points (LSAP) are specified in several CCE frames. LSAP addresses are one-byte fields specifying the logical point at which a network entity acquires the services of the Data Link Control layer. See Token-Ring Architecture Reference for a complete description of LSAP address fields. In CCE frames, these addresses should be specified as individual addresses (Individual/Group bit = B'0') and may be either User-defined or IEEE-defined (User-defined bit either B'0' or B'1'). **LSAPs must also be represented in canonical form in CCE frames.**

4.7.8 Correlator Subvector Format

Subvectors containing Correlators are specified in several CCE frames. Correlators are used to match NOTIFYs with QUERYs and FOUNDs with FINDs. The Correlator subvector is a 4-byte subvector generated and maintained at each CCE that initiates a search with a QUERY or a FIND. Server CCEs should echo the Correlator on NOTIFY and FOUND frames.

[10] "IEEE Standard 802-1990, Overview and Architecture" should be also consulted for additional information about the cononical representation of MAC addresses.

5. 5.0 CCE FINITE STATE MACHINES

These finite state machines (FSM) describe the actions of CCE in responding to primitives issued by network entities and the actions of CCE when each of the CCE frames is received. For clarity in the presentation and explanation of the architecture, FSMs are described as if the implementation is "single-threaded", i.e., the actions corresponding to a single invocation of a primitive are completed before another invocation of the same primitive. In practice for some operating systems, e.g., OS/2 and AIX, the FSM may be implemented in multi-threaded and/or multi-tasked environments, permitting asynchronous invocation of the same primitive. Such implementations are not precluded by the architecture.

5.1 Responding to Find_member Request

The FSM in Figure 12 implements the CCE responses upon invocation of the **Find_member** primitive. It specifies the conditions under which SOLICIT and FIND frames are transmitted and defines responses to receipt of ADVERTISE and FOUND frames. This FSM should be implemented by all CCE station, both servers and non-servers.

Input Name	Description
Find_member(find_flag, netid, group name, routing capabilities mask)	.CCE invoked to determine the name of one (or more) member of the specified group with the specified routing capabilities and netid and optionally (find_flag = 1) the SNPA of one such member
ADVERTISE received with correct group name, routing capabilities, and netid, but no resource SNPA	CCE receives ADVERTISE frame with group name matching one in the SOLICIT frame, with routing capabilities matching the mask in the SOLICIT frame, and with netid matching the one in the SOLICIT frame. The frame does not include the SNPA of the resource
ADVERTISE received with correct group name, routing capabilities, netid, and including resource SNPA	CCE receives ADVERTISE frame with group name matching one in the SOLICIT frame, with routing capabilities matching the mask in the SOLICIT frame, and with netid matching the one in the SOLICIT frame. The frame also includes the SNPA of the resource
ADVERTISE received with incorrect group name, routing capabilities, or netid	CCE receives ADVERTISE frame with either group name, routing capabilities, or netid not matching those in previously sent SOLICIT frame
FOUND received with the correct correlator	CCE receives a FOUND frame with a correlator matching the one on the previously sent FIND frame
FOUND received with an incorrect correlator	CCE receives a FOUND frame with a correlator not matching the one on the previously sent FIND frame
Solicit timer Expires	Solicit timer expires
Find timer Expires	Find timer expires
Deactivate	CCE receives a signal to deactivate
Activate	CCE receives a signal to activate

Figure 12 (Part 1 of 4). Responding to Find_member Request

Output Code	Function
a	Open nonserver functional address. Start Solicit Timer at Solicit Timer Maximum. Set BF = 0. Set CF = 0. Set FF = find_flag.
b	Add name from the ADVERTISE to the list of names for return to Higher Layer. Set CF = 1.
c	Send FIND frame to the CCE MAC address from the ADVERTISE. Start the Find Timer at Find Timer Maximum. Add name from the ADVERTISE to the list of names for return to the Higher Layer. Set CF = 1. Set DF = 0.
d	Send SOLICIT frame to Server Functional Address. Start Solicit Timer at Solicit Timer Maximum. Set BF = 1.
e	Signal Member_indication with reply code = 0. Stop the Find Timer.
f	Signal Find_member_complete with reply code=1.
g	Stop all timers. Deactivate the FSM. Signal that deactivation has occurred.
h	Set DF = 1.
j	Signal Member_indication with reply code = 0. Signal Find_member_complete with reply code = 0. Stop the Find Timer.
n	Start the FSM. Signal that activation has occurred.
p	Signal Find_member_complete with reply code=0.
q	Signal Find_member_complete with reply code=0. Signal Member_indication with reply code = 1.
r	Discard the ADVERTISE frame.
s	Discard the FOUND frame.
t	Signal Member_indication with reply code = 1.
u	Signal Member_indication with reply code = 0. Signal Find_member_complete with reply code=0. Stop the Solicit Timer.
v	Signal Member_indication with reply code = 0. Signal Find_member_complete with reply code=0. Stop the Find Timer. **-OR-** Add to list of advertises and continue to wait for the FOUND.

Figure 12 (Part 2 of 4). Responding to Find_member Request

Flag	Value	
	0	1
BF	Passive Listen (no SOLICIT frame sent)	Active Listen (SOLICIT has been sent)
CF	No ADVERTISEs received	At least one ADVERTISE received
DF	Find Timer active, Solicit Timer active	Find Timer active, Solicit Timer expired
FF	**Find_Member** Find Flag = 0	**Find_Member** Find Flag = 1

Figure 12 (Part 3 of 4). Responding to Find_member Request

Inputs	States			
	Waiting for Invocation	Waiting for ADVERTISE	Waiting for FOUND	Inactive
	01	02	03	04
Find_member(find_flag, netid, group name, routing capabilities mask)	2(a)	/	/	/
ADVERTISE received with incorrect group name, routing capabilities, or netid	/	-(r)	-(r)	/
ADVERTISE received with correct group name, routing capabilities, netid, and including resource SNPA; FF=1	/	1(u)	1 or 3(v)	/
ADVERTISE received with correct group name, routing capabilities, netid, and including resource SNPA; FF=0	/	-(b)	-(b)	/
ADVERTISE received with correct group name, routing capabilities, and netid, but no resource SNPA; CF=1	/	-(b)	-(b)	/
ADVERTISE Received with correct group name and routing capabilities, and netid, no SNPA; CF=0, FF=1	/	3(c)	/	/
ADVERTISE Received with correct group name and routing capabilities, and netid, no SNPA; CF=0, FF=0	/	-(b)	/	/
Solicit Timer Expires, CF=1	/	1(p)	-(h)	/
Solicit Timer Expires, CF=0, BF=0	/	-(d)	/	/
Solicit Timer Expires. CF=0, BF=1	/	1(f)	/	/
Find Timer Expires. DF=0, FF=1	/	/	2(t)	/
Find Timer expires. DF=1, FF=1	/	/	1(q)	/
FOUND received with incorrect correlator.	/	/	-(s)	/
FOUND received with correct correlator. DF=0, FF=1	/	/	2(e)	/
FOUND received with correct correlator. DF=1, FF=1	/	/	1(j)	/
Deactivate	4(g)	4(g)	4(g)	-
Activate	/	/	/	1(n)

Figure 12 (Part 4 of 4). Responding to Find_member Request

5.2 Responding to Find Request

The FSM in Figure 13 implements the CCE responses upon invocation of the **Find** primitive. It specifies the conditions under which FIND frames are transmitted and defines responses to receipt of FOUND frames. This FSM should be implemented by all CCE stations, both servers and non-servers.

Input Name	Description
Find(netid, name)	CCE invoked to determine SNPA of the specified name with the specified netid
FOUND received with the correct correlator	CCE receives a FOUND frame with a correlator matching the one on the previously sent FIND frame
FOUND received with an incorrect correlator	CCE receives a FOUND frame with a correlator not matching the one on the previously sent FIND frame
Find timer Expires	Find timer expires
Deactivate	CCE receives a signal to deactivate
Activate	CCE receives a signal to activate

Figure 13 (Part 1 of 3). Responding to Find Request

Output Code	Function
g	Stop all timers. Deactivate the FSM. Signal that deactivation has occurred.
k	Send FIND frame to the All CCE MAC address (or CCE MAC address if caching is used) Start Find Timer at Find Timer Maximum. Set Find Count = 1.
j	Resend FIND frame to the All CCE MAC address (or CCE MAC address if caching is used). Start Find Timer at Find Timer Maximum. Increment Find Count.
m	Signal Find_complete with reply code = 1.
n	Start the FSM. Signal that activation has occurred.
p	Signal Find_complete with reply code = 0. Stop the Find Timer.
s	Discard the FOUND frame.

Figure 13 (Part 2 of 3). Responding to Find Request

Inputs	States		
	Waiting for Invocation	Waiting for FOUND	Inactive
	01	02	03
Find(netid, name)	2(k)	/	/
Find Timer expires. Find Count = Find Count Maximum.	/	-(l)	/
Find Timer expires. Find Count <> Find Count Maximum.	/	-(J)	/
FOUND received with incorrect correlator.	/	-(s)	/
FOUND received with correct correlator.	/	1(p)	/
Deactivate	3(g)	3(g)	-
Activate	/	/	1(n)

Figure 13 (Part 3 of 3). Responding to Find Request

5.3 Responding to SOLICIT Frames

The FSM in Figure 14 implements the CCE response to the receipt of SOLICIT frames. It specifies the conditions under which ADVERTISE frames are transmitted, as well as the contents of the ADVERTISE frames as determined by the group names that have been specified on previously received SOLICIT frames. This FSM should be implemented by all stations that are servers.

Architectural Note: The server CCE implementation may choose whether or not to provide the Resource SNPA on ADVERTISE frames. This decision should be based on the need for providing the Rotary function or other load balancing functions on the station.

Input Name	Description
SOLICIT frame received with correct group name and netid	CCE receives a SOLICIT frame with group name and netid matching those recorded for a network entity at this station.
SOLICIT frame received with incorrect group name or netid	CCE receives a SOLICIT frame with group name or netid not matching any recorded for a network entity at this station.
Advertise Timer Expires	Advertise timer expires.
Deactivate	CCE receives a signal to deactivate
Activate	CCE receives a signal to activate

Figure 14 (Part 1 of 3. Responding to SOLICIT Frames

Output Code	Function
a	Activate the FSM. Add all group names to the Advertise List. Set Advertise Count = 0. Start the Advertise Timer at a time chosen randomly between 0 and the Advertise Timer Maximum
b	Deactivate the FSM. Signal that deactivation has occurred.
c	Discard the SOLICIT frame.
d	Add the group name from the SOLICIT to the Advertise List.
e	Send ADVERTISE frame to Non-Server functional address. Increment Advertise Count. Restart Advertise Timer = Advertise Timer Maximum.
f	Add the group name from the SOLICIT to the Advertise List. Set Advertise Count = 0. Start the Advertise Timer at a time chosen randomly between 0 and Advertise Timer Maximum.
g	Stop the Advertise Timer (Inactivate Advertise Timer). Send ADVERTISE frame to the Non-Server functional address. Increment Advertise Count. Delete all group names from the Advertise List.

Figure 14 (Part 2 of 3). Responding to SOLICIT Frames

Inputs	States	
	Active	Inactive
	01	02
SOLICIT frame received with incorrect group name or netid.	-(c)	/
SOLICIT frame received with correct group name and netid. Advertise Timer Inactive.	-(f)	/
SOLICIT frame received with correct group name and netid. Advertise Timer Active.	-(d)	/
Advertise Timer Expires. Advertise Counter = Advertise Count Maximum - 1.	-(g)	/
Advertise Timer Expires. Advertise Counter <> Advertise Count Maximum -1.	-(e)	/
Deactivate signal received	2(b)	-
Activate signal received	-	1(a)

Figure 14 (Part 3 of 3). Responding to SOLICIT Frames

5.4 Responding to FIND Frames

The fourth CCE FSM (see Figure 15) implements the CCE responses to the receipt of FIND frames. It specifies the conditions under which responding FOUND frames are transmitted. This FSM should be implemented by server CCE stations. Non-server stations may implement the FSM if the Record primitive is available to higher layer entities..

Input Name	Description
FIND frame received with correct netid and name.	CCE receives a FIND Frame with *netid.name* matching one recorded at this station.
FIND frame received with incorrect netid and/or name.	CCE receives a FIND Frame with *netid.name* not matching any recorded at this station.
Deactivate	CCE receives a signal to deactivate
Activate	CCE receives a signal to activate

Figure 15 (Part 1 of 3). Responding to FIND Frames

Output Code	Function
a	Choose which MAC address to use for return SNPA. Send a FOUND Frame to the address from the received FIND.
b	Deactivate the FSM. Signal that deactivation has occurred.
c	Activate the FSM. Signal that activation has occurred.
d	Discard the FIND frame.

Figure 15 (Part 2 of 3). Responding to FIND Frames

Inputs	States	
	Active	Inactive
	01	02
FIND frame received with incorrect netid and/or name.	-(d)	/
FIND frame received with correct netid and name.	-(a)	/
Deactivate signal received	2(b)	-
Activate signal received	-	1(c)/

Figure 15 (Part 3 of 3). Responding to FIND Frames

5.5 Responding to Search Request

The FSM in Figure 16 implements the CCE responses to the invocation of the **Search** primitive. It identifies the conditions under which QUERY frames are transmitted and the actions upon receipt of NOTIFY frames. This FSM should be implemented by both server and non-server CCE stations.

Input Name	Description
Search(netid, group name, routing capabilities mask)	CCE invoked to determine all network entities that are members of the specified group with specified routing capabilities and netid
NOTIFY received with correct netid, group name, and routing capabilities	CCE receives NOTIFY frame with routing capabilities matching mask in the QUERY frame, with group name matching that in the QUERY frame, and with netid matching that in the QUERY frame
NOTIFY received without correct routing capabilities, netid, or group name	CCE receives NOTIFY frame with either routing capabilities not matching mask in the QUERY frame, group name not matching that in the QUERY frame, or netid not matching that in the QUERY frame
Query Timer Expires	Query timer expires
Deactivate	CCE receives a signal to deactivate
Activate	CCE receives a signal to activate

Figure 16 (Part 1 of 4). Responding to Search Request

Output Code	Function
b	Store information from NOTIFY. Set AF = 1.
d	Send QUERY frame to Server Functional Address. Reset and start Query Timer at Query Timer Maximum. Set AF = 0.
f	Signal Search_complete with reply code=1. Stop Query Timer.
g	Stop all timers. Deactivate the FSM. Signal that deactivation has occurred.
k	Discard the NOTIFY frame.
n	Activate the FSM. Signal that activation has occurred.
q	Signal Search_complete with reply code=0. Stop Query Timer.

Figure 16 (Part 2 of 4). Responding to Search Request

Flag	Value	
	0	1
AF	No NOTIFYs received	At least one NOTIFY received

Figure 16 (Part 3 of 4). Responding to Find_member Request

Inputs	States		
	Waiting for Invocation	Waiting for NOTIFY	Inactive
	01	02	03
Search(netid, group name, routing capabilities mask)	2(D)	/	/
NOTIFY received with correct netid, group name, and routing capabilities	/	-(B)	/
NOTIFY received without correct routing capabilities, netid, or group name	-(K)	-(K)	/
Query Timer Expires. AF = 0.	/	1(F)	/
Query Timer Expires. AF = 1.	/	1(q)	/
Deactivate	3(g)	3(g)	/
Activate	/	/	1(n)

Figure 16 (Part 4 of 4). Responding to Search Request

5.6 Responding to QUERY Frames

The FSM in Figure 17 implements the CCE responses to receipt of a QUERY frame. It specifies the conditions under which NOTIFY frames are transmitted and. This FSM should be implemented by all CCE server stations.

Input Name	Description
QUERY frame received with matching netid, group name, and routing capabilities	CCE receives a QUERY Frame with netid, group name and routing capabilities matching those recorded for a network resource at this station.
QUERY frame received without matching **netid**, group name, or routing capabilities.	CCE receives a QUERY Frame with either netid, group name, or routing capabilities not matching those recorded for any network resource at this station.
Deactivate	CCE receives a signal to deactivate
Activate	CCE receives a signal to activate

Figure 17 (Part 1 of 3). Responding to QUERY Frames

Output Code	Function
a	Send a NOTIFY Frame to the CCE MAC address from the received QUERY.
b	Deactivate the FSM. Signal that deactivation has occurred.
c	Activate the FSM. Signal that activation has occurred.
d	Discard the QUERY frame.

Figure 17 (Part 2 of 3). Responding to QUERY Frames

Inputs	States	
	Active	Inactive
	01	02
QUERY frame received with correct netid, group name, and routing capabilities	-(a)	/
QUERY frame received without correct netid, group name, or routing capabilities.	-(d)	/
Deactivate signal received	2(b)	-
Activate signal received	-	1(c)/

Figure 17 (Part 3 of 3). Responding to QUERY Frames

5.7 Responding to Record and Erase Requests

The following FSM (see Figure 18) implements the CCE responses to a network entity's invocation of the **Record** and **Erase** primitives. This FSM does not result in the transmission of any CCE frames. It is merely a mechanism for a network entity to add and delete itself to/from the list of entities identified to the CCE. This FSM should be implemented by server stations. Non-server stations may also provide the functions.

Input Name	Description
Record (netid, name, routing capabilities, connection network id, list of SNPAs, list of group names)	CCE requested to record *netid.name* as available at this station, addressable through each of the list of SNPAs, having the specified routing capabilities, associated with the connection network id, and belonging to each of the list of group names.
Erase (netid, name)	CCE requested to erase *netid.name* from the list of network entities available at this station.
Deactivate	CCE receives a signal to deactivate
Activate	CCE receives a signal to activate

Figure 18 (Part 1 of 3). Responding to Record and Erase Requests

Output Code	Function
a	Add *netid.name* to list of network entities available at this station. Associate the specified characteristics with the specified resource. Signal Record_complete.
b	Remove *netid.name* from the list of network entities available at this station. Signal Erase_complete.
c	Deactivate the FSM. Signal that deactivation has occurred.
d	Activate the FSM. Signal that activation has occurred.

Figure 18 (Part 2 of 3). Responding to Record and Erase Requests

Inputs	States	
	Active	Inactive
	01	02
Record(netid, name, routing capabilities, connection network id, list of SNPAs, list of group names)	-(a)	/
Erase (netid, name)	-(b)	/
Deactivate signal received	2(c)	-
Activate signal received	-	1(d)/

Figure 18 (Part 3 of 3). Responding to Record and Erase Requests

6. 6.0 SAMPLE CCE LINE FLOWS

6.1 ADVERTISE/SOLICIT Flow

There are several possible variations of the ADVERTISE/SOLICIT flow that can result from the invocation of the **Find_member** primitive. The **Find_member** primitive can be invoked with the find_flag set to 1 to indicate that receipt of the first ADVERTISE frame is to be followed by transmission of a FIND frame to determine the SNPA of the entity identified in the received ADVERTISE. **Find_member** may also be invoked with the find_flag set to 0 to indicate that no FIND frame is to be transmitted. Another variation in ADVERTISE/SOLICIT flows can result because the CCE does not transmit a SOLICIT frame immediately upon invocation of the **Find_member** primitive. Instead CCE waits a period of time (the Solicit Timer Maximum) in an attempt to receive an ADVERTISE frame that might have been transmitted as a result of a SOLICIT from some other station.

Figure 19 illustrates the variation in which **Find_member** is invoked with the find_flag set to 1 and an ADVERTISE frame is received before the Solicit Timer expires. The sequence of flows is initiated by the invocation of **Find_member** by a network entity at a CCE station. The primitive specifies that CCE should find individual members of a given group name with given routing capabilities and netid. Because the find_flag is set to 1, the primitive also asks CCE to find the SNPA of some member of the group named. When CCE receives the **Find_member** invocation, the non-server functional address is opened (if not already open) for the receipt of ADVERTISE frames. The Solicit Timer is also started at its maximum value. The CCE waits to receive ADVERTISE frames that may be transmitted as the result of requests by other CCE stations. When an ADVERTISE is received with a group name, netid, and routing capability subvector that matches those on the **Find_member**, the CCE copies the information from the ADVERTISE and transmits a FIND frame to determine the SNPA of the individual named on the ADVERTISE. The FIND is addressed to the individual CCE MAC address given on the ADVERTISE. The FIND frame specifies the netid and name from the ADVERTISE as the target of the FIND. The Find Timer is started at its maximum value. The CCE continues to monitor the non-server functional address for additional ADVERTISE frames that match the **Find_member** specifications. Upon receipt of a FOUND frame that responds to the FIND (indicated by matching correlators), the CCE issues the return signal **Member_indication** to the network entity that invoked Find_member. The return signal gives the netid, name and SNPA of the target and echoes the group name and routing capabilities subvector. If no matching FOUND frame is received, the CCE issues **Member_indication** with a negative reply code. In any case, the CCE copies the information from any additional ADVERTISE frames that match the **Find_member** specifications as the frames are received. At the expiration of the Solicit Timer, the CCE issues the return signal **Find_member_complete**, providing a list of individual names that have been saved from the received ADVERTISE frames and echoing the group name, netid, and routing capability subvector from the **Find_member**.

Figure 20 illustrates the case in which the **Find_member** primitive is invoked with the find_flag set to 0 and an ADVERTISE frame is not received before the Solicit Timer expires. The sequence of line flows is very similar to that in Figure 19. The non-server functional address is opened and the Solicit Timer is started as before. In this case, however, no ADVERTISE frame is received with specifications that match the specifications in the **Find_member** before the Solicit Timer expires (No other CCE stations have recently requested servers to ADVERTISE.). At the expiration of the Solicit Timer, the CCE transmits a SOLICIT frame to request that servers send ADVERTISE frames and restarts the Solicit Timer. Because the find_flag is set to 0, the CCE does not transmit a FIND frame when the first matching ADVERTISE is received (as happened in Case 1). The CCE continues to monitor the non-server functional address for additional ADVERTISE frames that match the specifications of the **Find_member**. Information from such frames is copied and returned to the network entity at the expiration of the Solicit Timer as in Case 1.

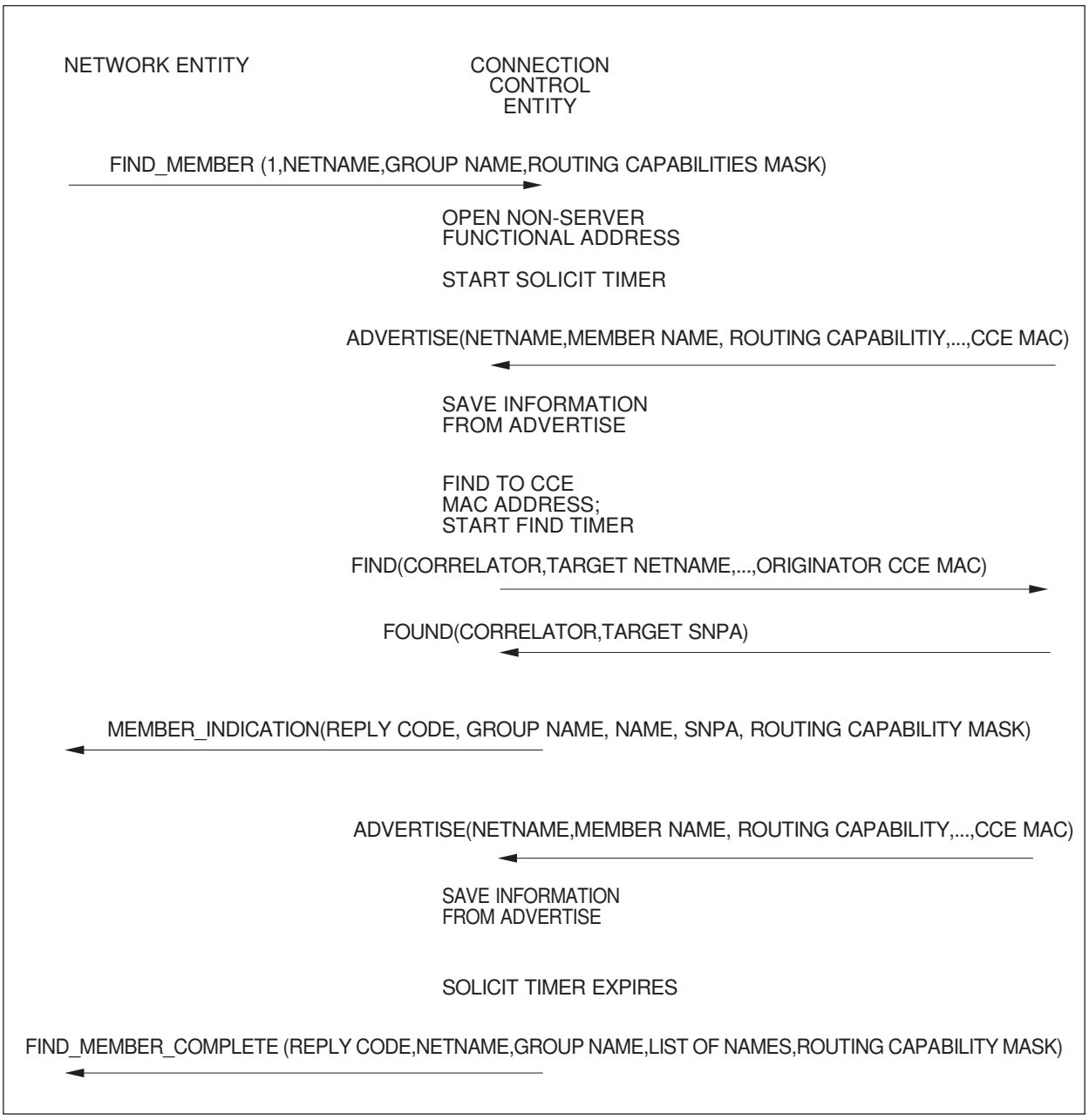


Figure 19. ADVERTISE/SOLICIT Line Flow at the Initiating CCE - Case 1

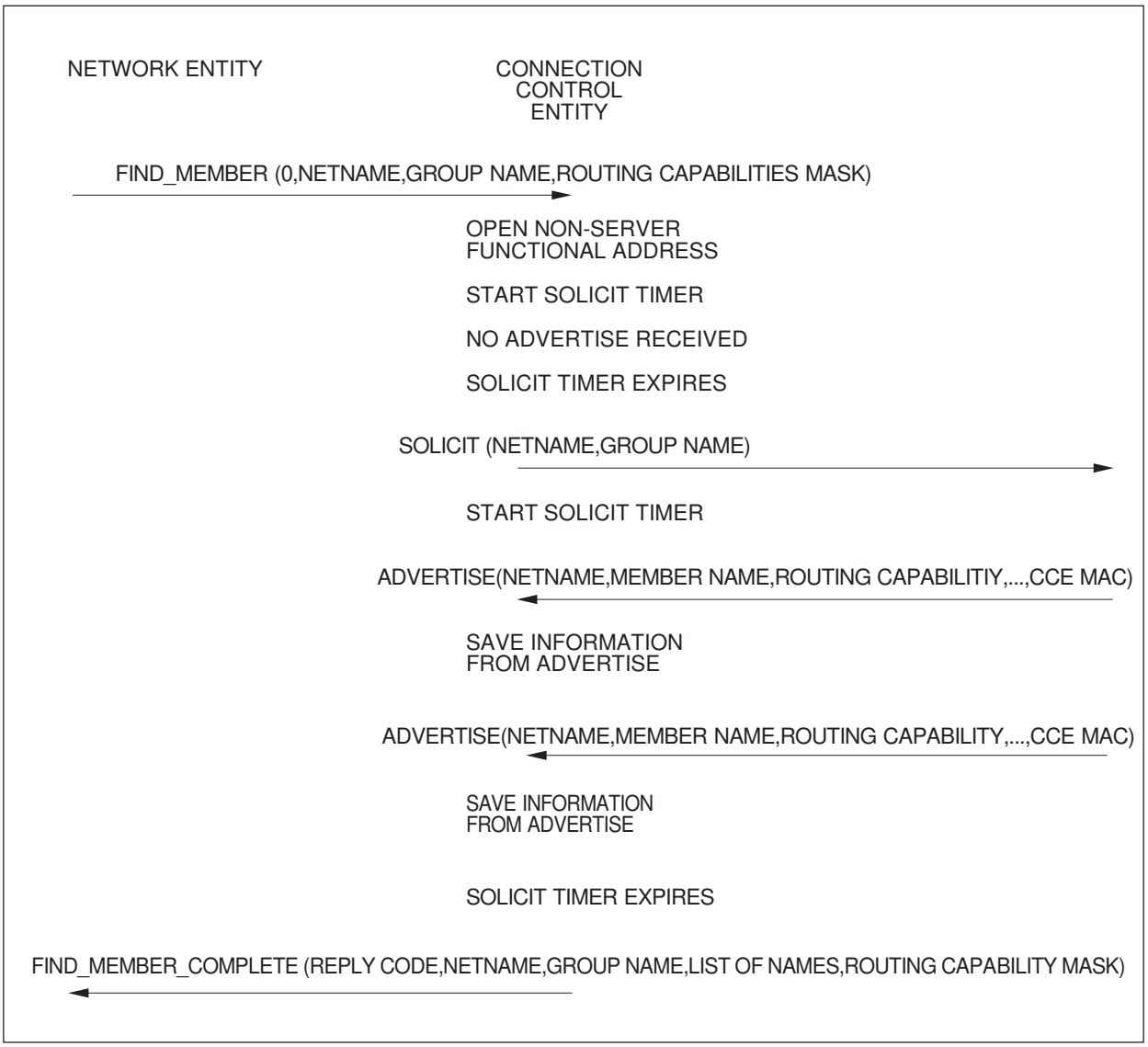


Figure 20. ADVERTISE/SOLICIT Line Flow at the Initiating CCE - Case 2

Figure 21 illustrates the behavior of CCE upon receipt of SOLICIT frames. The assumed initial condition at the CCE is that no ADVERTISE frames are pending, i.e., the Advertise Timer is not active. Upon receipt of a SOLICIT frame with a group name and netid matching those recorded for a network entity at the station (for group name A, say), the CCE chooses value randomly between 0 and the Advertise Timer Maximum minus the round trip delay and starts the Advertise Timer at that value.[11] The Advertise Count, indicating the number of times an ADVERTISE has been transmitted, is set to 0. A list of group names to be included in ADVERTISE frames (the Advertise List) is established with Group A as the first member. Upon expiration of the Advertise Timer, the CCE transmits an ADVERTISE frame, providing the list of group names, member name, and routing capability subvector). The Advertise Count is incremented and the Advertise Timer is restarted. When another SOLICIT frame is received with a matching group name, the group name (Group B, say) is added to the Advertise List. When the Advertise Timer expires, an ADVERTISE frame is again transmitted including the Advertise List and other necessary information. This process continues until the ADVERTISE frame has been transmitted a number of times equal to the Advertise Count Maximum.

[11] The purpose of this randomization is to disperse the ADVERTISE frames from different individual members (who would be responding to the same SOLICIT frame at nearly the same time) of the same group over the full interval from receipt of the SOLICIT to the next scheduled transmission of an ADVERTISE frame. This dispersal in turn helps prevent end systems that are merely monitoring for the first ADVERTISE (see Figure 21) from all receiving an ADVERTISE from the same server.

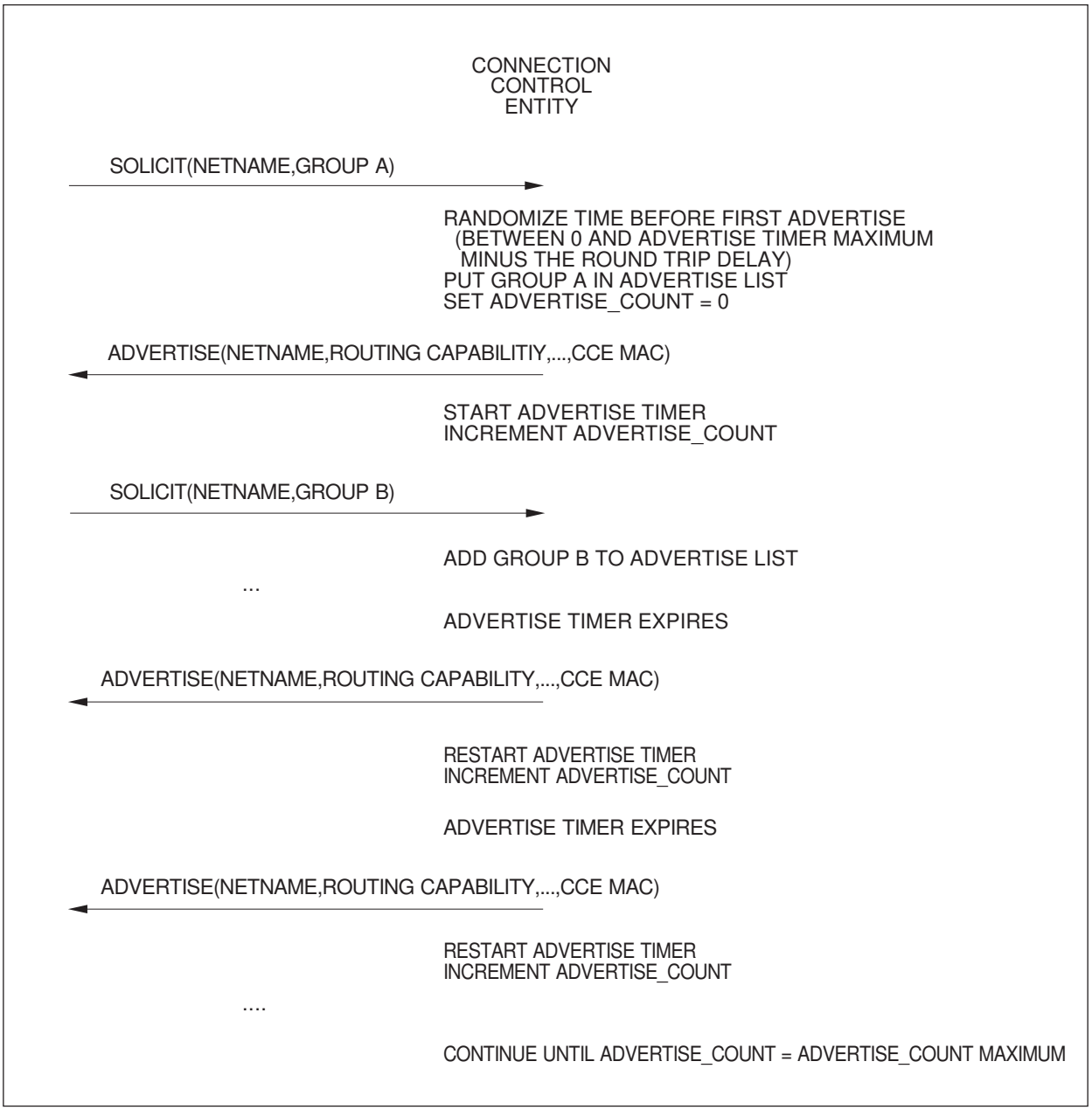


Figure 21. ADVERTISE/SOLICIT Line Flow at the Receiving CCE

6.2 FIND/FOUND Flow

The FIND/FOUND flow at an initiating CCE station is illustrated in Figure 22. The line flows are initiated by the invocation of the **Find** primitive by a network entity at the station. Upon receiving the primitive, the CCE starts the Find Timer and transmits a FIND frame. The frame may be addressed to an individual CCE MAC address if the individual CCE MAC address is known or to the All CCE functional address if the individual address is not known. (The individual CCE MAC address may be known, for example, because the address had been cached following receipt of an earlier NOTIFY frame.) The FIND frame includes a correlator, the netid, and name of the network entity that is the target of the FIND frame, and the CCE MAC address of the transmitting station. Upon receipt of a FOUND frame with a matching correlator, the CCE issues the return signal **Find_complete**, providing the SNPA of the target entity and echoing the target individual name to the network entity that invoked the **Find** primitive.

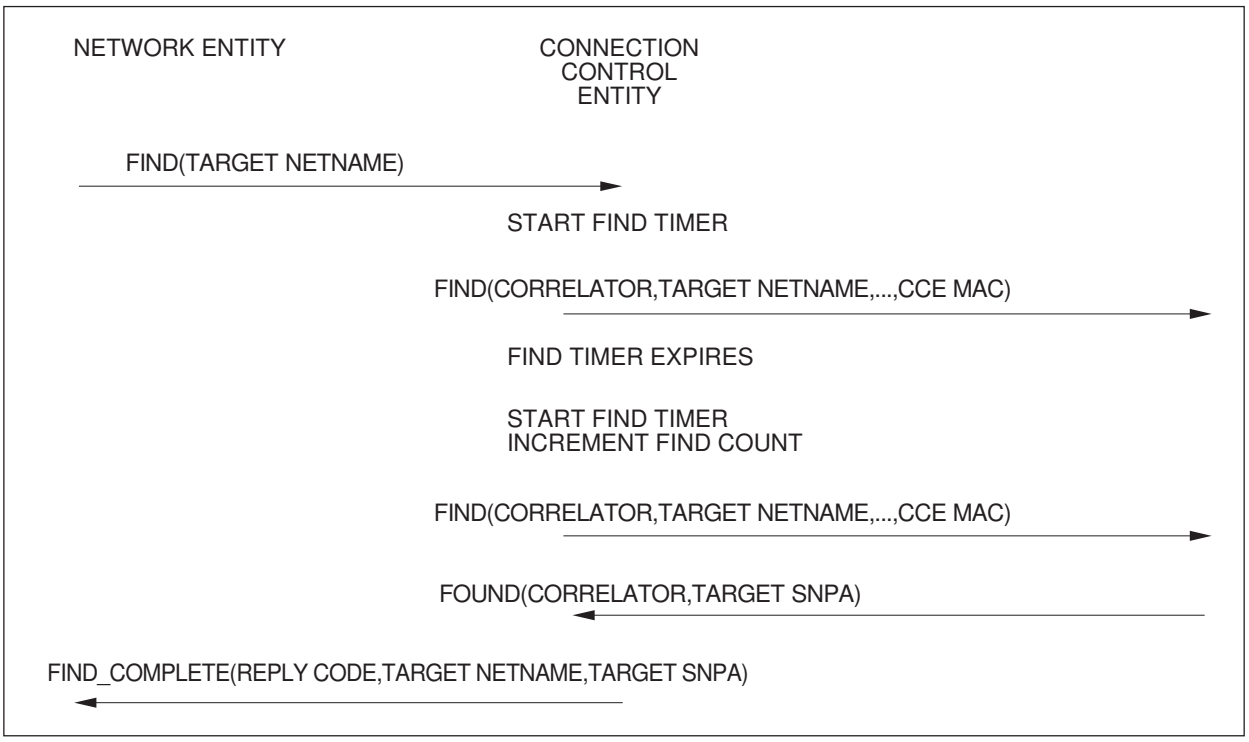


Figure 22. FIND/FOUND Line Flow

6.3 QUERY/NOTIFY Flow

The QUERY/NOTIFY flow at an initiating CCE station is shown in Figure 23. The line flows are initiated by the invocation of the **Search** primitive by a network entity at the station. The primitive specifies the routing capabilities, group name, and netid for which the network entity is searching. The CCE transmits a QUERY frame and starts the Query Timer. The QUERY frame is addressed to the server functional address, and it includes a correlator, the routing capability, group name, and netid from the **Search** primitive, and the individual CCE MAC address of the transmitting CCE station. Upon receipt of a responding NOTIFY frame (indicated by a matching correlator), the CCE copies the information in the NOTIFY. the CCE continues to monitor the CCE MAC address for additional matching NOTIFY frames, copying the information in each such frame. Upon expiration of the Query Timer, the CCE issues the return signal **Search_complete**, providing to the invoking network entity the list of received individual names that matched the **Search** specifications.

The QUERY/NOTIFY flow at a receiving CCE station is shown in Figure 24. Upon receipt of a QUERY frame with specifications matching those recorded for a network entity at the station, the CCE transmits a NOTIFY frame in response. The frame includes the correlator from the received QUERY, the individual name for which matching specifications were identified, and the CCE MAC address of the transmitting CCE.

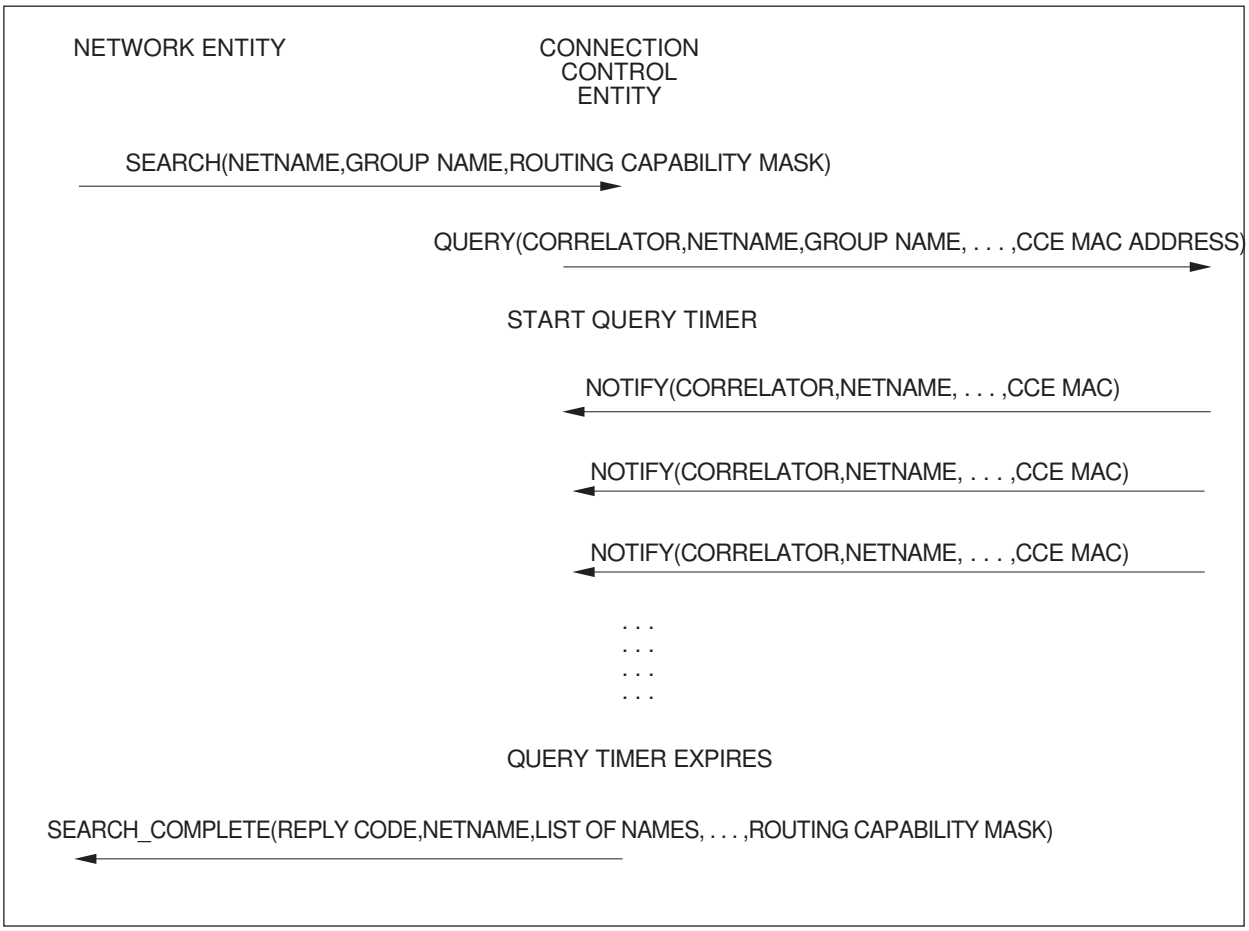


Figure 23. QUERY/NOTIFY Line Flow at the Initiating CCE

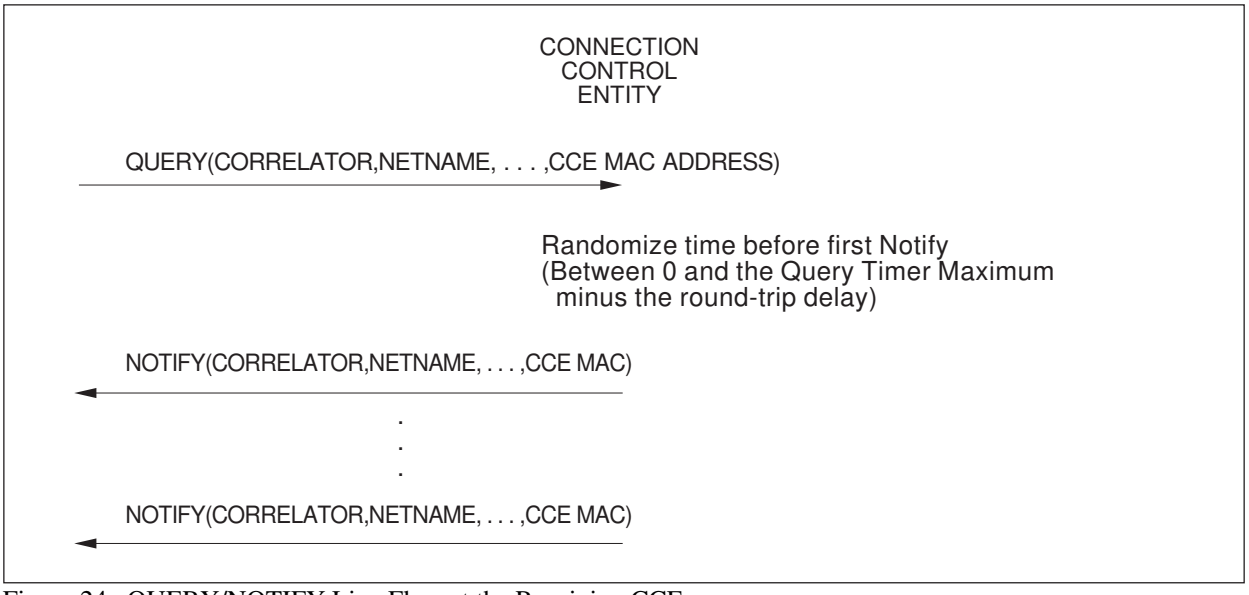


Figure 24. QUERY/NOTIFY Line Flow at the Receiving CCE

Appendix A. RELATED CONSIDERATIONS

A.1 Bridge Routing Considerations with CCE

IEEE 802.1 and 802.5 standards are changing relatively rapidly in the area of bridging of multi-segment LANs. Two activities in these committees will have significant effects on the present use of CCE and the migration of CCE as new standards are developed and implemented. Those activities are the introduction of a bridge route determination standard in IEEE 802.5 and the already approved IEEE 802.1 standard for the Source Route Transparent (SRT) bridge. Implementers of CCE should be aware of these evolving standards and plan their developments to minimize the redesign necessary when the new standards are implemented in the respective products. This appendix discusses some aspects of these standards that should be considered during CCE development.

The protocols defined in this document are used to dynamically bind a remote network entity to a SNPA through which LAN communications can occur with the entity. In a source-routing LAN, it is often desirable to also bind the pair (local SNPA, remote SNPA) to a bridge route through a multi-segment LAN. That bridge route will then be used on subsequent frames exchanged between the pair. This process of binding bridge routes in a source-routing LAN is called route determination. IEEE 802.5 is presently standardizing a technique for route determination, and the entity that performs the function is referred to as the Route Determination Entity or RDE. When the RDE becomes an IEEE 802 standard, products should implement the function in accordance with the standard. In the absence of the standard, many products have already implemented a route determination procedure called the **Resolve** procedure, documented in Local Area Network Internal Formats and Protocols. With the implementation of CCE, IBM products should continue to use the **Resolve** procedure for route determination until the RDE standard is completed.

The introduction of the SRT standard in IEEE 802.1 also has potentially significant effect on the development of CCE's LAN address resolution functions. The frame formats in Section 4.0 specify that the RII bit in CCE frames be set to B'1' to indicate that the Routing Information Field (RIF) is present and that the Routing Type (RT) field in the RIF be set to B'110' to indicate that the frames are Spanning Tree Explorer (STE) frames. These frame formats are appropriate for LANs such as IEEE 802.5 LANs that have source-routing end systems and source-routing (SR) bridges. In LANs such as Ethernet™ that use Transparent Bridges (TB) and do not support source routing, the RII bit should be set to B'0' and the RIF is not present. In existing bridge implementations, these methods are mutually exclusive in the sense that SR bridges do not forward transparent frames and vice versa. The frame formats in Section 4.0 are specified in SR formats because the vast majority of IBM implementations are source-routing token rings.

One of the principal objectives of the SRT standard is to describe a bridge (the SRT) that will forward both source-routing and transparent frames. Whenever a bridged LAN consists only of SR and SRT bridges, the CCE frame formats in Section 4.0 will be forwarded appropriately throughout the network by the source-routing mechanism of the SRT bridges. When TB bridges are introduced into the LAN, however, the CCE frames will be blocked by the TB bridges. A local network entity would be unable to discover the SNPA for a remote entity if the two were separated by a TB. If that function were required in a customer's network, it could be provided in two ways. The TB bridges could be replaced by SR/SRT bridges, allowing the source-routed CCE frames to flow throughout the network without change. The second method would replace all SR bridges in the network by SRTs, making the network a SRT/TB network. In that environment, the CCE frames should be transmitted with RII set to B'0' to allow the frames to flow throughout the network. Because the predominant bridging method can not be reliably predicted at this time, implementers are encouraged to incorporate in CCE implementations the capability to transmit CCE frames in either format. If that requirement emerges, however, the capability to easily modify the method of transmission without significant redesign would be of great benefit.

A.2 CCE Management Operations

Although beyond the objectives and scope of this document, a framework for CCE management operations has been developed in addition to the development of the fundamental CCE formats and protocols documented in the main body of the document. This section of the appendix presents that management operations framework. It is not a complete treatment of CCE management, and additional, separate work is required to complete the definitions of CCE managed objects and of the protocols that may be required to adequately manage the CCE.

The Connection Control Entity is comprised of two managed resources:

- the CCE Configuration.
- the CCE Protocol.

A.2.1 CCE Configuration Resources

The management operations for CCE Configuration inquire about the configuration of the CCE. The management operations that can be performed are **Get Configuration**, **Set Configuration**, **Get Identifiers**, and **Get Parameters**.

Get Configuration: The purpose of the **Get Configuration** operation is to obtain general information regarding the Connection Control Entity. There are no inputs to this operation. The outputs from the operation are:

- version number - the version number of the CCE software.
- service access point - the LSAP assigned to the CCE.
- CCE address - the MAC address for the CCE.

Set Configuration: The purpose of the **Set Configuration** operation is to modify configuration parameters in the CCE. There are no inputs to the operation. The outputs from the operation are:

- service access point - the new LSAP assigned to the CCE.
- CCE address - the new MAC address for the CCE.

Get Identifiers: The purpose of the **Get Identifiers** operation is to obtain a list of the individual and group identifiers (names) that the CCE responds to when a FIND, SOLICIT, or QUERY frame is received. There are no inputs to this operation. The outputs from the operation are:

- list of individual names - the list of the individual names of all network entities that the CCE recognizes.
- list of group names - the list of all the group names that the CCE recognizes.

Get Parameters: The purpose of the **Get Parameters** operations is to obtain information about the characteristics of a particular network entity that the CCE recognizes. The input to this operation is:

- individual name - the name of the network entity about which information is requested.

The outputs from the operation are:

- individual name - repeat of the network entity name from the input.
- connection network identifier - the CNID associated with this entity, if any.

- routing capabilities - the routing capabilities that the CCE associates with this entity.
- list of SNPA - the list of the LSAP and MAC addresses through which this entity is accessible.
- list of group names - the list of all the group names that the CCE associates with this entity.

A.2.2 CCE Protocol Resources

The CCE management operations for CCE Protocol inquire about parameters that define operation of the CCE protocol. The operations that can be performed are **Set Protocol Parameters** and **Get Protocol Parameters**.

Set Protocol Parameters: The purpose of the **Set Protocol Parameters** operation is to modify parameters in the CCE. The inputs to the operation are:

- Find Timer Maximum - the new value of the parameter.
- Advertise Timer Maximum - the new value of the parameter.
- Solicit Timer Maximum - the new value of the parameter.
- Query Timer Maximum - the new value of the parameter.
- Find Count Maximum - the new value of the parameter.
- Advertise Count Maximum - the new value of the parameter.

There are no outputs from the operation.

Get Protocol Parameters: The purpose of the **Get Protocol Parameters** operation is to obtain information about protocol parameters in the CCE. There are no inputs to the operation. The outputs from the operation are:

- Find Timer Maximum - the current value of the parameter.
- Advertise Timer Maximum - the current value of the parameter.
- Solicit Timer Maximum - the current value of the parameter.
- Query Timer Maximum - the current value of the parameter.
- Find Count Maximum - the current value of the parameter.
- Advertise Count Maximum - the current value of the parameter.